Reference Manual
Catalog of Programs for IBM Data Processing Systems
KWIC Index
April 1962
No. 1
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"**

<table>
<thead>
<tr>
<th>System Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAMAC 305 Data Processing System</td>
<td>35</td>
</tr>
<tr>
<td>650 Data Processing System</td>
<td>37</td>
</tr>
<tr>
<td>704 Data Processing System</td>
<td>41</td>
</tr>
<tr>
<td>705 Data Processing System</td>
<td>43</td>
</tr>
<tr>
<td>709 Data Processing System</td>
<td>47</td>
</tr>
<tr>
<td>1401 Data Processing System</td>
<td>49</td>
</tr>
<tr>
<td>1410 Data Processing System</td>
<td>55</td>
</tr>
<tr>
<td>1620 Data Processing System</td>
<td>59</td>
</tr>
<tr>
<td>1710 Data Processing System</td>
<td>63</td>
</tr>
<tr>
<td>7070 Data Processing System</td>
<td>65</td>
</tr>
<tr>
<td>7080 Data Processing System</td>
<td>69</td>
</tr>
<tr>
<td>7090 Data Processing System</td>
<td>71</td>
</tr>
</tbody>
</table>

**Program Abstracts, Section "B"**

<table>
<thead>
<tr>
<th>System Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAMAC 305 Data Processing System</td>
<td>73</td>
</tr>
<tr>
<td>650 Data Processing System</td>
<td>75</td>
</tr>
<tr>
<td>704 Data Processing System</td>
<td>179</td>
</tr>
<tr>
<td>705 Data Processing System</td>
<td>229</td>
</tr>
<tr>
<td>709 Data Processing System</td>
<td>245</td>
</tr>
<tr>
<td>1401 Data Processing System</td>
<td>253</td>
</tr>
<tr>
<td>1620 Data Processing System</td>
<td>265</td>
</tr>
<tr>
<td>7070 Data Processing System</td>
<td>287</td>
</tr>
<tr>
<td>7090 Data Processing System</td>
<td>307</td>
</tr>
</tbody>
</table>
INTRODUCTION

This catalog has been published as a service to computer users. It contains 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:

<table>
<thead>
<tr>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CARD SYSTEMS ERROR DETECTION AIDS</td>
</tr>
<tr>
<td>IDS</td>
</tr>
<tr>
<td>#CARD SYSTEMS ERROR DETECTION A</td>
</tr>
</tbody>
</table>

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>
</tbody>
</table>

A or B | The number of the IBM system for which the program is written | The IBM library code for filing and ordering a program.

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

<table>
<thead>
<tr>
<th>A</th>
<th>MODIFIED SUBPROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>NO SUBR</td>
</tr>
<tr>
<td>ADDS</td>
<td>NO, SUBROU</td>
</tr>
<tr>
<td>AN</td>
<td>NUMBER SUBROUT</td>
</tr>
<tr>
<td>AND</td>
<td>OF SYS</td>
</tr>
<tr>
<td>ANY</td>
<td>ON THE</td>
</tr>
<tr>
<td>AS</td>
<td>ONLY TO</td>
</tr>
<tr>
<td>AT</td>
<td>OR USING</td>
</tr>
<tr>
<td>ARITH.</td>
<td>OUT WITH</td>
</tr>
<tr>
<td>BY</td>
<td>PACKAGE I</td>
</tr>
<tr>
<td>DECK</td>
<td>POINT II</td>
</tr>
<tr>
<td>FOR</td>
<td>PROBLEM III</td>
</tr>
<tr>
<td>FROM</td>
<td>PROG V</td>
</tr>
<tr>
<td>GENERAL</td>
<td>PROGRAM VI</td>
</tr>
<tr>
<td>GENERATOR</td>
<td>PROGRAMS</td>
</tr>
<tr>
<td>IBM</td>
<td>PT</td>
</tr>
<tr>
<td>IF</td>
<td>PT.</td>
</tr>
<tr>
<td>IN</td>
<td>ROUTINE</td>
</tr>
<tr>
<td>INTO</td>
<td>ROUTINES</td>
</tr>
<tr>
<td>KIND</td>
<td>SOLUTION</td>
</tr>
<tr>
<td>Routine/Function</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>N OF N SIMULTANEOUS DIFFERENTIAL EQUATIONS</td>
<td>Solves simultaneous differential equations</td>
</tr>
<tr>
<td>GE DOUBLE PRECISION SIMULTANEOUS EQUATION SOLVER</td>
<td>Solves differential equations with high precision</td>
</tr>
<tr>
<td>FORTLESS SYS CALCUL AND PRINTING EVERYTHING</td>
<td>Performs calculations and printing</td>
</tr>
<tr>
<td>SLINEAR EQUATION SOLVER</td>
<td>Solves linear equations</td>
</tr>
<tr>
<td>SLEP1</td>
<td>Solves linear equations with pivoting</td>
</tr>
<tr>
<td>AST SQUARES SOL. OF SIMULTANEOUS EQUATIONS</td>
<td>Solves simultaneous equations using least squares method</td>
</tr>
<tr>
<td>#PRINCIPAL COMPONENTS PREDICTION EQUATION.</td>
<td>Performs principal components analysis</td>
</tr>
<tr>
<td>#LINEAR EQUATIONS SOLUTION FAP CODED</td>
<td>Solves linear equations with Fortran code</td>
</tr>
<tr>
<td>#DOUBLE PRECISION FLOATING POINT ADJUST.</td>
<td>Performs floating point adjustments with high precision</td>
</tr>
<tr>
<td>#DOUBLE PRECISION FLOATING POINT EXPONENTIAL</td>
<td>Performs exponential calculations with high precision</td>
</tr>
<tr>
<td>#DOUBLE PRECISION FLOATING POINT SUBROUTINE</td>
<td>Performs floating point subroutines</td>
</tr>
<tr>
<td>#NEWTONS METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#BENEDICT-WEBB-RUBIN EQUATIONS OF STATE.</td>
<td>Calculates state properties</td>
</tr>
<tr>
<td>#LINEAR MATRIX EQUATION SOLVER</td>
<td>Solves matrix equations</td>
</tr>
<tr>
<td>#DIFFERENTIAL EQUATION SOLVING SYSTEM</td>
<td>Solves differential equations</td>
</tr>
<tr>
<td>#SIMULTANEOUS EQUATIONS SOLVER</td>
<td>Solves simultaneous equations</td>
</tr>
<tr>
<td>#SIMULTANEOUS EQUATIONS COMPLEX</td>
<td>Solves complex simultaneous equations</td>
</tr>
<tr>
<td>#SIMULTANEOUS EQUATIONS REAL</td>
<td>Solves real simultaneous equations</td>
</tr>
<tr>
<td>#SIMULTANEOUS REAL SOLUTION</td>
<td>Solves real simultaneous equations</td>
</tr>
<tr>
<td>TRIPLE PRECISION EXPONENTIAL ROUTINE</td>
<td>Performs exponential calculations with triple precision</td>
</tr>
<tr>
<td>#PRINCIPAL AXIS FACTOR ANALYSIS</td>
<td>Performs factor analysis</td>
</tr>
<tr>
<td>#NORMALIZED VARIAX FACTOR ROTATION</td>
<td>Performs factor rotation</td>
</tr>
<tr>
<td>#FACTOR ANALYSIS BY THE CENTROID</td>
<td>Performs factor analysis by centroid method</td>
</tr>
<tr>
<td>#NORMALIZED RANGE COMPLEX ARITHMETIC</td>
<td>Performs arithmetic operations with normalized range</td>
</tr>
<tr>
<td>#HASTY EXPONENTIAL, FLOATING POINT</td>
<td>Performs exponential calculations</td>
</tr>
<tr>
<td>#GENERAL UTILITY PROGRAM</td>
<td>Performs various utility operations</td>
</tr>
<tr>
<td>#LONGAVERAGE FACTOR ANALYSIS</td>
<td>Performs factor analysis</td>
</tr>
<tr>
<td>#ERROR PROCEDURE FOR FORTRAN I I</td>
<td>Performs error procedures for Fortran I I</td>
</tr>
<tr>
<td>#NEWTONS METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#GENERAL ROOT FINDER FORTRAN SUBROUTINE</td>
<td>Performs root finding with Fortran subroutine</td>
</tr>
<tr>
<td>#GENERAL UTILITY PROGRAM</td>
<td>Performs various utility operations</td>
</tr>
<tr>
<td>#EXTREMUM OF UNIMODAL FUNCTIONS</td>
<td>Finds extrema of unimodal functions</td>
</tr>
<tr>
<td>#BACKSPACE FILE, FORWARD SPACE FILE</td>
<td>Performs file operations</td>
</tr>
<tr>
<td>#EXTRAPOLATION FROM DOUBLY TRUNCATED</td>
<td>Extrapolates from doubly truncated data</td>
</tr>
<tr>
<td>#NEWTONRAPHSON</td>
<td>Performs Newton-Raphson method</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#BENEDICT-WEBB-RUBIN EQUATIONS OF STATE.</td>
<td>Calculates state properties</td>
</tr>
<tr>
<td>#LINEAR EQUATIONS SOLUTION FAP CODED</td>
<td>Solves linear equations with Fortran code</td>
</tr>
<tr>
<td>#DOUBLE PRECISION FLOATING POINT ADJUST.</td>
<td>Performs floating point adjustments with high precision</td>
</tr>
<tr>
<td>#DOUBLE PRECISION FLOATING POINT EXPONENTIAL</td>
<td>Performs exponential calculations with high precision</td>
</tr>
<tr>
<td>#DOUBLE PRECISION FLOATING POINT SUBROUTINE</td>
<td>Performs floating point subroutines</td>
</tr>
<tr>
<td>#NEWTONS METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#FILE ORGANIZATION ROUTINES</td>
<td>Performs file organization routines</td>
</tr>
<tr>
<td>#LOAD AND UNLOAD DISK FILE</td>
<td>Performs disk file operations</td>
</tr>
<tr>
<td>#ZERO DISK FILE 3CCYCLIC</td>
<td>Performs cyclic disk file operations</td>
</tr>
<tr>
<td>#EXTRAPOLATION FROM DOUBLY TRUNCATED</td>
<td>Extrapolates from doubly truncated data</td>
</tr>
<tr>
<td>#EXTREMUM OF UNIMODAL FUNCTIONS</td>
<td>Finds extrema of unimodal functions</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#GENERAL UTILITY PROGRAM</td>
<td>Performs various utility operations</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#FILE UTILITIES</td>
<td>Performs file utilities</td>
</tr>
<tr>
<td>#EXTREMUM OF UNIMODAL FUNCTIONS</td>
<td>Finds extrema of unimodal functions</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#FILE UTILITIES</td>
<td>Performs file utilities</td>
</tr>
<tr>
<td>#EXTREMUM OF UNIMODAL FUNCTIONS</td>
<td>Finds extrema of unimodal functions</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#FILE UTILITIES</td>
<td>Performs file utilities</td>
</tr>
<tr>
<td>#EXTREMUM OF UNIMODAL FUNCTIONS</td>
<td>Finds extrema of unimodal functions</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#FILE UTILITIES</td>
<td>Performs file utilities</td>
</tr>
<tr>
<td>#EXTREMUM OF UNIMODAL FUNCTIONS</td>
<td>Finds extrema of unimodal functions</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#FILE UTILITIES</td>
<td>Performs file utilities</td>
</tr>
<tr>
<td>#EXTREMUM OF UNIMODAL FUNCTIONS</td>
<td>Finds extrema of unimodal functions</td>
</tr>
<tr>
<td>#NEWTON'S METHOD FOR FINDING ROOTS OF POLYNOMIALS</td>
<td>Solves polynomial equations using Newton's method</td>
</tr>
<tr>
<td>#POLYNOMIAL ROOT FINDER ROUTINES</td>
<td>Finds roots of polynomials</td>
</tr>
<tr>
<td>#FILE UTILITIES</td>
<td>Performs file utilities</td>
</tr>
</tbody>
</table>
NANTUATIONS OF SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.

DOUBLE-PRECISION FLOATING DECYM MATRIX CONVERSION PROG 8 0704-0601WHSN.

SIMULTANEOUS \[ T \cdot T \] VECT PROGRAM FOR THE 709/IMATR IX HAN I PULAT I NTERPRETI TION #UNIVERSAL MEMORY DUMP ANO CONDENSING ROUTINE 8 0704-0327GMITR.
SUBROUTINE SAVES THE CONSOLE TABLE(F).SUBRM.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
# MULTIPLE REGRESSION ANALYSIS.
null
UTINE #9X9 TEN MILLISECOND MULTIPLY SUBROUTINE 6E00-0314.001

ALIGNMENT PROGRAMS IDIGITAL TERRAIN MODEL SYSTEM VERTICAL 8 0650-09.2.041

ATA EDIT PROGRAM TD-1 D 0650-09.2.034

GENERAL PROGRAM ALIGNMENTS SUBROUTINE 1401-01.4.001

LOW, UNDERFLOW, AND DIVIDE CHECK TEST 6E00-0341.023

MULTIPLE TAPE TEST ROUTINE 6E00-0341.001

REGULAR SYSTEM TEST 1401-01.4.001

BINARY ARITHMETIC 6E00-0341.004

GROUPEMENT DE TÉLÉCHARGEMENT 6E00-0341.004

F PRODUCT TRANSFER 6E00-0341.004

FOURIER TRANSFORM 6E00-0341.004
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-AT-007

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 15,000 divisors.

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

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 $10^{-90}$ to $10^{90}$. Two versions of each routine are available, one utilizing the general purpose process control panel and the other requires a special wired panel.
Storage Requirements: Three drum tracks.
Remarks: 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-LM-006

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 82 x 97.
Storage Requirements: One disk.
Machine Requirements: Automatic division.
Additional Requirements: All arithmetic computations are performed by floating point subroutines. Data may be entered in fixed or floating point format.

IBM Application & Systems Programs Library Abstract File Number 0305-MI-003

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.

IBM Application & Systems Programs Library Abstract File Number 0305-LM-004

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 to 099999 on an IBM RAMAC 305 which utilizes six digit disk addresses.
GENERAL TRACING ROUTINE

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 \( Z + 3N + SM + K \) 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.

650 System: One 533 required.

Special Device: Alphabetic device for SOAP assembly.

FORTRANSIT I

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 \( (m_1, \ldots, m_n) \); IF; PAUSE; STOP; DO; CONTINUE; DIMENSION; READ; PUNCH; END.

Machine Requirements: Floating Point Arithmetic, Indexing Registers; 533 with alphabetic device.

FORTRANSIT II

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 \( (m_1, \ldots, m_n) \); IF; PAUSE; STOP; DO; CONTINUE; DIMENSION; READ; PUNCH; END.

Machine Requirements: Floating Point Arithmetic, Indexing Registers; 533 with special character device.

FORTRANSIT III

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 \( (m_1, \ldots, m_n) \); IF; PAUSE; STOP; DO; CONTINUE; DIMENSION; READ; PUNCH; END.

Machine Requirements: Floating Point Arithmetic, Indexing Registers; 533 with special character device; three 727 tape drives; standard 407.
IBM Application & Systems Programs Library Abstract  File Number  650-LM-004

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| < (2π · 10^7) / 2.

Accuracy:

<table>
<thead>
<tr>
<th>Range of Argument</th>
<th>Maximum error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>-2π ≤</td>
<td>A</td>
</tr>
<tr>
<td>2π ≤</td>
<td>A</td>
</tr>
<tr>
<td>2π ≤</td>
<td>A</td>
</tr>
</tbody>
</table>

Floating/Fixed: Uses floating point.

Mathematical Method: The Rand Approximation is used for Sin X expressed in radians.

Storage Required: Requires 49 locations. Execution time is 127 milliseconds.

Speed: 120 ms. for 10^9
127 ms. for e^t

Relocatability: Relocatable SOAP II form.

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

650 System: One 533, automatic floating decimal arithmetic, and three indexing registers.
Special Devices: For SOAP assembling, an alphabetic device is required.

IBM Application & Systems Programs Library Abstract  File Number  650-LM-005

SQUARE ROOT

Abstract:

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

b) Range: Any floating decimal argument

0 ≤ exponent ≤ 89. The error is less than one in the eighth place.

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

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

e) The program is in relocatable SOAP II form.

f) Alphabetic device used for SOAP II assembly.

IBM Application & Systems Programs Library Abstract  File Number  650-LM-006

5th ROOT FIXED POINT SUBROUTINE

Abstract:

a) Computes the 5th root of a single precision fixed point argument A.

b) Range: 0.000000001 ≤ A ≤ 999999999, N > 0. The number of significant places is approximately equal to ten minus the number of preceding zeros in A.

Minimum accuracy - nine digits.

c) Iteration of Bailey's function.

d) Relocatable SOAP II, occupies 78 locations. Speed is dependent upon N and the desired accuracy. The average speed is approximately 900 m. s.

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

f) Minimum 650.

IBM Application & Systems Programs Library Abstract  File Number  650-LM-007

FLOATING POINT EXPONENTIAL

Abstract:

Purpose: This routine computes 10^A and e^t for floating point arguments using automatic floating decimal arithmetic and three indexing registers.

Range: The routine accepts arguments for 10^t

| A | ≤ 49 |

The routine accepts arguments for e^t

| A | ≤ 112 | 82666 |

An error stop is provided for arguments outside this range.

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

Floating/Fixed: Floating decimal arithmetic.

Mathematical Method: (Adapted for floating decimal arithmetic and index registers from W. E. Stuart's "FRATS" library program 3. 1. 026)

e^t is reduced to 10^t x e^t = 10.43494464 x 10^t by computing in fixed point using a Hastings polynomial approximation over the range 0 ≤ t ≤ 1/10.

For negative exponents, e^t = 1/e^t.

Storage Required: Requires 86 drum locations within a group of 100 locations. The unused locations are available to the programmer.

Speed: 120 ms. for 10^t
127 ms. for e^t

Relocatability: Relocatable SOAP II form.

Remarks: Three indexing registers are used and not restored to their original values.

650 System: One 533, automatic floating decimal arithmetic, and three indexing registers.
Special Devices: Alphabetic device for SOAP II assembly.
NTH ROOT FLOATING POINT SUBROUTINE

Abstract:
Purpose: This routine computes the Nth root of a single precision floating point argument A.
Range: \(-.0000000000 \leq A \leq .9999999999, N > 0\).
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 \(xxxx:xxxxxxmm\), with floating zeros in the form \(00 0000 0000\).
Storage Required: 79 locations.
Speed: Speed is dependent upon \(N\) and the desired accuracy.
Relocatability: The subroutine is furnished in relocatable SOAP II form.
Remarks: The desired accuracy may be modified by the adjustment of a constant.

FLOATING POINT SQUARE ROOT SUBROUTINE

Abstract:
Purpose: This routine computes the square root of numbers in floating decimal form using an initial approximation and five iterations with Newton's method. This program was designed to use a minimum of drum space.
Range: This routine accepts floating point numbers of the form, \(DDD.DDDDMM\). Answers are in floating point form and all eight significant digits are exact.
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.
Storage Required: 21 permanent drum locations including a programmed stop for negative arguments. 3 temporary storage locations.
Speed: 140 ms.
The deck is in SOAP II form.
Remarks: The routine uses index register \(B\) 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.

FORTRAN SUBROUTINES

Abstract:
Purpose: This is a collection of subroutines to be used with the 650 FORTRAN programs. The subroutines are: absolute value, cosine, sine, and square root.

SORT 2

Abstract:
Purpose: Sort 2 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.
Method: 3-way merge.
Equipment Specifications: 4 727 Magnetic Tape Units
Additional Remarks: Routines for tape labelling, error corrections, restart procedures, record count, and hash totals are included.

BASIC SOAP 2A

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.
SOAP 1L

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-for-one machine language instructions. SOAP 2L processes LITERALS and three other pseudo-ops, not handled by SOAP II A.

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

Machine Requirements: 633 with alaphabetic device.

IBM Application & Systems Programs Library Abstract  File Number 0650-SP-203

SOAP 2L

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-for-one machine language instructions. SOAP 2L will process LITERALS and three other pseudo-ops, not handled by SOAP II A.

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

Machine Requirements: 633 with alaphabetic device.

IBM Application & Systems Programs Library Abstract  File Number 0650-SP-204

SOAP II A - 4000

Abstract:

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

Restrictions: A maximum of 1200 labels are processed per pass of card deck.

Machine Requirements: 633 with alaphabetic device. 4K drum.

IBM Application & Systems Programs Library Abstract  File Number 0650-SP-205

SOAP 42

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.

Machine Requirements: 633 with alaphabetic device.

IBM Application & Systems Programs Library Abstract  File Number 0650-SP-206

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 213 drum 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 numerical mode, the same speed will be maintained if not more than seven words are taken from each card. These rates apply to 533 input; if input is by means of a 537 or a 407, the maximum card reading rate (150 cards per minute) 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.

IBM Application & Systems Programs Library Abstract  File Number 0650-UT-002

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 SO 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 533 or one on-line 407 printer; one 727 tape unit.

Special Devices: None.

IBM Application & Systems Programs Library Abstract  File Number 0650-UT-003
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.

Simulation of the 1110 with the 704/709/7090

Abstract:

**Purpose:** The program enables the user to test and correct 1110 programs prior to installation of an IBM 1110 data processing system. The system will trace or dump simulated programs.

**Restrictions:** The program simulates standard card and tape systems. The simulated 1110 has 50,000 core storage positions. Using Basic Autocodes the simulator will assemble 1110 programs. A maximum of one disk of 1110 storage can be simulated.

**Timing:** The 704 takes approximately 50 times longer than if the program was running on a 1110.

**Equipment Specifications:**
- 30,076 words of core storage
- 4 tape units + 1 for simulated 1110 tape units + 2 for disk

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

IBM Application & Systems Programs Library Abstract  File Number  0704-F0-037

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

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

IBM Application & Systems Programs Library Abstract  File Number  0704-SI-044
IBM Application & Systems Programs Library Abstract  File Number 0705-AV-007

APTS 83

Abstract:

Purpose: An automatic program testing system for the IBM 7090 III, consisting of a coordinated set of the "83 series" utility programs that are used in testing, modified so that the utility programs themselves may be tested automatically from a utility tape, and their control cards from the card reader or other input device independent of the utility tape. With APTS 83, 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-046

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 705-1401 Assembly Program will run on a Model I, II, III, 1400, 1430, 1440.

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 0705-10-047

705 III JCS8

Abstract:

Purpose: JCS8 handles reading and writing, checkpoint and restart, error correction, beginning and end-of-reel and beginning and end-of-file processing, tape card blocking and de-blocking, and label checking. Macro-instructions and control parameters coded by the programmer cause generation of listings to JCS8 automatics, which in turn perform the specified functions.

An input/output memory restore system (UCAS BHR) operates in conjunction with JCS8 to restore program status from periodically recorded checkpoints, so that in the event of program interruption, previous processing need not be repeated.

Storage Requirements: Preassembled JCS8 occupies 17,074 locations.

Equipment Specifications: 705 Model III

IBM Application & Systems Programs Library Abstract  File Number 0705-AQ-009

LIST 76

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 Specifications: 705 Model I or Model II

IBM Application & Systems Programs Library Abstract  File Number 0705-MQ-050

LIST 77

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 Specifications: 705 Model I or Model II

IBM Application & Systems Programs Library Abstract  File Number 0705-PR-044

7056 PROCESSOR

Abstract:

Purpose: The 7056 Processor accepts six programming languages: Autocoder III; Decision-Making/Report/File Writing; Arithmetic; Table Creating; and FORTRAN. It will operate with any input/output device, on a 705, 706 III, or 7080 and announce programs for any model 705 or 7090.

7056 Processor languages, described below, permit 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 the incidence of clerical and logical errors, and will simplify problems of debugging and program modification. A statement in any of the languages may cause generation of an entire protected routine 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 instructions 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 operators or 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 necessary locations of data or 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/Files 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 or produce printed reports. Statements in this language describe the format of point or line or tape records by specifying the contents and spacing of record headings, page headings, and detail lines. A date and page numbering may be included in this report. Provisions are made also for accumulating counts or totals of any designated fields in the records being processed. Such tabulations may be printed, or inserted in the output file, or printed on a tape record, or used for printing these in stated formats upon tape or paper difficult, the columns 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 statement, TURE, is the vocabulary to the language of the Processor that signifies a Condition Decision-Making statement. The operand is composed of tags, literal constants, or special codes that express the relationship (e.g., lower than, equal to, etc.) that define the individual conditions. Conditions are listed within a statement by logical connectors and are grouped in a prescribed manner to form the complete conditional statement. Decision-Making statements are translated by the Processor into instruction sequences that will perform the necessary analyses 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 field. The word MATH is in the operation field signals that the operand contains a free-form arithmetic expression consisting of tags and/or literals separated by add, subtract, multiply or divide symbols, with possible parentheses. Specialized error protection, field modification, and redefinition of intermediate results are some optional features. These statements are translated by the Processor into automatic-decimal-point macro-instructions, chided to produce the most efficient machine coding.

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 TABLE in the operation field and consisting only of operands, the programmer specifies the table entries or ranges of entries. These entries are translated by the Processor into a table suitable for sequential or binary searching. Such a table may be utilized by other Processor languages for organizing data, or for printing these in stated formats.

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 1-9999. All the advantages of 7056 Processor assembly are available to the user.

Equipment Specifications: 7056 Processor programs may contain Autocoder statements at appropriate points.

IBM Application & Systems Programs Library Abstract  File Number 0706/1050 COBOL and COMMERCIAL TRANSLATOR PROCESSOR

Abstract:

Purpose: The processor translates programs written either in COBOL or Commercial Translator into machine language, and announces programs for the 705 Models I, II, III, and III, and the 7080. 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. For the Models III and IV the 7056 processor is used.
The 705/7068 COBOL and Commercial Translator Processor includes all the features of the 705 Processor, Version #2. It may be used to compile programs written in Autocoder, FORTRAN, Report Writer or the Decision, Arithmetic and Table languages as well as COBOL and Commercial Translator. Further, a COBOL or Commercial Translator program may utilize any of the languages available with the 705 Processor.

Machine Configuration: A 705 Model II, 705 Model III or 7068 with a minimum of eight tape units plus a card reader or additional tape unit for the source program. The availability of additional tape units will normally result in increased speed of compilation.

IBM Application & Systems Programs Library Abstract  File Number  0705-SM-048

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)
784 Tape Control
7 727 Tape Drives
717 Printer

Additional Remarks: Sort 54 incorporates checkpoints, 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-049

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)
784 Tape Control
7 727 Tape Drives
717 Printer

Additional Remarks: Sort 54T incorporates checkpoints, 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-050

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 705 (Model I or Model II)
7 727 Tape Record Coordinators
7 717 Printer

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

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 III
784 Tape Control
7 727 Tape Drives
717 Printer

Additional Remarks: Sort 54/ incorporates checkpoints, 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 III
7 727 Tape Record Coordinators
7 717 Tape Drives
717 Printer

Additional Remarks: Sort 54T/ incorporates checkpoints, 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 705 Model III
7 727 Tape Record Coordinators
7 717 Tape Drives
717 Printer

Additional Remarks: Sort 57/ incorporates checkpoints, 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 scheduling, 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 Sort mode for sorting particularly large files, and provisions for label processing and for the accumulation and checking of hash totals. Edits are provided at logical points in the program to allow the user to include additional routines. Sort 80 also provides checkpoints, interrupts, and restart procedures, and routines which facilitate the correction, or deletion and later recovery of uncodeable records.

Equipment Specifications:

735 Model III or 7039
977 Data Synchronizer
4 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
707 Data Synchronizer
4 tape drives

60 SERIES UTILITIES

Abstract:

Purpose: All "60 Series" utility programs except LOAD 80 and CLRM60 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 705 DS tape.

Clear Memory (CLRM60): Sets memory positions 00160 - 30999 (or 79999) to blanks, and resets the accumulator and ASS's 01 - 11 without interrupting automatic operation.

Expanded Loads (LOADB1 and LOADB2): 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 (TFA80): 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 (MEPR60): Produces a printed listing of the contents of any tape mounted on a 729 tape unit, either directly on a 731, 708 or 730 printer or on a 729 tape for later off-line printing.

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

Equipment Specifications: 705 Model III or 7090
Abstract: To make possible the execution of 704 programs on the 709 by assuming responsibility for all input/output functions, and to simulate 704 drum storage in cases if drums are not present in the 709 system.

Purpose: 9PAC is a collection of three systems, known as File Processor, Reports cores, 704/709 INPUT/OUTPUT COMPATIBILITY PROGRAM responsibility for all input/output functions, and to simulate 704 drum storage in cases if drums are not present in the 709 system.

Purpose: To make possible the execution of 704 programs on the 709 by assuming responsibility for all input/output functions, and to simulate 704 drum storage in cases if drums are not present in the 709 system.

System Programs Library Abstract File Number 0709-PR-064

SHARE OPERATING SYSTEM - SHARE MONITOR VERSION

Abstract: Purpose: S0S is a set of components controlled by a three-phase monitor operating on stacked jobs. The system compiles symbolic machine-oriented language into condensed squoze form and/or performs one-pass loading of squoze decks with symbolic modification. The output includes absolute decks, listings, and new squoze deck. Features include programmer macros, library facilities, system macros, and routines for symbolic debugging. Tape assignments and system references are symbolic.

SHARE OPERATING SYSTEM - IB MONITOR VERSION

Abstract: Purpose: S0S is a set of components controlled by a one-phase monitor operating on stacked jobs. The system compiles symbolic machine-oriented language into condensed squoze form and/or performs one-pass loading of squoze decks with symbolic modification. The output includes absolute decks, listings, and new squoze deck. Features include programmer macros, library facilities, system macros, and routines for symbolic debugging. Tape assignments and system references are symbolic.

SIMULATE PERIPHERAL EQUIPMENT

Abstract: Purpose: This is a collection of programs for conversion of card formats. They are:

1. IBM01 Hollerith to BCD, or Column Binary to Row Binary
2. IBM02 BCD to Hollerith
3. IBM03 Row Binary to Column Binary
4. IBM04 BCD to live image

Restrictions: Hollerith input may contain only those characters listed in Appendix I of The Share 709 System (SOC) Manual, Part I, Preliminary Edition, July, 1958, including the symbols "normally not used". Any other character will cause an error return.

Column binary input must be identified by a 1 in the sign positions of the 9-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 column as specified by the calling sequence.

Timing:

- IBM01 80 - 100 ms
- IBM02 38 ms
- IBM03 118 ms
- IBM04 50 - 60 ms

Storage Requirements:

- IBM01 256 + 1/O words
- IBM02 151 + 1/O words
- IBM03 66 + 1/O words
- IBM04 182 + 1/O words

IBM Application & Systems Programs Library Abstract File Number 0709-PR-063

SHARE OPERATING SYSTEM - IB MONITOR VERSION

Abstract: Purpose: S0S is a set of components controlled by a one-phase monitor operating on stacked jobs. The system compiles symbolic machine-oriented language into condensed squoze form and/or performs one-pass loading of squoze decks with symbolic modification. The output includes absolute decks, listings, and new squoze deck. Features include programmer macros, library facilities, system macros, and routines for symbolic debugging. Tape assignments and system references are symbolic.

SHARE OPERATING SYSTEM - SHARE MONITOR VERSION

Abstract: Purpose: S0S is a set of components controlled by a three-phase monitor operating on stacked jobs. The system compiles symbolic machine-oriented language into condensed squoze form and/or performs one-pass loading of squoze decks with symbolic modification. The output includes absolute decks, listings, and new squoze deck. Features include programmer macros, library facilities, system macros, and routines for symbolic debugging. Tape assignments and system references are symbolic.

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

26 IBM 700/709 FORTRAN PROGRAMMING SYSTEM

Abstract: Purpose: The IBM Formula Translating System, 26 IBM 700/709 FORTRAN, is an automatic coding system for the IBM 709/7090 Data Processing System. More precisely, it is a 709/7090 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 709 or 7090. The system also contains the FAP Assembler and FORTRAN Monitor, enabling jobs to be compiled, assembled, and executed automatically.

IBM Application & Systems Programs Library Abstract File Number 0709-PR-062

737/95 SPAC

Abstract: SPAC is a collection of three systems, known as File Processor, Reports Generator and 8PAC Sort. They respectively maintain, write reports from, and sort a file. The source language is written in a series of specialized forms and describes the function to be performed or a pictorial view of the output reports. I/O is handled by the system and need not concern the programmer. The mode of operation may be either complete and execute, or load and execute.

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

SORT 709

Abstract: Purpose: This is a generalized sort program. This program uses a 5 through 5-way merge, input to binary or BCD from tapes. The tape may consist of one or more reels of fixed-length records. Input tape is sorted into ascending order sequence based upon 1 through 5 control fields arbitrarily arranged within the record. The control fields may have a total of up to 50 bits.

Usage: Control cards specify record length, input and output blocking, control fields, memory available, merge order, and tape unit. 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 on 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 splitting 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.

709 UTILITIES

Abstract:

Purpose: This is a collection of 8 utility routines:

1. RAWG generates a file of random binary or BCD digits.
2. 90AL loads instructions punched in absolute octal with their alphabetic mnemonic operation codes.
3. YMSG prints on-line messages.
4. TCMP compares two tapes word for word.
5. XMSG 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 prints upon completion of program.
7. T0L2 builds short tapes for testing and other special purposes.
8. T0 provides an octal or BCD print of tape.

709 DATA PROCESSING PACKAGE

Abstract:

Purpose: The 709 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.
Purpose: To provide a simple 1401 system for checking out programs.

Abstract:

2. Print selected areas of storage at selected times.

Equipment Specifications:

1. Halt the program

2. Means for conveniently removing the patches are also provided.

Purpose: To provide more powerful tools for programmers to enable them to concentrate their efforts on the problems of program logic rather than coding.

Abstract:

The FORTRAN compiler translates such descriptions, established FORTRAN programming language, the principal use of which is to describe solutions in scientific and engineering problems. The FORTRAN compiler translates such descriptions, or source programs, into 1401 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. Symbolic origins
3. Advanced Programming Features
4. A - 1401

Use of program:
The user's FORTRAN program statements, punched on cards, are entered into the 1401 DFS, followed by the FORTRAN compiler, which may be on cards or tape. The source program is translated by the compiler into the 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 report 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-Equal Compare Feature
   Multiply-Divide Feature

2. Card Read-Punch
   1402
   1403 Printer (Model 1 or 2)

One Tape Unit Model 729 II, 729 IV, 729 V, 729 VI, or 7330 may be used if installed to store and load the 1401 FORTRAN compiler

Some switches may be used if installed to provide a 1403 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-Equal Compare Feature
   Multiply-Divide Feature

2. Card Read-Punch
   1403 Printer (Model 1 or 2)

Tape Units Model 729 II, 729 IV, 729 V, 729 VI, or 7330 - only as required for input and output data.

Some switches - may be used if installed.
1401 INPUT/OUTPUT CONTROL SYSTEM

Abstract:
Purpose: The 1401 I/0CS consists of a set of library routines which, when called for in a 1401 Assembler 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/0CS routines are to be placed in the 1401 Assembler system Version 3 or later Version through a librarian run.

Machine Configuration: The 1401 I/0CS 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 any of the above I/O units, 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-IC-001

1401 CARD SYSTEM SUBROUTINES

Abstract:
Purpose: To provide a few frequently used arithmetic subroutines.
Method: Does not apply.
Restrictions, Range: Does not apply.
Remark: Programs provided:
Multiply I (for storage space economy)
Multiply II (for speed economy)
Divide
Decimal-to-Units Conversion
Units-to-Decimal Conversion
Note: Closed subroutine linkage instructions provided.

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

FARGO (Fourteen-O-One Automatic Report Generating Operation)

Abstract:
Purpose: To provide a simple-to-learn, easy-to-use method of converting accounting reports from unit record equipment (802A - 402 - 514 - 604 - 407 - 519 types) into an IBM 1401 Data Processing System.

Programming Language: 1401 Symbolic Programming System

Machine: Load & Go, which means there is no interactive symbolic assembly operation.

Features: Data Sorting and Cross-heading, a feature that may be used to sort the data according to any number of fields.

Range:
1. Add or Tabulate with or without Summary Punching.
2. Print one full line of Report Heading on the first line of each page of the report.
3. Print 1 or 2 full lines of Columnar or Field Headings on each page.
4. Control on a maximum of four fields of any length.
5. Group Indicate a maximum of four fields on the first line of each minor control group.
6. Recognize up to 10 types of detail cards by any single column character. If more than one card column must be tested to identify a given type of card, a patch is required. Note: Each of the 10 types may be in separate card columns
7. Add, Subtract, Multiply*, Divide* operations may be performed on Detail or Total lines. *These operations require Multiply/Divide feature.
8. Print multiple lines from one card (MLP).

IBM Application & Systems Programs Library Abstract
File Number 1401-3G-045

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 or disk storage file. The language facilities specifying the classic report writing functions of heading and detail lines, total lines controlled by control field breaks, offset total printing, summary punching, cross-footing and calculation, 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, punched 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 (SPS-1, SPS-2, or Automatic) to obtain the machine language report writing program ready for loading.

Machine Configuration:
For report program generation:
1. 1401 Processing Unit (any model with 6000 or more Core storage positions)
2. 1400 Card Read Punch
3. 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)
2. 1400 Card Read Punch
3. Tape Units (Model 729 II, 729 IV, 729 V, 729 VI, or 7330), 1403 Printer (Model 1 or 2), 1405 Disk Storage Unit - only as required for input data file and output report media.
4. Multiply-Dividc Special Feature - may be used if installed.
5. Sense Switches Special Feature - may be used if installed.

IBM Application & Systems Programs Library Abstract
File Number 1401-3G-049

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 sequential output records. SORT I may also be used as a merge program if input tapes are already ordered.

Method: Does not apply.

Restrictions, Range: Does not apply.

Storage Requirements: Does not apply.

Specifications:
4000 positions of storage
High-Low-Equal compare
Minimum of four (4) tape drives

Additional Remarks:
1. SORT I may handle single or blocked records.
2. The sort will be on a maximum of five (5) control fields.
3. SORT I will allow a maximum of 800 character blocking for single control field records and 755 for multiple control field records.

(Continued on next page)
4. Restart procedure is provided before each pass.
5. Output can be relabeled.
6. SORT 1 will process input labels and provides the insertion of a different output label if desired.
7. Three (3) options are provided for disposing of unreadable records:
   (1) Accept record by correcting invalid character
   (2) Punch unreadable block
   (3) Write unreadable block on fifth tape (if available).

IBM Application & Systems Programs Library Abstract File Number 1401-SM-043

1401 SORT II

Abstract:
Purpose: To provide a sort program for advanced 1401 systems. The program consists of an internal sort, which orders a large block of records internally, and a two or three way merge which creates an ordered sequence as output.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications:
Minimum of 8000 positions of memory
High-Low-Equal Compare
Advanced Programming Features
Minimum of four (4) tapes

Additional Remarks:
1. SORT II is a generalized sort program adapted for a particular application by use of a control card. It will adapt for 8K, 16K or 16K machine, and may be used as either a two or three way merge.
2. Input records may be singly or multiply blocked.
3. A maximum of ten (10) control fields can be specified by the user.
4. The user may specify size of patch area desired. The program will modify itself to reserve space for any specified patch. Convenient exits are provided in the program.
5. The allowable blocking is dependent on machine size and patch size. Maximum blocking for a 16K machine with no patch area is 3,999 characters.
6. Restart and unreadable record procedures are similar to those of SORT 1.

IBM Application & Systems Programs Library Abstract File Number 1401-SM-044

1401 MERGE II

Abstract:
Purpose: To provide a two, three, four or five way generalized merge program for advanced 1401 systems.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications:
Minimum of 6000 positions of memory
High-Low-Equal Compare
Advanced Programming Features
Minimum of three (3) tapes

Additional Remarks:
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 maximum 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.

IBM Application & Systems Programs Library Abstract File Number 1401-SP-021

SYMBOLIC PROGRAMMING SYSTEM 1 (SPS-1)

Abstract:
Purpose: To provide a basic symbolic programming language and processor for the IBM 1401.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications:
1400 positions of storage
1420 Reader-Punch
1430 Printer, Model 1

Additional Remarks:
1. SPS-1 is designed to run on a machine with minimum hardware specifications.
2. Additional storage, up to 4000 positions, is used if available.
3. Read release option used if available.

IBM Application & Systems Programs Library Abstract File Number 1401-SP-020

SYMBOLIC PROGRAMMING SYSTEMS 2 (SPS-2)

Abstract:
Purpose: To provide a symbolic language processor for machines with greater than 4000 positions of core storage.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications:
4000 positions of storage
1420 Reader-Punch
1430 Printer, Model 1

Additional Remarks:
Additional storage, up to 16,000 positions, is used if available.

IBM Application & Systems Programs Library Abstract File Number 1401-UFT-001

1401 CARD SYSTEM UTILITY PROGRAMS

Abstract:
Purpose: Utility Programs to load or to 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:
Purpose: 1. To enable the printing of various tape configurations in map print
configurations without the need for specific programs.
2. To simulate the 717, 720 and 730 off-line printers for tapes prepared
on 700-7000 series computers.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications: 1401 Model C3

Additional Remarks:
1. Maximum block size allowable 1197 characters.
3. Multiple file reels and multi-file reels may be handled.
4. Sequence checking and exception testing are provided.
5. Header and trailer labels are optionally treated.

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

IBM 1401 CARD-TO-TAPE UTILITY PROGRAM

Abstract:
Purpose: The Card-to-Tape program provides for writing information contained
in punched cards onto magnetic tape.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications: 1401 Model C3

Additional Remarks:
1. Input record in from 1 to 99 cards.
2. Rearrangement of input prior to output is allowed,
3. Up to 15 fields may be selected for output.
4. Blocking of 1496 characters of BCD records and 1599 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. 
9. A count of the number of data cards read and of the records written,
exclusive of header and trailer cards and records, is printed out at
the end of each file.

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.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications: 1401 Model C3

Additional Remarks:
1. To simulate the 700 series equipment, and
allow any combination of Tape-to-Card, Tape-to-Printer and Card-to-Tape
operations to be performed at the same time.
2. The following rates are applicable:
3. Exception record processing and card sequence numbering is allowed.
4. 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, Tape-to-Printer and Card-to-Tape
operations to be performed at the same time.
Method: Does not apply.
Restrictions, Range: Does not apply.
Storage Requirements: Does not apply.
Equipment Specifications: 1401 Model C3

Additional Remarks:
1. High-Low-Equal Compare
2. Maximum speed will be effected when any one single operation is
being performed. Tape-to-Printer 600 lpm, Card-to-Tape
800 c.p.m., Tape-to-Card 250 c.p.m.
3. When more than one operation is desired simultaneously, the
following time speeds are applicable:

(Continued on next page)
Multiple Utility Program for IBM 1401 Tape System

Card-to-Tape - Tape-to-Printer, single space printing 510 c 1 ppm.
Card-to-Tape - Tape-to-Printer, 1st character forms control 300 c 1 ppm.
Card-to-Tape - Tape-to-Card, 100 card read, 100 card punch ppm.
Tape-to-Printer - Tape-to-Disk, 315 c 1 ppm, 100 cpm.
Card-to-Tape - Tape-to-Printer, Tape-to-Card 315 c 1 ppm, 516 card read, 140 card punch ppm.

1. High and low densities may be intermixed on the several tape drives while running simultaneous operations.
2. Binary and BCD operations may be processed at the same time except that the same operation (i.e., Tape-to-Card BCD as operation #1, and Tape-to-Card Binary as operation #2) is not permitted.
3. Any combination of the following may be processed at the same time considering the restrictions stated in 2 above: Tape-to-Card BCD, Tape-to-Card Binary, Card-to-Tape BCD, Card-to-Tape Binary, Card-to-Tape processing files containing both Binary and BCD records, and Tape-to-Printer. Only Tape-to-Printer may be blocked up to a maximum of 1000 characters.
4. Interrupt (switch I) allows interruption of processing to delete or activate additional functions after which the program continues governed by the new sense switch settings.

IBM Application & Systems Programs Library Abstract File Number 1401-UT-051

FILE UTILITIES

Abstract:

Purpose: 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 Disk, Disk to Tape, Card to Disk, and Card to Card.

Note: Coarse cards are used to specify the affected portions of the disk file.

Restrictions: The Tape to Disk and Card to Disk are companion programs to the Disk to Tape and Disk to Card programs, and are designed to load the data generated by these programs. Memory requirements are from 2K to 8K, depending upon which program is used.

Equipment Specifications: 1401 Model F, 1402, 1403, 1405, tape drives as required.

IBM Application & Systems Programs Library Abstract File Number 1401-UT-057

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 1401-1405 File Organization Program will efficiently load and maintain a chained disk file so 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 as to provide the most efficient program for any size 1401, 2) ability to make additions and deletions to a chained file, 3) lead and add trailer records to a disk, 4) unload a file onto 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 Program: The Load and Additions programs are used in conjunction with the edit program.

IBM Application & Systems Programs Library Abstract File Number 1401-UT-066

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.

Equipment Configuration:

1. 1401 Processing Unit (4000 core storage positions are minimum)
2. 1402 Card Read-Punch
3. 1403 Printer (Model 1 or 2)
4. 1405 Disk Storage Unit (Model 1 or 2)

IBM Application & Systems Programs Library Abstract File Number 1401-UT-065

Abstract:

Purpose: The 1400-1009 Utility transmits data to or receives data from another terminal or other cards or magnetic tape.

Use of Program: The four uses are:

1. Transmit data from cards - blocked or unblocked.
2. Transmit data from magnetic tape.
3. Receive blocked or unblocked data on cards.
4. Receive data on tape.

Equipment Configuration:

1. 1401 Processing Unit with 4000 or more Core Storage positions
2. Sense Switch special feature is required
3. 1402 Card Read-Punch
4. 1009 Data Transmission Unit
5. Serial I/O Adapter
6. Tape Unit (Model 729 II, 729 III, 729 V, or 7330) is optional
IBM Application & Systems Programs Library Abstract  
Number 1410-AU-106

1410 AUTOCODER

**Purpose:** The 1410 AUTOCODER replaces the user from writing his routines in machine language. He can write his routines using a well defined set of mnemonic operation codes in conjunction with useful and significant labels, which he defines, and then process them with Autocoder to produce an operating object deck. 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 Program:** 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 10 K storage.
2. Four IBM 729 IL, IV, or 7330 Magnetic Tape units.
3. An IBM 1402 Card Read-Punch.
4. An IBM 1403 Printer, model 2.

*Options are available to trade 1, 2, or 3 magnetic tape units for the 1022 and 1023 unit record devices.

IBM Application & Systems Programs Library Abstract  
Number 1410-CB-012

1410 COBOL PROCESSOR

**Purpose:** 1410 COBOL Processor accepts programs written in the COBOL 61 language as input and produces complete object programs to perform the functions specified in the source statements.

**Use of Program:** This 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.
2. An IBM 1402 Card Read-Punch, model 2.
3. An IBM 1403 Printer, model 2.
4. Four IBM 729 IL, IV, or 7330 Magnetic Tape Units (may be determined).

IBM Application & Systems Programs Library Abstract  
Number 1410-FO-013

1410 FORTRAN II PROCESSOR

**Purpose:** The 1410 FORTRAN II (FORMula TRANslating 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 1410. The FORTRAN processor that 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 this program are:

1. 20,000 positions of core storage.
2. One 1402 Card Read-Punch, Model 2
3. One 1403 Printer, Model 2
4. IBM 729 IL, IV, or 7330 Magnetic Tape Units (may be determined).

---

IBM Application & Systems Programs Library Abstract  
Number 1410-AT-002

1410 PAT UTILITY SYSTEM (10/20K)

**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. In addition, the automatic testing facility, the PAT system provides a number of 1410 card, tape and disk utility programs.

**Use of Program:** At the direction of the user and under control of a PAT program, the PAT routines are arranged on a PAT tape in conjunction with the programs to be tested on. The routines and programs are arranged in the order they are to be executed. Testing the program merely requires the loading of the PAT tape and an identification card for each program to be tested.

**Machine Configuration:** The 1410 PAT System requires on IBM 1410 System with the following minimum configuration:

- 10,000 positions of core storage
- IBM 1402 Card Read-Punch
- IBM 1403 Printer, Model 2
- IBM 729 IL, IV, or 7330 Magnetic Tape Units on Channel one (1)

---

IBM Application & Systems Programs Library Abstract  
Number 1410-AT-126

1410 PAT UTILITY SYSTEM (40K)

The 1410 PAT System comprises a series of program testing routines and utility programs that, at the discretion of the user and under control of the PAT programs, are arranged in conjunction with the program to be tested on a PAT tape.

**Use of Program:** The source symbolic program is set up in a prescribed manner and is operated on by the PAT System deck.

**Machine Configuration:** The machine configuration required for the use of the program are:

1. Minimum of 20 K storage.
2. Four IBM 729 IL, IV, or 7330 Magnetic Tape Units.
3. An IBM 1402 Card Read-Punch.
4. An IBM 1403 Printer, model 2.

*Options are available to trade 1, 2, or 3 magnetic tape units for the 1022 and 1023 unit record devices.

IBM Application & Systems Programs Library Abstract  
Number 1410-AU-008

1410 BASIC AUTOCODER

**Purpose:** The 1410 Basic Autocoder relieves the user from writing his routines in machine language. He may now write his routines using a well defined set of mnemonic operation codes in conjunction with useful and significant labels, which he defines, and then process them with Autocoder to produce an operating object program.

**Use of Program:** The source symbolic program is set up in a prescribed manner and is operated on by the Basic Autocoder Processor.

**Machine Configuration:** The machine configuration required by the Basic Autocoder processor is:  
1. Minimum of 10,000 core locations.
2. One 1402 Card Reader-Punch.
3. One 1403 Printer.
IBM Application & Systems Programs Library Abstract  
File Number: 1410-IO-910

1410 INPUT/OUTPUT SYSTEM FOR 1405 DISK STORAGE

Abstract:

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

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

Machine Configuration: The machine configuration required by the 1405 Disk IOCS 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-IO-911

1410 INPUT/OUTPUT SYSTEM FOR 1406 DEK STORAGE

Abstract:

Purpose: The 1406 Disk IOCS provides several macro-instructions and related routines that handle the scheduling of 1406 input and output operations for random and/or sequential processing.

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

Machine Configuration: The machine configuration required by the 1406 Disk IOCS is:
1. Minimum of 20K storage
2. 1406 Disk storage
3. Processing Overlap and Priority special features.

IBM Application & Systems Programs Library Abstract  
File Number: 1410-PR-106

PROCESSOR OPERATING SYSTEM TAPE

Abstract:

Purpose: This is a systems tape containing the following 7 programs:
1410-SY-907 System Supervisor
1410-AD-900 Autocoder
1410-DX-900 I/OCS Card/Type
1410-DX-911 I/OCS 1405 Disk
1410-RG-910 Report Program Generator
1410-GB-911 CCSDS II
1410-FV-915 FORTRAN II

IBM Application & Systems Programs Library Abstract  
File Number: 1410-RG-910

1410 REPORT PROGRAM GENERATOR (CARD/TAPE/1400 - DEK 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 (see only) and performing calculations at any point in the reporting process. RPG-generated programs utilize the 1410 IOCS.

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...
2. For Autocoder (to assemble the report program) - 1410 system...
3. For the report program (to produce the report) - 1410 system...
4. 20 K storage... 1402 Card Read Punch... other I/O units appropriate to the program.

IBM Application & Systems Programs Library Abstract  
File Number: 1410-IO-106

SIMULATION OF THE IBM 650 ON THE IBM 1410

Abstract:

Purpose: The 1410 RPG condensed deck accepts specifications and produces a symbolic deck in Basic Autocoder for a report program. Processing is sequential, without allowance for overlap and priority, both in RPG itself, and in the generated report program. The latter can produce reports in a wide range of formats, extracting its data from a card file and performing calculations very much after the fashion of an IBM 650 Accounting Machine, save that multiply, divide and compare, in addition to more basic calculations, may be performed at any point in the total reporting process.

Use of Program: A control card and specifications cards must be placed at definite points in the RPG condensed deck. The standard card loader is used.

Machine Configuration: The 1410 Card RPG will handle card input and card-printer output only. Machine requirements are:
1. 10K storage
2. 1402 printer (allow 100 or 132 character positions)
3. The report program generated by RPG will have machine requirements dependent on the specifications provided. The minimum would be:
1. 10K storage
2. 1402 card/reader punch

IBM Application & Systems Programs Library Abstract  
File Number: 1410-OE-042

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 30,000 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 709 takes approximately 20 times longer than if the program was running on a 1410.

Equipment Specifications:
52,000 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-E-101

SIMULATION OF THE IBM 650 ON THE IBM 1410

(continued on next page)
Abstract:

**Purpose**: The 650 Simulation provides means to run 650 programs on a production basis on the 605, 1401. If the user requires a more detailed description on the program, he may obtain it by requesting the Simulation of IBM 650 on IBM 1410 Bulletins.

**Use of Program**: The 650 Simulation is to be entered into the 1410 along with control information indicating the system being simulated. Then the 650 program is run monitored through the 650 Simulator Program.

**Machine Configuration**: The machine configuration required by the simulation of IBM 650 on IBM 1410 program is:

1. Minimum of 40,000 core locations.
2. One 1620 Reader-Punch.

---

**IBM Application & Systems Programs Library Abstract**  
**File Number**: 1410-SS-110

**1410 SORT 10**

**Abstract**:

**Purpose**: Sort 10 is a generalized sorting program which employs from 1 to 5 IBM 1405 Disk Storage Units and the Processing Overlap and Priority Special Features. Input records can be either on tape or in disk storage and can be fixed or variable length, single or blocked. Output will be on tape in ascending order.

**Use**: A minimum of four 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 data characteristics and for the machine configuration.

**Machine Configuration**: Sort 10 requires an IBM 1410 Data Processing System with the following minimum configurations:

- 20,000 positions of core storage.
- 1 IBM 1405 Disk Storage Unit.
- 1 IBM 729 II, 729 IV, or 7330 Magnetic Tape Unit.
- IBM 1402 Card Read-Punch Model 2.

If storage size is 40K, Sort 10 will use the additional storage, when necessary, to increase the size of its input/output area and work area.

---

**IBM Application & Systems Programs Library Abstract**  
**File Number**: 1410-SS-112

**SORT/MERGE 11**

**Abstract**:

**Purpose**: Sort-Merge 11 is a generalized tape sorting and merging program designed to permit either the sorting or the merging of data or 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.

**Use**: 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 function 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:

- 20,000 positions of core storage.
- Processing Overlap and Priority Special Features
- 1 IBM 729 II, 729 IV, and/or 7330 Magnetic Tape Units (may be inter-mixed) if Sort/Merge 12 is to function as a Sort. (To perform a 3-way Merge, only three tapes are needed.)
- IBM 1402 Card Read-Punch Model 2.

---

**IBM Application & Systems Programs Library Abstract**  
**File Number**: 1410-SS-107

**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 System Tape, calls in the required phase 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 in the Processor Operating System Tape.

**Use of Program**: 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 20 K storage.
2. Two IBM 1329 II, IV, or 7330 Magnetic Tape Units.
3. IBM 1402 Card Read Punch.

The machine configuration for the individual programs on the Processor Operating System Tape are specified in the Abstracts of the programs. The 1410 Autocoder has the largest minimum requirement.

---

**IBM Application & Systems Programs Library Abstract**  
**File Number**: 1410-SS-106

**1410 UTILITY PROGRAMS**

**Abstract**:

**Tape File Generator A**, this program prepares unblocked tape files from variable-length card records.

**Tape File Generator B**, this program generates blocked and unblocked tape files from fixed-length card records.

**Tape Compare Program**, this program compares the contents of two magnetic tapes, each of which may 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 comparison 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 tape 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-tape reel can be duplicated.

**Snapshop Program**, the Snapshot Program is a program testing aid. It prints out in a specific 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. Subtitled characters are used in place of those not available on the user's 1405 Printer. Word marks are represented by the digit '1' printed above the character with which the word mark is associated.

---

**Continued on next column**
IBM Application & Systems Programs Library Abstract File Number 1410-UT-107

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 61 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 that 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 Specification 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 1405 Disk-to-Tape, Model 1 or 2
- 1 IBM S932, S934, or 7330 Magnetic Tape Unit

1410-1405 Tape-to-Disk Program
- 1 IBM S932, S934, or 7330 Magnetic Tape Unit

IBM Application & Systems Programs Library Abstract File Number 1410-UT-117

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 that 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 user's 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 25%) than the DISK-TO-TAPE utility program. This program is primarily written to be used in conjunction with the user's 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 to conjunction with a Machine Specification 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.

59
IBM Application & Systems Programs Library Abstract  File Number 1620-AT-013

1620 FLOW TRACE PROGRAM

Abstract:

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, typen the address of the branch instruction and the address to which it branches.

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

Storage Requirements: 631 positions of core storage. Program is relocatable.

Equipment Specifications: 1620 with paper-tape reader. No restriction on 1620 core storage (20K, 40K, 60K). Trace output via typewriter. Cannot be used on machines with Indirect Addressing feature.

IBM Application & Systems Programs Library Abstract  File Number 1620-AT-014

1620 SELECTIVE TRACE PROGRAM

Abstract:

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 the data fields as they are manipulated.

Method: Not applicable.

Restrictions, 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 with a BT or BTM and a BB instruction.

Storage Requirements: Program requires 2440 core locations. The small parameter table containing start trace & stop trace addresses is located at the end of the program and the additional storage required by the table will vary depending upon the number of parameters specified. The program is completely relocatable.

IBM Application & Systems Programs Library Abstract  File Number 1620-FO-003

1620 FORTRAN (Tape)

Abstract:

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 300. The program will process FORMAT statements.

Storage Requirements: Requires 20,000 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.

IBM Application & Systems Programs Library Abstract  File Number 1620-FO-002

1620 FORTRAN (Card)

Abstract:

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 300.

Storage Requirements: Requires 20,000 storage positions 1620.

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

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

FORTRAN WITH FORMAT

Abstract:

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 300. The program will process FORMAT statements.

Storage Requirements: Requires 20,000 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.

IBM Application & Systems Programs Library Abstract  File Number 1620-FO-005

FORTRAN PRE-COMPILE FOR PAPER TAPE

Abstract:

Purpose: This program detects and permits correction of errors in a FORTRAN source program before the object program is compiled. The Pre-Compile detects many of the more common programming errors in individual source statements, and indicates possible logical errors in the source program as a whole.

Storage Requirements: Requires 20,000 positions.

Equipment Specifications: 1620 CPU 1621 Paper Tape Reader

IBM Application & Systems Programs Library Abstract  File Number 1620-FO-006

FORTRAN PRE-COMPILE FOR CARD

Abstract:

Purpose: This program detects and permits correction of errors in a FORTRAN source program before the object program is compiled. The Pre-Compile detects many of the more common programming errors in individual source statements, and indicates possible logical errors in the source program as a whole.

Storage Requirements: Requires 20,000 positions.

Equipment Specifications: 1620 CPU

59
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 logical search technique applied to the stepping-stone method.

**Restrictions:** Problems sizes are dictated by the formula:

\[
6,000 + (161) (19 MODS + 19 + 90 MODC + MODS + MODC + MODC + 193 + MODC + 19) = 19040000
\]

where

- \( M \) = number of warehouses
- \( N \) = number of retailers
- \( MODS \) = maximum number of digits used to specify units
- \( MODC \) = maximum number of digits used to specify cost
- \( CORES \) = number of positions of core memory

**Typical sizes:**

- 40 x 50 with both MODS and MODC equal to 2 digit fields,
- 40 x 50 with MODS and MODC reduced to 3 digit fields, or
- 40 x 50 problem may be solved using 3 digit fields.

**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 13,000 to over 35,000. The total card input required approximately 2 1/2 additional min. while the output added another 1/2 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.

**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 fast approximation method.

**Restrictions:**

- There can be as many as 99 zones.
- The maximum number of jobs per zone is 91 to 99 depending on the average number of precedence jobs per job.
- The maximum number of cards to do jobs is 80.

**Timing:** A problem with 350 input cards and 150 jobs took about 3 minutes of computer time.

**Equipment Specifications:**

- Paper tape reader or card reader.

**IBM Application & Systems Programs Library Abstract**

**File Number:** 1620-LM-018

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

**Abstract:**

**Purpose:** This subroutine package can be used with 1620 FORTRAN with FORMAT, Tape System (Program #1620-FS-005) 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 at compilation or may be loaded separately prior to the execution of the object program. Messages are automatically typed during compilation and loading, indicating appropriate action by the user. This subroutine package 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-LM-023

**1620 HASH TOTAL PROGRAM**

**Abstract:**

**Purpose:** The purpose of this program is to determine quickly and to a high probability whether a duplicate 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 is read in, it is split into fields of twenty digits and lines each of these fields, in turn, is subtracted 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 408 to 1116 and 19850 to 19899. 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-ME-018

**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 in 8, 650. Also, corrections may not increase or decrease the length of any record.

**Storage Requirements:** Program is loaded into memory from 00400 to 02300. Each record to be duplicated is loaded from 02301. The program also uses an area of core storage, ending in 19999 and equal to the length of the record, as a dump area.

**Equipment Specifications:** 1620 with paper tape and 28K memory.

**IBM Application & Systems Programs Library Abstract**

**File Number:** 1620-ME-025

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

**Abstract:**

**Purpose:** This subroutine package can be used with 1620 FORTRAN with FORMAT, Card System (Program #1620-FS-005) 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 at compilation or may be loaded separately prior to the execution of the object program. Messages are automatically typed during compilation and loading, indicating appropriate action by the user. This subroutine package 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 read-punch and the Automatic Floating Point feature.

**IBM Application & Systems Programs Library Abstract**

**File Number:** 1620-LM-028

**1620 G0TRAN (Tape)**

**Abstract:**

**Purpose:** A relatively fast compiler for programs which will generally be executed only once.

**Method:** G0TRAN stores the compiled program in memory during compilation. The object program is then executed in an interpretive mode. No object tape 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 G0TRAN is a modified subset of FORTRAN, including the functional subroutines. Arithmetic statements are restricted to one arithmetic operation per statement.

(Continued on next page)
Data is handled in the form of 10-digit floating point numbers or 3-digit fixed point numbers. Input-output is the same form as FORTRAN.

Storage Requirements: Not given.

Rame Specifications: Basic 1620 Tape.

IBM Application & Systems Programs Library Abstract
File Number 1620-PR-001

1620 Symbolic Programming System - Tape I/O

Abstract:
Purpose: This program system compiles 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.

Method: Does not apply.

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

IBM Application & Systems Programs Library Abstract
File Number 1620-SP-001

IBM 1620 Symbolic Programming System - Card I/O

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

Restrictions, Range: The system can accommodate 313 labels.

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-002

IBM 1620 Symbolic Programming System - One-Pass Processor

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

Restrictions, Range: The system processes the following declarative operations: DB, DC, DBA, DSR, and DYN. A maximum of one hundred and fifty-five labels can be handled. Multiplication is not allowed in address arithmetic.

Equipment Specifications: The system is designed to operate on a basic 1620 with tape I/O.

Method: Does not apply.

IBM Application & Systems Programs Library Abstract
File Number 1620-SP-003

IBM 1620 Symbolic Programming System - Card I/O

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

Restrictions, Range: The system can accommodate 313 labels.

Equipment Specifications: The system is designed to operate on a basic 1620 with card I/O.

Method: Does not apply.

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

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

IBM 1620 Symbolic Programming System - Tape I/O

Abstract:
Purpose: This programming system compiles 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.

Method: Does not apply.

Storage Requirements: Not given.

IBM Application & Systems Programs Library Abstract
File Number 1620-SP-005

IBM 1620 Symbolic Programming System - Card I/O

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

Restrictions, Range: The system can accommodate 313 labels.

Equipment Specifications: The system is designed to operate on a basic 1620 with card I/O.

Method: Does not apply.

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

IBM Application & Systems Programs Library Abstract
File Number 1620-SP-006

IBM 1620 Symbolic Programming System - Tape I/O

Abstract:
Purpose: This program system compiles 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.

Method: Does not apply.

Storage Requirements: Not given.

IBM Application & Systems Programs Library Abstract
File Number 1620-SP-007
SPS is an extension of 1620 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:

1 \times 10^{-100} \text{ to } 1 \times 10^{+100}

For variable length subroutines, the fractional part of the floating point number may vary from 9 to 65 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 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: 1710-SP-002

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)
- 6 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 (AOC/TAS 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. Process Interrupt Indicators
11. 1621-1624 (Paper Tape I/O)
12. 1625 (Card I/O)
13. Indirect Addressing
14. Additional Instruction (TNF-TNF-MF)
15. Store
16. 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 program:

- A 1620 or 1710 System with either paper tape or card I/O and those optional features acquired by the Source Language Program, such as the 1620 additional instructions or 1710 Random Addressing Feature.
PAT - PROCEDURE FOR AUTOMATIC TESTING

Abstract:

Purpose: The PAT System has been designed to standardize testing procedures so that they may be as efficient as a customer installation as they are in a

7070 Data Center with no change in test procedures.

The testing of a program by the PAT System is accomplished in three phases. The first phase is the creation of the data file by the Tape File Generation program. The second phase is the processing of the object program. The third

is the recording of the results of the test through the use of Storage Print and Tape Print programs.

PAT testing enables the processing of undebbuged programs by remote testing yet under programmer control. The results including the output from the

Utility programs would be returned to the programmer for data debugging.

The PAT System provides for the testing of programs by card or tape processing.

IBM Application & System Programs Library Abstract File Number 7070-AT-081

7070 AUTO-TEST GENERATOR SYSTEM

Abstract:

The Auto-Test Generator System provides a highly flexible and efficient method of creating tapes for automatic tape 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 5,000 words, 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 specifies 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 object program packet (the TD 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.

IBM Application & System Programs Library Abstract File Number 7070-AT-083

7070 BASIC AUTOCODER

Abstract:

The 7070 Basic Autocoder 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 Basic Autocoder 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 IOCS macro-instructions.

Machine Requirements: 4 tape units.

IBM Application & System Programs Library Abstract File Number 7070-AT-084

7074 AUTOCODER

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 Basic Autocoder 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 IOCS macro-instructions.

Machine Requirements: 4 tape units.

IBM Application & System Programs Library Abstract File Number 7070-AT-081

7074 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 F28-9531-I 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 de-

ferred:

Procedure 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 ROUND OPTION is standard; truncation is allowed.

(Continued on next page)
IBM Application & Systems Programs Library Abstract  File Number  7070-793

IBM COMMERCIAL TRANSLATOR

Abstract:
Program: 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 eighth tape or a card reader.

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

BASIC FORTRAN

Abstract:
Program: The IBM FORmula 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 used in the solution of many business problems which can be expressed in a mathematical formula.

Machine Configuration: 5,000 words of core storage
IBM 7500 Card Reader (Utility Panel)
IBM 7500 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.

IBM Application & Systems Programs Library Abstract  File Number  7070-2/4 FORTRAN LOADER

Abstract:
Purpose: The 7070/2/4 FORTRAN Loader provides users of 7070/2/4 FORTRAN and users of 7090/2/4 Basic FORTRAN with the principle of relocatability to insure that several routines can be compiled separately but used together at object time.

Use of Program: The 7070/2/4 FORTRAN Loader has been designed specifically to load the FORTRAN object programs, the 7090/2/4 FORTRAN Package, and the user's compiled subprograms, and sub­ routines (written in the FORTRAN or Autocoder language) to produce a relocated program (within storage or on some output medium) available for object time processing.

Machine Configuration: The 7090/2/4 FORTRAN Loader may be utilized with any of the following configurations:
1. 7070, 7070 or IBM 7074
2. Card oriented, Card/Tape or Tape oriented system
3. 6 or 10K Magnetic Core Storage
4. The Floating Decimal Arithmetic device is optional.

The program is adaptable to each user's requirements by changing the control information in the Loader. The 7070/2/4 FORTRAN Loader relocates itself into upper core storage as specified by the user. The Loader zeros itself out once all programs required for a particular object run have been relocated.

Capabilities and Limitations: FORTRAN object programs which are of such size that they overlap the Loader but which do not exceed core storage capacity, as defined by the user, may be executed by writing out the relocated program on some output medium. This is done through the use of an Alteration Switch. The relocated program should be read back into core storage with the IBM 7070/2/4 Converted Card Lead Program which, together with a zero storage program, is written out preceding the relocated program. Storage is zeroed up to the point indicated by the user in the Loader option.

This option is available to any program - regardless of size, but not exceeding core storage capacity. Programs which exceed core storage capacity are not executable and must be rewritten.

Under control of another Alteration Switch, the user has the option to type out a map showing the locations of programs and their data areas.

IBM Application & Systems Programs Library Abstract  File Number  7070-FO-090

FORTRAN 7070

Abstract:
Purpose: The IBM FORmula 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, completely or partially, 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:
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 7590 Card Reader (Utility Panel)
2. IBM 7550 Card Punch (Utility Panel)
3. IBM 7400 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: Programs may be compiled for any configuration of 7070 equipment. 7070/2/4 FORTRAN accepts all FORTRAN II features in a source program.

IBM Application & Systems Programs Library Abstract  File Number  7070-D2-076

SPOOL SYSTEM

Abstract:
Purpose: The SPOOL system provides two programs which may be run simultaneously with the main programs. This system provides tape-to-card, card-to-tape, and tape-to-printer operations. One or two of these operations may take place while the user's main program is running.
Restrictions: Operates in conjunction with 7070 IOCS.
Storage Requirements: 400 words + IOCS requirements.

**IBM Application & Systems Programs Library Abstract**  
**File Number:** 7070-50-004

**INPUT/OUTPUT CONTROL SYSTEM 7070**

**Abstract:**

**Purpose:** To provide users of the IBM 7070/2/4 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. Machine requirements at compile time are dictated by the specifications for the program which is 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 defined in the Input/Output Control System vary from 765 to 2100 words depending upon the number of files specified and the parameters in the DDICS 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 running of SPQOOL programs with programs using the Input/Output Control System.

**IBM Application & Systems Programs Library Abstract**  
**File Number:** 7070-50-005

**7070 DISK IOCS**

**Abstract:**

**Purpose:** To provide users of the IBM 7070/2/4 Data Processing Systems with routines for reading and writing 7300 Disk.

**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. Machine requirements at compile time are dictated by the specifications for the program which is 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 defined in the Input/Output Control System vary from 765 to 2100 words depending upon the number of files specified and the parameters in the DDICS 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 running of SPQOOL programs with programs using the Input/Output Control System.

**IBM Application & Systems Programs Library Abstract**  
**File Number:** 7070-50-006

**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:**
1. 7500 Card Reader and necessary I/O.

**IBM Application & Systems Programs Library Abstract**  
**File Number:** 7070-50-007

**COMPILER SYSTEMS TAPE**

**Abstract:**

**Purpose:** The 7070 compiler system provides Autocoder, Report Program Generator, FORTRAN, COBOL, Commercial Translator, and IOCS on a common tape for ease of usage.

**Equipment Specifications:** 6 magnetic tape units.

**IBM Application & Systems Programs Library Abstract**  
**File Number:** 7070-50-008

**REPORT PROGRAM GENERATOR 7070**

**Abstract:**

**Purpose:** Programs for writing reports from data on magnetic tape 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 7070/2/4 Autocoder. Input consists of the layout of the data tape, 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 779 Model II, IV, V, VI or 7330 tape units. 3. Channel 1 or Channels 1 and 2.

- Optional 1. IBM 7559 Card Reader (Utility Panel) 2. IBM 7559 Card Punch (Utility Panel) 3. IBM 7402 Printer 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 1 to 23 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 20 characters.

**IBM Application & Systems Programs Library Abstract**  
**File Number:** 7070-50-009

**SIMULATE 660 ON 7070**

**Abstract:**

**Purpose:** Programs written for the 650 (except 650 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 650 system for the program to be simulated. The simulation program was written for standard 650 systems.

**IBM Application & Systems Programs Library Abstract**  
**File Number:** 7070-50-010

**SORT 80**

**Abstract:**

**Purpose:** Tape files containing records from 1 through 9999 words in length can be sorted according to a control word that may have from 1 through 150 characters located in from 1 through 10 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 4 half reels of tape.

(Continued on next page)
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.

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

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 Merge 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-US-080

RAMAC UTILITIES

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: 7300 Disk Storage Unit
7500 Card Reader
750 Tape Units

IBM Application & Systems Programs Library Abstract  File Number 7070-US-081

7070 UTILITIES

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
Load Program Relocator
Zero Storage Program
Tape Mark Program
Tape File Generator Program
SNAPSHOT Program
Storage Print Program
Tape Print Program
Branch Trace Program
Tape Duplication Program
Unload Storage Program

IBM Application & Systems Programs Library Abstract  File Number 7070-US-085

UTILITY PROGRAMS FOR ADDITIONAL STORAGE

Abstract:

Purpose: This is a collection of 5 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.

Load 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 load 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 File Generator Program for Additional Storage: This program is used to write a tape file 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 7050-CV-080

INT580

Abstract:

Purpose: INT580 enables a program coded for an IBM 705 I, II or III will serial input/output equipment to operate on the IBM 7090, utilizing communication channels and 729 tape units. The 754, 756 I and II, 777, 778, 779 and 784 are simulated in memory. 721, 720A, 710A, 711, 712, 713 and 714 units are simulated on 729 tape units. Restrictions to full simulation are indicated in the detailed description of interpretation of each unit, starting at page 19 of the enclosed preliminary manual (as amended by 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, or leave there until that memory is reused for another application. Loading of an object program is initiated after INT580 housekeeping 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 7090, working with the Nonstop 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 7090 configuration of 40K memory 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 read/write simulation requires additional 80K memory if many sections are required. Four communication channels are required for efficient simulation of simultaneous PRW-WR operations on two TRC's.

IBM Application & Systems Programs Library Abstract  File Number 7050-CV-086

7050 I/OCS

Abstract:

Purpose: To provide the user a complete 7080 Input/Output control system. The 729 tape units and a means of obtaining two channel and minimal versions of this system.

Use: To see the 7080 I/OCs, the first file of the distribution tape should be punched out and a Processor librarian run should be made using the 7080 I/OCS. All programs using 7080 I/OCS 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 Ident in columns 75 to 80 of the cards, the new system tape, these cards. All programs using 7080 I/OCS should be assembled from the tape which have a "C" in column 74.

The deck with Ident IOCS502 will produce a complete system for two channels.

The deck with Ident IOCS805 will produce a minimal system for four channels. The checkpoint routine may be included by removing the change card which have a "C" in column 74.

The deck with Ident IOCS805 will produce a minimal system for two channels. The checkpoint routine may be included by removing the change card which have a "C" in column 74.

The deck with Ident IOCS805 and with a "D" in column 74 will produce an I/OCS to run with 79K and VTape. This deck may also be collated by index numbers in columns 1 to 5 with any of the three above decks.

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

The 7080 I/OCS 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 I/OCS program deck may be placed in front of the object program deck and loaded as one block.
2. The I/OCS Program deck may be loaded first and then the object program loaded.
3. The I/OCS 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 00 card produced by the processor should be discarded.

Machine Configuration: The 7080 I/OCS complete version for four channels will occupy memory locations 500 to 80,000 with erasable housekeeping occupying memory locations 20,000 to 24,996. The minimal system for 2 channels will occupy memory locations 500 to approximately 11,500 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 I/OCS Library.
2. The second file is the reassembly master for I/OCS65.
3. The third file contains 4 change decks.
4. The fourth file is the preassembled I/OCS80 deck.
5. The fifth file is the IOCS80 Listing.

Each file is preceded by a standard header and a tape mark.

IBM Application & Systems Programs Library Abstract  File Number 7050-CV-10-121

CSMRS

Abstract:

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

Use: The CSMRS program tape must be placed in a program tape, indicated to the 7080 I/OCS at the time of the loading of the object program. This tape will be unrolled and loaded by the checkpoint load control record, as previous 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 7080 program tape cards and will be loaded automatically if CS509 is indicated to 7080 I/OCS.

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 may be used approximately 60,000 memory positions. If the machine is 140K, the memory positions used will be 0 to 40,000 and 125,000 to 160,000.

IBM Application & Systems Programs Library Abstract  File Number 7050-33-114

IBM SORT 80 FOR 7080 UNDER SUPERVISORY CONTROL: 5B00SC

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" form C4A-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 79999 must be reserved for use by 5B00SC and 5B00SC 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 2.00 cpi density. The external label reads, "IBM Sort 80 for 7080 Under Supervisory Control; 5B00SC. Program Number 7050-33-114, Version 1, Modification Level 6." The first file, preceded by a standard IBM header label, contains the 5B00SC program deck, including INSTR command and DEFINE cards. This file may be used as input (Change Tape) to the 7080 Librarian. The second file is a listing of the 5B00SC 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: 5B00SC" form 526-6181.
3. INCL command card to be used on a master program tape for unmodified sort applications.
4. INCL-61 command card and dummy 00 TCO cards to be used on the master program tape for modified sort applications.
5. EXEC command card enabling loading of 5B00SC from the common program tape.
IBM Application & Systems Programs Library Abstract

**Abstract:**

**Purpose:** The NOSTP macro-instruction and a set of associated subroutines enable 700 and 7080 programs, running on the 7080, to utilize the non-stop operation feature of the machine. The use of these routines, in conjunction with the non-stop operation feature, will permit continuous operation of the 7080 in automatic mode.

**Additional Remarks:** When the 7080 is running in non-stop mode (i.e., interrupt mode with the non-stop switch or and is not in automatic interrupt mode), the non-stop operation feature would normally cause the 7080 to enter manual status. If this condition would normally cause the 7080 to cease output, you must use the corresponding switch for the interrupt word specified by the interrupt word 250. The conditions which will result in this automatic interrupt are:

1. Any halt instruction
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 250 would be the entry to those routines, and the automatic interrupt would transfer program control to them.

**Equipment Specifications:**

- 7080

IBM Application & Systems Programs Library Abstract

**Abstract:**

**Purpose:** To reduce the time and effort required to perform the set-up functions for "production" 7080 runs. SCG80 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 (used at object time) will contain a copy of the Memory Print (MP7080) at the beginning of each reel. This program will be placed at this location at 7080 user request to assist them when a production 7080 job encounters trouble.

SCG80 will also assist 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:** SCG80 provides:

1. A program library maintenance facility.
2. Ability to select "current" programs.
3. An Object Time Routine.
4. Data to be handled by SCG80 is normally supplied by the user and constitutes his programs, interspersed with SCG80 command cards.
5. Initially, however, data is being supplied as input to the first run. Input to the maintenance program is converted to a memory image of 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-EV-088. That program is obsolete and will not be distributed or maintained in the future. The Preliminary Reference Manual, IBM 7080 Supervisory Control System SCG80, dated September 1965, is also obsolete.

**Machine Configuration:**

- **A. The Library Maintenance Program**
  - Memory Size - 80K (minimum)
  - IBM 729 Magnetic Tape Units (minimum) Console Card Reader
- **B. The Production of a Correct Tape**
  - Memory Size - 80K (minimum)
  - 5 IBM 729 Magnetic Tape Units (minimum) Console Card Reader
- **C. SCG80 Object Time Routine**
  - Memory @ 80 to 8199
  - Plus 2700 characters beginning at a 0 or 5 locations above @499
  - 1 IBM 729 Magnetic Tape Unit (minimum) Console Card Reader

IBM Application & Systems Programs Library Abstract

**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 tapes. There is provision for attaching the input data to the correct data set and then processing through to an "End" card. On output, the data created at DA7080 may be of fixed or variable length, blocked or unblocked, multilined or single line files and labeled or unlabeled.

**Expanded Load Program (EL7080):** The expanded load program for the 7080 will be capable of locating a program record 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 5000 positions of memory and the lower 380 positions. If the input is from tape, the processing will be stopped by the reading of the next program card.

**Expanded Load Program (UL7080):** UL7080 provides for loading information between memory positions 000000 and 100000 on an 80K 7080 or between 000000 and 097999 on an 80K 7080. Otherwise, this program is the same as EL7080.

**Load Program (LD7080):** The Load Program for the 7080 will provide the following functions:

1. Clear Memory from 0460 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, 260, 251, 252, and 253 so as to prevent the machine from hanging following the loading operation due to an unanswered interrupt signal.
4. Modify itself to load an object program from any card reader or channel tape.
5. Load an object program into an 80K or a 160K 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 outputs of the alter switches, and memory from positions 000 through 100000. Memory areas may be defined as constant, instruction, and/or bit switch areas. The constant and instruction areas will be marked sequentially through 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 processing each segment to the 7080.
2. **Referencing** - The Referencing option allows for two functions:
   - 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.
   - 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 (FC7080):** The patch conversion program provides for the use of certain mnemonic operations when an expanded patch card is being punched.

**Data Conversion (DC7080):** This 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 unblocked records, reblocking blocked records, deblocking blocked records, and changing IBM standards for variable length records on files containing variable length records. Multilined and/or multi-reel tapes may be created and tapes may be duplicated by DC 7080.
Purpose: To facilitate the reduction of time and effort required to program commercial problems by providing a user to compose programs written in the Commercial Translator language, and to load and execute these programs.

Use: Commercial Translator, Version 3, is a subsystem of the IBM Processor, 7090-PR-130, operating under the control of the Basic Monitor (IBSYS). All input and output functions are performed through the 7090 IOCS system.

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

The following minimum configuration is required:
1. 37348 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 punch for punch output.

Abstract:

Purpose: The IOCS Version C is designed to relieve programmers of the necessity of writing input and output routines. Some of these subroutines add L-Language elements to the string without regard to their logical validity. The 7090 Input/Output Control System (IOCS) is used to transmit information from core storage to tape.

Use: IOCS Version C is used under the Basic Monitor Operating System. For example, reference should be made to the 7090 IOCS Reference Manual, SC8-4499-2.

Machine Configuration: IOCS Version C requires at least one tape unit (for the system tape), one on-line printer, and the Data Channel Trap.

Abstract:

Purpose: The D-Program consists of interdependent subroutines for writing L-Language string output. Some of these subroutines add L-Language elements to the string without regard to their logical validity. The 7090 Input/Output Control System (IOCS) is used to transmit information from core storage to tape.

Abstract:

Purpose: The L-Program consists of interdependent subroutines for writing L-Language string output. Some of these subroutines add L-Language elements to the string without regard to their logical validity. The 7090 Input/Output Control System (IOCS) is used to transmit information from core storage to tape.

IBM Application & Systems Programs Library Abstract File Number 7090-SM-920

SORT (709-Fixed Length)

Abstract:

Purpose: To sort and/or merge signed or unsigned binary and BCD files in logical or algebraic sequence.

Use: The 7090/7091 Sort is run under control of the IBSYS operating system. Instructions are supplied to the program via control card statements. The formats for these statements, details of their preparation, and instructions for loading and operating the system are explained in the 7090/7091 Sort bulletin, J2B-6217.

Machine Configuration: The program operates on a 32K machine. It requires a minimum of two channels and five magnetic tape units, two of which must be on the same channel. (The system tape must be on A1.) Additional tape units can be utilized to provide up to a 10-way merge. An on-line printer is necessary; an on-line card reader is optional.

IBM Application & Systems Programs Library Abstract File Number 7090-SM-920

IBSFAP

Abstract:

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

Use: IBSFAP is used under the Basic Monitor Operating System. For example, reference should be made to the Fap Supplement #F2B-1226.

Machine Configuration: 7090/7094 IBSFAP 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 card reader for input.
5. One tape or card punch for punched output.
6. One tape for printed output.
7. Two tapes for work tapes.

IBSFAP works under IBSYS and thus will obtain its tape units from IBSYS.

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

7090 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 effect a substantial time saving in computer operation, and will allow greater flexibility in programming.

Use of Program: Basic Monitor, IBSYS, provides:
1. An Editor routine to modify, add, and/or delete programming systems to satisfy the requirements of any user.

(Continued on next page)
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. IBSYS makes it possible to have system maintenance, assembly, and selection of current systems each passing information as needed to the next system to be executed. IBSYS control cards are used to obtain the desired results with the minimum of computer time.

A complete set of instructions on the usage of IBSYS is in the IBM 7090 Basic Monitor Manual J205-8286.

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 system tape.
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.

IBSYS 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 IBSYS control cards as needed.
IBM 305 PROGRAM LIBRARY ABSTRACT

305 RAMACODER

Henry L. Coon

Direct Inquiries to: Henry L. Coon
IBM Corporation
320 Church Street
New York 13, 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.

IBM 305 PROGRAM LIBRARY ABSTRACT

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 Dick File uses general Purpose Control Panel

Additional Remarks: Timing - 45-70 seconds per station
OPTIMIZING PROGRAM
B. Gordon and A. Dalton
Equitable Life, New York
July 15, 1955

a) Automatically assigns optimum locations to the instructions and data of a program.
b) Does not apply.
c) Does not apply.
d) The program occupies approximately 500 storage locations in addition to 1216 locations for tables. Both input and output are one word per card.
e) Addresses may be left fixed or optimized. Addresses being optimized are 4 digit decimal numbers but are symbolic in the sense that they are assigned new optimum locations. A flow chart is included.
f) Minimum 650.

AN AUTOMATIC METHOD OF OPTIMUM PROGRAMMING
Elmer F. Shepherd
John Hancock, Boston, Mass.
April 8, 1955

a) Automatically assigns optimum locations to the instruction and data of a program.
b) Does not apply.
c) Does not apply.
d) The program occupies approximately 250 storage locations in addition to 1700 locations for tables. Both input and output are one word per card.
e) Addresses being optimized are written as a pseudo address in the 9000 series. Drum locations available to the optimizing program are indicated by manually removing the restricted addresses from a deck of 2000 cards numbered 0000 to 1999 and running those remaining through the 533 as part of the load deck. A flow chart is included.
f) Minimum 650.

SOAP
Texas Highway Department
Austin, Texas

a) SOAP is a modification of the original SOAP so that it may be used on a numeric 650.
b) Allows up to 900 symbolic addresses. Includes all the features of original SOAP.

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

a) 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.
b) Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c) Mathematical Method: Does not apply.
d) Storage Required: Both programs occupy most of the drum.
Speed: The edit program reads at the rate of 180 to 200 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.
Re relocatability: Not relocatable.
e) Remarks: This system will accommodate 60 regions 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.
f) IBM 650 System: One 533 required.
Special Devices: None required.

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

a) Purpose: This program is a modification of SOAP II which permits use of 300 general symbols throughout the program, plus an unlimited number of sets of 100 symbols used only in a particular section.
b) Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c) Mathematical Method: Does not apply.
d) Storage Required: Entire drum and immediate access storage.
Speed: Not given.
Re locatability: Not given.
e) Remarks: None.
f) IBM 650 System: One 533, IAS, and indexing registers.
Special Devices: Group II special character devices are required.
IBM 650 Library Program Abstracts

NO SOAP

G. M. Clemence
IBM Applied Science
Madison, Wisconsin

a. Purpose: NO SOAP is a Numerically-Operated Symbolic-Ortho-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.
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: Same as Basic SOAP IIA.
e. Remarks: NO SOAP is similar to SOAP II in its design and operation; however,
only numerical symbolic addresses are used.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

A MODIFIED SOAP IIA FLOATING POINT PACKAGE FOR THE IBM 650

E. Vernon Griffith
IBM Applied Science
Madison, Wisconsin

a. 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.
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: Same as Basic SOAP IIA.
Speed: Same as Basic SOAP IIA.
Relocatability: Same as Basic SOAP IIA.
e. Remarks: Has all the features of Basic SOAP IIA except that on reading a floating
point 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.
f. IBM 650 System: One 533 equipped with a total of 12980 contacts.

STANOSPYCE

Curtis E. Stevens
Standard Oil Company (Indiana)
Des Plaines, Illinois

a. Purpose: Using the 650 without the alphabetic device, this routine translates English
sentences into a symbolic program language.
(Continued on next column)
IBM 650 Library Program Abstracts

1401 ASSEMBLY ON THE 650 TAPE SYSTEM

Henry La Badie
U.S. Army Ordinance
Frankford Arsenal

a. Purpose: 1401 S. P. S. Assembly on the 650 Tape System

b. Ranae: None

c. Mathematical Method: None

d. Storage Required: 200 Words; 1995 to 1999; Locations 1951 to 1960 are used as the read band. Cards are loaded at 200 per minute.

e. Remarks:
1. Only mnemonic op codes.
2. Comments, DC and DCW Cards must have 11-X punch in Col. 75.
3. Above cards must have no invalid 650 punches in Cols. 8-32.
4. Signs in Col. 23 may not be used with a constant. The units position of the constant may be aligned.
5. All other 1401 S. P. S. Rules must be followed for this program.

f. IBM 650 System:
1. T. L. E.
2. Set Format
3. I Tape Unit
4. Index Registers
5. Both Alpha Devices
6. 12 Pilot Selectors
7. 6 Connectors
8. Rd Side - 2 Digit Selectors (or 1 digit and 1/2 time emitter, H extra pilot Sel. available)

---

FIVE-PER-CARD LOADING ROUTINE

J. M. Kibbee
IBM, Houston

1-1-56

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

b) Does not apply.

c) Does not apply.

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

e) Self-loading.

f) Minimum 650.

---

SEVEN-PER-CARD LOADING ROUTINE

J. M. Kibbee
IBM, Houston

IBM, New York

11/16/55

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

b) Does not apply.

c) Does not apply.

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

e) Self-loading.

f) Minimum 650.

---

EIGHT PER CARD LOADING ROUTINE

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

June 16, 1956

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

b) Does not apply.

c) Does not apply.

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

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

f) Minimum 650.
LD,  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 9.2.000.)
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.
Relocatability: Does not apply.
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 location of the links. This makes for flexibility in arranging subroutines, replacing subroutines, or adding to the lengths of modular sections of programming.
f. IBM 650 System: One 533 required.
Special Devices: None.

IBM 650 Library Program Abstracts
File no. 1.2.007
Utility Programs

LOAD DECK AUDITOR
T. S. Gemmell
Ohio Department of Highways
Columbus, Ohio

a. Purpose: These two routines load the seven word per card instruction card format using any band other than the 1900 - 1950 band as the location of the loading routine, and are used in the Ohio Department of Highways engineering programs. (See classification 9.2.000.)
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
Relocatability: Program is non-relocatable.
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 location of the links. This makes for flexibility in arranging subroutines, replacing subroutines, or adding to the lengths of modular sections of programming.
f. IBM 650 System: One 533 required.
Special Devices: None.

IBM 650 Library Program Abstracts
File no. 1.2.008
Utility Programs

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: The format is the same as most 7/card loaders. This program will load the output of "7/Card Punch," File Number 1.3.012.
f. IBM 650 System: One 533, IAS, and indexing registers are required.

IBM 650 Library Program Abstracts
File no. 1.2.009
Utility Programs

DEMP AND LOAD ROUTINE FOR IBM 650
(HGSP)
H. G. Vanderschu
Princeton University
Princeton, N. J.

a. Purpose: This routine is designed to load programs for the IBM 650. "HGSP".
b. Restrictions, Range: Does not apply.
c. Method: Does not apply.
d. Storage Requirements: 100 locations relocated by the symbolic terms G, Routine is in HGSP.
e. Remarks: Will clear one read band for unnecessary blanks. Therefore, two or more read bands are used, they must be free of blanks.
f. IBM 650 System: 650 with Index Registers.
INDEPENDENT TABLE LOADER

T/Sgt. J. D. Fry
Directorate of Statistical Services
Ft. Af cam Force Base, Florida

a. Purpose: Independent Table Loader - loads tables, permits origin 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.

IBM 650 System: Minimum 650.

SEVEN-PER-CARD PUNCH ROUTINE

D. W. Sweeney
IBM, New York

11-16-55

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

b) Does not apply.

c) Does not apply.

d) Storage required is 27 locations, 1950 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 1.3.002.

f) Minimum 650.

STORAGE DUMP

R. Haberman
G. E., Schenectady

January 29, 1956

a) Punches a specified block of storage, 6 words per card.

b) Does not apply.

c) Does not apply.

d) Storage required is 15 locations, 1900 to 1950, and 1961 to 1964. No speed information given.

e) The upper limit of the block being punched must be less than 1900. The block may be specified by a master card or entry may be programmed. If the number of locations being punched is not an even multiple of 6, additional storage will be punched to fill the last card with 6 words. The first card punched in a master card for use when these cards are loaded with L-2, see Technical Newsletter No. 4, pp. 50-52.

f) Minimum 650.
Remarks: The output of this program may be reloaded with the program, "7/Card Loader," File Number 1.2.009.

f. IBM 650 System: One 533, IAS, and indexing registers.
   Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

SEVEN/CARD PUNCH BY LARRY ZIRKLE
It was discovered that the program does not perform as indicated in the writeup under program entry.
A corrected relocatable deck and new listing are available upon request.
Listing and decks mailed on or after March 1, 1961 have been corrected.

650 LIBRARY PROGRAM ABSTRACT

FLOW TRACER

E. Doley
IBM, New York 5-15-56

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

b) Does not apply.

c) Does not apply.

d) Storage required is 80 locations and two successive bands should be designated as an assembly area for the routine. The symbolic deck contains 32 cards.

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

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

650 LIBRARY PROGRAM ABSTRACT

TRACING ROUTINE

D.W. Hagebarger
Bell Telephone Laboratories, Murray Hill, New Jersey July 27, 1956

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

b) Does not apply.

c) Does not apply.

d) Storage required is 150 locations, 1900 to 1949 (or 0600 to 0949). Tracing is at 100 card per minute.

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

f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT

SELECTIVE TRACING ROUTINE

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

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

b) Does not apply.

c) Does not apply.

d) Does not apply.

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

f) Minimum 650

650 LIBRARY PROGRAM ABSTRACT

SYMBOLIC TRACING ROUTINE FOR A 650 SYSTEM WITH INDEXING REGISTERS

D. J. Hall
Research Computing Center
Indiana University
Bloomington, Indiana

a. Purpose: This routine is designed to be assembled by SOAP II, along with an unrelated main program, in anticipation of utilizing tracing as an aid in debugging.

b. Range: Does not apply.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: 60 locations in addition to eight successive words of any punch area.

Speed: Tracing proceeds at 100 instructions per minute.

Relocatability: Not given.

e. Remarks: For each instruction traced a card is punched with the location of the instruction, the instruction itself, the contents of the distributor, upper and lower accumulators, and the contents of the three indexing registers. The location of the first instruction to be traced is set in the storage entry switches. A SOAP II symbolic deck listing with a sample absolute listing is included in the write-up.

f. 650 System: One 533 and indexing registers required.

Special Devices: Alphabetic device if SOAP II symbolic version is used.

650 LIBRARY PROGRAM ABSTRACT

GENERAL TRACING ROUTINE

J.W. Burgeson
IBM, Akron, Ohio

a. Purpose: This routine traces all instructions, or only those with a minus sign.

b. Range: Does not apply.

Accuracy: Does not apply.
IBM 650 Library Program Abstracts

MODIFIED SYMBOLIC TRACING ROUTINE

J. May
Hudson Laboratories
Columbia University
Dobbs Ferry, New York

a. 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, 001.
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 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.
e. Remarks: For each instruction traced, a card is punched with the location
of the instruction, 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.
f. 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, D. C.

a. 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.
b. Range: Will preset one to six tapes.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
Speed: Approximately that of tape reading.
Relocatability: May be relocated to any band.
e. 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.
f. 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, Ohio

a. Purpose: These routines write any number of programs on a single tape.
Any required program can be reloaded 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.
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: All routines are fixed.
c. Mathematical Method: Does not apply.
d. Storage Required: The maximum storage requirement for any routine is
0000 - 0049 plus the first ten locations of IAS and a read band.
Speed: Not given.
Relocatability: Not given.
e. Remarks: These routines will destroy instructions located in IAS and
indexing registers.
f. 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. Ballantyne
Crown Life Insurance Company
Toronto, Ontario

a. 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.
b. Range: Sorts on a single word index only. Program has two phases.
Phase I block 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.
c. Mathematical Method: Does not apply.
d. Storage Required: Requires bands 0450 to 1950 for the internal block
sorting in Phase I, and there are seventy-seven free locations between
0400 and 0449.
Speed: Not given.
Relocatability: Not given.
e. Remarks: None.
f. 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
National Biscuit Company
New York, New York

a. 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.
           (Form 32A-0415). The "High" and "Low" exits of the original comparison
           blocks have been interchanged.
   IBM 650 System: An IBM 650 system with four 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 program (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 other. 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 IAS and tapes.

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.
   d) Storage required is 8 locations, 1951 - 1958.
   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)
   b) Does not apply.
   c) Does not apply.
   d) The deck contains 47 cards. Output is 100 cards per minute.
   e) Self-loading. A trailer card placed at the end of the condensed deck makes
      it self-transferring.
   f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT  FILE NUMBER  1.6.009

ONE TO SEVEN CONVERTER

P. S. Herwitz
IBM, Washington

a) Converts single-word load cards to seven-per-card load cards which may
   be used with the seven-per-card loader, file number 1.2.002.
   b) Does not apply.
   c) Does not apply.
   d) Storage required is 37 locations, 0000 to 0035 and 1950. In addition, 25
      locations are used in the 1950 and 1950 bands for reading, punching, and
      loading. Cards read at 200 per minute and punch at approximately 26 per
      minute.
   e) Loading routine not included in listing.
   f) Minimum 650.

ERRATA  650 Library Program - File No. 1.6.009

"One to Seven Converter," by P. S. Herwitz

In the one-page listing appended to the detailed write-up for 1.6.009, instruction
number 29 (location 0028) should read:

65 0028  0030

instead of

65 0028 0039

This is a typographical error in the preparation of the listing; the program deck
is not affected.

April 1958, Bulletin 18 - 37

650 LIBRARY PROGRAM ABSTRACT  FILE NUMBER  1.6.011

SEVEN TO ONE CONVERTER

P. S. Herwitz
IBM, Washington

a) Converts seven-per-card load cards to single instruction load cards.
   b) Does not apply.
   c) Does not apply.
   d) Storage required is 8 locations 1951 to 1958. The 1950 band is
      used for a read area, punch area, and self-loading routine. Cards are
      punched at 100 per minute.
   e) Self-loading.
   f) Minimum 650
A PROCEDURE FOR USING SOAP WITH A NUMERIC 650

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

a) Enables SOAP to be used with a minimum 650 provided a 407 with summary punch is available.
b) Does not apply.
c) Does not apply.
d) Approximately 850 storage locations are required.
e) A SOAP deck is partially converted to 650 alphabetic code using the 407 and summary punch. This routine completes the conversion at which time the regular SOAP program performs the assembly. No special characters may be used for any part of symbolic addresses.

SOAP TO SEVEN

James D. Chappell
IBM, Washington

a) Will convert single instruction load cards to seven-per-card load cards. SOAP output cards may be converted immediately without removing special type cards. Only those locations from the FWA to the LWA are punched with the further provision that no output card shall begin with an unused location.
b) Does not apply.
c) Does not apply.
d) Uses entire 1950 band. Running time is approximately read and punch speed.
e) The 1.2.002 loader is punched along with the 1.6.001 stop number routine prior to punching the converted program deck. A 1.2.002 transfer card is the last card punched. No single instruction load cards can be processed (or loading into the area used by the 1.2.002 loader.

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

R. W. Klopfenstein
RCA Laboratories
Princeton, New Jersey

a) Designed as an adjunct to the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter No. 11.
b) Floats a fixed point number or fixes a floating point number. Rounds in the last place in both floating and fixing.
c) Not applicable.
d) Programmed for locations 001-049. (Note: Interpretive system proper occupies locations 1000-1999).

d) Storage required including tables is approximately 220 locations. Timing is approximately 100 cards per minute.
e) It is assumed that errors detectable by SOAP I have been corrected and that relocatable addresses are in the range 0000 - 0999. Only the first ten columns of the remarks field will be retained. A SOAP II symbolic deck listing and a four-per-card absolute deck listing are included.

IBM 650 Library Program

"SOAP to Seven," by J. D. Chappell

ERRATA

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 31-42."

a) This routine is a modification of the Trimble interpretive floating decimal system described in IBM Technical Newsletter No. 8. It is designed for the 650 installation equipped with the automatic floating decimal device to provide a compromise between rewriting infrequently used programs which incorporate the Trimble routine and inefficient machine utilization while running such programs.
b) Floating arithmetic.
c) Modification of methods in Trimble routine.
d) Uses 243 storage locations in a block of 390 locations. The routine is 11% faster than the Trimble routine with no recoding required.
e) None
f) 650 with automatic floating decimal device.

INTERPRETIVE FLOATING DECIMAL ROUTINE

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

a) Translates symbolic cards prepared for SOAP I into symbolic cards acceptable to SOAP II.
b) Does not apply.
c) Does not apply.
(Continued on next column)
IBM 650 Library Program Abstracts

DATES 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: 69 words plus 10 words for each time the subroutine is used.
e. Remarks: The earliest date must be used as the first date and the most recent date as the second date. 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.
Special Devices: None.

IBM 650 Library Program Abstracts

FIVE-PER-CARD CONDENSING ROUTINE

J. H. Cooper
R. B. Fraser
T. B. Green
Shell Oil Company
P. O. Box 2527
Houston, 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 650 drum locations are required for program and storage.
e. Remarks: The entire drum is available to object program since object program instructions, which overlap 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.
Special Devices: None.

IBM 650 Library Program Abstracts

MICROSERVICES UTILITY ROUTINES

C. S. Stevens
Standard Oil Company (Indiana)
Detroit, Michigan

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 and IAS to Minus 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.
Special Devices: None.

IBM 650 Library Program Abstracts

LOAD DECK GENERATOR

C. E. 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.

(Continued on next column)
IBM BSD Library Program Abstracts

STOP NUMBER DRUM AND IAS

J. B. Reid
Trans-Canada Air Lines
Quebec, Canada

a. Purpose: This program reads all drum locations (except 1951-1960) and IAS locations with: 01 aaaa 9999, where aaaa is the address of the location.
b. Range: Does not apply.
   Accuracy: Does not apply.
   Floating/Fixed: Does not apply.
   Mathematical Method: Does not apply.
d. Storage Required: There are two sections to the subject program. The first section is read into the last band and punches seven words per card for locations 0000-1950. The second section, if used, requires a second loading of the program to be punched. This section is read into the first two bands and punches two instructions per card for locations 1951-1999.
   Speed: Punching speed for both sections of the program is 100 cards per minute. Loading speed of the seven-per-card deck output is 200 cards per minute.
   Relocatability: Not given.
e. Remarks: None.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

Reference 1.6.027
Utility Programs

UNIVERSAL MEMORY DUMP AND CONDENSING ROUTINE

B. M. Taylor, Jr.
North Carolina State College
Raleigh, North Carolina

a. Purpose: This program dumps entire contents of drum, accumulator, and distributor as a numbered, self-reloading, self-starting, condensed re-entry deck of not more than 360 cards. Any operating program may be interrupted and dumped at any point; reloading the output automatically restarts the operating program at the point of interruption. An operating program beset with a validity error may be dumped and repaired for re-entry and restarted at the point of interruption. A program being debugged and beset with anomalies may be dumped and listed for inspection. A debugged ready-to-operate program may be condensed for permanent use, without reserving any special area on the drum for the condensing routine itself. The dump program is read into any single available read band of ten words, and does not disturb any other locations.
b. Range: Does not apply.
   Accuracy: Does not apply.
   Floating/Fixed: Does not apply.
   Mathematical Method: Does not apply.
d. Storage Required: Any read band - ten words.
   Speed: Not given.
   Relocatability: Relocatable.
e. Remarks: If operating program is stopped following division without reset (14), the upper accumulator will be restored with the sign of the lower. If invalid information (blank bits, etc.) is present on the drum, special steps may be taken.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

Reference 1.6.029
Utility Programs

CDGB

D. A. D’Esopo
P. H. Butterfield
Stanford Research Institute
Menlo Park, California

a. Purpose: This program permits the use of the command difference method of address modification in the SOAP language. This command difference editing technique can save initialization and modification instructions when it is used on a series of variable commands which have a common modification increment and which are modified as a group.
b. Range: Does not apply.
   Accuracy: Does not apply.
   Floating/Fixed: Does not apply.
   Mathematical Method: Does not apply.
d. Storage Required: This program requires 23 storage locations plus that needed for parameters.
   Speed: Not given.
   Relocatability: Not given.
e. Remarks: The 23-card symbolic deck can be punched from the listing included in the write-up.
f. IBM 650 System: One 533 required.
   Special Devices: Alphabetic device required.
IBM 650 Library Program Abstracts

Self-Checking Load Deck Generator

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

a. Purpose: With the 650 doing all the work, this program will produce, for any read area of the drum, a condensed load deck consisting of the following sections:

1. Drum servicing routine
2. Seven-per-card, self-checking load routine
3. Seven instructions per card, 0000-1950
4. Self-checking card, 0000-1950
5. Load routine erasing card
6. Two instructions per card, 1951-1999
7. Self-checking card, 1951-1999

Many zero locations are bypassed in producing the seven-per-card and two-per-card sections, reducing the size of the load deck. The entire output is loaded in the same order as punched with one console setting.

If loading stops with 01 2345 0789 in the program register, something is wrong with the load deck; cards are missing, or have been added or altered.

b. Range: Does not apply.

c. Accuracy: Does not apply.

d. Floating/Fixed: Does not apply.

(Continued on next column)

IBM 650 Library Program Abstracts

Matrix Translation A/O Transposition

R. L. Freeman
Portsmouth Naval Shipyard
Portsmouth, New Hampshire

a. Purpose: This program was designed to separate, translate, or transpose matrices. The matrix to be manipulated may be stored on the drum or in a form to be loaded by the standard four-per-card loader or the n-per-card loader (IBM 650 Library Programs number 1. 6. 051 or 1. 6. 052). The repositioned matrix is stored in cards in a form to be reloaded by the n-per-card loader. This program is written to prepare data output of one routine in forms suitable for use in other routines.

b. Range: Does not apply.

c. Accuracy: Does not apply.

d. Mathematical Method: Does not apply.

e. Storage Required: The entire drum is used.

f. IBM 650 System: One 533 required.


Special Devices: None required.
IBM 650 Library Program Abstracts

**MATRIX PACKAGE**

V. Kahan
W. G. Thorpe
V. Sears
L. E. Green
Computing Centre, University of Toronto
Toronto, Canada

**a.** Purpose: These programs are a compatible set of utility routines for many different configurations of 650 systems. They use standard console settings throughout. The routines range from those useful with basic machines through those which may be used with systems (e.g., RAMAC). They are useful both as program error-detection aids and utility programs. The routines included, and the LDAPAC number for each are:

<table>
<thead>
<tr>
<th>Routine</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LADPAC SOAP</td>
<td>1513</td>
<td>Includes in both SOAP relocatable and SOAP symbolic. An explanation of the numbering system used in identification of these routines is included, together with symbolic and absolute listings. Nextpage routines always include, as the first cards of the output, a routine to load that disk. This loader will operate from the same storage locations as the punchout. Most of the punchout and loader routines are written for the basic machine.</td>
</tr>
</tbody>
</table>

**b.** Ranges: 37 rows X 50 columns. Accuracy: Dependent on matrices being processed by matrix operations.

**Storage Required:** Dependent on size of matrices used.

**Speed:** Not given.

**Relocatability:** Not given.

**c.** Mathematical Method: The inversion subroutine uses Jordan’s Elimination Method.

**d.** Remarks: The package contains the following operations:
- 10 Type
- 20 Transpose
- 30 Add Transpose
- 39 Linear combination
- 49 Linear combination
- 51 Add Transpose
- 66 Row augmentation
- 71 Column augmentation
- 81 Pivot
- 91 Checksum output

**IBM 650 System:** Tape system consisting of one 535, indexing registers, one 727 Magnetic tape unit.

---

**LM 650 FORTRAN SYMBOL EQUVALENCE TABLE**

W. C. Compton
Arabian-American Oil Company
New York City, N. Y.

**a.** Purpose: This program automatically prepares SOAP II “EQUS” cards defining the storage locations of each non-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.** Ranges: Does not apply.

**Accuracy:** Does not apply.

**Floating/Fixed:** Does not apply.

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

**d.** Storage Required: Not given.

**e.** Remarks: All routines have been tested and put to use at the Los Angeles Data Processing Center. In addition to the routines, an extensive commentary is included to fully explain the standard procedures employed. A trace table is included to assist the customer in choosing the proper trace. Descriptions in detail of the LADPAC Utility Read/Punch panel (largely 80-80) and the LDAPAC 407 Online Print panel are included. Most routines will operate with only a load hub wired to column 1, or with a ten word print panel. Standard card formats are described. Floating point mathematical routines for the basic functions are included in both SOAP relocatable and SOAP symbolic. An explanation of the numbering system used in identification of these routines is included, together with symbolic and absolute listings.

Nextpage routines always include, as the first cards of the output, a routine to load that disk. This loader will operate from the same storage locations as the punchout. Most of the punchout and loader routines are written for the basic machine.

**f.** IBM 650 System: One 535 and indexing registers. Special Devices: Group II special character device required.

---

**IBM 650 Library Program Abstracts**

**LADPAC UTILITY ROUTINES**

Los Angeles Data Processing Center
Los Angeles, California

**a.** Purpose: These programs are a compatible set of utility routines for many different configurations of 650 systems. They use standard console settings throughout. The routines range from those useful with basic machines through those which may be used with systems (e.g., RAMAC). They are useful both as program error-detection aids and utility programs. The routines included, and the LDAPAC number for each are:

<table>
<thead>
<tr>
<th>Routine</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LADPAC SOAP</td>
<td>1513</td>
<td>Includes in both SOAP relocatable and SOAP symbolic. An explanation of the numbering system used in identification of these routines is included, together with symbolic and absolute listings. Nextpage routines always include, as the first cards of the output, a routine to load that disk. This loader will operate from the same storage locations as the punchout. Most of the punchout and loader routines are written for the basic machine.</td>
</tr>
</tbody>
</table>

**b.** Ranges: 37 rows X 50 columns. Accuracy: Dependent on matrices being processed by matrix operations.

**Storage Required:** Dependent on size of matrices used.

**Speed:** Not given.

**Relocatability:** Not given.

**c.** Mathematical Method: The inversion subroutine uses Jordan’s Elimination Method.

**d.** Remarks: The package contains the following operations:
- 10 Type
- 20 Transpose
- 30 Add Transpose
- 39 Linear combination
- 49 Linear combination
- 51 Add Transpose
- 66 Row augmentation
- 71 Column augmentation
- 81 Pivot
- 91 Checksum output

**IBM 650 System:** Tape system consisting of one 535, indexing registers, one 727 Magnetic tape unit.
a. Purpose: This package includes subroutines for ABSF, COSF, SINF, ATANF, SQRTF, EXPF, LGNF, ANTF, CLOGF.


Floating/ Fixed: Floating decimal arithmetic is used.

Mathematical Method: Standard iterative techniques are employed.

c. Storage Required: Not given.

Speed: Not given. Relocatability: Not given.

e. Remarks: All subroutines are in 5/card format.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

AUTOMATIC PERSONAL IDENTIFICATION CODE (AUTOPIC)

Jack Melnick
IBM - Trenton
25 West State Street
Trenton 8, New Jersey

a. Purpose: To numerically code alphabetic names of individuals and assign unique identifying data to each individual.

b. Range: Not applicable.

Accuracy: Expected accuracy of 85-95% alphabetic sequence with an expectancy of .01-.02 duplications.

c. Mathematical Method: Not applicable.

d. Storage Required: 17.27 words for tables; Z67 words for program, constants, and input-output areas; 6 words available.

Speed: 100 cards per minute. Relocatability: Non-relocatable.

e. Remarks: Limits of tables: 768 first names; 9590 last names broken into 10 phases of 959 words each.

f. IBM 650 System: Minimum 650 with alphabetic device.

IBM 650 Library Program Abstracts

SWCH SUBROUTINE FOR 650 FORTRAN

David L. Grobstein
Concepts and Applications Laboratory
Picatinny Arsenal
Dover, New Jersey

a. Purpose: This subroutine makes available to 650 FORTRAN a statement resembling the IF (SENSE SWITCH i) n 1, n instruction available in 704-709 FORTRAN.

b. Range: Does not apply.

Accuracy: Does not apply.

Floating/ Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: 28 drum locations

Speed: Varies from 10 to 65 milliseconds depending on the degree of optimization. (Continued on next column)

Relocatability: SWCHF is written in SOAP II and is used in symbolic form during 650 FORTRAN PASS II assembly. Available locations are assigned by the FORTRAN PASS II deck, and may be anywhere on the drum.

e. Remarks: The subroutine uses the rightmost three Storage Entry Switches on the 650 console to stimulate sense switches, and control program branching.

f. IBM 650 System:Same as needed for 650 FORTRAN.

IBM 650 Library Program Abstracts

UTILITY SUBROUTINES

George Rado
Daniel Saloff
New York University College of Engineering
University Heights
New York, N. Y.

a. Purpose: The package has the advantage of offering a system with uniform linkage, 4-character local addresses, and index-register preserving routines.

b. Range, Accuracy, Floating/Fixed: Not applicable.

c. Mathematical Method: Not applicable.

d. Storage Required: Does not apply.

e. Remarks: Does not apply.

f. IBM 650 System: 650 with Floating Decimal and Index Register.

IBM 650 Library Program Abstracts

GOCF II A

A. Wachowski
J. L. Overbey
Research Department
Automatic Electric Laboratories, Inc.
400 North Wolf Road
Northlake, Illinois

a. Purpose: This program with associated 533 and 407 control panels form a unified system of programmed input and output both in numeric and alphabetic form for the scientific use of the IBM 650.

b. Range, Accuracy, Floating/ Fixed: Not applicable.

c. Mathematical Method: Not applicable.

d. Storage Required: 177 locations.

Speed: Maximum read and punch speed. Relocatability: Not relocatable.

e. Remarks: The 533 Control Panel may also be used as a General Utility Board with 80-80 Read and Punch, as Load or Non-Load cards.


IBM 650 Library Program Abstracts

AUTOMATIC SOAP CONVERSION UTILITY PROGRAM (ASCUP)

T/Sgt. Robert D. Drury
5755 Hickam Drive
Dayton 31, Ohio

(Continued on next page)
a. **Purpose:** Program automatically converts sequentially coded 650 programs to strip HA input for optimization.

b. **Restrictions, Range:** Does not apply.

c. **Method:** Does not apply.

d. **Storage Requirements:** Load deck contains 164 cards - 100 card per minute output.

e. **Remarks:** Program must be reloaded for each program being converted.

f. **IBM 650 System:** Alphanumeric device necessary.

**IBM 650 Library Program Abstracts**

**BLOCK CORRELATION - COR2**

Numerical Computation Laboratory
Ohio State University Research Center
Columbus, Ohio

**Purpose:** COR2 will produce all the correlations for a block of variables, which are to be correlated with themselves or with another block of variables. Results include means, sums of squares, sums of crossproducts, mean, standard deviation, variance, covariance, correlation coefficient, and its square.

**Restrictions, Range:** Does not apply.

**Mathematical Method:** COR2 uses the following formula in the computations:

\[ r_{ij} = \frac{\sum (x_i - \bar{x})(y_j - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_j - \bar{y})^2}} \]

**Storage Requirements:**
- **Permanent locations:** 0000 and 1067 thru 1999.
- **Reserved for assembly:** 0001 thru 1066.

**Remarks:**
- **Accuracy:** Not given.
- **Translated correctly and every address referred to, but not used as a location, will be reserved when assembling. Other formats require hand checking in order to ascertain that they have been treated as intended.

If it is desired, a group of constants may be held fixed by preceding them with a load card containing all nines in the first word.

A post-SOAP and source word per card listing is included.

f. **IBM 650 System:** Minimum 650 equipped with alphanumeric device.

**IBM 650 Library Program Abstracts**

**SHIFT**

Richard E. Chandler
Research Computing Center
Florida State University
Tallahassee, Florida

**Purpose:** SHIFT is a FORTRAN I (1) subroutine designed to shift a fixed point number a desired number of places right or left (or both).

**Restrictions, Range:** Fixed point.

**Method:** Does not apply.

**Storage Requirements:** 37 locations plus 1454 and 1999 thru 1999.

**Remarks:** SHIFT 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.

**IBM 650 System:** Minimum 650 with alphanumeric and special character devices.

**IBM 650 Library Program Abstracts**

**TRANSLATOR - OTHER FORMATS TO SOAP RELOCATABLE (TYPE 2) DEC60**

W. H. Lowellen
D. L. Wainer
Ohio Department of Highways
Columbus, Ohio

(Continued on next column)
IBM 650 Library Program Abstracts

537 SIMULATOR GENERATOR

Q. J. Malloy
North American Life Assurance Co.
Toronto, Ontario, Canada

a. Purpose: Generates an SOAP II input card format subroutine for use within a program. The subroutine generated, after assembly within a program will simulate the SOAP II program, the operation of a 537 input-output unit to the extent of punching the output on the input cards. Misalignments between reading and punching are detected.
b. Range: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: 650 Set up for SOAP II, reads at 40.50 cards per minute.
e. Remarks: This diagnostic will not check all possible errors (i.e. transcribing and keypunching) but will provide a fairly thorough check for the most common errors. The program is open ended and future plans include checking more of the statements and programs in this system.

IBM 650 Library Program Abstracts

560 DIAGNOSTIC

T. L. Yates
Oregon State Highway Department
Salem, Oregon

a. Purpose: A program to detect irregularities in IBM 650 routines.
b. Range: Does not apply.
c. Mathematical Method: Does not apply.
e. Remarks: Input to this program consists of load cards in the SOAP output format. Output consists of 30 columns of alphabetic from punch words 1-6.
f. IBM 650 System: Minimum 650 with alphabetic device.

IBM 650 Library Program Abstracts

565 FORTRAN EDITOR

Jon Pegg
IBM Advanced Systems Development
Monterey, California

b. Range: Does not catch all errors.
c. Mathematical Method: Does not apply.
d. Storage Required: Speed about 100 cards per minute.
e. Remarks: None.
f. IBM 650 System: IBM 650, indexing registers, alphabetic device.

IBM 650 Library Program Abstracts

FORSCAN

G. A. Irvine
Monte G. Smith
Continental Oil Company
P. O. Drawer 1257
Ponca City, Oklahoma

a. Purpose: This routine will scan a program written in the "650 FORTRAN" language and will examine the program for forty-seven types of errors. These errors fall into four 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: 1849 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 processing cycle.
f. IBM 650 System: Minimum 650.

IBM 650 Library Program Abstracts

GENERAL PURPOSE 407 CONTROL PANEL

Robert C. Heasing
Citizen's Service Research and Development Company
900 East Third Street
Tulsa 20, Oklahoma

a. Purpose: This control panel allows the 407 user to list all card formats which are in normal 650 programming and data processing. FORTRAN, (Continued on next page)
SOAP, and machine language processing (see below). FORTRAN statement cards, data cards, answer cards, SOAP instruction cards, machine language cards, and five per card punched decks are examples of formats which may be printed. In addition to the above, any title of 32 characters (or less) may be stored and subsequently printed on the first line of each form.

b. Range, Accuracy, Floating/Fixed: Does not apply.

c. Mathematical methods: Does not apply.

d. Storage: Does not apply.

e. Remarks: Standard 407 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.

Cards must contain identifying punch where necessary.

f. Equipment specifications:

1) Standard 407 accounting machines (56 co-selectors, 15 pilot selectors, and 2 digit selectors) allow printing of all card formats (except FORTRAN statement cards).

2) 407 accounting machines equipped with 16 additional co-selectors, 5 additional pilot selectors, and 2 additional digit selectors allow printing of all card formats (except FORTRAN statement cards).

IBM 650 Library Program Abstracts

SIR: SOAP INTERPRETIVE ROUTINE*  
B. G. Oldfield  
W. Hemmerle

IBM, New York

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

b. Range: Does not apply.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: The program is separated into 9 sections and only those required for a particular problem need be assembled. Storage for individual sections varies from 11 to 184 locations.

Space: Timing is a function of the operation being performed.

Relocatability: Relocatable SOAP program cards are available.

e. Remarks: Included, in addition to the arithmetic operations, are trace, print, file, square root, sin, cos, tan, and arctan. Entry and exit from the interpretive routine are at the discretion of the programmer.

The program is available from the Program Library in 3 forms: an absolute 7-per-card condensed deck a symbolic deck in SOAP I format a symbolic deck in SOAP II format

Modified SOAP I and SOAP II decks are also available from the library and must be used in assembling the SIR symbolical decks. If possible, use of the condensed deck is advised.

f. 650 System: One SIR required.

Special Devices: Alphabetic device necessary.

*This abstract, which has been revised to reflect the current status of the system, should be substituted for the existing abstract for 2.001.

IBM 650 Library Program Abstracts

"SIR: SOAP Interpretive Routine," by B. G. Oldfield and W. Hemmerle

The original SIR write-up has been rewritten by Dr. J. A. Kazar and Mrs. Helga Sharebas, 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)

Copies of the new write-up are available (either separately or combined with the original report) from the IBM 650 Program Librarian.

IBM 650 Library Program

FILE NUMBER 2.0001

ERRATA


In the SOAP listing of the compiler the following changes should be made:

Card No. Should read
1. 0141 SUP A0001 0189 11 0383 1137
2. 0141 STU NEWAB 1137 21 0665 0987
3. 0601 BS LDD DROPU 0987 09 0690 0893
4. 0603 RAL NEWAB 0690 65 0645 0298
5. BS03 NEA BSA 0298 45 0786 0406
6. 0606 STL A0001 BSA 1695 20 0383 0766
7. 0607 BSA BAU N BNI 0786 60 0694 1059
8. 0650 LDD LOB 1611 09 1377 3556

The above changes are corrections to the compiler and do not represent misprints in the listing. Changes 1 - 7 are necessary since the compiler, as distributed, would incorrectly erase an entry in the acon table at every time a floating point constant with a negative exponent was compiled, regardless of whether the exponent had previously been stored as a constant. Changes 8 and 9 are necessary to make room for the insertions.

The above changes have been made in all decks supplied on or after June 1, 1958.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 2.0003

MITHILAC

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

MIT, Boston

September, 1955

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

b) Does not apply.

c) Does not apply.

d) The complete routine requires all but 390 locations 0010 to 0399. This amount may be increased to approximately 900 by not using all the features of MITHILAC. Timing is a function of the operation being performed.

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

f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 2.0003

COMPLEX ARITHMETIC INTERPRETIVE ROUTINE

Tsun H. Lee

Detroit Edison, Detroit

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

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

c) Does not apply.

(Continued on next page)
A floating-point interpretive routine using 5 digit instructions so that program changes are very easy to make.

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

c) Does not apply.

d) Storage required for the interpretive system is 1100 locations, 0900 to 1999.

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

f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 2.0.006

NINE OPERATION SPLIT INSTRUCTION ROUTINE: NOSIR

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

August 3, 1956

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

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

c) Does not apply.

d) Storage required is in 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.

ERRATA

650 Library Program - File No. 2.0.008

"General Purpose System for the 650: L2" by R. W. Hamming and Miss R. A. Weiss

An error has been discovered in certain copies of the L2 program deck furnished to 650 users. In the main deck, column 18 of card 30 should contain a zero punch; in the incorrect copies, this column is blank.

It is recommended that all copies of this deck be examined and, if necessary, corrected. L2 decks furnished by the 650 Program Library on or after March 5, 1958, have been corrected.
ERCO FLOATING DECIMAL POINT SUBROUTINES

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

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

b) Range: \(-10^{50} < X < 10^{50}\). Accuracy: 6 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.

---

DOPHIR: DOUBLE PRECISION FLOATING POINT SOAP

INTERPRETIVE ROUTINE

Hebron E. Adams
IBM, Washington

January 2, 1957

a) DOPHIR is both a system of coding in 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: \(10^{-29} \text{ to } 10^{+50}\). Accuracy: 18 places. Floating point.

c) Conventional floating point methods.

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

e) DOPHIR is, in most ways, analogous to IR, and all IR operations are included in DOPHIR. In addition, such features as interpretive floating decimal addition and decimal and floating decimal in floating decimal commands, are implemented in interpretive tracing system, and an addressable pseudo-accumulator are included. Inasmuch as DOPHIR 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 sent into a complex mode of operation by a special command; however, previous results and looped operations are preserved, and there is also a complex move; all other operations send the 650 back to the usual mode. Complex numbers are stored in two floating decimal parts in successive registers.
IBM 650 Library Program Abstracts

SIMULATION OF AN INDEXING REGISTER IN SIR
B. Leavenworth
American Machine & Foundry Company
Greenwich, Connecticut

a. Purpose: This program is a modification in SIR ("SOAP Interpretive Routine", File No. 2, 0, 001) to simulate an indexing register.
b. Range: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: Requires 47 storage locations.
e. Remarks: Must be loaded after the SOAP deck. Should be used with non-standard SOAP II deck.
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. Schorre
E. I. Organick
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.

b. Range: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: Does not apply.
e. Remarks: The card formats for a WIT program and its associated data files are identical to those specified for the corresponding WIT system. For this reason, it is possible to check out programs using the TRACE mode of the interpretive system before translation by WIT.

(Continued on next page)
IBM 650 Library Program Abstracts

File No. 1. 0. 011

FULLY FLOATING INTERPRETIVE COMPATIBLE OPERATION ROUTINE

S. I. Schlesinger

Aeronutronic Systems Incorporated

a. Purpose: This routine is a special interpretive system designed for use in the process industry.

b. Restrictions, Range: Does not apply.

c. Method: Does not apply.

d. Storage Requirements: 1350 drum locations are available for interpretive programs intended for use with the IBM 630 equipped with floating decimal device and indicating 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.

e. Remarks: Tagging for address modification is interpreted for the data locations only of the instruction word. The subroutine (arithmetic) mentioned are independent of the main routine in operation, and may be assembled separately.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

File No. 2. 0. 022

ID-3 INTERPRETIVE SYSTEM

Bonner and Moore Engineering Associates

a. Purpose: This routine is a special interpretive system designed for use in the process industry.

b. Restrictions, Range: Does not apply.

c. Method: Does not apply.

d. Storage Requirements: 1550 drum locations are available for interpretive instruction.

e. Remarks: The ID-3 system is used to write the executive program for the Bell Operations Simulator. Operation code of ID-3 are of the type that greatly reduce the programming time for the Process engineer.

f. IBM 650 System: Basic 650 is required.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 2. 1. 001

INTERNAL TRANSLATOR (ITI)
A COMPILER FOR THE 650

A. J. Perlis
J. W. Smith
K. B. Van Zwaren
Carnegie Institute of Technology, Pittsburgh 13, Pa.

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

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

c) Does not apply.

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

e) The SOAP I type programs produced are assembled by a modified SOAP I deck which output in a machine language program which produced 3 words/card. These machine language programs require, during operation, an auxiliary package of subroutines which include floating point, input-output, and optional logarithm, power and exponential routines. Depending on the option, these packages require from 270 to 500 locations. The remainder of the drum is available for program and data. A general technique may be used to incorporate additional subroutines.

The system includes a programming manual, 503 wiring diagram, the translation program, the modified SOAP I program, subroutine packages, and sine, cosine, and square root floating point subroutines.

1) Alphabetic device is required.
MODIFICATIONS OF THE INTERNAL TRANSLATOR (IT) COMPILED FOR USE OF SPECIAL CHARACTERS

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

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 Parentheses</td>
<td>(</td>
</tr>
<tr>
<td>Right Parentheses</td>
<td>)</td>
</tr>
<tr>
<td>Equality (substitution sense)</td>
<td>=</td>
</tr>
<tr>
<td>Comma</td>
<td>,</td>
</tr>
<tr>
<td>Addition</td>
<td>+</td>
</tr>
<tr>
<td>Division</td>
<td>/</td>
</tr>
<tr>
<td>Decimal Point</td>
<td>.</td>
</tr>
<tr>
<td>Exponentiation</td>
<td>^</td>
</tr>
<tr>
<td>Negation</td>
<td>-</td>
</tr>
</tbody>
</table>

A sample statement would appear as below:

\[ Y2 = (3 \times Y1) - ((2.05 + C) (12 + 10)) / 5.82 \]

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.

---

IBM 650 Library Program Abstracts

FILE NUMBER 2.1.002

**IBM 650 Library Program Abstracts**

**FILE NUMBER 2.1.002**

**BIT 2**

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

a) Programs written as a sequence of statements in IT language (see 650 Abstract 2.1.001) are translated directly into machine language represented in standard 5 instructions/card form.
b) Same as 2.1.001.
c) Does not apply.
d) The translator requires the entire drum. Output is approximately 50 cards per minute (100 instructions per minute).
e) The machine language programs produced require, during operation, an auxiliary package of subroutines which include floating point, input-output, and optional logarithm, power and exponential routines. Depending on the option, these packages require from 270 to 500 locations. The remainder of the drum is available for program and data. A general technique may be used to incorporate additional subroutines.

The system includes the translation program, relocation routine and subroutine packages, and associated function subroutines.
f) Alphabetic device is required.

April 1958, Bulletin 18 - 13

---

**IBM 650 Library Program Abstracts**

**FILE NUMBER 2.1.003**

**IT - 2**

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

a) Programs written as a sequence of statements in IT language (see 650 Abstract 2.1.001) are translated directly into machine language represented in standard 5 instructions/card form.
b) Same as 2.1.001.
c) Does not apply.
d) The translator requires the entire drum. Output is approximately 50 cards per minute (100 instructions per minute).
e) The machine language programs produced require, during operation, an auxiliary package of subroutines which include floating point, input-output, and optional logarithm, power and exponential routines. Depending on the option, these packages require from 270 to 500 locations. The remainder of the drum is available for program and data. A general technique may be used to incorporate additional subroutines.

The system includes the translation program, relocation routine and subroutine packages, and associated function subroutines.
f) Alphabetic device is required.

April 1958, Bulletin 18 - 15
IBM 650 Library Program Abstracts

IBM 650 FORTRAN MODIFIED FOR THE 4000 WORD 650

Dr. H. Klein
Max. Ann. Miller
Union division
AVCO Corporation
Stratford, Conn.

Purpose: To provide a FORTRAN system for the 4000 word 650. The system consists of two major parts:
1. The compiler, 650 FORTRAN, which accepts FORTRAN statements and compiles 650 instructions in symbolic SOAP II language.
2. The assembler, a modified version of SOAP 5A-4000, which produces an optimized machine language program from the symbolic instructions.

Range: Does not apply.

Mathematical Method: Does not apply.

Storage Required: Not given.

Reliability: Not given.

Remarks: The translation is accomplished in a single pass and the resulting program is produced on fixed-point format cards. Intersections called for by the source program are selected by means of a symbolic linkage and located at the time of execution.

IBM 650 Systems: One 535, automatic floating decimal arithmetic feature and indexing registers are required.

Special Devices: Group II special character device is required.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: Not given.

e. Speed: Not given.

Floating/Fixed: Does not apply.

IBM 650 FORTRAN MODIFIED FOR THE 4000 WORD 650

Dr. H. Klein
Max. Ann. Miller
Union division
AVCO Corporation
Stratford, Conn.

Purpose: To provide a FORTRAN system for the 4000 word 650. The system consists of two major parts:
1. The compiler, 650 FORTRAN, which accepts FORTRAN statements and compiles 650 instructions in symbolic SOAP II language.
2. The assembler, a modified version of SOAP 5A-4000, which produces an optimized machine language program from the symbolic instructions.

Range: Does not apply.

Mathematical Method: Does not apply.

Storage Required: Not given.

Reliability: Not given.

Remarks: The translation is accomplished in a single pass and the resulting program is produced on fixed-point format cards. Intersections called for by the source program are selected by means of a symbolic linkage and located at the time of execution.

IBM 650 Systems: One 535, automatic floating decimal arithmetic feature and indexing registers are required.

Special Devices: Group II special character device is required.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: Not given.

e. Speed: Not given.

Floating/Fixed: Does not apply.

IBM 650 Library Program Abstracts

IBM 650 Library Program Abstracts

IBM 650 FORTRAN MODIFIED FOR THE 4000 WORD 650

Dr. H. Klein
Max. Ann. Miller
Union division
AVCO Corporation
Stratford, Conn.

Purpose: To provide a FORTRAN system for the 4000 word 650. The system consists of two major parts:
1. The compiler, 650 FORTRAN, which accepts FORTRAN statements and compiles 650 instructions in symbolic SOAP II language.
2. The assembler, a modified version of SOAP 5A-4000, which produces an optimized machine language program from the symbolic instructions.

Range: Does not apply.

Mathematical Method: Does not apply.

Storage Required: Not given.

Reliability: Not given.

Remarks: The translation is accomplished in a single pass and the resulting program is produced on fixed-point format cards. Intersections called for by the source program are selected by means of a symbolic linkage and located at the time of execution.

IBM 650 Systems: One 535, automatic floating decimal arithmetic feature and indexing registers are required.

Special Devices: Group II special character device is required.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: Not given.

e. Speed: Not given.

Floating/Fixed: Does not apply.
be translated or not. Error stops are branches to 9999, and a display and
restart procedure is given.
d. Storage Requirements: Not given.
e. Remarks: Precautions: Instructions to be translated must be in the range
0000-9999. The program is for use with the Bell III Interpretive Program
as it checks for 3 return addresses to Bell III and translates then to the
-corresponding Bell III Systems locations. There are no provisions for
RAMAC or tape instructions.

Regulation Specifications: Basic IBM 650 and 533 card input-output device,
and the 653 auxiliary SAS unit with 60 words of core storage and 3 Index
Accumulators.

IBM 650 Library Program Abstracts

THREACS

S. Nakai
Applied Science Dept.
IBM - Japan, Ltd.
Tokyo, Japan

a. Purpose: This system is a compiler, which accepts THREACS
instructions which are in three address form and produces 650
instructions in synthetic language. Some synthetic instructions
can be assembled by the standard SOAP II. This system has two
main advantages: first, it can be directly written in the source program together with
THREACS instructions which can be interpreted for higher efficiency and flexibility than other compilers.
The other is that it is possible to translate a program written in the
L3 interpretive form into a SOAP program.
b. Range: Does not apply.
c. Mathematical method: Does not apply.
d. Storage required: This system requires all of from storage.
e. Relocatability: Not relocatable.
f. 650 System: One 533, indexing registers and the floating arithmetic
device are required.
Special device: Alphabetic device.

EXPONENTIAL

S. Fleming March 28, 1956
G. E., Schenectady

a) Computes e^x for a single-precision fixed-point number.
b) Range: -10.45 < x < 23.02585092.
c) Method not given.
d) Storage required is 50 locations, 0000 to 0049; the routine may be trans­
lated an even number of locations. Not more than 6 iterations are required.
e) None.
f) Minimum 650.
<table>
<thead>
<tr>
<th>Library Program Abstract</th>
<th>File Number: 3.1.005</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXPONENTIAL</strong></td>
<td></td>
</tr>
<tr>
<td>S. Fleming</td>
<td>March 28, 1956</td>
</tr>
<tr>
<td>G. E., Schenectady</td>
<td></td>
</tr>
<tr>
<td>a) Computes $e^x$ for single-precision fixed-point number.</td>
<td></td>
</tr>
<tr>
<td>b) Range: $-20 &lt; x &lt; 25.02365632$. Accuracy: error is less than one in the eighth significant digit.</td>
<td></td>
</tr>
<tr>
<td>c) Method not given.</td>
<td></td>
</tr>
<tr>
<td>d) Storage required is 40 locations, 0000 to 0048; the routine may be translated by an even number of locations.</td>
<td></td>
</tr>
<tr>
<td>e) None.</td>
<td></td>
</tr>
<tr>
<td>f) Minimum 650.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Library Program Abstract</th>
<th>File Number: 3.1.009</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SINH X AND COSH X</strong></td>
<td></td>
</tr>
<tr>
<td>Barbara Martin</td>
<td>August 8, 1955</td>
</tr>
<tr>
<td>Detroit Edison, Detroit</td>
<td></td>
</tr>
<tr>
<td>a) Computes sinh $x$ or cosh $x$ for a single-precision fixed-point number.</td>
<td></td>
</tr>
<tr>
<td>b) Range: $0 &lt; x &lt; 2$. Accuracy information not given.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>d) Storage required is 62 locations, 0000 to 0061, including the $e^x$ subroutine. The routine may be translated an even number of locations.</td>
<td></td>
</tr>
<tr>
<td>e) The $e^x$ subroutine is not included in the deck listing.</td>
<td></td>
</tr>
<tr>
<td>f) Minimum 650.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Library Program Abstract</th>
<th>File Number: 3.1.010</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SIN-COS SUBROUTINE</strong></td>
<td></td>
</tr>
<tr>
<td>G. R. Trimble, Jr.</td>
<td>1-30-55</td>
</tr>
<tr>
<td>IBM, Houston</td>
<td></td>
</tr>
<tr>
<td>a) Calculates sin $x$ or cos $x$ for a single-precision fixed-point number.</td>
<td></td>
</tr>
<tr>
<td>b) Range: For sin $x$, $-1.23x10^{-7.2}$; for cos $x$, $-8.8$ to $10^{-8.4}$. Maximum error is $2 \cdot 10^{-5}$.</td>
<td></td>
</tr>
<tr>
<td>c) Method: 12th power in Taylor series. Reference: Technical Newsletter No. 9, p. 34.</td>
<td></td>
</tr>
<tr>
<td>d) LWA is 0099 with one word open in the relocatable version. Running time is 125 ms.</td>
<td></td>
</tr>
<tr>
<td>e) Both absolute and SOAP relocatable deck listings are included.</td>
<td></td>
</tr>
<tr>
<td>f) Alphabetic device if relocatable version is used.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Library Program Abstract</th>
<th>File Number: 3.1.013</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LOG_{10} A, Ln A</strong></td>
<td></td>
</tr>
<tr>
<td>E. B. West and A. G. Gardner</td>
<td>3-20-56</td>
</tr>
<tr>
<td>IBM, Houston</td>
<td></td>
</tr>
<tr>
<td>a) Computes $\log_{10} A$ or $\ln A$ for single-precision fixed-point numbers.</td>
<td></td>
</tr>
<tr>
<td>b) Range $10^{-5} &lt; A &lt; 10^5$. Accuracy: maximum error is $2 \cdot 10^{-7}$.</td>
<td></td>
</tr>
<tr>
<td>c) Method: polynomial approximation by Hastings.</td>
<td></td>
</tr>
<tr>
<td>d) LWA is 0099 with 34 words open in the relocatable version. Running time is 130 ms.</td>
<td></td>
</tr>
<tr>
<td>e) Both absolute and SOAP relocatable deck listings are included.</td>
<td></td>
</tr>
<tr>
<td>f) Alphabetic device if relocatable version is used.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Library Program Abstract</th>
<th>File Number: 3.1.014</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NATURAL LOGARITHM</strong></td>
<td></td>
</tr>
<tr>
<td>S. Fleming</td>
<td>3-28-56</td>
</tr>
<tr>
<td>G. E., Schenectady</td>
<td></td>
</tr>
<tr>
<td>a) Computes $\ln x$ for a single-precision fixed-point number.</td>
<td></td>
</tr>
<tr>
<td>b) Range: $10^{-6} &lt; x &lt; 10^9$. Accuracy: error is less than 2 in the 7th decimal.</td>
<td></td>
</tr>
<tr>
<td>c) Method not given.</td>
<td></td>
</tr>
<tr>
<td>d) Storage required is 54 locations, 0000 to 0053.</td>
<td></td>
</tr>
<tr>
<td>e) None.</td>
<td></td>
</tr>
<tr>
<td>f) Minimum 650.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Library Program Abstract</th>
<th>File Number: 3.1.015</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POLAR TO CARTESIAN COORDINATES</strong></td>
<td></td>
</tr>
<tr>
<td>Barbara Martin</td>
<td>7-27-55</td>
</tr>
<tr>
<td>Detroit Edison, Detroit</td>
<td></td>
</tr>
<tr>
<td>a) Converts single-precision fixed-point polar coordinates to single-precision fixed-point cartesian coordinates.</td>
<td></td>
</tr>
<tr>
<td>b) Range: $r &lt; 100$, $0 &lt; \theta &lt; 2\pi$.</td>
<td></td>
</tr>
<tr>
<td>c) Method is to use the sin-cos subroutine in Technical Newsletter No. 9, page 30 and then to use the standard conversion formulas.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>e) The sin-cos subroutine is not included in the deck listing.</td>
<td></td>
</tr>
<tr>
<td>f) Minimum 650.</td>
<td></td>
</tr>
</tbody>
</table>
FLOATING POINT LOG \( |A| \) AND LN \( |A| \)

Prepared by IBM 650 Applied Programming

G. J. Porter
IBM, New York

a) This subroutine computes Log \( |A| \) and Ln \( |A| \) utilizing the floating decimal arithmetic device and indexing register A. This routine has maximum range and accuracy with running time minimized as much as possible.

b) Range: \(|A| > 0\)
Accuracy: Error < 10^{-8}

Floating Point

c) Method:
\[ A = M \times 10^p \]
where \( p \) is an integer
Multipliers \( A_i \) are found such that \( m = A_i \)
The \( A_i \) are chosen so that \( 1 < m < 1.1 \)

Log \( M \) is computed by use of a relaxed Taylor series for
\[ \log(1 + c), \quad 0 < c < 1 \]

Finally, Log \( M \) = Log \( 10^p \) - \( \sum \) Log \( A_i \)

Ln \( A \) is secured by multiplying Log \( A \) by 10.

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

d) Storage requirements: 100 locations with 15 open.

Speed: Log: 170 m. s.; Ln: 140 m. s.

Relocatable SOAP II cards.

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

f) 650 equipped with floating decimal 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) Subroutines for \( e^A \), \( 10^A \), Sinh A and Cosh A utilizing the floating decimal arithmetic device and indexing register A. Maximum accuracy and range have been secured with reasonable running time and storage requirements.

b) Range: \( e^A : A < 100; \quad 10^A : A < 43 \).

Accuracy: Relative accuracy of 10^{-8}

Floating Point

c) Mathematical methods:

\( e^A \): By several reductions \( A \) is reduced to the range \( |A| < 10^4 \).
A relaxed Taylor series is then used.

sine, cosine, sinh, and cosh are obtained from the same table used for the exponential function.

d) Speed: \( e^A \): 180 m. s.; \( 10^A \): 186 m. s.; Sinh A and Cosh A: 240 m. s.

Storage: 159 Locations for the entire routine. If only \( e^A \) or \( 10^A \) are desired, 26 Locations can be omitted. For convenience these 25 are located at the end of the program.

Input: Relocatable SOAP II cards.

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

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

FRATS
(Fast, Reallocatable, Arithmetic and Transcendental Subroutines)

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

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

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

Scientific notation

The range of the exponent, xx, is therefore \( 0 \leq xx \leq 99 \). Unnormalized numbers may be used as input to any of the routines. Results are normalized, except for FIX and unnormalized ADD. Given exact, normalized input, the maximum result error is about \( 0.06 \) units of the last result digit, except for logarithms of numbers near unity, which are correct within \( 3 \times 10^{-10} \) before normalization and rounding. Unnormalized input is handled with equal precision, except when added or used as numerator in division.

c) Square root is computed from the Newton iteration method, using three iterates. The exponential function, \( e^A \) or \( x^A \) (\( a \leq 10 \)), is evaluated using a table of \( y = 10^m \) at intervals \( \Delta y = 0.1 \), and a sixth-degree polynomial interpolation; the 650 table lookup operation is not used. The logarithmic function, \( \ln z \), is evaluated using a seventh-degree expansion in odd powers of \( z - 1 \). Values of \( y \) and \( \log y \) are obtained, by table lookup, from the same table used for the exponential function.

d) The complete set of routines occupies 356 locations including temporary storage, and can be loaded in locations 0001 - 0399 or any 8 consecutive bands on the drum. The routines are relocatable by SOAP II in any higher region on the drum, except that the address increment for Natural Logarithm must be evenly divisible by 50. Any block of routines may be omitted without affecting the others, except that Multiply-Add requires Blocks 1 and 2.

April 1958, Bulletin 18 - 17
### IBM 650 Library Program Abstracts

#### Mathematical Functions

**ARCSIN X, ARCCOS X, SQUARE ROOT X**

- **Purpose:** Computes arcsin X, arccos X, square root X for a single-precision floating-point number.
- **Range:** Arcsin: \(-1 \leq X \leq 1\)
- **Accuracy:** Maximum error \(<1.5 \times 10^{-7}\)
- **Floating/Fixed:** Floating.
- **Mathematical Method:**
  - Arcsin / Arccos: Polynomial approximation by Hastings.
  - Square Root: First approximation involving a table look-up followed by three iterations with Newton’s formula.
- **Storage Required:** 140 locations are required.

**PARABOLIC INTERPOLATION**

- **Purpose:** To interpolate the \(f(x)\) value corresponding to a given \(x\) value by fitting a parabola through 3 given points which define the curve on which \(f(x)\) lies. All values must be in normalized floating-point form.
- **Range:** The routine will use any set of numbers supplied.
- **Accuracy:** The region of the curve under consideration must be parabolic, and the base of symmetry of the assumed parabola must be perpendicular to the x-axis for most accurate results.
- **Floating/Fixed:** Floating.
- **Mathematical Method:** The three given points are used to set up 3 simultaneous linear equations. Solution of these equations yields the equation of the parabola from which \(f(x)\) is calculated.
- **Storage Required:** 80 locations in 2 adjacent bands plus a previously defined region K of b 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 531, automatic floating decimal arithmetic, and indexing registers are required.
- **Special Devices:** None.

**CUBE ROOT X**

- **Purpose:** Computes the cube root of any single-precision normalized floating-point number.
- **Range:** Any positive floating point argument.
- **Accuracy:** Maximum error \(<1.5 \times 10^{-7}\)
- **Floating/Fixed:** Floating decimal.
- **Mathematical Method:** Square root subroutine uses a Newton-Raphson iteration. All other routines 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 23.
- **Speed:** For the subroutines given in (a) above, the average computation times are, respectively: 156, 187, 175, 130 and 188 milliseconds.
- **Relocatability:** The program decks are in relocatable SOAP 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.
- **650 System:** One 531, indexing registers, and automatic floating decimal arithmetic feature are required.
- **Special Devices:** None.
IBM 650 Library Program Abstracts

PRIME NUMBER GENERATOR

J. J. Di Giorgio
New York Telephone
New York City

a. Purpose: To generate prime numbers within a given range.
b. Range: 1-2048, 000, 000.
c. Accuracy: Does not apply.
d. Floating/Fixed: Not given.
e. Mathematical Method: A number is tested for primeness by dividing by all prime numbers up to the square root of the number tested.
f. Storage Required: The program is stored in the first 2048 drum locations. A table is created from 500 upwards, depending on the range of numbers desired.
g. Speed: Is a function of the range. For example, program execution time for the range 20,000 to 31,000 is ten minutes.
h. Relocatability: Not given.
i. Remarks: None.

IBM 650 System: One 650 required.

IBM 650 Library Program Abstracts

STANDARDIZED UTILITY DECK OF SUBROUTINES (SUDS)

T. A. Wall
Raytheon Company
Wayland, Mass.

b. Restrictions: Range: Floating point throughout, angles In radians.
c. Method: Square root uses 3 iterations

d. Storage Requirements: Speed is from 125 to 350 ma. depending upon the

wayland, Mass.

e. Remarks: All entries, exits, and atops are standardized. Although execution time is slightly longer than separate relocatable subroutines, time is saved overall through reduced card handling. All of the functions have been thoroughly tested. The Library Program File SUDS is in absolute and as it had been programmed in SOAP II format.
f. IBM 650 System: 650 with floating point. SUDS uses no index registers. Since SUDS is In absolute, the alphabetic device is not required.

IBM 650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 3.2.001

CIRCULAR AND HYPERBOLIC FUNCTIONS: REGULAR BESSEL FUNCTIONS

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

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

b. Arguments are fixed-point in the form xx.xxxxxxxx; answers are given in both fixed and floating-point form. Range for sin x and cos x is 1 < x < 100; for sinh x and cosh x, 1 < x < 5.29; sinh x < 6.32; J0(x), x < 6.32; J1(x), x < 6.77; 

j2(x), x < 7.15; J3(x) and J4(x), x < 7.82, J6(x), x < 8.62, J8(x), x < 9.84. The series is summed until the new term is <10^-4.

c. Series expansions are used.
d. Storage required in 150 locations, 0000 to 0149, and may be translated by an even amount.
a) None.
f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 3.2.002

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 designed for use with the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter No. 11.

b) Floating point input and output. Arguments are fixed-precision. Form xx.xxxxxxxx; answers are given in the interpretive system, i.e.,

10^-50 < |x| < 10^50, and x = 0.

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

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

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

(Continued on next page)
**IBM 650 Library Program Abstracts**

**BESSEL FUNCTIONS SUBROUTINE**

An error in the Bessel Functions subroutine, File Number 3.2.005 has been noted. The error is such that a Bessel function is calculated incorrectly if the argument is less than minus infinity. The subroutine calculates all Bessel functions correctly. This error may be corrected by removing card No. 605 and replacing it with two cards.

**TYPE** | LOC | OP | OA | IA
--- | --- | --- | --- | ---
2 | 0394 | 0DF | 023 | 0084

In the original deck, the upper accumulator was not cleared following the execution of the instruction in FUP. For the Be and Th functions, the succeeding instruction was a FAD instruction. The amount of relocation UN, module 105, was then treated as the argument of the number remaining in the accumulator. Thus, when UN was greater than about 40, a significant error was introduced.

---

**Mathieu and Modified Mathieu Functions Subroutine**

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

a) Computes Mathieu and modified Mathieu Functions

b) Range: n = 0, 1, 2, 3

Accuracy: 5 significant figures.

---

**Mathematical Functions**

---

**IBM 650 Library Program Abstracts**

---

**AN INTERPRETIVE SUBROUTINE FOR THE SINE INTEGRAL AND COSINE INTEGRAL FUNCTIONS**

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

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

b) Floating point input and output. Accepts any argument (positive and negative) accepted by the interpretive system, viz., locations 1000-1999.

c) The maximum error is designed for use with the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter No. 11.

d) 490 storage locations are required. SOAP

---

**BESSEL FUNCTIONS SUBROUTINE**

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

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

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

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

d) 650 storage locations are required - SOAP II relocatable or fixed in locations 0000-0989.

e) None.

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

---

**IBM 650 Library Program Abstracts**

---

**ASSET OF INTERPRETIVE SUBROUTINES FOR CYLINDRICAL AND SPHERICAL BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS AND THEIR DERIVATIVES**

R. E. Kulawad
RCA Laboratories
Princeton, New Jersey

a) Purpose: Subroutines compute any or all of the Bessel functions J_n(x), Y_n(x), I_n(x), I_n'(x), K_n(x), K_n'(x), J_n'(x), and Y_n'(x). Three...
IBM 650 Library Program Abstracts

IBM 650 Library Program Abstracts

Multiple Numerical Integration

Laplace Transformation

The following supplement to the program write-up has been submitted:

This program solves the algebraic equation entered on data card #1 prior to returning control to the console to read the second data card. Therefore, it has been found useful at times to replace the second data card with a self-loading program to read out or operate upon the coefficients without performing the transformation.
AN INTERPRETIVE SUBROUTINE FOR THE SOLUTION OF SYSTEMS OF FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS

Franz Edelman
RCA, David Sarnoff Research Center, Princeton

a) Solve systems of first order ordinary differential equations.

b) Systems of up to 50 equations may be solved. Floating decimal arithmetic is used. Precision is specified by the programmer.

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

d) The interpretive routine occupies locations 0000 to 0099. Execution time per point is about 4 - 5n seconds for the RKG method and about 7.5 - 12n seconds for the Milne method where N is the number of equations to be solved.

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

f) Minimum 650.

ESLIPIC INTEGRALS

B. Pootos
B. Carpenter
University of California Radiation Laboratory
Livermore, California

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

b) The elliptic integrals contain two parameters whose ranges are: 0  $\leq k$  $1.0$; 0  $\leq Q$  $\pi/2$. $k$ is defined as the modulus and $Q$ is defined as the amplitude of the elliptic integrals.

Magnitude of parameters are expressed in floating point notation. The two high order digits determine the location of the decimal point. $XXXYYYYY.$ $Q$ is measured in radians.

The results are accurate to seven decimal digits when the parameters are in the following ranges: $0.25 k < 0.5$ and $0.25 Q < 1.6$ (p=97). Outside this range, the accuracy decreases, particularly when both parameters are close to their upper bounds.

c) Repeated application of Legendre's transformation permits one to replace a numerical integration process with an algebraic expression whose members are rapidly evaluated. The magnitudes of the algebraic members rapidly converge to a constant value ($0.7 
$) and hence only a few terms are required for the desired accuracy.

(Continued on next column)

IBM 650 Library Program Abstracts

FILE NUMBER 4.0.005

RELAXATION PROGRAM: LAPLACE'S EQUATION IN RECTANGULAR COORDINATES

D. Dorfman
Lycoming Division of AVCO Mfg. Corp., Gas Turbine Department
Stratford, 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 900 interior points distributed as follows:

1. Up to 50 vertical distances, including boundaries.
2. Up to 30 horizontal distances excluding boundaries.
3. Up to 10 interior points along any of the vertical coordinate strips (11 including the boundary). Where $N$ is the number of equations to be solved, the execution time is used. Precision is specified by the programmer.

c) Mathematical Method: Finite difference method for unequal spacing, allowing both over-relaxation and under-relaxation.

d) Storage Required: Full drum storage required.

Speed: Speed is approximately .35 seconds per interior point per iteration.

Locatability: Not relocatable.

f) Minimum 650.

Special Devices: None.

IBM 650 Library Program Abstracts

FILE NUMBER 4.0.006

"Relaxation Program: Laplace's Equation in Rectangular Coordinates," by D. Dorfman.

The following changes in the deck and listings should be made:

Restrictions as types of parabolic points:

Experience in using the relaxation programs indicates 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:

(Continued on next page)
Differential and Integral Equations

RELAXATION PROGRAM: LAPLACE’S EQUATION IN THE CYLINDRICAL COORDINATE SYSTEM

D. Dorfman
Lycoming Division of AVCO Mfg. Corp.
Stratford, Connecticut

a. Purpose: Solves axisymmetric incompressible flow problems with variable $r$ (radial distances), and $u$ (azimuthal coordinate 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 30 interior points along any radial coordinate strip (32 including the boundaries).

c. Mathematical Method: Finite difference method for unequal spacing, allowing both over-relaxation and under-relaxation.
d. Storage Required: Full drum storage required.
e. Remarks: Program must be reloaded for each new case.

IBM 650 Library Program Abstracts

ERRATA/ADDENDA

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:

<table>
<thead>
<tr>
<th>Location</th>
<th>In</th>
<th>Should Be</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290</td>
<td>24 1958 0134</td>
<td>24 1958 0134</td>
</tr>
<tr>
<td>1453</td>
<td>24 1953 1807</td>
<td>24 1953 1807</td>
</tr>
<tr>
<td>1503</td>
<td>24 1958 1800</td>
<td>24 1958 1800</td>
</tr>
</tbody>
</table>

The following changes in the deck and listings should be made:

Restrictions on types of parabolic points:

Experience in using the relaxation programs dictates that parabolic points should be avoided wherever possible, because error 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 the following 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: Laplace’s Equation in the Cylindrical Coordinate System, by D. Dorfman.

The following addenda should be made to the program write-up:

 Sony games

<table>
<thead>
<tr>
<th>Location</th>
<th>In</th>
<th>Should Be</th>
</tr>
</thead>
<tbody>
<tr>
<td>0540</td>
<td>24 1958 0134</td>
<td>24 1958 0134</td>
</tr>
<tr>
<td>1653</td>
<td>24 1953 1807</td>
<td>24 1953 1807</td>
</tr>
<tr>
<td>1700</td>
<td>24 1958 1800</td>
<td>24 1958 1800</td>
</tr>
</tbody>
</table>

(Continued on next column)

IBM 650 Library Program Abstracts

ERRATA/ADDENDA

Relaxation Program: Laplace’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>0540</td>
<td>24 1958 0134</td>
<td>24 1958 0134</td>
</tr>
<tr>
<td>1653</td>
<td>24 1958 0134</td>
<td>24 1958 0134</td>
</tr>
<tr>
<td>1700</td>
<td>24 1958 1800</td>
<td>24 1958 1800</td>
</tr>
</tbody>
</table>

(Continued on next column)
IBM 650 Library Program Abstracts

NUMERICAL SOLUTION OF LAPLACE, POISSON, AND HEAT FLOW EQUATIONS

J. B. Annable
Jack & Milton, Incorporated
Cleveland I, 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 to be studied is represented by a grid approximation and $T$ is found for each intersection by a finite difference approximation $K$ applicable to that point. Output is both $T$ and the residual at each point.

b. Range: The size of the field is limited such that $7 \leqslant 704$; and $E \leqslant 50$.

Accuracy: Not given.

Floating/Fixed: Both input and output data are fixed-point form.

c. Mathematical Method: The numerical method used, based on a finite difference approximation to the partial differential equation, yields equations of the form:

$$A_{ij}T_{m} + B_{ij}T_{m-1} + C_{ij}T_{m+1} + D_{ij}T_{n-1} + E_{ij} + F_{ij} = 0$$

The values of the coefficients are determined by an analysis of the properties of the region at each intersection point. The equations are solved for $T_{m}$ at each point by setting $E = 0$ and using an iterative process. Convergence is controlled by:

$$\sum_{i=1}^{m} \left| \frac{T_{m} - T_{m-1}}{T_{m}} \right| \leq 10^{-\nu}$$

where $\nu$ = iteration number, $m$ = point number, $n$ = number of points and $0 \leq \nu \leq 5$.

d. Storage Required: The entire drum is used; however, locations may be used with a consequent 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 convergence criteria specified above. Consequently, the error analysis is extremely difficult.

d. 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

SOLUTION OF N SIMULTANEOUS DIFFERENTIAL EQUATIONS

R. R. Haefner
Savannah River Laboratory
E. L. 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 $\frac{dy}{dx} = Ay$, 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: $N \leq 30$.

Accuracy: Not given.

Floating/Fixed: Computation is in floating decimal arithmetic.


d. Storage Required: 2000 storage locations are required.

Speed: 3.9 sec/pt for $N = 7$; 9.5 sec/pt for $N = 14$; for $N = 18$.

Relocatability: Non-relocatable.

e. Remarks: None.

f. 650 System: One 533, automatic floating decimal arithmetic, and indexing registers.

Special Devices: None.

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

a. Purpose: This program consists of two separate routines for solving differential equations. One makes use of Runge-Kutta-Gill over the whole range of integration. The other uses the Milne method as a main process and uses the Runge-Kutta-Gill as a starting procedure and as an auxiliary process for changing the mesh size when desired.

b. Range: See the program write-up for detailed information.

Accuracy: The programmer specifies the number of significant figures to be retained when using the Milne methods. The routine automatically checks the truncation error at each step to see that it is not significant enough to affect the desired accuracy. The routine also checks to see whether the truncation error is so slight that a significantly larger interval may better be used.

Floating/Fixed: Floating decimal.


d. Storage Required: The RKG routine requires 288 storage locations including printout subroutines. The BEGM routine requires 195 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. There also exists a facility for punching out errors involved at each step. This punchout consists of the round-off error at each step when using BEGM and the truncation error at each step when using Milne. The routine is written in SOAP II and may be used as an extension for any SOAP II version of the Carnegie Tech Compiler (IT) in the usual automatic way. However, it may also be used as a Compiler I extension or as a separate SOAP II subroutine. In this case the programmer must make the following provisions:

1) Reserve an adequate block of storage.
2) Insert the subroutine variables into the 1950 read band as indicated in the write-up.
3) Make the necessary regional and symbolic address assignments as indicated by the main program.

(Continued on next page)
The printout subroutine used is Compiler Extension 3 and may be used by any other part of the program by making the usual reference.

f. IBM 650 System: Once 533, automatic floating decimal arithmetic feature, and indexing registers.

Special Devices: Alphabetic device required.

650 Library Program Abstracts

NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS OF ORDER N

Dennis M. Sinnett
University of Michigan
Willow Run Laboratories
Computation Department
Ann Arbor, Michigan

a. Purpose: The routine solves differential equations of order N.

b. Restrictions: Range 1 < N.

Accuracy: Specified by user.

c. Method: Combined Runge-Kutta Milne method, with an option for Runge-Kutta solution only.

d. Storage Requirements: 620 locations 00100...00730, with 100 or less storage locations (0001...0099) depending on the order of the equation.

e. Remarks: The user specifies the function to be integrated, its order, and the initial conditions.

Time: Milne - .1N seconds per point. Runge-Kutta - .6N seconds per point. Plus - .3 seconds per card punched.

f. IBM 650 System: Uses index registers and floating decimal arithmetic.

650 Library Program Abstracts

MATRIX INVERSION

A. O. Garder and J. M. Kilbee
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. Dora Sweeney's, and performs Gaussian Elimination using eight-digit floating-point arithmetic.

d) The program with storage space for the matrix utilizes essentially the complete drum. For a matrix of order n = .00004 (.01n) 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 Abstracts

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

A. O. Garder
IBM, Houston

a) Solves b systems of n 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)
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: Fixed arithmetic is used.

Mathematical Method: The Jacobian Matrix Diagonalization method is used in these routines.

Storage Required: Part I and Part II require all 2000 locations for a maximum size matrix.

Speed: The time requirement for a well conditioned matrix may be computed as follows:
Part I: \((2.5 \times 10^{-8} \times n^2)\) minutes, where \(n\) is the size of the matrix.
Part II: \((0.006n^3)\) minutes, plus punch-out time.
Part III: \((0.006n^2)\) minutes to reassemble vectors from rotation punch-out of Part II.

Relocatability: The program is not relocatable.

Remarks: None.

IBM 650 System: One 533 required.

Special Devices: None.

---

**PATTERN QUARTIMAX ROTATION OF A FACTOR MATRIX**

Miss Ruth W. Bredon
Educational Testing Service
Princeton, New Jersey

**a. Purpose:** This program employs a modification of the quartimax computation for factor rotation. In this modification a hypothesized factor pattern is given to the computer to serve as a guide in the choice of factor loadings. The result is a factor matrix with elements rounded to 8 decimal places.

**b. Range:** The program will handle a matrix up to 500 elements.

**Accuracy:** Elements are rounded to 8 decimal places.

**Floating/Fixed:** Fixed decimal arithmetic is used.

**Mathematical Method:** The quartimax method is used.

**Storage Required:** The entire drum is used.

**Speed:** The speed of computation depends on the number of reflections needed in this method, this sum has been called the Matrix-Vector Product or M-VP. A square matrix, herein 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 sum of these matrices.

**Remarks:** Over 200 problems have been successfully run to date, the largest involving a product matrix of order 48 x 48.

**IBM 650 System:** The basic IBM 650 computer is required.

---

**IBM 650 Library Program Abstracts**

**IBM 650 Library Program Abstracts**

**IBM 650 Library Program Abstracts**

---

**MATRICES - VECTOR PRODUCT**

Reverdy Wheat
Agricultural Experiment Station
University of Florida
Gainesville, Florida

**a. Purpose:** To compute the portions of the total sum of squares of deviations of \(n\) observations from their mean, appropriate to the \(i\) th independent contributions in that sum. To accomplish this, the products of each row, after the first, of an \(n \times n\) matrix and the \(n\) x \(n\) matrix of deviations vector are computed and summed. In the development of this method, the sum has been called the Matrix-Vector Product or M-VP. A square matrix, herein 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 sum of these matrices.

**b. Restrictions, Range:** All computations are done in either single or double precision fixed-point arithmetic.

**c. Method:** Sums of squares are obtained to 4 places of decimals in single precision.

**d. Storage Requirements:** The program is non-relocatable, consists of approximately 550 instructions and is reasonably fast in execution.

**e. Remarks:** None.

**f. IBM 650 System:** The basic IBM 650 computer is required.

---

**IBM 650 Library Program Abstracts**

**IBM 650 Library Program Abstracts**

---

**FACTOR ANALYSIS BY THE CENTROID METHOD**

S. O. Navarro
University of Kentucky
Lexington, Kentucky

**a. Purpose:** This program computes the factors of a symmetric matrix with unknown communalities by assuming each communality equal to the largest element in each column.

**b. Range:** Not given.

**Accuracy:** Not given.

(Continued on next column)
Note that the location given is relative to the submat, not the matrix itself.

1. IBM 650 System: Minimum 650 with alphabetic and special character devices. Of course, this subroutine can be modified for use as a strict machine language program.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.001

MATRIX INVERSION

D. W. Sweeney
IBM, New York

October 6, 1955

a) Inverts matrices of order n ≥ 42 or solves b sets of simultaneous equations for n² + nb ≥ 1764

b) Matrix elements are in floating-point form.

c) Method not given.

d) Storage required is 336 locations. 1764 to 1999. The matrix inversion, exclusive of input and output time, is executed in approximately 0.72n² seconds.

e) Locations 0000 to n²-1 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. Garder
IBM, Houston

April 2, 1956

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

b) All numbers are of the form c e aaaaaaa = a.aaaaaaa 10ee-50.

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

d) Storage required is approximately 350 locations 1650-1999. Time required for one inversion, or solution, is 0.00003n²h in hours.

e) Storage limitations require that n²+(n+1)b ≥ 1650. The inverse of the coefficient matrix is obtained with solution of a system of simultaneous linear equations. This is a modified version of a program originally written by Dura Sweeney which is now self-restoring on the drum.

f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.005

COMPLEX AND REAL EIGENVALUES

R. W. De Sin
IBM, Schenectady

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

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

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.007

LARGE SCALE MATRIX INVERSION

a) Computes the inverse of large order matrices.

b) Matrix elements are floating-point of the form x.xxxxx xxxxx ee, where ee represents an exponent modulo 50. A matrix of order n ≥ 508 may be handled.

c) The Jordan method is used.

d) Approximately 350 storage locations are used for the program. Time required is n²+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 kth elimination step is supplied as input for the k +1st step. A total of n passes is necessary.

f) Alphabetic device if SOAP symbolic version is used.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.008

MATRTEX INVERSION

R. L. Norman
IBM, Washington

December 31, 1956

a) This program has modified 5.2.002 to include load and punch routines so that any number of matrices may be loaded, inverted and punched out without reloading the program. This program will invert a matrix of order N or will solve b systems of simultaneous linear equations with b constant column vectors on the righthand side of a common coefficient matrix of order N, where N² + (N+1)b ≥ 1650.

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 1650 - 1999. Locations 0000 - (N+1)(N-b) are used for storage of matrix elements and temporary storage. Loading and punching are at full speed; the calculation requires approximately .0012N (N+b)² minutes. The program is no 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

December 31, 1956

(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 x 25 may be inverted and V systems of N equations may be solved where 2 (N+1) (N+V) ≤ 1990.

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

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

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

f) Minimum 650.

ERRATA

650 Program Library - File No. 5.2.005

"Double Precision Matrix Inversion," by J. D. Chappell

The following correction should be made in the detailed write-up:

On page 3, in the paragraph headed "Deck Description," the last sentence should read: "The deck consists of 106 cards serially numbered from 001 to 106."

The program deck is correct as distributed.

D. R. Williams
University of California Radiation Laboratory
Livermore, California

B. N. Carr
IBM Corporation

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

b) Square matrices, (nxn), can be inverted where n (n+1) (n+m) ≤ 1990. Rectangular arrays, (m x n), can be solved where (m + 1) (m + n) ≤ 1990. As with any similar procedure, error due to accumulated roundings may be large.

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

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

e) The inversion program fails if a1,1 or any element which takes its place during the calculation is zero. The program is written in machine language.

f) This routine requires a 650 equipped with the floating decimal device, index registers, and immediate access storage.
d) Speed is that of fastest standard method to size 12 x 12, but from this point
the necessarily complex address modification increases running time as n, and
hence the number of iterations, increases.

e) Since the product of a matrix and its transpose is a symmetrical matrix, the
routine can be extended to non-symmetrical matrices to size 34 x 34.

f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 5.2.014
VECTOR BY SYMMETRICAL MATRIX MULTIPLICATION
S. Young
Detroit Edison Company
Detroit, Michigan

a) Performed 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 modifica­
tions 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, \( \frac{n^2 \times 2}{2} \) storage locations are needed
to store the matrix
in memory.

e) None

f) Minimum 650.

April 1958, Bulletin 18 - 23

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 5.2.015
MATRIX INVERSION
J. C. English
F. K. Townsend
E. I. du Pont de Nemours & Co., Inc.
Savannah River Laboratory
Aiken, South Carolina

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

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

c) Gaussian Elimination.

d) Approximately 350 storage locations are used. The code is given in SDAP II
format. Computation time for 40th order matrix is about 0.029 sec.

e) If a matrix system has b constant vectors, then a+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 alpha­
abetic device is desirable.

April 1958, Bulletin 18 - 25

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 5.2.016
LATENT ROOTS AND VECTORS OF A MATRIX
W. Granet
Boston University
Boston, Massachusetts

(Continued on next column)
GENERAL SIMULTANEOUS EQUATIONS SOLUTION

J. H. Scheck
Coriell-Wright Corporation
Prepilator Division
Cambridge, 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.


d. 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 2 minutes for 4 equations.

Relocatability: Program is not relocatable.

Remarks: None.

IBM 650 System: One 533, automatic floating decimal arithmetic, and indexing registers are required.

Special Devices: None.

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 39th 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.

c. Mathematical Method: The Gauss-Jordan elimination method is used. Pivotal elements are selected according to size. Zero elements may appear on the main diagonal.

d. Storage Required: The entire drum is used.

Speed: The time required for the inversion process is approximately 0.044n^3 seconds, where n is the order of the system.

Relocatability: Not relocatable.

Remarks: A matrix check program is included.

IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic feature.

MATRIX INVERSION WITH ITERATIVE IMPROVEMENT OF ACCURACY

R. B. Dean
M. B. Higgins
Development Department
Union Carbide Chemicals 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 minimizes 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.

Relocatability: Not relocatable.

Mathematical Method: The following method is used for the iterative improvement of the inverse:

\[ A_k^{(-1)} = (I - A_k^{-1})^{-1} \]

where \( A \) is the original matrix

\( A_k^{(-1)} \) is the kth approximation of the inverse

\( I \) is the unit or identity matrix

Speed: The inversion time, excluding input, is approximately 0.031n^3 seconds. The calculation time for the improvement iterations is approximately 0.09n^3 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 automatically and duplicate storage locations used in the first deck. 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.

IBM 650 System: One 533, IAS, and automatic floating decimal arithmetic feature are required.

MULTIPLICATION OF MATRICES

MOLECULAR SPECTROSCOPY
IBM 650 Library Program Abstracts

MOLAR SPECTROSCOPY
LATENT ROOTS AND VECTORS OF A MATRIX

George J. Joss
Rensselaer Polytechnic Institute
February 29, 1956

*Purpose*: Calculation of latent roots and their associated eigenvectors for real matrices.

*Restrictions, Range*: Floating decimal arithmetic.

*Method*: An adaptation of a method of Werner Frank for the calculation of the roots of \( f(x) \) in a matrix reduction method due to Givens.

*Storage Requirements*: Machine language program handles a 3 x 3 up to a 20 x 20 matrix. With more memory larger matrices can be handled by changing the dimension statement in the Fortransit II III program. (Continued on next column)

- Remarks: This program can obtain multiple eigenvalues and their associated eigenvectors.
- This program can be used in conjunction with the Fortransit system.

IBM 650 System: IBM 650.
### Correlation Coefficient Routine

**J. W. Robinson, III**  
IBM, Houston  
July 9, 1956

- Computes the means, standard deviations, and product moment correlation coefficients of $n \leq 50$ variables.
- 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.
- All locations except $N! + 1$ to $1274$ are used; for $n = 50$ the entire drum is used. Approximate time for 100 observations is 8 min. for $n = 10$; 29 min. for $n = 20$; 71 min. for $n = 30$; 125 min. for $n = 40$; 195 min. for $n = 50$. For other cases assume that the time varies linearly as the number of observations and as the square of the number of variables.
- Self-loading and self-restoring.
- Minimum 650.

### Analysis of Variance Program

**W. Andrus**  
IBM, Endicott

- Computes the sums of squares, with the exception of the high-order interaction term, necessary in an analysis of variance.
- 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 at most 16,500 individual digits in all data cells.
- Does not apply.
- Storage required is approximately 341 locations, 0000 to 0340. Timing information not given.
- 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.
- Minimum 650.

### Auto-Correlation Program

**W. E. Andrus, Jr.**  
IBM, Endicott  
May 31, 1956

- 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.
- Arithmetic is fixed-point in the form $x.xxxx$.
- The standard formulas are used.
- Storage required for the program and load routines is 301 locations 0000 to 0300; data locations are 0600 to 1999. Timing is 6 to 109 seconds where $n$ is the total number of data elements.

### Errata

**“Polynomial of Best Fit by Least Squares Method,” by M. A. Kelly and M. S. Dyrkacz**  
April 2, 1956

The following error has been noted in the program deck:
- In part 1 of the deck, card 001 should have an 12-punch in column 1 in addition to the 7-punch.

Copies of the program deck furnished by the 650 Program Library on or after March 3, 1956, have been corrected.

### Multiple Correlation for 50 Variables

**J. D. Hall**  
University of Indiana, Bloomington  
May 31, 1956

- Computes all possible correlations (1225) of 50 variables of 3 digits each.
- The maximum number of observations for each variable is 10,000. Arithmetic is fixed-point.
- The standard formulas are used.
- Storage required is approximately 350 locations. Timing information not given.
- 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.
- Minimum 650.
IBM 650 Library Program


ERRATA

Page 41:

Page 42: The following instructions are missing at the bottom of the page.

Page 45: Decks supplied on or after May 1, 1958 include the appropriate changes shown above.

IBM 650 Library Program

POLLY: POLYNOMIAL FIT BY LEAST SQUARES

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

September, 1956

a) Obtains a least squares fit of a polynomial \( \sum_{i=0}^{N} a_i x^i \) to data obtained in a subcritical reactor. The relative activities of foils corrected or uncorrected for epithermal neutron background may be obtained.

b) Fixed point arithmetic is used.

c) Least squares.

d) Storage required is approximately 1550 locations. An average speed for an nth order polynomial with n data points.

f) Minimum 650.
AUTOCORRELATION AND POWER SPECTRUM

Esser Moso 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.

Minimum 650.

CORRELATION ANALYSIS WITH ANNOTATED OUTPUT

Staff, Scientific Computing Center
IBM, Washington
December 31, 1956

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

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

c) The standard formulas are used.

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

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

Alphabetic 650.

CHI SQUARE AND PHI FOR 2x2 CONTINGENCY TABLE

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

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

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

c) Standard formulas with option for correction.

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


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

A STATISTICAL INTERPRETIVE SYSTEM FOR THE IBM 650 MAGNETIC DRUM CALCULATOR

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

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

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

c) Does not apply.

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

e) The trigonometric functions and negative multiply have been removed and the following operations added: float, mean, covariance, $a^2$, $a^4$, random number, negative, gamma function, normal probability, Poisson probability, binomial probability, cumulative Poisson, cumulative binomial, $X^2$ test, $t$ test, if test, clear, store loop box, restore loop box, general exponentiation, and two statistical read commands.

f) Minimum 650.

RAP - A REGRESSION ANALYSIS PROGRAM

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

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

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

(Continued on next page)
c) Normal least squares techniques.

d) Program is in 2 parts, each of which uses the entire drum. Output from Part I in 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, Dept. of Education
W. H. Peirce, Numerical Analysis Laboratory
University of Wisconsin
Madison, Wisconsin

a. Purpose: Using an n x n (symmetrical) 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, 38 x 38.

Accuracy: Not given.

Floating/Fixed: Computation is in fixed point.

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 18 x 18 matrix which was processed took 14 hours to meet the conditions of the Lawley test.

Relocatability: Not given.

e. 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.

CURVE AND SURFACE FITTING ON EQUALLY OR UNEQUALLY SPACED POINTS

C. Hobby
A. Newhouse
L. Giezl
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 m of points allowed varies with the degree n ≤ 10 of the polynomial, e.g., for n = 2 or 3, m ≤ 99; n = 4, m ≤ 43.

Accuracy: Not given.

Floating/Fixed: Calculations are in floating point.


d. Storage Required: The entire drum is used.

Speed: Not given.

Relocatability: Not relocatable.

Machine: 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.

IBM 650 Library Program Abstracts

FACTOR ANALYSIS

When loading the "Woods Dispatched by Punch Drum Routine" deck, insertion hole 10 is not properly restored: This may be remedied by adding to the deck an extra card as shown below. Also it is necessary to add a wire on board 14, from AL-95 to C-84.

Decks received on or after March 1, 1964 have been corrected.

Addenda/Errata

IBM 650 Library Program Abstracts

CURVE AND SURFACE FITTING ON EQUALLY OR UNEQUALLY SPACED POINTS

C. Hobby
A. Newhouse
L. Giezl
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 m of points allowed varies with the degree n ≤ 10 of the polynomial, e.g., for n = 2 or 3, m ≤ 99; n = 4, m ≤ 43.

Accuracy: Not given.

Floating/Fixed: Calculations are in floating point.


d. Storage Required: The entire drum is used.

Speed: Not given.

Relocatability: Not relocatable.

Machine: 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.
IBM 650 Library Program Abstracts

MULTI-VARIABLE CORRELATION
K. Glaser
J. Taylor
General Electric Co.
Utica, New York

- 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.
- Range: The order of the matrix $n \leq 100$.
- Mathematical Method: The "multiple-correlation" is built from the simple correlation coefficient as described in Crowen & Cowden's "Applied General Statistics", Second Edition, Chapter XXI.
- Storage Required: Not given.
- Speed: The approximate computation time for a five variable correlation is twelve seconds.
- Relocatability: Not given.
- Remarks: The program may be used in conjunction with A. Cohen's "Multiple Regression Analysis", Phase II (File No. 6.0.011) or "Correlation Analysis with Annotated Output" using Option 9 (File No. 6.0.014).
- Special Devices: None.

LEAST SQUARES CURVE FITTING WITH ORTHOGONAL POLYNOMIALS
F. K. Chapman
Case Institute of Technology
Cleveland, Ohio

- Purpose: A best polynomial fit is obtained using orthogonal polynomials.
- Range: The present program is restricted to 100 points maximum and 10th degree maximum, for the sake of optimization. It may be easily changed to allow for perhaps 200 points or a degree of 20 or more.
- Mathematical Method: Cholesky's scheme is used.
- Storage Required: The program occupies approximately 750 drum locations and 60 IAS locations for maximum number of variables.
- Speed: Not given.
- Relocatability: Not given.
- Remarks: The program is self-restoring, hence may be used to obtain fits for many sets of data without reloading.
- Special Devices: None.

IBM 650 Library Program Abstracts

MODEM II
A. Oldehoeft
J. W. Hamblen
Oklahoma State University Computing Center
Stillwater, Oklahoma

- Purpose: To accept the output of COR IV (IBM 6.0.025) and build the entire "sum of squares" or correlation matrix in a manner such that it can be loaded with MA INV III (IBM 5.2.01). B. N. Carr
(Continued on next page)
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: Entire program is contained in IAS.
Speed: Not given.
Relocatability: Not given.
e. Remarks: None.
f. IBM 650 System: One 533, indexing registers, and IAS are required.
Special Devices: None.

**IBM 650 Library Program Abstracts**

**GENERAL LEAST SQUARES ANALYSIS**

J. Spector
Huntsville Arsenal
Dover, New Jersey

a. Purpose: Determines the polynomial of any degree up to 6 which best fits a set of observed data points.
b. Range: Determination of coefficients of polynomials up to 6th degree.
Accuracy: Not given.
Floating/Fixed: Floating point.
c. Mathematical Method: Does not require that all terms be present. Polynomials can be specified as having only odd powers, etc.
d. Storage Required: Requires approximately 1460 locations.
Speed: Speed is dependent upon the number of input data points being considered and the degree of the polynomial desired.
Relocatability: Not given.
e. Remarks: Program actually consists of two parts so that large quantities of data need not be kept on drum: Part 1 provides coefficients of the desired polynomial. Part 2 uses these coefficients to obtain calculated ordinates, residuals, and square-errors.
f. IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic are required.
Special Devices: Alphabetic device.

**IBM 650 Library Program Abstracts**

**THE WHERRY-WINER METHOD OF FACTOR ANALYSIS**

H. R. Brenner
Miss Frances Dallow
The Standard Oil Company of Ohio
Midland Building
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 communalities,
d. Storage Required: Part 1 (obtaining observations' subset scores and correlation between subsets) requires approximately 2,000 locations. Part 2 (obtaining item-subset correlations) requires 450 locations. Part 3 (calculating projections on group centroid vectors) requires 600 locations. Part 4 (an alternative procedure to obtain summations of inter-item correlations for each subset which failed to converge) requires 1100 locations.
Speed: Not given.
Relocatability: Not given.
e. Remarks: Maximum number of subsets 15
Maximum number of items in a subset 19
Maximum number of items in observations 300
f. IBM 650 System: One 533 required.

**IBM 650 Library Program Abstracts**

**FITTING OF THE GAMMA-DISTRIBUTION TO RAINFALL DATA**

W. T. Lewish
Iowa State College
Ames, Iowa

a. Purpose: This program will obtain the parameters \( q \) and \( y \) for the Gamma distribution.
b. Range: Input data must be in the form \( x, x', y < 10.0 \).
Accuracy: The parameters are accurate to four decimal places.
Floating/Fixed: Fixed point input and output.
d. Storage Required: Storage locations 1400-1999 are not used.
Speed: The input cards are read at 200 cards per minute.
Relocatability: Not in relocatable form.
e. Remarks: Special remarks are contained in the program description.
f. IBM 650 System: One 533 required.

**IBM 650 Library Program Abstracts**

**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 an on-line 407. TRAP is for the 650 equipped as 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 31 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 25 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)
MULTIPLE REGRESSION ANALYSIS

IBM 650 Library Program Abstracts

"Multiple Regression Analysis Programs: RAP, RAPA, TRAP" by J. E. Nichols.

The following correction has been submitted for the addenda sheet of the above writeup. It affects only the page entitled IDENTIFICATION OF CARDS; the card deck is accurate.

The column reading 7-001 - 7-025 Sample Data - TRAP, RAPA, RAP should be changed to read 7-001 - 7-075 Sample Data - TRAP, RAPA, RAP

IBM 650 Library Program Abstracts

MULTIPLE REGRESSION ANALYSIS

Mrs. Emma E. Iulo
State College of Washington
Pullman, Washington

a. Purpose: This program completes a multiple regression analysis and provides related statistics in concise form, utilizing a minimum number of control cards.

b. Remarks: Maximum number of variables is 25. Maximum number of observations is 9999. The maximum size of any single variable is eight digits. All output (except identification and number of observations) is in floating decimal notation.

Accuracy: Not given.

Floating/Fixed: Floating decimal.

Mathematical Method: See the program write-up.

Storage Required: The entire drum.

Speed: See timing chart in the program write-up.

Relocatability: Not relocatable.

(b. Remarks: Input data is checked for proper sequence of card number within observation number. Any number of selected independent variables may be eliminated from the regression equation, if desired. The program utilizes the "Matrix Inversion Routine I (MIR I)" by K. B. Williams, IBM 650 Library Program No. 6.0.012.)

f. IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic feature.

IBM 650 Library Program Abstracts

10 x 90 CORRELATION COEFFICIENTS

J. E. Farmer
Computing Center
State College of Washington
Pullman, Washington

a. Purpose: This program provides simple correlation coefficients and related data for up to ten dependent variables correlated with up to 90 independent variables.

b. 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 \( = \frac{2(1 + d) + 54 + 8i + 51d}{108} \) seconds per observation, where \( i \) is the number of independent variables and \( d \) is the number of dependent variables.

For punching, time required \( = \frac{(1 + d + 14i)2d}{60} \) seconds per problem.

Relocatability: Not relocatable.

e. 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.

f. IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic feature.
IBM 650 Library Program Abstracts

"10 x 90 Correlation Coefficients," by J. E. Farmer

The following correction has been submitted for the write-up, page 4, paragraph e., subparagraph 3. The last sentence there should be changed to read:

"If not, the unpacking routine must be loaded separately and behind the main program deck."

IBM 650 Library Program Abstracts

ANALYSIS OF VARIANCE OR COVARIANCE AND ADJUST MEANS PROGRAM

G. Ingram
State College of Washington
Computing Center
Pullman, Washington

a. 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.
b. Range: Maximum number of variables is six. Maximum number of observations is 9999. Maximum number of sources of variation is 60. All output is in floating decimal form. There can be no missing observations.

Accuracy: Not given.
Floating/Fixed: Floating decimal.
c. Mathematical Method: See the program write-up.
d. Storage Required: The entire drum is used.

Speed: See the timing table in the program write-up.

Relocatability: Not relocatable.
e. Remarks: None.
f. IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic feature.

---

CORRELATION ANALYSIS WITH ANNOTATED OUTPUT - PART III

Marlene Hirsch
The Service Bureau Corporation
Washington, D. C.

a. Purpose: This program computes the sums of squares necessary to compute an analysis of variance, as well as the mean and a measure of dispersion for each variable.
b. Range: Maximum number of variables is 99. Maximum number of observations is 9999. Maximum size of any single variable is eight digits. Maximum number of components (without special identification procedures) is 96. Corrected sums of squares for all interactions obtained are corrected with the grand total correction term only, and not for any main effects. One pass of input data through the machine is required for each component except "Total."

Accuracy: Not given.
Floating/Fixed: Floating decimal.
d. Storage Required: The entire drum.

Speed: Not given.

Relocatability: Not relocatable.
e. Remarks: Corrected sum of squares for any given level represents the "within" corrected sum of squares for that particular level. Zero may be significant or nonsignificant through use of a control card.
f. IBM 650 System: One 533, indexing registers and automatic floating decimal arithmetic feature.

---

1. IBM 650 System: One 533 and automatic floating decimal arithmetic feature are required.

CORRELATION ANALYSIS WITH ANNOTATED OUTPUT - PART III

---

IBM 650 Library Program Abstracts

RANDOM NORMAL DEVIATES

R. A. Conger
The Emerson Electric Mfg. Co.
St. Louis 21, Missouri

a. Purpose: This is a relocatable subroutine which will generate a random number upon entry. A sequence of these numbers produced by repeated entry will be approximately normally distributed with mean X and variance \( \sigma^2 \) supplied by the user. The Central Limit Theorem is utilized to produce a t-distribution with N degrees of freedom. The sequence is asymptotically pseudo-Gaussian as the value of N, supplied by the user, becomes increasingly large.
b. Range: \( -\infty < X < \infty \)

Accuracy: Does not apply.

Floating/Fixed: Floating decimal.
c. Mathematical Method: A sequence of uniformly distributed random numbers is generated by the multiplicative congruence method. A group of N of these is then added to produce a single random deviate having zero mean and unit variance. These random deviates are then modified so that they have mean \( \mu \) and variance \( \sigma^2 \). For most problems a value of 10 for N is sufficiently large. However, when sampling from the tails of the distribution is fairly important, \( \mu \) should be larger.
d. Storage Required: 35 storage locations are used.

Speed: The time required is approximately (25 N + 96) milliseconds.

Relocatability: Relocatable.
e. Remarks: Values of \( X = 0 \), \( \sigma^2 = 1 \), and \( N = 10 \) are incorporated into the program. The user need only change any of these which are unsatisfactory for his needs. A fourth parameter, \( R_0 \), which determines all subsequent random numbers generated by the subroutine, must be changed if different sequences are desired.

(Continued on next column)
PAIRED COMPARISONS FROM BALANCED INCOMPLETE BLOCKS

H. Gulliksen
L. Tucker
Educational Testing Service
and Princeton University
Princeton, New Jersey

a. Purpose: This program utilizes input data from a questionnaire involving 31 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.

c. Accuracy: Proportions are rounded to four decimals; The approximations for the normal deviate, the arc sine, and logistic have a maximum discrepancy of 0.0005 for proportions between 0.98 and 0.02.

d. Storage Required: The program uses 1,904 drum storage locations.

H. O. Hartley
W. T. Lewish
Iowa State University
Ames, Iowa

a. Purpose: This program computes three decimal digit probabilities and is a sequel to "Fitting the Gamma-Distribution to Rainfall Data" by H. O. Hartley and W. T. Lewish (file #6.0.023).

b. Range: The parameter \( \gamma \) must be less than 100, but \( \eta \) must be greater than 0.2 and be less than 100.

c. Mathematical Method: For \( \gamma < 1.9 \) probabilities are computed from a stored table of the incomplete Gamma function. Linear and/or quadratic interpolation is used within the table. For \( \gamma > 7.9 \), Wilson-Hilferty approximation, requiring a table of Normal Probabilities, was used.

d. Storage Required: Entire drum is used.

e. 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 cards are in consecutive numerical order and that each item contains some permutation of the rank orders 1 to 6. Errors here may produce misleading results.

IBM 650 System: One 533 required.

ORTHOGONAL POLYNOMIAL CURVE FITTER

E. McCauley
J. Kocher
Wayne State University
Detroit, Michigan

a. Purpose: The program finds least square polynomial of 1 point to degree m.

b. Range: \( 2 \leq m \leq 99; 1 \leq m \leq 19 \).

c. Accuracy: The coefficient output is computed to double precision accuracy.

d. Storage Required: The program uses 1,964 drum storage locations.

IBM 650 System: One 533 required.

SEASONAL ADJUSTMENT OF ECONOMIC TIME SERIES

S. H. Haeckel
IBM, St. Louis, Mo.

a. Purpose: This program is 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.


d. Storage Required: The entire drum is used.

e. Remarks: The original source program was written in FOR TRANSIT, and may take considerable time to execute.

IBM 650 System: One 533 required.

PROGRAM TO CALCULATE SEASONALLY ADJUSTED INDICES

W. Moli
Prudential Life Insurance Company
Jersey City, New Jersey

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.

(Continued on next page)
b. Range: The program will process series of from 9 years through 21 years duration. No original observations may be missing.
Accuracy: Final moving seasonal indices to 0.1%
Floating/Fixed: Fixed decimal arithmetic is used.
 Mathematical Method: The method is a modification of the Bureau of Census Method I.
Storage Required: The entire drum is used.
Speed: 10 year series (120 input items) - approximately 6 minutes.
21 year series (260 input items) - approximately 15 minutes.
Relocatability: Not relocatable.
Remarks: Due to storage space requirements, it is necessary to reload the data lines with each series to be adjusted.
IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

MINIMAX POLYNOMIAL APPROXIMATION ON A FINITE POINT SET

D. W. Marquardt
Mary Ann Bornfeldt
E. L. duPont de Nemours & Co., Inc.
Wilmington, Delaware

a. Purpose: To compute the polynomial of specified degree n which approximates to the minimum sense in a finite set of points (values of some function f(x) on a finite interval).
b. Range: Up to 100 values of f(X); where the x,, 1=1, 2, ..., N may be spaced as desired on any finite interval.
Degree of polynomial: 1 <= n <= 12
Accuracy: Program normalizes range of x to -1 <= x _<= 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. Fitch, as described in the Preprints of papers presented at the June 1954 meeting of the Association for Computing Machinery, pages 23-1 to 23-9.
Storage Required: Most of drum, all of immediate access storage.
Speed: Depends upon N, n, and number of iterations required.
Typical cases: N = 33 n=3 Time =3 min.
N = 60 n=3 Time = 5 min.
N = 90 n=5 Time = 5 min.
N = 120 n=5 Time = 6 min.
Relocatability: Not relocatable.
Remarks: Output includes: coefficients of minimax polynomial, minimum error of the approximation, normalization constants. Utility board is used.
IBM 650 System: One 533 (or one on line 407), indexing registers, 1AS and automatic floating decimal arithmetic feature.

IBM 650 Library Program Abstracts

AN ANALYSIS OF VARIANCE PROGRAM FOR THE IBM 650

J. W. Johnson
Canadian Army Operational Research Establishment
Ottawa, Ontario
Canada

a. Purpose: This program calculates the analysis of variance table including the components of variance for crossed, nested, or mixed experiments with three or fewer factors.
b. Range: The restrictions imposed by use of this program are:
   - (number of digits in X, x, y, z) 19
   - (number of digits in p, q, r, s) 23
   - The sizes of p and q are restricted only by word size.
   - The number of replications must be constant.
Accuracy: Double precision arithmetic is used in summing squared terms to preserve accuracy.
(Continued on next column)
Floating/Fixed: Fixed decimal arithmetic.
Storage Required: Not given.
Speed: The example problem required about 15 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 0 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 call sums of squares and variances.
IBM 650 System: One 533 required.
Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

COMPLETE PAIRED COMPARISONS SCHEDULE

PARCOPLT-2.21

Herald Gulliksen
Psychology Department
Princeton University
Princeton, New Jersey

a. Purpose: This program utilizes input data from a paired comparisons questionnaire of N objects or less (with or without the Like-Dislike section) and processes the summary data for each subject and the total values. The detail paired comparison matrix may be punched out or omitted as desired.
b. Range: Accuracy: Floating/Fixed: The program will handle a maximum of 1999 subjects in a single group. Fixed point is used throughout.
Proportions are rounded to least decimals. The approximation for the normal deviate, arc sine, and logistic have a maximum discrepancy of .005 for proportions between .00 and .50.
c. Mathematical Method: The least squares solution for scale values is used. Scale values are computed, using the normal deviate, the arc sine, and the logistic transform.
Storage Requirements: Speed: Relocatability: The analysis program utilizes 1024 drum locations, and is not relocatable. Depending on the number of stimuli in the questionnaire the program requires each subject to obtain three to 15 seconds and the final paired comparisons computations for the total group take five to six minutes.

Additional Remarks: Precautions at Restrictions: It is desirable to see the utility checking program to insure that the input cards are in correct form. This program checks to see that the cards are in consecutive numerical order and that each item response is a 1 or a 2. Errors here may produce misleading results.
Equipment Specifications: It requires the minimum 650 installation and use the standard 60-60 board for eight ten-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

a. Purpose: This program performs the multiple regression analysis under the hypotheses:

\[ y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n \]

The x's are the observable independent variables, the y is the observable dependent variable, and the b's, called the regression coefficients, are the constants to be estimated.
b. Range: Not given.
Accuracy: Not given.
Floating/Fixed: All input data must be described by six digit fixed point numbers of the form XXX.XXX.
c. Mathematical Method: The method used is a standard one for multiple regression analysis. Details are contained in the program write-up.
d. Storage Required: This program utilizes the entire drum and high speed storage.

(Continued on next page)
IBM 650 Library Program Abstracts

**SIMPLE CORRELATION - COR**
Numerical Computation Laboratory
Ohio State University Research Center
Columbus 18, Ohio

a. **Purpose**: COR computes simple correlations between two variables, \( r_{xy} \) and \( r_{yz} \). Results include means, sums of squares, sums of cross-products, means, standard deviations, variance, covariance, correlation coefficient, and its square.

b. **Range**: This routine will handle up to 60 variables at a time and compute up to 211 correlations.

c. **Accuracy**: Not given.

d. **Floating/Fixed**: Fixed point data forms - see write-up for details.

e. **Mathematical Method**: The computations of COR are based on the formula:

\[
12 \left( \frac{\sum x_i \sum y_i}{N} \right) - \frac{\left( \sum x_i \right)^2}{N} - \frac{\left( \sum y_i \right)^2}{N}
\]

f. **Storage Requirements**: COR occupies essentially the entire drum.

g. **Speed**: Time required for accumulation of sums is approximately (in minutes)

\[
\text{Drum time} = 62.5 \times \frac{k}{60} \quad \text{where} \quad k = \text{number of variables} \quad h = \text{number of observations} \quad c = \text{number of correlations}
\]

h. **CorrelationTime required for approximation (in seconds)**

\[
i = \text{where} \quad n = \text{number of correlations}
\]

i. **Remarks**: Not relocatable.

j. **Remarks**: See write-up for restrictions of input deck.

k. **650 System**: Minimum 650; no special equipment required.

---

IBM 650 Library Program Abstracts

**GENERAL TABULATION PROGRAM**

V. H. Nicholson
Agricultural Marketing Service
U.S. Dept. of Agriculture
Washington 15, D. C.

**Purpose**: The purpose of this program is to tabulate any desired field of 10 digits or less controlling on minor, intermediate, and major fields.

---

IBM 650 Library Program Abstracts

**CALCULATION OF THE CROSS-CORRELATION FUNCTION AND THE SPECTRAL DENSITY**

M. V. D. Mihaileni
Battelle Memorial Institute
505 King Avenue
Columbus 18, Ohio

**Purpose**: This computer program computes the cross-correlation function and the spectral density. The program is divided into two phases as follows:

Phase I: Calculation of the auto-correlation function, \( R(x) \)
Phase II: Calculation of the spectral density, \( W_1(w) \)

The two phases are used independently. The output of Phase I is the input for Phase II.

**Limitations of Program**:

b. **Range**: Phase I - The input data must not exceed 1350 in the range -1000 - 1000, where the decimal point may be arbitrary. The number of observations, \( N \), must be less than 10,000.

**Phase II** - The range of the discrete variable \( K \) must be less than 1350. In general the range of \( K \) is defined as \( 0 \leq K < 1350 \).

**Accuracy**: Phase I - The mean value is calculated to the same number of significant figures as the given function. The cross-correlation function is computed to one more significant figure than the given input.

**Phase II** - The spectral density is evaluated to one more significant figure than the cross-correlation function.

c. **Mathematical Method**: Formule are given in the write-up.

d. **Storage Requirements**: Phase I - Approximately 500 drum locations are used.

**Phase II** - Almost the entire drum is used. Locations 6000-11500 are, however, reserved for storage of input data. For open memory locations of both phases see the availability table included in the write-up.

e. **Speed**: Computation speed of the computer program is dependent on the number of input data. Approximate formulae are given in the write-up.

f. **Remarks**: Storage is not required.

---

IBM 650 Library Program Abstracts

**CALCULATION OF THE CROSS-CORRELATION FUNCTION AND THE CROSS-SPECTRAL DENSITY**

M. V. D. Mihaileni
Battelle Memorial Institute
505 King Avenue
Columbus 18, 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: Calculation of the cross-correlation functions \( R_{xy}(K) \) and \( R_{xz}(K) \).

**Phase II**: Calculation of the cross-spectral density, \( W_{xy}(w) \).

The two phases are used independently. The output of Phase I is the input for Phase II.

**Limitations of Program**:

b. **Range**: Phase I - The input data must not exceed four significant digits over the range 0.2 ± 0.04, 0 <= \( K <= 1000 \) where the decimal point may be arbitrary. The number of observations, \( K \), must be

---

IBM 650 Library Program Abstracts

---
FURTHER ANALYSIS OF THE TWO PARAMETER GAMMA DISTRIBUTION WITH SPECIAL REFERENCE TO RAINFALL DATA

M. O. Hartley
W. T. Leath
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 covariances.

b. Range: Input - 4 digits or less and less than 20,000 observations

   Output: $\alpha$, $\beta$, and 2,000

   Variance and covariance scaled: 1

Accuracy: $H_{\alpha}(5.772)$ maximum error $= 0.00005$

   $H_{\alpha}(5.772) \leq x$ maximum error $= 0.006$

   for additional information see reference in the program description.

Floating/Fixed: All calculations in fixed.

c. Mathematical Method: Greenwood and Dunord's polynomial approximations to the maximum likelihood method.

d. Storage Requirements: Entire drum (2,000 words).

   Speed: 4 digits input data about 170/min. 3 digits or less at 208/infra.

   Punch loop of about 2 seconds.

e. Remarks: Test example and answers contained in description.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

54 X 54 CORRELATION COEFFICIENTS

James E. 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 error of any variable is eight digits (positive or negative).

(Continued on next column)
MISSING DATA CORRELATION COEFFICIENTS
James E. Farmer
Computing Center
Washington State University
Pullman, Washington

- **Purpose**: This program computes and prints the F-value, regression coefficients, and related statistics for all combinations of up to 23 variables. It handles missing data as non-significant or missing data, the zero variable and its pairs are eliminated from the computations for this observation. The program makes maximum utilization of data not missing (4, 9).

- **Range**: Maximum number of variables is 23. Maximum size of any variable is eight digits (positive or negative).
- **Floating/Fixed**: Floating decimal.
- **Accuracy**: Not specified.
- **Storage Requirements**: Entire 2,000-word drum is used.
- **Speed**: Timing approximation: Input—second/observation: 2.08. Output—seconds = 0.75 \( \frac{(V)(V-1)}{2} \) where \( V \) is number of variables.

IBM 650 Library Program Abstracts

ANALYSIS OF VARIANCE, DISPROPORTIONATE SUBCLASS NUMBERS
Glenn R. Ingram
Assistant Computing Analyst
Washington State University
Pullman, Washington

- **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.

- **Restrictions, Range**: No restrictions except those required by the floating point device.
- **Accuracy**: Not specified.
- **Floating/Fixed**: Floating point arithmetic is used.
- **Method**: The method of “fitting constants” is used.
- **Storage Requirements**: The entire 2,000-word drum is used.
- **Speed**: Speed is a function of the number of factors and number of levels within factors.
- **Relocatability**: Not relocatable form.
- **Remarks**: (1) This routine used IBM 650 Library Program No. 05.3.02, Matrix Inversion Routine. (2) Special remarks are contained in the program write-up.

IBM 650 Library Program Abstracts

ANALYSIS OF VARIANCE OR COVARIANCE FOR NON-ORTHOGONAL DATA AND FOR ANY STATISTICAL DESIGN
John R. Isowell
Agricultural Experiment Station
University of Florida
Gainesville, Florida

- **Purpose**: In writing a general analysis of variance program, one is confronted with the problems of (1) devising a general systematic scheme for retrieving from the computer storage the elements that occur in each of the many sums necessary for the analysis desired, (2) making the program general enough to be useful for analyzing the data from as many possible non-orthogonal designs as possible and (3) providing for the situation where there are missing data or unequal subclass numbers. The purpose of this program is to analyze the variances in such a way that all three problems stated above are answered.

- **Range**: All computations are in double precision fixed-point arithmetic. Some of squares can be obtained to approximately 16 significant digits.
- **Mathematical Method**: The mathematical method used in adjusting for disproportionate frequencies (involving a set of least squares equations) is an iterative scheme which does not require that the matrix of coefficients be stored in the computer. For this reason up to 300 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 page)
IBM 650 Library Program Abstracts

**CONTOUR CODE FOR THE IBM 650**

L. N. Shapiro & W. W. Marks  
IBM Corporation  
3244 Wiltshire Blvd, Los Angeles 5, California

- **Purpose:** The Contour Code for the IBM 650 accepts data in three coordinates (x, y, z) and yields contour (or representative) lines for given z values.
- **Range:** Accuracy: Floating/Fixed: The range for the results are as follows: Interpolation - Full range (no limit) Extrapolation - Limit is a function of the data. The accuracy for linear interpolation is dependent on the significance of the data. A trial run involving an exponential, trigonometric function showed an average interpolation error of 2.45 fixed point arithmetic is used exclusively.
- **Mathematical Method:** Linear algebra is the basis for the arithmetic.
- **Storage Requirements:** Speed: Availability tables are included for the Contour Code which requires three passes through the 2000 work 650, The time for a maximum problem (49 points) is 12 minutes for loading, calculating, and punching the first contour and 15 seconds for each additional contour.
- **Remarks:** None.
- **Equipment Specifications:** IBM 650 with Index Registers - Standard 80-80 board for 533.

IBM 650 Library Program Abstracts

**EXPANDED SIMPLE CORRELATION COEFFICIENT ROUTINE FOR THE BASIC AND AUGMENTED 650**  
F. P. Fisher  
International Business Machines Corporation  
3244 Wiltshire Blvd, Los Angeles, California

- **Purpose:** To provide the ability to obtain simple correlation coefficients of k dependent variables against several combinations of independent variables, to include: linear terms, quadratic terms and interaction terms. This information will serve as an aid in Regression Analysis by giving the analyst more information on which to determine the form of the regression equation.

(Continued on next column)

b. **Range:** All computations are performed in single precision floating point. There is no restriction on the amount of data that may be processed. The program is available in two versions:
   1. Up to 6 independent variables and one dependent variable.
   2. Up to 13 independent variables and one dependent variable.

The restriction on dependent variables is not rigid. Any of the independent variables could be dependent variables provided the output is interpreted accordingly.

- **Mathematical Method:** Notation and methods are largely derived from "Statistics in Research", by Bernard Ostle.
- **Storage Requirements:** Because FORTRAN was used as coding media, peculiar input or storage requirements were not determined. However, the following information about the test problem will serve as a guide:
  - **Problem:** k dependent variables
  - **Segment 2 input requires about** n output gives all two-way tables in conventional arrangement plus corrected sums of squares, mean 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 2 minutes plus 40-50 seconds per seven experimental values fed. Segment 2 runs 3-6 minutes per set of measured values.
- **Remarks:** Porttrans 1
- **IBM 650 System:** Basic IBM 650.

IBM 650 Library Program Abstracts

**CARP - A CORRELATION AND REGRESSION PROGRAM**

R. E. Bacon  
International Harvester Company  
Wisconsin Steel Works  
Chicago 17, Illinois

- **Purpose:** The program computes means, standard deviations, simple correlation coefficients, partial correlation coefficients, and multiple correlation coefficients. It is intended for use by experimenters who wish to add or subtract observations, interchange and remove variables, and combine results of problems of equal dimensions.
- **Range:** Up to 39 variables are permitted, of which any number may be designated as dependent.
- **Accuracy:** Not given
- **Floating/Fixed:** Data may be entered in 8HR, floating-point format. Interval operation and output are in 8HR floating-point.

(Continued on next page)
SOLUTION OF SIMULTANEOUS EQUATIONS

IBM 650 Library Program Abstracts

File no. 7.0.003
Mathematical Routines

C. Mathematical Method: Least squares.
D. Storage Required: The entire drum is used.
Speed: Reading time for a 9 variable observation is 2.5 minutes; for a 50 variable observation 1.32 hours are required. Calculation and output times are from 1 to 108 minutes, depending on size of problem.
Relocatability: Not relocatable.
E. Remarks: Transformations are accomplished on the input variables by IBM's FOR TRAMIT program of the IBM, Part I Transformation program (File No. 6.6.021).
F. 650 System: One 533 required. Alphabetic device if available.

ROOTS OF A FUNCTION OF A REAL VARIABLE

IBM 650 Library Program Abstracts

File no. 7.0.002
Mathematical Routines

E. Mathematical Method: Bisection.
F. Storage Required: The entire drum is used.
Speed: Processing time required is approximately 1 second per interval for each variable.
Relocatability: Not given.
G. Remarks: The method fails in the following cases. However, these cases give additional information about the problem, as failure indicates one of the following:
1) No 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 1533 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.
H. IBM 650 System: One 533 indexing registers, and automatic floating decimal arithmetic feature.

RUNGE-KUTT A ROUTINE FOR SOLVING DIFFERENTIAL EQUATIONS ON THE IBM 650

IBM 650 Library Program Abstracts

File no. 7.0.005
Mathematical Routines

A. S. Rosenthal
Naval Air Development Center
Johnsville, Pennsylvania

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 to a specified fixed location. Information such as number of equations, expected duration of problem, allowable terminal error, and initial conditions is supplied to the system by the programmer. The system then computes, choosing its own time intervals and processing variables and derivatives at each time interval.

Range: The routine solves up to 35 simultaneous first order ordinary differential equations.

Accuracy: The routine provides automatic time interval control designed to keep total estimated accumulated errors in certain of the variables designated by the programmer.

Floating/Fixed: Floating decimal arithmetic is used.

Mathematical Method: Integration is by standard Runge-Kutta formulas. Special formulas are derived for error estimation.

Storage Required: The main program uses 178 drum storage locations, plus the number used to compute f(x). Additional storage is required.

Remarks: In addition to the main program the system contains an input-output routine, a program which allows reading or punching any chosen number of drum locations as a fixed increment. The input-output routine, which is extremely flexible, may be used independently, as well as with the system.

IBM 650 System: None and indexing registers required.

Special Devices: None.
**IBM 650 Library Program Abstracts**

**ZEROS OF COMPLEX POLYNOMIALS**

Lawrence S. Andrews

Technical Staff

Greenwich Engineering Division

American Telephone & Telegraph Company

a. Purpose: This subroutine will find the complex roots of the general


degree n equation of the form:

\[ a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0 \]

where the coefficients are complex numbers, \( a_n \neq 0 \) and \( n \geq 20 \).

b. Range: \( N \) must be less than or equal to twenty.

c. Mathematical Method: Successive approximations toward a particular root are obtained by limiting 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. Roae, 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 \( \text{of Compound Interest} \)

2. Present Value of 1 at Compound Interest

3. Amount of an Annuity of 1

4. Present Value of an Annuity of 1

5. The Annuity That 1 Will Purchase

b. Range: 9X10-10 \( \leq x \leq 9X10,9 \)

c. Mathematical Method: Refer to "The Functions En(x)

f. IBM 650 System: Minimum 650.

**IBM 650 Library Program Abstracts**

**FLOATING POINT SQUARE ROOT SUBROUTINE**

Texas Technical

Hiroshi Takahashi

Japan Atomic Energy Institute

Tokai, Ibaragi Pref., Japan

a. Purpose: To compute any of the following functions: \( x, \exp(x), \sin(x), \cos(x), \tan(x), \cot(x), \sec(x), \) and \( \tan(x) \).

b. Range: The range of the argument must be: \( -5.00 \leq x \leq 5.00 \).

Accuracy: Whenever any term of the infinite sum becomes less than \( 10^{-7} \), the subsequent terms are neglected.

Floating/Fixed: The computation is done in fixed point arithmetic.

c. Mathematical Method: Refer to "A Short Table of the Functions \( K_n(x) \),

from \( n = 0 \) to \( n = 6 \)" by W. G. Bickley, D. Sc., and John Hayler, A.C.G.I.,


d. Storage Required: 500 locations (0000 through 0499) are used.

Speed: The average execution time is as follows:

- For \( \text{ln} x \) 1 sec.
- For \( \exp (x) \) 2 sec.
- For \( \sin (x) \) 2 sec.
- For \( \cos (x) \) 2 sec.
- For \( \tan (x) \) 6 sec.
- For \( \cot (x) \) 6 sec.
- For \( \sec (x) \) 6 sec.

Relocatability: Non-relocatable.

e. Remarks: IBM 650 Library Program #3.2.002 for \( \ln x \) is incorporated as a subroutine.

f. IBM 650 System: Minimum 650.

**IBM 650 Library Program Abstracts**

**NUMERICAL INTEGRATION OF THE DOUBLE INTEGRAL**

A. Anastasio

C. Cassidy

Columbia University

Stevens Institute of Technology

A. Anastasio

New York, N.Y.

a. Purpose: To approximate the integral having the general form

\[ \int_{a}^{b} \int_{c}^{d} f(x, y) \, dx \, dy \]

b. Restrictions, Range: Region of integration over the annulus with outer

radius \( R \) and inner radius \( r \).

c. Mathematical Method: Numerical Integration over the Planar Annulus, a method by

H. Peirce.

Accuracy: The subsequent terms are neglected.

Floating/Fixed: The computation is done in fixed point arithmetic.

d. Storage Requirements: Does not apply.

e. Remarks: None.

f. IBM 650 System: Uses floating point and index register.

**IBM 650 Library Program Abstracts**

**FLOATING POINT SQUARE ROOT SUBROUTINE**

Charles Goldberg

IBM 650 Applied Programming

Time & Life Building

New York, New York

a. Purpose: This routine computes the square root of numbers in floating

point format using an initial approximation and five iterations with

Newton's method. This program was designed to use a minimum of drum

space.

(Continued on next page)
a. Purpose: This program generates a set of random numbers. Answers are in floating point form and all significant digits are exact.

b. Range: Up to 99,999 numbers may be generated by each computer pass.

c. Mathematical Method: 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: 25 Permanent drum locations including a programmed area for negative arguments. Temporary storage locations are used.

Speed: 160 ms.

The deck is in SOAP II form.

Remarks: The routine uses index register B which is not reset.

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

IBM 650 Library Program Abstracts

CLEBSCH-GORDAN COEFFICIENT SUBROUTINE

S. C. C. Computer Science Institute

Purpose: The subroutine computes the Clebsch-Gordan or vector-coupling coefficients \( C(J_1 J_2 | m_{12} m_2 m_{12}) \).

Range: \( |J_1 J_2 | m_{12} m_2 m_{12} \leq 100 \).

Accuracy: /part/ in 8th decimal place.

Mathematical Method: Not applicable.

Storage Required: 35 consecutive locations are required. The subroutine is written in SOAP-II relocatable format.

Remarks: None.

IBM 650 System: Minimum 650 with alphabetic unit (minimum SOAP requirements).

IBM 650 Library Program Abstracts

PYRAMID OF RANDOM NUMBERS

John Burgesson, Robert Bashor

IBM

Purpose: This program generates a set of random un-correlated numbers which span a predetermined range or field size.

Range: Up to 99,999 numbers may be generated for each computer pass. Any field size from a minimum of five "cells" may be used. Normal use of the program calls for a field size of CC columns ( \( 5 \leq CC \leq 99 \) by 12 rows, the "cells" being numbered 000 to 10CC-1).

Mathematical Method: Does not apply.

Storage Required: About 650 words of memory optimally scattered in lower memory.

Speed: Depends on field size used and the number of 3-above-3-numbers desired. Usually runs close to 1/2 punch speed.

Reliability: The program deck is furnished on SOAPed single instruction load cards and is therefore relocatable by further SOAPing.

Remarks: 1. The program is furnished in SOAP form so that modifications can be made easily.

2. This program was designed to give a "dictionary" of numbers for use in an information retrieval system centering about 300. It is possible to generate a set of 3-above-3-numbers, use them, then run the program again, obtaining a new and completely different set of 3-above-3-numbers, none of which duplicate any number in the first run. For practical applications, this process can repeat itself indefinitely.

IBM 650 System: Minimum 650.

IBM 650 Library Program Abstracts

COMPLEX I

AN INTERPRETIVE PACKAGE FOR COMPLEX ARITHMETIC

(Located on next column)
IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.
e. Remarks: None

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.

IBM 650 Library Program Abstracts

PRESENT VALUE AND RATE OF RETURN (PVAR)

Martin R. Solomon, Jr.
University of Kentucky

a. 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.
b. Range: Life can range from 1 to 50 years.
c. Mathematical Method: PV = \sum_{i=1}^{n} \frac{R_i}{(1+r)^i} = \frac{R_1}{1+r} + \frac{R_2}{(1+r)^2} + \cdots + \frac{R_n}{(1+r)^n}
d. Storage Required: Optimized by SOAP II so program is scattered throughout drum.

Speed: Computes present value in a few seconds. Rate of return is computed by successive approximations. Requires about 6 seconds for each percent computed.
Relocatability: Not relocatable.
A Ray Tracing Program

J. May
Columbia University
Hudson Laboratories
Debra Ferry, N. Y.

a. Purpose: Trace the path of a ray in a layered inhomogeneous medium with regular boundaries.

b. Range: Maximum of 40 different velocity points.

Floating/Fixed Floating Point Arithmetic

c. Mathematical Method: Snell's law is used at the boundaries between
sections. J. L. Webber, Hudson Laboratories Technical Report No. 47
dated June 4, 1957.

d. Storage Requirements: Approximately 150 unused drum locations.

Speed: Depends upon number of layers. Up to 100 points per minute.
Reliability: Not relocatable.

Remarks: None.

Special Devices: Automatic Floating Point, Three Indexing Registers.

IBM 650 Library Program Abstracts

File No. 8.1.004

Solutions of Heat Diffusion Equation

R. R. Herber
Theoretical Physics Division
E. L. du Pont de Nemours & Co.
Savannah River Laboratory
Aiken, S. C.

a. Purpose: Equations and a routine are presented to obtain the temperature distribution in a section of a spherical heat source. The solution of the heat diffusion equation in spherical geometry is approximated by the solution of a set of appropriate difference equations. Three regions with possible inhomogeneity in the radial direction, e.g., inner cladding, fuel, and outer cladding, lead to a form of a spherical shell with radial inhomogeneity. The program can be used to study the effects of inhomogeneity in the surface heat transfer and in the fuel source.

b. Range: Floating.


d. Storage Requirements: 2000 locations. Speed depends on number of grid points used.

Remarks: Not given.

IBM 650 System: Model 2 with Floating decimal & index registers.
560 LIBRARY PROGRAM ABSTRACT  
FILE NUMBER 8.2.004

LOST, A CROSS-SECTION AVERAGING PROGRAM

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

a) Computes cross-section integrals over specified lethargy groups.
b) Input is in floating-point form. The maximum number of lethargy points is 50.
c) Integrations are performed using the trapezoidal rule.
d) Storage required for the program is 424 locations, 1571 to 1994. The rest of the drum is used for data storage. Time required for a typical computation with six materials and self-shielding for 170 point and 15 point files is 12.5 minutes and 1.24 minutes respectively.
e) The program distinguishes between the absorption of moderator or non-fissionable materials and those of fissionable or associated fission product materials.
f) Minimum 650.

560 LIBRARY PROGRAM ABSTRACT  
FILE NUMBER 8.2.005

DONATE

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

May 1956

a) Distribution of neutrons at thermal energies - a solution for the energy distribution of neutrons in equilibrium with an infinite homogeneous medium of pure monatomic hydrogen undergoing thermal motion. Allowing varying cross sections, elements other than hydrogen and a buckling turn for leakage from a finite volume.
b) Floating point.
c) Milne's Predictor-corrector formulas, 3 point Lagrangian interpolation, 5 and 6 point integration formulas.
d) 3 runs.
e) None.
f) Minimum 650.

560 LIBRARY PROGRAM ABSTRACT  
FILE NUMBER 8.2.006

MUFT III

R. L. Hellens
R. W. Long, and R. H. Mount

July 1956

a) Computes the energy distribution of neutrons, having a given Fourier mode in an infinite medium.
b) Four approximations are provided with the inclusion of isotropic and anisotropic scattering, resonance capture, and fast fission. Fixed point arithmetic is used.
c) The output includes flux, current, and slowing density spectra and computes the fast constants for a variety of few group schemes.
d) Solution requires two runs through the computer. The entire drum is used.
e) Twenty is the maximum number of elements that can be used as input for any one problem.
f) Minimum 650.

560 LIBRARY PROGRAM ABSTRACT  
FILE NUMBER 8.2.007

LIL ABNPR: A FEW-GROUP ONE-DIMENSIONAL CODE

H. Bohl
G. Gelbard
R. Suarez

a) LIL Abner is a one-to-eight group code designed, primarily, to treat one-dimensional reactor and cell problems.
b) This code will handle a maximum of ten regions and one hundred mesh points. Floating point arithmetic is used.
c) The method is an iterative process.
d) None.
e) All physical parameters in the Few-Group equations as well as the mesh width must be constant within each region. In the fast groups these parameters may be obtained directly from MUFT III (8.2.006) calculations or from microscopic cross sections fitted to match MUFT III results. Sample problem is enclosed.
f) Minimum 650.

560 LIBRARY PROGRAM ABSTRACT  
FILE NUMBER 8.2.008

K-CODE

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

December, 1955

a) Obtains the transients of neutron flux in response to a change in the reactivity of a reactor.
b) Eleven delayed groups of neutrons and two power coefficients of different relaxation times are allowed. Floating decimal arithmetic is used.
c) Theoretical treatment is given in a paper by B. D. Brown, submitted for the journal "Nuclear Science and Engineering" under the title, "A General Treatment of Flux Transients."
d) Storage required is approximately 1600 locations. One time increment requires 20 seconds.
e) A very general change in reactivity as a function of time can be made by proper input parameters. The set of differential equations is solved by integration of the associated difference equations.
f) Minimum 650.

560 LIBRARY PROGRAM ABSTRACT  
FILE NUMBER 8.2.009

BEEHIVE AND HORNET
REACTOR CODES FOR SPHERICAL GEOMETRY

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

a) "Beehive" is a five energy group, two region, time independent, spherical reactor code. It considers the problem of a reactor system in which the core material is assumed to be at a higher energy (temperature) than the reflector. The companion code, "Hornet," computes the neutron fluxes for the critical assembly determined by the Beehive calculations.
b) The majority of arithmetic is performed in interpretive floating point.
c) The code obtains a closed solution for the critical reactor assembly by a procedure which is a logical extension of normal two group theory. The solution is obtained by an iterative process.

(Continued on next page)
**UNCLE I**

THE DIFFUSION EQUATION IN CYLINDRICAL GEOMETRY

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

a) UNCLE I - Solution of the Neutron Diffusion Equation in Cylindrical Geometry.
b) Uses network of 9 points in the r-direction and 16 in the z-direction. Fixed decimal.
c) Extrapolated Liebmann Method.
d) 20 seconds per iteration.
e) One group only.
f) Minimum 650.

**UNCLE II**

THE DIFFUSION EQUATION IN (x, y) SPACE

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

a) UNCLE II - Solution of the Neutron Diffusion Equation in (x, y) Space.
b) Uses network of 9 points in the x-direction and 16 in the y-direction. Fixed decimal.
c) Extrapolated Liebmann Method.
d) 20 seconds per iteration.
e) One group only. \( \frac{A_2}{x} \) at \( x = 0 \) is a restriction on the types of problems that can be solved.
f) Minimum 650.

**UNCLE III**

THE DIFFUSION EQUATION IN ONE DIMENSION

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

a) UNCLE III - Solution of the Neutron Diffusion Equation in One Dimension.
b) Uses network of \( K \times 1 \) points. Fixed decimal.
c) Extrapolated Liebmann Method.
d) 20 seconds per iteration.
e) One group only.
f) Minimum 650.
"P - 3 Flux Distribution," by J. W. Well and P. Cahal

Part I 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

Error:
P - 3 Flux Distribution," by J. W. Well and P. Cabral

Part I 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

**BALL**

**A Reactor Code for Spherical Geometry**

E. F. Stone
T. B. Kerr
University of California
Livermore, California

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

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

c) Iterative solution.

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

e) None.

f) Minimum 650.

650 Library Program Abstract

**NED**

**A Program for Computing the Wigner-Wilkins Kernel**

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

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

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

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

d) The program uses the whole drum and is not relocatable. For H moderator, sample calculations required 38 minutes. For R moderator, sample calculations required 58 minutes.

e) None.

f) Minimum 650.

650 Library Program Abstract

**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 8.2.008), 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 reactivity; (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.

c) Mathematical Method: Integration is accomplished by the fourth order Runge-Kutta.

d) Storage Required: Approximately 1930 storage locations are required.

e) Accuracy: Not given.

f) Minimum 650.

Special Devices: None.
ART-1

F. Narin
E. J. Voltaggio
Armour Research Foundation of Illinois Institute of Technology
Chicago 16, Illinois

a. Purpose: ART-1 evaluates the analytic solution of the equations describing the time dependent temperature distribution in a three region composite slab during a nuclear power excursion. The slab typifies that nuclear reactor fuel elements immersed in a coolant, and consists of a homogeneous heat source which varies exponentially with time, followed by two consecutive slabs of non-source material. Heat transfer is by conduction only.

b. Range: Not given.
Accuracy: Not given.
Floating/Fixed: Floating point arithmetic is used.
Mathematical Method: The equation for the conservation of neutrons is expressed in difference form as the matrix equation $N = KN$, which is solved by iteration.

Speed: The matrix $Q$ is obtained in about twenty minutes. Distributions with three digit precision are obtained with about twenty-five minutes of iteration.
Relocatability: Not given.
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 2.0 008.

Special Devices: None.

IBM 650 System: One 533 required.

f. IBM 650 System: One 533 required.

Special Devices: None.

RAYTHEON REACTOR SURVEY CODES 2G 2RI, 2G 2RII, AND 2G 3R

L. Holday
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 2RI and all values of km greater than 1 in 2G 2RII and 2G 3R.

Accuracy: Depends upon the number of iterations as determined by the comparison constant used.

Floating/Fixed: Floating point arithmetic is used.

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 2RI) or km (2G 2RII and 2G 3R) which makes the determinant equal to zero.

Speed: For 2G 2RI and 2G 2RII the running time is about 45 seconds per point. Loading time of interpretive system deck with program is 2.25 minutes.
Relocatability: Not relocatable.
Remarks: Since fixed point arithmetic is used, limits must be set on the input. Even with these limits, an overflow condition may occur. Also, many restrictions are placed upon the magnitudes of the parameters.

IBM 650 System: One 533 is required.

IBM 650 Library Program Abstracts

File no. 8, 2, 024

Physical Sciences

NEUTRON ENERGY SPECTRA IN WATER

J. C. English
E. I. du Pont de Nemours and Company
Waltham, Massachusetts

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.
Accuracy: Not given.
Floating/Fixed: Computation is in fixed decimal arithmetic.
Mathematical Method: The equation for the conservation of neutrons is expressed in difference form as the matrix equation $N = KN$ which is solved by iteration. The Rand fit to the erf function is used in the evaluation of elements of the matrix.

Storage Required: Not given.

Speed: The matrix $Q$ is obtained in about twenty minutes. Distributions with three digit precision are obtained with about twenty-five minutes of iteration.
Relocatability: Not given.
Remarks: The code as written assumes that the input parameters are in the range of those for $H_2O$ and $D_2O$ moderators.

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

File no. 8, 2, 021

Physical Sciences

ENSIGN CODE

B. L. Anderson
H. Sohl, Jr.
Bettis Atomic Power Division
Westinghouse Electric Corporation
Pittsburgh 30, Pennsylvania

a. Purpose: ENSIGN is a few-group, one-dimensional code designed to handle symmetric slabs, nonsymmetric slabs, and cylinders.

b. Range: Problems may not exceed 4 groups, 10 regions, and 100 points.

Accuracy: Not given.
Floating/Fixed: Floating point arithmetic is used.
Mathematical Method: Fluxes and eigenvalues are computed by means of an iterative scheme in which it is necessary to make an initial source 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.

Storage Required: The program requires 2000 words of storage.

Speed: The time required for a 2-group, 100-point, 1-iteration problem is 20 minutes.

Relocatability: Not relocatable.
Remarks: Since fixed point arithmetic is used, limits must be set on the input. Even with these limits, an overflow condition may occur. Also, many restrictions are placed upon the magnitudes of the parameters.

IBM 650 System: One 533 is required.

IBM 650 Library Program Abstracts

File no. 8, 2, 022

Physical Sciences
A MULTIGROUP P3 PROGRAM FOR THE NEUTRON TRANSPORT EQUATION

Richard B. Haseler
E. I. du Pont de Nemours & Co.
Exploratory Department
Atomic Energy Division
Savannah River Laboratory
Aiken, South Carolina

a. Purpose: An IBM 650 routine that computes the spherical harmonic approximation of the neutron transport equation in five energy groups, for rectangular, and for cylindrical geometry. The P3 approximation is used for the lowest energy group and the P3 approximation is used for the higher energy groups.


d. Storage Requirements: 2,000 words, 10 minutes/region.

e. Remarks: None.

f. IBM 650 System: Model 2 computer with automatic floating decimal and indexing registers.

IBM 650 Library Program Abstracts

LOC SURFACE FITTING PROGRAM FOR BASIC 650

W. C. Krantz
Department of Geology
Northwestern University
Evanston, Ill.

&

C. E. Faulkner
IBM, UK, Ltd.
London, England

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.

Accuracy: Double precision used in matrix inversion and computation of coefficients. Other computations in single precision.

Floating/Fixed: Input in fixed point, Program converts to floating point. Output 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 1 data card per 9 seconds. The output is in the form of 10 x 10, 6 x 6, 4 x 4, and 3 x 3 matrices and their corresponding vectors. Part II computes S, Q, and C matrices and outputs the coefficients at the rate of 10 minutes per mapped variable.

Part III computes 3 answer cards per data card every 4 seconds (Observed value, computed value, and deviation). Some of squares exceed at end.

Reliability: Not relocatable.

Remarks: Full description of data and output cards is program write-up.

f. IBM 650 System: Basic 650 and 533.
IBM 650 Library Program Abstracts  

STRUCTURE FACTORS  
B. Shiono  
University of Pittsburgh  
Pittsburgh, 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 the "Differential Fourier Synthesis" program (File No. 8, 4, 001). Six individual programs were prepared for centric and noncentric 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: 6  
   - Number of temperature factors:  
     1. Isotropic temp. factor for each kind: 5  
     2. Individual anisotropic temp. factor: 90  
   - Index of reflexions:  
     1. Centrosymmetric: 99  
     2. Non-centrosymmetric: no limit  

Accuracy: Not given.  
Floating/Fixed: Fixed point.  

Mathematical Method: Geometrical structure factors are computed with simplified expressions in the International Tables for X-ray Crystallography. Trigonometric functions are computed with Trigg's subroutine (IBM Technical Newsletter No. 9, 1956). Atomic scattering factors are stored in table form and linear interpolation is used.  

d. Storage Required: Most of the ZOOO storage locations are used.  

Speed: The following examples of speed are given:  
- P 31/4: 9 atoms, 2 kinds  
  3.5 sec/reflexion  
- P 21/4: 7 atoms, 7 kinds  
  8 sec/reflexion  
- P 3: 28 atoms, 2 kinds, anisotropic temp. factors  
  20 sec/reflexion  
Relocatability: Not given.  
Remarks: The necessary modification cards for each space group are listed (except for FddZ and Fddd).  
IBM 650 System: One 533 required.  

IBM 650 Library Program Abstracts  

"Structure Factors," by B. Shiono  

The following corrections have been submitted in the writing of the above program:  

<table>
<thead>
<tr>
<th>PAGE</th>
<th>LOCATION</th>
<th>LINE</th>
<th>WORD</th>
<th>SHOULD BE</th>
<th>ACTUAL</th>
<th>ERRATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0262</td>
<td>309</td>
<td>69 0214 0410 should be</td>
<td>69 0235 0410</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IBM 650 Library Program Abstracts  

DIFFERENTIAL FOURIER SYNTHESIS  
B. Shiono  
University of Pittsburgh  
Pittsburgh, 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 at a given coordinate, and solves the shifts 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.  

Speed: The following examples of speed are given:  
- P 1: 14 atoms, 1 kind  
  400 reflexions  
  approx. 40 minutes/atom  
- P 1/4: 1200 reflexions  
Relocatability: Not given.  
Remarks: The necessary modification cards for each space group are listed.  
IBM 650 System: One 533 required.
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 and 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.

---

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 9.2.005

MOMENT DISTRIBUTION

J. D. Hutchinson
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Computes the bending moments in structural members of a rigid frame, given fixed end moments.

b) Meets all engineering requirements. The program is written in fixed point.

c) The "Moment Distribution" method of Hardy Cross is used. (See Paper 1793, Trans, A.S.C.E., 1932.)

d) Program requires 540 memory locations; data require 10 words per member in the frame. Speed: 1/8 to 1/10 seconds per member per joint per iteration. Relocatability: Program is written in SOAP, but all data locations are in absolute.

e) Handles frames with up to 100 members. Not more than 8 members can meet at any given joint.

f) Minimum 650.

---

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 9.2.006

TRUSS ANALYSIS

A. A. Aucoin
J. D. Hutchinson
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Computes axial forces in statically loaded, simple, determinate, pinned trusses.

b) Range: Loads varying from 1 to 99999 (units arbitrary). Accuracy: Depends on number of significant figures in data; 1 part in 500 accuracy can be obtained on large trusses. Program is written in fixed point.

c) The "Method of Joints" is used. (See any standard text on truss analysis.)

d) The program requires 1200 memory locations; data require six locations per member. Speed: Approximately 3/4 seconds where J is the number of joints in the truss. Relocatability: Since the program and data occupy most of the drum, it is not convenient to relocate. The program is written in SOAP, however.

e) The program is self restoring and will process either many loading configurations for the same truss or many trusses, or any combination, in sequence, automatically. For indeterminate trusses, see Abstract 9.2.007, "Connector and Redundancy Programs for Indeterminate Truss Analysis."

f) Minimum 650.

---

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 9.2.007

CONNECTOR AND REDUNDANCY PROGRAMS
FOR INDETERMINATE TRUSS ANALYSIS

Irwin Tong
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Designed to compute true axial forces in all members of indeterminate trusses from output of "Truss Analysis" program.

b) Fixed point except the Sweeney Matrix Inversion routine which is incorporated.

c) Castigliano's Theorem of Least Work is applied. (See any standard text on indeterminate structures.)

d) The Connector requires 750 locations for program and data. The Redundancy Program requires 1725 locations for program and data. The programs are written in SOAP in fixed point except the Sweeney Matrix Inversion program which is incorporated.

e) Up to 24 redundants in a truss can be handled.

f) Minimum 650.

---

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 9.2.008

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.

April 1958, Bulletin 18 - 5
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.

ANALYSIS OF LATERALLY LOADED PILES

C. B. Rader, Sr.
C. R. Hobby
E. R. Organick

University of Houston
Computing and Data Processing Center
Houston, Texas

a) Computes lateral deflection, bending moment, shear, fiber stress due to vertical as well as horizontal loading, and soil pressure for 1 + 1 positions along a pile divided into t sections (t ≤ 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.
REVIEWED TRAVERSE AND TRAVERSE ADJUSTMENT COMPUTATION

J. A. Haller
California Division of Highways
Sacramento, California

a) This routine calculates traverse data for the typical highway survey, right of way, or design problem. Input is in the form of one card per course. Any two unknowns within a traverse may be accepted. Results are punched one course in 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 are considered optimum and is not relocatable form.

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 thousands of feet and bearings to seconds. Functions are computed to nine decimal places.

c) Library subroutines used are from Technical Newsletter #9 for sine, 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.

THE ENTIRE PROGRAM REQUIRES ABOUT 300 LOCATIONS, BUT THIS NUMBER MAY BE REDUCED IF THE PUNCHING PHASE IS SEPARATED FROM THE READING PHASE. THE PROGRAM SHOULD NOT BE RELOCATED EXCEPT TO SEPARATE PUNCHING FROM READING PHASES. SPEED VARIES WITH THE CONCENTRATION OF TRIPS WITHIN THE PARTICULAR SWATH BEING PROCESSED.

Reading of trip cards may be suspended and the trip values for each coordinate point may be punched out at any time so that the 650 does not need to be reserved for the entire time necessary to compute a given swath.

f) Minimum 650.

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 plot contour values. The listing may then be used to draw a contour chart of trip desires.

b) Up to approximately 1750 contour points may be posted in one pass of the trip cards. Coordinate boundaries for each pass must be set up.

c) The x and y coordinates of each point along a straight line from origin to destination are computed. The number of points computed for any one trip will be one more than the number of ordinates crossed by the longer axis of the trip.

(Continued on next column)
a. Purpose: This program will compute steel girder size and all other factors needed to complete the design of a concrete-steel composite girder.

b. Range: 138 plate sizes from 10" x 5/8" to 28 x 3-1/4" are available as trial sizes.

Accuracy: Not given.

Floating/Fixed: Fixed decimal arithmetic is used.

c. Mathematical Method: The routine picks a trial size of top and bottom flange, computes the stresses on each, and then modifies top and bottom flange sizes separately as a result of the test of the stresses. When both top and bottom flanges are within the proper stress band, the program computes reductions in flange sizes, end reactions, or shear stress, and punches results. A single card input produces a single card output for each beam to be designed. AASHO recommendations are observed.

d. Storage Required: Approximately 1700 locations of table, instruction, and temporary storage are used.

Speed: Varies, but the average beam will be designed in 25 to 60 seconds.

Relocatability: Not given.

e. Remarks: Provision is made to compute initial factors which are not specified by the engineer. The minimum data include span length, spacing between girders, structure depth, and steel stress. If other data are given, these data will be used in place of values computed from the minimum. When both top and bottom flanges are within the proper stress band, the program computes reductions in flange sizes, end reactions, or shear stress, and punches results. A single card input produces a single card output for each beam to be designed. AASHO recommendations are observed.

Plate girders without composite action may also be designed by the program.

f. IBM 650 System: One 533 required.

Special Devices: None.

*This program supersedes the original program bearing the same name and title number.

---

**650 LIBRARY PROGRAM ABSTRACT**

**FILE NUMBER** 9.2.021

**TRAVERSE AND COORDINATE PROGRAM**

**ENGINEERING APPLICATIONS**

**C. Travis**

**Washington Department of Highways**

**Computer Section**

**Olympia, Washington**

a) Using either Stations and Deflection Angles right or left, Length of Courses and Deflection Angles right or left, or Stations and Azimuths as input, the Bearings, Stations, Length of Courses, Course Lat., and Dips, and Coordinates of angle points are computed. Using P. I. Numbers and Coordinates as input, the Bearings, Delta Angles, and Length of Courses are computed. In all, fourteen different problem types are computed.

b) Coordinates CC, CCC, CCC, CC, Bearings N. or S. DDMMSS E. or W., Stations SSE/SSE, Deflection Angles DDMMSS R. or L., Delta Angle DDMMMS, P. I. Numbers PP, PPP, PPP, and Course Lengths LLL, LLL, Lll, (i), (i), Ll when using coordinates as input). The subroutines used are SR-3 (Square Root), SC-1 (Sine-Cosine), and AS-1 (Arcsine). Program is in fixed point.

c) Does not apply.

d) Storage required is about 1000 locations between 0000 and 1836. Speed is 60 courses per minute.

e) Program is written in SOAP.

f) IBM 650 with alphanumeric device.

---

**IBMM 650 Library Program Abstracts**

**FILE NO.** 9.2.022

**ENGINEERING APPLICATIONS**

**C. Travis**

**Washington Department of Highways**

**Computer Section**

**Olympia, Washington**

---

**IBMM 650 Library Program Abstracts**

**FILE NO.** 9.2.023

**ENGINEERING APPLICATIONS**

**C. Travis**

**Washington Department of Highways**

**Computer Section**

**Olympia, Washington**

---
SLOPE TOPOG PROGRAM

K. F. Kohler
K. E. DeClark
Bureau of Public Roads
Portland, Oregon

a. Purpose: Converts cross section slope topog (slope in percent or degrees and slope distance) to H. I. and rod topog.
b. Range: Input is Station (SSSS + SS), Base Elevation (EEEE.EE), and Rod Distance (DDDD.DDD). Output is Station (SSSS + SS), Elevation of contours, and Rod Distance from centerline (DDDD.DDD). The subroutines used are SC-1 (Sine-Cosine) and SR-3 (Square Root).

d. Storage Required: 1397 positions

IBM 650 System: One 533 required.

e. Remarks: None.
f. IBM 650 System: One 533 required.

SURVEY TRAVERSE PROGRAM

S. E. LaMota
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) Animeth

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 thousand.

Contour interpolation is ideally suited for plotting purposes. The output for each section is the station, the elevation and distance of the left-most topog point, all contours as elevation and distance from centerline that lie between the left-most topog point and centerline, the elevation of centerline, all contours as elevation and distance from centerline that lie between centerline and the right-most topog point, and the elevation and distance of the right-most topog point.

c. Mathematical Method: Does not apply.
d. Storage Required: Approximately 560 locations between 0000 and 1800 are required.

e. Remarks: None.

IBM 650 System: One 533 required.

IBM 650 System: One 533 required.

Special Devices: None.

CONTOUR INTERPOLATION

K. F. Kohler
K. E. DeClark
Bureau of Public Roads
Portland, Oregon

a. Purpose: This program develops the location of each contour within any highway topog cross section that is in the H. I. and rod distance form. The contour interval desired is selectable between 00.0 and 99.9 feet.

IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts
Accuracy: Output data is accurate to the nearest one-tenth foot.

Floating/Fixed: Computation is made in fixed point arithmetic.

c. Mathematical Method: The angle is first converted to an azimuth and then added to the previous sum. Latitudes and departures are computed with the use of the sine-cosine subroutine, SC 2.

d. Storage Required: Memory locations 1 - 50 and 200 - 400 approximately, are used.

Speed: Speed is approximately the maximum for card reading and punching.

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 Ohio Department of Highways.

b. Range: The maximum X value is 999.9 feet. The maximum Y value is 99.9 feet.

Accuracy: Values are rounded to the nearest tenth from the field notes. In the simple process of one subtraction of these values, the difference remains significant to the nearest tenth.

Floating/Fixed: The decimal is fixed in all calculations.

c. Mathematical Method: Simple arithmetic is used.

d. 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.

e. Remarks: None.

f. 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

ROAD DESIGN PROGRAM

T. S. Gemmell
Ohio Department of Highways
Columbus, Ohio

a. Purpose: Computes the coordinates of the surface of the pavement for Ohio Department of Highways.

b. Range: X ordinates have a maximum value of 100 feet, the Y ordinates are limited to ±10,000 feet.

Accuracy: All values are to the nearest 1/100th of a foot.

Floating/Fixed: Fixed point arithmetic is used.

c. Mathematical Method: The methods used incorporate analytical geometry plus comparison on design criteria.

d. Storage Required: 850 consecutive memory locations.

Speed: Punches at approximately maximum speed.

Relocatability: Not given.

e. Remarks: Program will compute through a station equation, allow shrinkage factor to apply to cut and fill, and will either compute or not compute seeding area.

f. 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

SUPERELEVATION TABLES

C. R. Caylor
Ohio Department of Highways
Columbus, Ohio

a. Purpose: Computes the super-elevations for the surface of the pavement for Ohio Department of Highways.

b. Range: The X ordinates have a maximum value of 100 feet, the Y ordinates have a maximum value of 10,000 feet.

Accuracy: All values are to the nearest 1/100th of a foot.

Floating/Fixed: Computation is in fixed point arithmetic.


d. Storage Required: 850 consecutive memory locations.

Speed: Punches at approximately maximum speed.

Relocatability: Program is relocatable by multiples of 50, plus the last 100 locations which cannot be transferred.

e. Remarks: None.

f. 650 System: One 533 required.

Special Devices: None.
DESIGN TEMPLATE PROGRAM
C. R. Caylor
Ohio Department of Highways
Columbus, Ohio

a. Purpose: Computes the design template for any given station.
b. Range: The maximum X value is 1000 feet. The maximum Y value is 10,000 feet.
d. Storage Required: 1099 consecutive memory locations.
e. Remarks: None.
f. 650 System: One 533 required.
Special Devices: None.

MOMENT DISTRIBUTION AND INFLUENCE LINE CALCULATION
P. Yeager
L. C. McReynolds
E. D. Lee
Computer Section
Washington State Highway Department
Olympia, Washington

a. Purpose: Computes final end moments in beams and columns tops of single story continuous frames. The beams may be integral with the columns. Computer influence line ordinates for loads at all the tenth points or for loads at the 1/3, 1/2, and 2/3 points. These ordinates are the final moments at the beam ends and at the respective points in the span. Shear values are also computed. Information required for input is the distribution coefficients and carry-over factors, fixed end moments if they are to be distributed, and span lengths and load to be used if influence line ordinates are to be computed. When influence line ordinates are to be computed, a table of fixed and moment coefficients must be supplied only if the beams are not prismatic.
b. 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.
c. Mathematical Method: Uses Exact (Sine Series) Method wherein deflected shape is represented by a Fourier series.
d. Storage Required: Program occupies 1869 positions in memory storage.
e. Remarks: Self-loading 5 instruction per card deck is available. Written in SOAP.
f. 650 System: One 533 required.
Special Devices: None.

APPROXIMATION OF FUTURE TRIP TRANSFERS
E. A. Radsliff
California Division of Highways
Sacramento, California

a. 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.
b. 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.
d. Storage Required: Essentially the entire drum is used by the program. Only 480 locations are used for instructions or constants, but 1484 fixed locations are required for storage of data.
e. 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.
f. 650 System: One 533 required.
Special Devices: None.
**GENERAL FREEWAY ASSIGNMENT**

M. Brubaker
Sacramento, California

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.

Accuracy: Not given.

Floating/Fixed: The entire routine is processed in fixed point.

Mathematical Method: Does not apply.

Storage Required: The entire drum is used. 1000 locations are used to store cumulative time and distance between zones. For problems not requiring this many zonal interchanges, additional locations can be made available.

Speed: Not given.

Relocatability: Not given.

**Points of Interest:**
- 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 I.
- California Wheel Base Law and "AASHO", American Association of State Highway Officials, recommendations.

**Special Devices:** Alphabetic device.

---

**LOADOMETER W-6 TABLE**

J. H. Harbour
California Division of Highways
Sacramento, California

The purpose of this routine is to edit data and calculate per cent of overload on total weight and each axle of trucks and truck combinations with one or more axles 18,000 pounds or more, and single unit trucks weighing 13 tons or more per California State Law and "AASHO", American Association of State Highway Officials recommendations.

**Points of Interest:**
- Accuracy: Not given.
- Floating/Fixed: The entire routine is processed in fixed point.
- Storage: A maximum of 7 axles per vehicle.

**Special Devices:** Alpha betic device.

---

**Addenda/Errata:**
- This is a revision of the block diagram for Block 610 to replace page 31 of the program write-up.
- 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 each computation is rounded up by one half, the number of trips assigned to the basic 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:

<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>441</td>
<td>0</td>
<td>1702</td>
<td>65 1104 1652</td>
</tr>
<tr>
<td>430</td>
<td>445</td>
<td>0</td>
<td>1652</td>
<td>16 1026 1602</td>
</tr>
<tr>
<td>430</td>
<td>461</td>
<td>0</td>
<td>1602</td>
<td>16 1028 1652</td>
</tr>
<tr>
<td>430</td>
<td>471</td>
<td>0</td>
<td>1502</td>
<td>20 1027 1402</td>
</tr>
<tr>
<td>430</td>
<td>681</td>
<td>0</td>
<td>1402</td>
<td>65 1106 1352</td>
</tr>
<tr>
<td>430</td>
<td>451</td>
<td>0</td>
<td>1352</td>
<td>45 1030 1703</td>
</tr>
<tr>
<td>430</td>
<td>501</td>
<td>0</td>
<td>1924</td>
<td>16 1032 1927</td>
</tr>
<tr>
<td>430</td>
<td>511</td>
<td>0</td>
<td>1927</td>
<td>20 1031 1777</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>420</td>
<td>010</td>
<td>0</td>
<td>1024</td>
<td>69 1109 1101</td>
</tr>
<tr>
<td>430</td>
<td>015</td>
<td>0</td>
<td>1105</td>
<td>24 1151 1101</td>
</tr>
<tr>
<td>420</td>
<td>046</td>
<td>0</td>
<td>1501</td>
<td>69 1107 1101</td>
</tr>
<tr>
<td>430</td>
<td>075</td>
<td>0</td>
<td>1071</td>
<td>60 1002 1752</td>
</tr>
<tr>
<td>430</td>
<td>081</td>
<td>0</td>
<td>1752</td>
<td>30 0002 1702</td>
</tr>
<tr>
<td>430</td>
<td>085</td>
<td>0</td>
<td>1752</td>
<td>15 1551 1652</td>
</tr>
<tr>
<td>430</td>
<td>086</td>
<td>0</td>
<td>1652</td>
<td>20 1551 1259</td>
</tr>
<tr>
<td>430</td>
<td>091</td>
<td>1</td>
<td>1259</td>
<td>21 1821 1074</td>
</tr>
<tr>
<td>401</td>
<td>036</td>
<td>2</td>
<td>1053</td>
<td>21 1821 1074</td>
</tr>
<tr>
<td>430</td>
<td>341</td>
<td>0</td>
<td>1249</td>
<td>45 1102 1777</td>
</tr>
</tbody>
</table>

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 each computation is rounded up by one half, the number of trips assigned to the basic 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:

<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>441</td>
<td>0</td>
<td>1702</td>
<td>65 1104 1652</td>
</tr>
<tr>
<td>430</td>
<td>445</td>
<td>0</td>
<td>1652</td>
<td>16 1026 1602</td>
</tr>
<tr>
<td>430</td>
<td>461</td>
<td>0</td>
<td>1602</td>
<td>16 1028 1652</td>
</tr>
<tr>
<td>430</td>
<td>471</td>
<td>0</td>
<td>1502</td>
<td>20 1027 1402</td>
</tr>
<tr>
<td>430</td>
<td>681</td>
<td>0</td>
<td>1402</td>
<td>65 1106 1352</td>
</tr>
<tr>
<td>430</td>
<td>451</td>
<td>0</td>
<td>1352</td>
<td>45 1030 1703</td>
</tr>
<tr>
<td>430</td>
<td>501</td>
<td>0</td>
<td>1924</td>
<td>16 1032 1927</td>
</tr>
<tr>
<td>430</td>
<td>511</td>
<td>0</td>
<td>1927</td>
<td>20 1031 1777</td>
</tr>
</tbody>
</table>

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 each computation is rounded up by one half, the number of trips assigned to the basic 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.
STRESS ANALYSIS OF OPEN-WEB STRUCTURES

IBM 650 Library Program Abstracts

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, 61 panels;
"IW" type, 88 panels.

Accuracy: Not given.
Floating/Fixed: Fixed point arithmetic is used.
c. Mathematical Method: The standard formulas are used.
d. Storage Required: The entire drum.
Speed: Not given.
Relocatability: Not relocatable.
e. Remarks: This routine consists of several packages: Load Routine; Indexing Register Simulator; Reaction program; True Geometry and Stresses; Simple Web, Subdivided Panel, and "IW" System. Mathematical subroutines include:
SINE, COSINE, SINH, COSH, ex, LOGe, ARCSINE, ARCTAN, √X, √(1 - X).

In the right triangle a, b, c, any of the following are computed, with or without their natural functions:

\[ \begin{aligned} \sqrt{a^2 + b^2} & , & \sqrt{a^2 - b^2} & , & \sqrt{b^2 - a^2} & , \\
2ab & , & 2ab \cos \frac{\pi}{2} & . 
\end{aligned} \]

Also, \[ \sqrt{a^2 + b^2} , \sqrt{a^2 - b^2} , \sqrt{b^2 - a^2} \].

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

DIGITAL TERRAIN MODEL SYSTEM HORIZONTAL ALIGNMENT PROGRAMS

HA-1, 2, 3, and 4.

b. Range: Maximum number of vertical curves is 98.

Accuracy: All lengths and distances are computed to three decimal places.
Floating/Fixed: Fixed point arithmetic is used.
c. Mathematical Method: Coordinate transformations and trigonometry are used.
d. Storage Required: HA-1, 2, 3, and 4 are loaded together. There are 200 locations available.
Speed: Not given.
Relocatability: Not relocatable.
e. Remarks: HA-3 and HA-4 are options of HA-1. HA-2 is a separate program but is loaded with HA-1.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

DIGITAL TERRAIN MODEL SYSTEM VERTICAL ALIGNMENT PROGRAMS

VA-1 and VA-2.

a. Purpose: VA-1, Basic Vertical Alignment Program:
This program computes the geometry of the vertical alignment of a highway and computes the profile elevation at each cross section. The input is the profile definition data and the output of the DTM HA-1 program.

VA-2, Highway Profile Geometry Program:
This program computes the geometry of the vertical alignment of a highway and computes the profile elevation at even stations along the alignment. The input is the profile definition data and the increment between even stations. Can be used independently of the DTM System.

b. Range: Maximum number of vertical curves is 98.

Accuracy: Per cent violation to 1/10 of one per cent which is converted to feet.

Floating/Fixed: Fixed decimal point.
c. Mathematical Method: Does not apply.
d. Storage Required: Not given.
Speed: Operates at read speed (100 cards per minute).
Relocatability: Not in relocatable form.
e. Remarks: None.
f. IBM 650 System: One 533 required.
Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

DIGITAL TERRAIN MODEL SYSTEM TERRAIN DATA EDIT PROGRAM TD-1

Massachusetts Department of Public Works
C. L. Miller
R. A. Laflamme
Photogrammetry Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

a. Purpose: The Digital Terrain Model (DTM) System Series of computer programs requires the terrain data to be in a certain format and to meet a set of specifications. This program checks the terrain data to insure that it is in the proper format and meets the required specifications. Error cards are punched to identify terrain data cards and points which are not in proper format or sequence.

b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.

(continued on next column)
DIGITAL TERRAIN MODEL SYSTEM PRELIMINARY EARTHWORK PROGRAM
EW-Z

Massachusetts Department of Public Works
C. L. Miller
Photogrammetry Laboratory
Cambridge, Massachusetts

Purpose: This is the basic program for computing earthwork quantities in
construction 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
purpose is to determine the volumes of earthwork required.

Range: Maximum number of points per cross section is 200.
Accuracy: In standard vertical curves are used.
Storage Required: VA-1 and VA-2 are loaded together and use 600 locations.
Speed: Not given.
Relocatability: Not relocatable form.

IBM 650 System: One 533 required.

Addenda/Errata

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
comparisons 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.

Add to the program the following instructions:

Block Card Code Loc. Instruction
430 050 0 1047 69 1309 1452
430 055 0 1852 24 1762 1994
430 060 0 1994 69 1327 1580
430 075 0 1994 60 8002 1546
430 080 0 1546 10 0002 1664
430 085 0 1546 15 1902 1596
430 090 0 1596 20 1902 1599
430 095 0 1599 21 1010 1413
430 100 0 1646 15 1906 1746
430 105 0 1646 15 1906 1746
430 110 0 1814 20 1814 1995
430 115 0 1995 65 1364 1046
430 120 0 1846 16 1816 1996
430 125 0 1846 15 1816 1996
430 130 0 1996 15 1816 1996
430 135 0 1996 20 1817 1471

Addenda/Errata

This is a revision of the block diagram for Block 430 to replace page 12
of the program write-up.

(Continued on next column)
Earthwork Data Check

R. R. Declark
K. F. Kohler
Bureau of Public Roads
Portland, Oregon

Purpose: 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 error or error.

Range: Minor errors are not detected. The break-point between major 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 1, 2, 3, 4 or 5 separately
2. Type 0 or 2 combined with type 1, 3, 4 or 5

Accuracy: Not given.

Floating/Fixed: Fixed point.

Mathematical Method: Simple arithmetic is used.

Storage Required: The program occupies 1860 storage locations.

Speed: The computations for each intersection require approximately 30 seconds. Distance and bearing computations proceed at about 30 per minute.

Relocatability: Not relocatable.

Remarks: None.

IBM 650 System: One 533 required.

Special Devices: Alphabetic device required.

Profile Grade

J. Oakes
Oregon State Highway Department
Salem, Oregon

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 of major problems based on this spiral may be calculated. An unlimited number of distances and bearings computations is possible.

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.

Accuracy: Distances are given to thousandths 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 30 seconds. Distance and bearing computations proceed at about 30 per minute.

Relocatability: Not relocatable.

Remarks: None.

IBM 650 System: One 533 required.

Special Devices: Alphabetic device required.

Contract Bid Computations

T. L. Yates
Oregon State Highway Department
Salem, Oregon

Purpose: 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 $999,999,999.99. Item and job totals up to $99,999,999.99. This routine can handle up to 95 items and 50 bidders per job.

Accuracy: As indicated above.

Floating/Fixed: Not given.

Mathematical Method: Does not apply.

IBM 650 System: One 533 required.

Special Devices: Ten extra pilot selectors (for a total of 20) are required.
Time Series Trend Equations

R. A. Bieber
California Division of Highways
Sacramento, California

a. Purpose: This program is designed to solve the equations \( Y = A + Bx \), \( \log Y = A + Bx \), and \( Y = ABx \) for a value of \( A \) and \( B \) and using this value determine \( Y \), 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.

b. Range: The linear equation may be based on increasing or decreasing trends. The semilog equation may be based on increasing or decreasing trends as long as the values of \( T \) do not become negative. The exponential equation may only be solved for increasing trends.

c. Accuracy: The log and scaling routines used are accurate to \( 2 \times 10^{-7} \) and the square root routine is accurate to \( 10^{-8} \).

d. 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.

e. Storage Required: The program requires the entire 2000 storage locations.

Speed: The time required for solving the three types of equations is approximately 4-3/4 minutes.

Relocatability: Not relocatable.

(Continued on next column)
AUTOMATIC MINIMUM WEIGHT DESIGN OF STEEL FRAMES

R. L. Stone
Division of Applied Mathematics
Brown University
Providence, Rhode Island

a. Purpose: Given the centerline 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 4-bay, 3-storey.
c. Mathematical Method: A method which was devised by J. Heyznan and W. Prager of the Division of Applied Mathematics of Brown University.
d. Storage Required: The entire drum is used.
e. Remarks: The program is completely automatic, requiring no intermediate intervention by the operator. It consists of 15 subroutines (a total of about 2400 instructions).
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

LAND AREA - SURVEY TRAVERSE

W. L. Anderson
State Highway Department of Georgia
Atlanta, Georgia

a. Purpose: This program is designed to calculate the following:
   - Cut, fill, fill plus shrinkage volumes
   - Mass ordinates
   - Slope selection
   - Slope state offset and elevation
   - Summarization of cut and fill volumes at five station intervals

For the Design Problem:

b. Range: The program handles any traverse with up to 200 courses.

c. Mathematical Method: Based on Talbot Spiral using "Arch" definition of circular curve.

d. Storage Required: Approximately 1688 storage locations are used.

e. Remarks: To obtain correct areas, the courses must be in order; and in any case all the cards for a given traverse must be together. Except for double punch and blank columns, there should be no insercetable machine stops. Error cards are punched and the program proceeds to the next traverse automatically.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device is required.

IBM 650 Library Program Abstracts
THREE-POINT SOLUTION

D. Geister
Oregon State Highway Department
Salem, Oregon

a. Purpose: This program is designed to compute the coordinates of a point by the three-point method. It can handle from three to nine known points computing a solution for every combination of three known points. The selection of the most desirable solution is left to the engineer submitting the data.
b. Range: From three to nine known points are acceptable in the input data. The output will include every combination of three points.
c. Mathematical Method: Three-point solution; see the program write-up for further details.
d. Storage Required: 1,700 storage locations.
e. Remarks: Subroutines used in SIR are Float, Fix, Sin, and Cos. For best results, angles greater than 20° should be used. Three-point problems in which all points including unknowns are on a circle have an infinite number of solutions, any one of which the program may produce as its result.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

a. Purpose: This program computes the necessary information needed for detailing a tangent bridge. The information calculated includes the following:
1. The station of a point.
2. The P. G. 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.
b. Range: The maximum number of points on any substructure element is 20. Any number of substructure elements are allowed.
d. Storage Required: The program requires the first 725 drum storage locations; subroutines included require about 350 additional locations.
e. Speed: The time required by the program is approximately as follows:
- 50 + 0.5n seconds, where n is the number of points to be computed.
f. IBM 650 System: One 533 required.

Remarks: Some precautions which should be observed are:
1. Negative information must be identified by a negative overpunch in the units 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 B). In these words, the overpunch serves to identify the card as having ten words of information in it.
3. Of course, one cannot exceed the problem format. Any D1 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

R. N. Boden
Ohio Department of Highways
Columbus, Ohio

a. 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.
b. 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.
c. Mathematical Method: Primarily, algebra and trigonometry. Manning's formula is used to compute the hydraulic elements of conduits flowing full. Chezy's formula is the basis for computing the hydraulic elements of partially full conduits.
d. Storage Required: 999 drum storage locations are reserved for tables, subroutines and loading routines; 1054 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.
e. Speed: This is a function of the type of analysis chosen by the program to compute the hydraulic elements of the conduit.
f. IBM 650 System: One 533 required.

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. SOAP symbolic deck listing is included.

IBM 650 Library Program Abstracts

3-SPAN CURVED CONCRETE SLAB BRIDGE PROGRAM

a. Purpose: This program calculates moment and reaction influence line ordinates for symmetrical 3-span or 4-span continuous girder bridges.
b. Range: Maximum number of points on any substructure element is 20.
d. Storage Required: Not given.
e. Accuracy: All calculations are accurate to at least three decimal places.
f. IBM 650 System: One 533 required.

Remarks: None.

Special Devices: None required.
D. L. Herke
Ohio Department of Highways
Columbus, Ohio

a. Purpose: This program is designed to generate and compute a station number, a profile grade elevation in X and Y coordinates; and a final surface elevation for a number of specified and given points on the abutments and piers of a 3-span curved concrete slab bridge.

b. Range: The range of the important portion of the input data is as follows:

For \( B_1 - B_2 \) incl., 0.01 ft. \( \leq B_1 \leq 316526.00 \) ft.

\[ 90^\circ < \theta < 90^\circ + 55^\circ \]

where \( D \) Degree of Curvature

For \( S_1 - S_2 \) incl., 0.000 \( \leq S \leq 99.999 \)

\[ \theta < 90^\circ + 55^\circ \]

Accuracy: The accuracy of the program, the profile grade and the final surface elevations calculations are to \( \pm 0.01 \) 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 \( Y = f(X, \theta) \). 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 \( TX + TR \), convert that to an angle 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 \( S = \pi R \). Using this, we may compute B.L. Survey \( = (\theta - \pi)R \). 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
C. 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 station equations and odd stations (not even multiples of 25) combined is 600. The maximum number of PVI points is 100.

Accuracy: Percent grade is accurate to the nearest 0.001 ft. Other values are accurate to the nearest 0.01 ft.

Floating/Fixed: Fixed decimal.


d. Storage Required: 154 locations.

Speed: Not given.

Relocatability: Not relocatable.

e. Remarks: None.

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

DIGITAL TERRAIN MODEL SYSTEM FOUR POINT POLYNOMIAL INTERPOLATION PROGRAM DA-3

Massachusetts Department of Public Works
C. L. Miller
R. R. Buggett
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-Z Program, IBM 650 Library Program File Number 9.2.064).

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.

Accuracy: The output has as many significant digits as the input.

Floating/Fixed: Fixed decimal arithmetic is used.

c. Mathematical Method: Aitken's method of iteration is used to compute the polynomial.

d. Storage Required: About 200 locations are required for program and storage.

Speed: The interpolation of a point requires 1.4 seconds. Therefore 43 points per minute are computed and punched.

Relocatability: Not relocatable.

e. 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 panel is used.

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

**DIGITAL TERRAIN MODEL SYSTEM PROFILE SMOOTHING PROGRAM DA-3**

Massachusetts Department of Public Works
C. L. Miller
R. R. Buggett
Photogrammetry Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

a. Purpose: The DA-3 program applies curve smoothing formulas to terrain profiles obtained from DTM programs HA-1, 2, or 3 (IBM 650 Library Program File Number 9.2.064).

The output of the DA-3 program is a smoothed profile which can then be used for selecting a vertical alignment. This program can also take as input its own output so that any particular profile can be remoothed as many times as desired. Either the 7 points or 11 points smoothing formulas may be selected.

b. Range: No practical restrictions.

Accuracy: The input data are treated as integers. Therefore the output has the same scaling and significant figures as the input.

Floating/Fixed: Fixed decimal arithmetic is used.

c. Mathematical Method: Standard smoothing formulas using a third degree polynomial over 7 or 11 points are used.

d. Storage Required: The program uses approximately 1000 locations.

Speed: The program requires approximately 6 seconds per profile point. Assuming points at 200 foot intervals, the program will smooth 12 miles of profile per hour.

Relocatability: Not relocatable.

e. Remarks: This program operates in conjunction with 9.2.040 DTM Horizontal Alignment Program and is one of a series of programs in the Digital Terrain Model System.

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

**CONTINUOUS BEAM DESIGN PROGRAM**

J. C. Porter
Nebraska Department of Roads
Lincoln, Nebraska

a. Purpose: This program calculates moments and shears in a 2- to 5-span continuous or framed structure.

(Continued on next page)
COMPUTER ANALYSIS OF CONTINUOUS BEAMS AND FRAMES

E. D. Lee
Washington State Highway Department
Olympia, Washington

Purpose: This program analyzes a single story frame with from one to five spans when given the frame dimensions and the H-S wheel load. Output is influence lines for end moments, moments at tenth points and shears at supports for loads at the tenth points. Dead load moments and shears are computed. Moment curve due to unit cantilever moment at either end is computed. Live load moments due to an H-S truck are computed and combined with dead load moments to give the total moment curve.

b. Range: One to five span structures.

Accuracy: Does not apply.

Floating/Fixed: Not given.

Relocatability: Not relocatable.

Remarks: This program was modified from "Continuous Bridge Analysis" by L. H. Deh, Oregon State Highway Department, Salem, Oregon. There is a program deck for each one, two, three, four and five span structures. A bootstrapping procedure is followed wherein one portion of the program is read in and used and then replaced with additional program instructions until the problem is completed.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

File no. 9.2.067

Engineering Applications

CONTINUOUS BEAMS AND FRAMES

Washington State Highway Commission

An error has been detected in one of the program decks of the Continuous Beams and Frames program (9.2.067). This error affects cantilever moments in a three-span beam program. Make the following changes in Part A of the three-span program:

NEW
OLD
Inst. 6
Loo. of Def. Instruciion
1560 STL RZ STL C1
1562 MPT RZ MPT C1
20 0912 1562
19 0912 0468

IBM 650 Library Program Abstracts

File no. 9.2.068

Engineering Applications

FRAME CONSTANTS

E. D. Lee
Washington State Highway Department
Olympia, Washington

Purpose: Given span lengths and variation in section, this program will compute the following: carry-over, stiffness, and distribution factors around each joint; concentrated and uniform load fixed end moment coefficients for each span.

b. Range: One to five spans for joint distribution factors; any number of spans for beam constants.

Accuracy: Not given.

Floating/Fixed: Not given.

Relocatability: Not relocatable.

Remarks: This program was developed separately, they are specifically designed such that a part or all of the output from one can be used as input to another.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

File no. 9.2.067

Engineering Applications

CONTINUOUS BRIDGE ANALYSIS

T. L. Yates
Oregon State Highway Department
Salem, Oregon

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-S Loading, and (3) Deflections.

b. Range: Two to five span structures are accommodated.

Accuracy: In calculating dead load moments, an error of approximately 1/2 ft. is allowed, Floating/Fixed: Not given.

Mathematical Method: Principle of Muller-Breslau and numerical procedure of Newmark.

Storage Required: All but six storage locations are used in the routine Live Load and Total Moments Due to H-S Loading.

Speed: A complete frame analysis, including total moments and deflection, requires approximately 15 minutes per span.

(Continued on next page)
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.

Accuracy: 1 Unit (100 cubic yard stations of overhaul).

Floating/Fixed: Not relocatable.

e. Remarks: 600 ft. used for fresh haul areas.

IBM 650 System: One 533 required.

Special Devices: Alphabetic device required.

STAGE CONSTRUCTION PROGRAM

G. J. Kellenbenz
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 150 points on new cross-sections.

Accuracy: Not given.

Floating/Fixed: Not given.

e. Remarks: Input and output cards are of the type used by the Washington State Highway Department.

IBM 650 System: One 533 required.

W-6 TABLE SUMMARY

(Continued on next column)
b. Range: 1. The increment between even stations may be any number greater than zero.
2. A profile having any number of points may be used and as many profiles as desired may be processed in the same run.

Accuracy: Since the program creates the input data as integers, the output has as many significant figures as the input.

Floating/Fixed: Fixed decimal arithmetic is used.

Mathematical Method: Aitken’s method of iteration is used to compute the interpolating polynomials.

Storage Required: Approximately 160 locations are required for the program and storage.

Speed: The program will compute approximately 47 points per minute.

Relocatability: Not relocatable.

Remarks: The program has been written for a utility (80-80) control panel. The board must have the facility of setting word sizes equal to or more of the word (16 columns) in length; this is necessary for words through 8.

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

C. L. Miller - Project Director
R. A. LaFlamme - Programming Supervisor

PROFILE COMPARISON AND STATISTICAL ANALYSIS PROGRAM DA-1

Department of Civil and Sanitary Engineering
Massachusetts Institute of Technology
Cambridge, Mass.

a. Purpose: Compares 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: 1. A map data profile cannot exceed 600 points.
2. A profile having any number of points may be used and as many profiles desired may be used.

Accuracy: (1) Differences have as many significant digits as the input data.
(2) Statistics are rounded to two decimal places.

Floating/Fixed: Fixed.

Mathematical Method: Aitken’s method of iteration is used to compute the interpolating polynomials.

Storage Requirements: 400 locations are reserved for the map profile and the program occupies the remaining 1400 locations.

Speed: Differences are computed in 2 seconds, therefore 30 points per minute are computed and printed. Profile or map statistics require 25 seconds, independent of the number of points in the profiles.

Relocatability: Not relocatable.

Remarks: Input data consists of elevations recorded at intervals of 10 foot along four screed lines. Successive spaces are considered one at a time with no limitation on the number of spaces.

f. IBM 650 System: Equipment Required is a minimum 650.

Special Devices: None.

COMPUTATION OF BRIDGE SCREED ELEVATIONS

Z. L. Math
C. E. Cooper
Bridge Bureau
State Highway Department of Indiana
Indianapolis, Ind. Indiana

a. Purpose: This program computes the elevations for setting screeds for concrete slabs on continuous steel beam or steel girder bridges.

b. Range: Elevations are given at ten foot intervals along four screed lines. Successive spaces are considered one at a time with no limitation on the number of spaces.

(Continued on next column)
IBM 650 Library Program Abstracts

GENERAL FREEWAY ASSIGNMENT, STOCKTON DIVISION

S. F. Perssell
California Division of Highways
210 N Street
Sacramento, California

a. Purpose: The purpose of this program is to compute time and distance between access numbers and an additional 40 locations are used to store time and distance between zones. Speed is approximately 2000 input cards per hour. The program is written in SOAP I terminology and can be relocated.

b. Restrictions, Range: There is no restriction to as to the number of input cards per small interchange. Three years of trip data can be handled at one time.

c. Method: Floating/Fixed: Fixed decimal,

Freezer Adjustment

S. F. Perssell
California Division of Highways
1120 N Street
Sacramento, California

a. Purpose: The purpose of this program is to provide a transition from the California Minimum Path Program to the California Freeway Assignment Program.

b. Restrictions, Range: There is no restriction as to the number of path nodes in any interchange. An input card may have a maximum of 20 path nodes, and an input card must have a maximum of six entrance-rout segments. A node may have a maximum of 50 access points. The program accommodates as many as 699 nodes and 1400 access points.

c. Method: The principle involved is one of search and compare.

d. Storage Requirements: Table storage requires 1,000 locations. Other program and temporary storage requirements see an additional 500 locations. Speed is approximately 1,600 input cards per hour. The program is written in SOAP I terminology and can be calculated.

e. Remarks: The program contains an error in input routine which identifies the error and the input card thereby eliminating machine stops during processing.

f. IBM 650 System: A basic 450 with special shift is used.

IBM 650 Library Program Abstracts

FREEWAY ASSIGNMENT

S. F. Perssell
Calif., State Div. of Highways
1120 N Street
Sacramento, Calif.

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.

b. Restrictions, Range: Fixed point arithmetic is used.

c. Method: Not applicable.

d. Storage Requirements: 1,000 locations are used to store time and distance between access numbers and an additional 44 locations are used to store time and distance between zones. Speed is approximately 2000 input cards per hour. The program is written in SOAP I and can be relocated.

Freezer Adjustment

S. F. Perssell
California Division of Highways
1120 N Street
Sacramento, California

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.

b. Restrictions, Range: Fixed point arithmetic is used.

c. Method: Not applicable.

d. Storage Requirements: 1,000 locations are used to store time and distance between access numbers and an additional 44 locations are used to store time and distance between zones. Speed is approximately 2000 input cards per hour. The program is written in SOAP I and can be relocated.

Freezer Adjustment

S. F. Perssell
California Division of Highways
1120 N Street
Sacramento, California

a. Purpose: The purpose of this program is to provide a transition from the California Minimum Path Program to the California Freeway Assignment Program.

b. Restrictions, Range: There is no restriction as to the number of path nodes in any interchange. An input card may have a maximum of 20 path nodes, and an input card must have a maximum of six entrance-rout segments. A node may have a maximum of 50 access points. The program accommodates as many as 699 nodes and 1400 access points.

c. Method: The principle involved is one of search and compare.

d. Storage Requirements: Table storage requires 1,000 locations. Other program and temporary storage requirements see an additional 500 locations. Speed is approximately 1,600 input cards per hour. The program is written in SOAP I terminology and can be calculated.

e. Remarks: The program contains an error in input routine which identifies the error and the input card thereby eliminating machine stops during processing.

f. IBM 650 System: A basic 450 with special shift is used.
a. Purpose: This routine adjusts traverses by the compass or the translt to the nearest 100 feet, or both, as requested by the engineer. Input is in the form of one course per card and output is 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 98 regular courses. All linear alignments are given in thousands of feet and bearings are computed to seconds. All trigonometric functions are computed to an accuracy of six decimal places.

c. Method: The trigonometrical functions used are from Technical Newsletter No. 9. Area is calculated using the criss-cross method.

d. Storage Requirements: One hundred locations each are required for storage of input coefficients, input data, and distance. Thirty-three hundred locations are required for storage of the description. Program and temporary storage requirements are approximately 650 core locations. Speed is approximately 2500 courses per hour. The program is written in SOAP I form.

e. Remarks: No provision has been made for computing area of circular segments because no provision has been made to keep certain courses constant.

f. IBM 650 System: A 650 with half-time emitters and alphabetic device is used.

IBM 650 Library Program Abstracts

REVISED TRAVERSE AND HORIZONTAL ALIGNMENT

E. F. Perssell

2. Vikes

California Division of Highways

Sacramento, California

a. Purpose: This routine will calculate traverses with two unknowns or with no unknowns in each traverse. Input is in the form of one course per card. Speed is approximately two thousand courses per hour. The program is written in SOAP I form.


c. Method: The mathematics used closely follows the hand calculated procedure making numerous decisions following the standard rules of the program. IBM Library SIN routine is utilized.

d. Storage Requirements: Does not apply.

e. Remarks: This program operates in conjunction with 9. Z. 040 DTM Distance, Alignment Program and is one of a series of programs in the Digital Terrain Model System.

f. IBM 650 System: Alphabetic device and special shift utilized although not necessary. Otherwise minimum 650.

IBM 650 Library Program Abstracts

REVISED TRAVERSE AND HORIZONTAL ALIGNMENT

R. E. Weisberg

Civil Engineering Computer Laboratory

S. F. Perssell

Civil Engineering Computer Laboratory

S. F. Perssell

Sacramento, California

a. Purpose: The routine will calculate traverses with two unknowns or with no unknowns in each traverse. Input is in the form of one course per card. Speed is approximately two thousand courses per hour. The program is written in SOAP I terminology.

b. Restrictions, Range: Each traverse may have a maximum of 20 regular courses.

c. Method: The routine will calculate traverses with two unknowns or with no unknowns in each traverse. Input is in the form of one course per card. Speed is approximately two thousand courses per hour. The program is written in SOAP I terminology.

d. Storage Requirements: Does not apply.

e. Remarks: This program operates in conjunction with 9. Z. 040 DTM Distance, Alignment Program and is one of a series of programs in the Digital Terrain Model System.

f. IBM 650 System: Alphabetic device and special shift utilized although not necessary. Otherwise minimum 650.

IBM 650 Library Program Abstracts

REVISED TRAVERSE AND HORIZONTAL ALIGNMENT

V. F. Perssell

2. Vikes

California Division of Highways

Sacramento, California

a. Purpose: This routine will calculate traverses with two unknowns or with no unknowns in each traverse. Input is in the form of one course per card. Speed is approximately two thousand courses per hour. The program is written in SOAP I form.


c. Method: The mathematics used closely follows the hand calculated procedure making numerous decisions following the standard rules of the program. IBM Library SIN routine is utilized.

d. Storage Requirements: Does not apply.

e. Remarks: This program operates in conjunction with 9. Z. 040 DTM Distance, Alignment Program and is one of a series of programs in the Digital Terrain Model System.

f. IBM 650 System: Alphabetic device and special shift utilized although not necessary. Otherwise minimum 650.

IBM 650 Library Program Abstracts

REVISED TRAVERSE AND HORIZONTAL ALIGNMENT

V. F. Perssell

2. Vikes

California Division of Highways

Sacramento, California

a. Purpose: This routine will calculate traverses with two unknowns or with no unknowns in each traverse. Input is in the form of one course per card. Speed is approximately two thousand courses per hour. The program is written in SOAP I form.


c. Method: The mathematics used closely follows the hand calculated procedure making numerous decisions following the standard rules of the program. IBM Library SIN routine is utilized.

d. Storage Requirements: Does not apply.

e. Remarks: This program operates in conjunction with 9. Z. 040 DTM Distance, Alignment Program and is one of a series of programs in the Digital Terrain Model System.

f. IBM 650 System: Alphabetic device and special shift utilized although not necessary. Otherwise minimum 650.

IBM 650 Library Program Abstracts
IBM 650 Library Program Abstracts

Thermodynamic Properties and Phase Behavior of Light Hydrocarbon Mixtures

W. B. Edwards
E. J. Organick
L. Larrey
Computing Center
University of Houston
Houston, Texas

a. Purpose: Computes density, compressibility factor, enthalpy, entropy, and equilibrium ratios of single and two phase systems.
b. 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 (binary floating point interpretive routine).
c. Mathematical Method: Rigorous thermodynamic solution based on:
1. Benedict, Webb, Rubin Equation of State for pure components and mixtures; and
2. Zero pressure thermal properties of pure components.
d. Storage Required: Approximately 100 unused drum locations.
Speed: Speed depends upon number of phases, number of components, and options to compute enthalpy and entropy.
Relocatability: Program is non-relocatable.
Remarks: None.
f. IBM 650 System: One 533 required.
Special Devices: None.

IBM 650 Library Program Abstracts

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

a. Purpose: This program calculates the half-wave potential and slope of a polarographic wave,

\[ E = E_1/2 + k \log (i_{1/2}/i) \]

by the method of least squares using current-voltage data taken from a polarogram.
b. 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.
d. 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.
a. Remarks: This program, written in the Bell Labs Interpretive System (see TNL Nu. 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.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

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 gas phase for the rigid rotor • simple vibrator model. The contributions for hindered internal rotation cannot be handled by this program. The mathematical accuracy is ± 0.0001 unit.

c. Method: The calculation of the exponential and the logarithmic functions are made by the use of the sub-routine.

d. Storage Requirements: The number of storage used for the whole computation is 500. When the number of the fundamental frequencies is n, the time required for the computation for an assigned temperature is 1 ± 2 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

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 9.4.001

ELECTRICAL POWER SYSTEM TRANSIENT STABILITY CALCULATIONS

J. E. Rowe and J. L. Gabbard, Jr. November 1, 1956
Union Carbide Nuclear Co., Oakridge, Tenn.

a) It is possible to make the transient stability calculations for any system that can be represented by 18 equivalent machines or less. However, if the number of equivalent admittances required to represent the network does not exceed 200, a program limit of approximately 50 machines is possible (a 30 machine system has been studied). Induction machines as well as synchronous machines can be handled.

b) Uses fixed decimal arithmetic.

c) Uses transient stability theory, symmetrical component theory, and network theory. Makes use of Starr's equivalent circuit for the n-terminal network expressed in matrix form and as admittances rather than impedances. Calculations are made in the per unit system and care must be exercised in selecting the system base in order to avoid field excursions with the fixed decimal program. The transient stability differential equations are solved using the method of 1st order forward differences.

d) Uses 718 words plus data and output. Time approximately 1 1/2 - 2 1/2 hours depending on variables.

e) Contains an excellent flow chart.

f) Minimum 650.

IBM 650 Library Program Abstracts

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 9.4.002

NETWORK REDUCTION

P. E. Scott and E. M. Kidd October 19, 1956
Union Carbide Nuclear Co., Oak Ridge, Tenn.

a) A network reduction program - describes an automatic method of reducing an electrical power network to a smaller equivalent network.

b) Limitations as to size of matrix to be handled are n ≤ 20, n ≤ 20 is a size of M and K which represents those junctions to be eliminated and these to remain respectively.

c) Matrix theory and network theory.

d) Approximate time (5.96e6 - 1.273e+ 0.726) seconds storage required - 460 words plus data and output.

e) Number of output words = 1 + Whv+1.2

f) Minimum 650.
PROBABILITY OF LOSS OF LOAD
H. D. Limmer
Public Service Electric & Gas Co.
Newark, New Jersey
a. 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.
c. Mathematical Method: The nodal iterative method of solution is used.
d. Storage Required: Almost entire drum.
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: Alphanumeric device.

IBM 650 Library Program Abstracts
File no. 9.4.006
Engineering Applications

CALCULATION OF ELECTRIC POWER SYSTEM SHORT-CIRCUIT CURRENTS
L. W. Coombs
The Detroit Edison Company
Detroit, Michigan
a. Purpose: This program computes the total fault current and the currents in the lines connected to the faulted bus. The real and imaginary components and the magnitude of the currents are punched out together with the X/R ratios. 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 150 lines.
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-Seidel iteration method with acceleration.
d. Storage Required: Not given.
e. Remarks: The 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.
f. IBM 650 System: One 533 required.
special Devices: None.

IBM 650 Library Program Abstracts
File no. 9.4.007
Engineering Applications

OVERHEAD ELECTRICAL DISTRIBUTION SYSTEMS ANALYSIS
J. E. Jones
E. J. Farris
IBM, Houston, Texas
G. W. Opres
Houston Lighting and Power Company
Houston, Texas
a. Purpose: This program calculates voltages 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.
c. Mathematical Method: The nodal iterative method of solution is used.
d. Storage Required: Not given.
e. Remarks: Both absolute and SOAP listings are included.
f. IBM 650 System: One 533 required.
special Devices: Alphanumeric device.

IBM 650 Library Program Abstracts
File no. 9.4.008
Engineering Applications

ECONOMIC CONDUCTOR STUDY
R. P. 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: ±A 10^10 , where 15 A ≤ 10 and -50 ≤ A ≤ 49.
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: Not given.
e. Remarks: Card 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
File no. 9.4.009
Engineering Applications

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 10 equal increments of percentage moisture at one

(Continued on next page)
location (X), the corresponding values of percentage moisture at the other location (Y) are calculated.

b. Range: The input data consists of up to 39 pairs of measured values of percentage moisture in coal. All measurements are considered to be of equal weight in the computation.

Accuracy: The output consists of corresponding values of (X) and (Y) with (X) ranging from 0.10 to 12.00 in increments of 0.10.

Floating/Fixed: The input and output data are in fixed point decimal form. Computations are performed in the G. E. floating decimal mode.

c. Mathematical Method: The Method of Least Squares is used. The equation found is of the form: \( Y = a + bX \).

d. Storage Required: The program, including data storage, uses locations 0000-0007.

Speed: For 16 pairs of input data, total machine time is approximately 1.5 minutes.

Relocatability: Not given.

e. Remarks: The program includes an interpretive routine to perform the floating decimal arithmetic. The number of values, increment size, and range of the output data can be easily modified.

f. IBM 650 System: One 533 required.

---

**IBM 650 Library Program Abstracts**  
**File Number** B-650  
**5D SERIES BIG LOAD FLOW PROGRAM**

Carlos O. Love  
Texas Power & Light Co.  
P. O. Box 631  
Dallas 21, Texas

a. Purpose: Studies service conditions on radial and series distribution systems and supplements system load flow studies.

b. Restrictions, Range: 30 buses maximum including source bus. Calculation and print time is approximately 6 seconds/bus/problem with a tolerance of 0.10%.

c. Method: Per unit notation on an equivalent single phase system is used for all internal calculations. Input and output data are noted in standard electrical units. Directive solution.

d. Storage Requirements: Complete 2000 drum locations are required for program and data.

e. Remarks: Only three phase load may be considered. May be used to supplement system load flow studies. The absolute and SOAP deck listings are included.

f. IBM 650 System: Basic IBM 650, standard 60 columns, 4 word panel.

---

**IBM 650 Library Program Abstracts**  
**File Number** B-650  
**RADIAL SHORT CIRCUIT PROGRAM**

Carlos O. Love  
Texas Power & Light Co.  
P. O. Box 631  
Dallas 21, Texas

a. Purpose: Computes three phase, phase-to-phase, and phase-to-ground short circuit currents on a radial or tree system.

b. Restrictions, Range: Up to 50 fault points per problem.

c. Method: Based on mathematical system of symmetrical components.

d. Storage Requirements: Approximately 1000 drum locations are required for program and data. Average calculation time is 4 seconds/bus/problem.

e. Remarks: The absolute and SOAP deck listings are included.

d. IBM 650 System: Standard 60 columns, 4 word panel.
**IBM 650 Library Program Abstracts**

**STRAIN ROSETTE DATA REDUCTION**

J. A. Stone

IBM, Boston

- **Purpose:** This program reduces the data taken from delta or rectangular rosettes. The normal input is in strain in micro inches per inch. Precision is adequate when computing strains in the form y = A(x+B), where y is the strain, x is the data, and A and B are constants. The output is the maximum stress, minimum stress, shear stress, and angle to the principle axis.
- **Range:** This routine will compute up to a stress level of 500,000 PSI.
- **Accuracy:** Stresses to ± 12 PSI and the angle to ± 6, 01 degrees.
- **Floating/Fixed:** Computation is done in fixed point form.
- **Mathematical Method:** A seven-term approximation is used for the arctangent. Newton's method is used to evaluate the square root. The first value of the iteration is obtained from a table included in the program.
- **Storage Required:** The program occupies locations 0000-0060.
- **Speed:** Using the normal input the speed is 100 reductions per minute. With modified input, speed is greater than 85 per minute.
- **Relocatability:** May be relocated except for storage locations 0000-0004.
- **Remarks:** The program is self-loading.
- **IBM 650 System:** One 553 required.

**EVALUATING COMPRESSION PERFORMANCE**

R. W. Evans

R. L. Smith

R. A. Semrad

Sinclair Oil and Gas Company

Tulsa, Oklahoma

- **Purpose:** Sinclair's purpose in 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.
- **Range:** Not given.
- **Accuracy:** Not given.

**WELL BORE DEVIATION RECORD**

I. T. Ablin and G. E. Mitchell

IBM, Houston

- **Purpose:** This subroutine is designed for use with the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter #11 for explanation of this step.
- **Mathematical Method:** See pages 8 through 14 of the write-up.
- **Storage Required:** The program requires 2000 storage locations.
- **Speed:** The average is one minute for each set of operating pressures.
- **Relocatability:** Not relocatable.
- **Remarks:** The step most frequently encountered is 7777. This is caused by cards missing or out of order in the input deck.
- **IBM 650 System:** One 553 is required.

**CAM LEADER CO-ORDINATE ROUTINE**

Marie T. Casanov

Henry M. Schaele

United Shoe Machinery Corporation

Research Division

Engineering Department

Beverly, Massachusetts

- **Purpose:** Calculates the cam follower center and y coordinates for any angular position of the cam from the outer most position of the roll.
- **Mathematical Method:** See pages 8 through 14 of the write-up.
- **Storage Required:** The program occupies locations 0000-0060. It takes approximately 5 to 4 seconds to calculate the co-ordinates for each degree of cam rotation.
- **Remarks:** A conditional stop may be programmed at the conclusion of each loop to facilitate the removal of the output cards and to assist in monitoring the progress of the 650 through the problem. See write-up or IBM Technical Newsletter #11 for explanation of this step.
- **IBM 650 System:** Basic 650.
P-V-T DATA CALCULATIONS

A. Cohen
IBM, NY DPC

a) Program uses the Benedict equation to compute the density roots, entropies, volumes and heat capacities of a given component at pre-selected temperatures and pressures given in either English or c. g. s. units.

b) Fixed point arithmetic with different scaling for English and c. g. s. units. Accuracy depends on quantity considered.

c) Uses Benedict equation. Exponential and logarithmic routines are employed.

d) Program scattered optimally over the whole drum. A temperature-pressure combination takes 3-4 seconds, depending on number of iterations required.

e) None.

f) IBM 650 System: One 533 required.

EQUILIBRIUM FLASH CALCULATION

M. E. Klecka
R. Y. Seaber
Shell Oil Company
Houston Research Laboratory
Houston, Texas

a) Calculates isothermal equilibrium flash vaporizations where the feed composition and K values are specified.

b) A maximum of 30 components can be used. Floating point arithmetic is employed, and closure accuracy is ± 0.0001 mole fraction, based on the liquid product from the flash stage.

c) Conventional isothermal equilibrium flash calculation equations are used.

d) 1400 locations are used for program and data. The time per calculation depends upon number of components and the system but is generally 3-6 minutes per completed calculation.

e) Three check features are incorporated into the program: 1. The system must be above the bubble point. 2. The system must be below the dew point. 3. The sum of the mole fractions of the feed must equal 1.

A violation of any one of the above conditions will cause rejection of the particular problem by the machine. The name card identifying the problem will be punched followed by another card which gives the reason for rejection.

f) 650 equipped with alphanumeric device.

ABSORBER CALCULATION

J. M. Morris
Warren Petroleum Corporation
Tulsa, Oklahoma

a) Purpose: This program computes the lean oil rate to the absorber necessary to achieve a predetermined percent extraction of a key component. It also calculates a complete material and heat balance for the absorber.

(Please note: The remainder of the text is not visible in the image.)
Purpose: This program computes several residuals and secound derivatives of gravity at each regularly spaced grid intersection where sufficient data exists.

Remarks: None.

Storage Required: 1100 locations.

Speed: Average running time for each datum point is .016 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.

IBM 650 System: Basic

RESIDUALS AND DERIVATIVES OF GRAVITY

E. 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 exists.

b. Range: Maximum size of each map is limited to 100 rows by 9999 columns.

Accuracy: Not given.

Floating/Fixed: Not given.

Mathematical Method: Not given.

Storage Required: The program requires 1472 drum locations, of which 700 are for map storage, 500 for program instructions, 100 for temporary storage, and the remaining 172 are for constants, corrections, read and punch, etc.

Speed: Average running time for each datum point is .016 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.

IBM 650 System: One 533 required.

GRID SYSTEM VOLUME DETERMINATION

O. P. Shinn
Citicorp Service Oil Company (Del.)
Bartlesville, Oklahoma

a. Purpose: This routine computes sand volumes and accumulates volume totals by lease or company.

b. Range: The program will handle up to 490 leases.

Accuracy: Not given.

Floating/Fixed: Fixed decimal arithmetic is used.

Mathematical Method: Given in writeup.

Floating/Fixed: Not given.

Restrictions: Essentially the only data necessary are the relative permeability relationships, the saturation of oil, gas, and water at the beginning of the drive, and the saturation of oil, gas, and water at the beginning of the drive.

Mathematical Method: Given in writeup.

Storage Required: N/A

Remarks: This program makes use of the SIR interpretive routine to permit the computer to perform floating point arithmetic.

IBM 650 System: Basic

THE BUCKLEY-LEVERETT, WEILGE CALCULATIONS

C. R. McEwen
C. A. Rogers
Union Oil Company of California
Research Laboratory
Brea, California

a. Purpose: This program is a method of predicting the recovery of oil when it is being displaced by gas or water.

b. Range: This routine is limited to the relative permeability relationships and the saturations of oil, gas, and water at the beginning of the drive.

Mathematical Method: Given in writeup.

Storage Required: N/A

Remarks: This program uses the SIR interpretive routine to permit the computer to perform floating point arithmetic.

IBM 650 System: Basic

CALCULATION OF RATE OF RETURN USING THE IBM 650 COMPUTER

G. D. Smith
Union Oil Research Laboratory
Brea, California

a. Purpose: This program may be used to calculate the rate of return of an investment. In essence, a discount or interest rate is found which will make the present worth of the future income equal to the investment.

b. Range: The size of the cash flow changes more than once during the life of the investment. Cash flows must be in floating point notation (5120000000•1.0)

Mathematical Method: N/A

Storage Required: N/A

Remarks: A result of 7000/o was obtained for a test problem in less than three minutes of computer time.

IBM 650 System: One 533 required.
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:** IBM 650 with alphanumeric device and a 533.

A PROGRAM FOR PARTITIONING OF ARBITRARILY SHAPED AREA

D. C. Schiller
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, Relocatability:** N/A
e. **Remarks:** Two limitations exist. First, no more than 99 intersections can be counted around any area. Second, the area in square vars and the distance in vars may not exceed 99,999,999,99.
f. **IBM 650 System:** N/A

A PROGRAM FOR THE GAUSS-SOUTHWELL RELAXATION METHOD

H. C. Garnay
D. C. Schiller
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 sets 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 about 17,950 storage spaces where \( m \) is the number of off diagonal elements.
e. **Remarks:** The program is divided into three parts.
E. J. Assiter  
D. H. Eckhardt  
W. Williams  
Mobil De Venezuela  
Caracas, Venezuela

**LEAST SQUARES DETERMINATION OF THE VELOCITY FUNCTION FOR REFRACTION TIME-DEPTH DATA**

- **Purpose:** This program is designed to compute the refraction (V, a) and plot a time-distance curve for these parameters. Since there exist relationships between the refractions (V, a) and the reflections (V, a) it is very useful for velocity determination to be used with the reflection seismograph.
- **Range and Accuracy:** Not given.
- **Mathematical Method:** Least squares.
- **Storage:** Instructions are stored in 0400 to 0800.
- **Speed:** Not given.
- **Remarks:** None.
- **System:** 533, 655 (core and indexing registers).

**IBM 650 Library Program Abstracts**

---

---
Bonner and Moore Engineers
Houston, Texas

a. Purpose: The simulator is a series of thirteen modules for making certain chemical engineering calculations involving vapor-liquid separations with heat and material balances. Its purpose is to permit a process design engineer to write a computer program to simulate the design of many types of equipment and combinations of equipment where vapor-liquid equilibrium and heat and material balance are the unit operations involved.

b. Restrictions, Range: Up to approximately 15 component systems may be handled by reassembly of the program.

Accuracy: Does not apply.

Floating/Fixed: Fixed point.

c. Standard chemical engineering formulas are used.

d. Storage Requirements: 630 drum locations are available for the executive program with the 10 component system while with 2.0 components 480 drum locations are available.

e. Remarks: The ID-3 Interpretive System is an integral part of the Unit Operations Simulator and must be used to write the executive program instructions.

f. IBM 650 System: Basic 650 Required.

IBM 650 Library Program Abstracts

GAS NETWORK ANALYSIS PROGRAM

F. L. Duffy
The Cincinnati Gas & Electric Co.
Cincinnati, Ohio

a. Purpose: This program provides a very flexible method for computing the solution of low, intermediate or high pressure gas networks. Variations in network conditions to arrive at the optimum system development may be entered with a minimum of effort.

Range: Networks with 1800 main sections may be analyzed and any flow formula which can be reduced to the form

\[ h(\text{or } P_a - P_b) = ALQ^2 \]

can be used. The main length and flow may be in any units whatsoever.

Accuracy: The network may be balanced to a predetermined limit of accuracy.

Floating/Fixed: Computations are in a fixed point.

Mathematical Method: The 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.

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

HARDY-CROSS SOLUTION OF WATER FLOW NETWORK

C. G. Fultz
A. A. Lea
IBM, Atlanta, Georgia

Purpose: This program solves for flow in a water network. Given the initial estimates of the flow in each pipe, the routine produces a corrected flow for the system.

Range: A network of up to 99 loops, containing up to 199 pipes, can be handled by this program. The pipes may be up to 99,999 yards in length and of any diameter.

Accuracy: The user may control the accuracy of the solution.

Floating/Fixed: Fixed decimal arithmetic is used.


Storage Required: Virtually the entire drum is used.

Speed: Approximately one second per pipe per iteration, plus two minutes for read-in, punchout and initialization.

Relocatability: Not relocatable.

Remarks: If the initial estimate of flow is too poor, the Hardy-Cross method will not converge, in which case the program stops.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

HYDRAULIC NETWORK ANALYSIS

(Continued on next column)

IBM 650 Library Program Abstracts

BACKWATER CURVE ANALYSIS

(Continued on next page)
FLUID FLOW DISTRIBUTION: HARDY CROSS METHOD

G. Hamilton Harrison
Philadelphia Gas Works
Division of Duble Gas Improvement Company
Philadelphia, Pennsylvania

a. Purpose: This routine computes, by means of successive corrections, the distribution of flow in a gas distribution network.

b. Restrictions, Range: The program can be used for low pressure system networks consisting of up to 150 separate mains, or allowing for mains within a single loop of which the diameter is up to 48 inches. Resistance coefficients are calculated for gas of 0.65 specific gravity.

Accuracy: Undistributed pressure drop within any one loop less than .004 in. flow correction factor for any single loop less than .003 Mbf per hour.

Floating/Fixed: Fixed point arithmetic is used.

c. Method: Procedure of successive corrections (slightly modified Hardy Cross Method) is used.

d. Storage Requirements: Maximum storage requirement for the program is 353 locations.

Speed: Speed depends on the number of internal iterations required.

Relocatability: Not relocatable.

e. Remarks: Resistance constants are calculated and stored in table form for main diameters of 4 to 42 inches, inclusive. The length of mains, in feet, must be within certain limits (see program write-up).

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

FLUID FLOW DISTRIBUTION: HARDY CROSS METHOD

Win. F. Atkinson, Nash
Rich Electronic Computer Center
Georgia Institute of Technology
Atlanta, Georgia

a. Purpose: This program was written to solve gas network problems for the Public Service Electric and Gas Company. The program, using the modified Hardy Cross technique, will be used to supplement the studies being made on the McIlroy Pipeline Network Analyzer. This presentation discusses and exemplifies the intermediate or high pressure network, hence no range limitations exist. Systems with a maximum of 375 pipe sections in each loop.

b. Restrictions, Range: The program utilizes a floating point representation, hence no range limitations exist. Systems with a maximum of 375 pipe sections in each loop.

c. Method: The Hardy Cross Method of successive corrections is used. Energy loss calculations are based on the Darcy-Weisbach equation for energy loss in a straight pipe.

d. Storage Requirements: Does not apply.

Speed: The computer requires approximately 2 seconds per loop per iteration.

Relocatability: Not relocatable.

e. Remarks: The tolerance is set at 0.23%. The tolerance is maintained when all corrections applied in the network during one iteration cycle are within a prescribed tolerance. It is also possible to halt computations after any complete iteration cycle.

f. IBM 650 System: Minimum 544.

IBM 650 Library Program Abstracts

A GAS NETWORK ANALYSIS PROGRAM WITH AUTOMATIC RECYCLING (IBM 650)

Ardit James
Public Service Electric and Gas Company
Newark, New Jersey

a. Purpose: This program was written to solve gas network problems for the Public Service Electric and Gas Company. The program, using the modified Hardy Cross technique, will be used to supplement the studies being made on the McIlroy Pipeline Network Analyzer. This presentation discusses and exemplifies the intermediate or high pressure network.

b. Restrictions, Range: The program utilizes a floating point representation, hence no range limitations exist. Systems with a maximum of 375 pipe sections in each loop.

c. Method: The Hardy Cross Method of successive corrections is used. Energy loss calculations are based on the Darcy-Weisbach equation for energy loss in a straight pipe.

d. Storage Requirements: Does not apply.

Speed: The computer requires approximately 2 seconds per loop per iteration.

Relocatability: Not relocatable.

e. Remarks: The tolerance is set at 0.23%. The tolerance is maintained when all corrections applied in the network during one iteration cycle are within a prescribed tolerance. It is also possible to halt computations after any complete iteration cycle.

f. IBM 650 System: Minimum 544.
b. A comparison of the largest correction (Q) with the desired limit of accuracy, causes the program to perform additional iterations or punch results. This feature permits the problem to be solved during other than prime machine time. A punch of the largest |(Q) at the end of each iteration provides a check on convergence. When the desired accuracy is obtained, flows and pressure drops 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 Spillanes co-efficients, 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. Wiring is for the 533.

---

**IBM 650 Library Program Abstracts**

**ROOT AND GAIN LOCUS**

R. D. Blosser
Firestone Tire & Rubber Co.
Los Angeles, California

a. 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.

b. Range: Degree of forward and feedback loop must be less than 14.


d. Storage Required: The program occupies approximately 1500 drum storage locations.

e. Remarks: The program is self-loading. It does not always work for multiple roots. Transfer functions must be linear polynomials with constant coefficients.

f. IBM 650 System: One 533 required.

**IBM 650 Library Program Abstracts**

**BPR PARALLAX REDUCTION PROGRAM**

K. F. Kohler, Highway Engineer
R. B. DeLalla, Engineering Tech.
D. L. Tingey, Photogrammetric Engineer
Bureau of Public Roads
Region 8
Portland, Oregon

a. Purpose: Reduces distances manually scaled from aerial vertical photographs to actual elevations and distances.

b. Range: Control Stationing (0.00-50.00), and Elevations (0.0000 EE).

C. Cross-section survey data (0.000-8.80), Distance (0.000-8.00), and Base Elevations (0.0000 EE).

c. Accuracy: Computed with manual methods.

d. Floating/Fixed: Fixed.

**IBM 650 Library Program Abstracts**

**LINEAR PROGRAMMING**

H. F. Smith
IBM, Chicago

a) Solves a linear programming problem.

b) All numbers are of the form xxx.xxx.xxx.xxx. An M by N system may be solved where M ≤ 20, N ≤ 50 and M(No1) ≤ 1000. These values pertain to the system after the slack vectors and artificial vectors have been added.

c) Method not given.

d) The entire drum is used. Time required is approximately 90 MN seconds for one iteration.

e) Input consists of matrix elements, cost coefficients, indices of basis, and constants. At the end of each iteration the program punches out the number identifying the variables in the basis, the values of these variables, the value of the functional, and an iteration count.

f) Minimum 650.

---

**IBM 650 Library Program Abstracts**

**IBM 650 Library Program Abstracts**
By changing one instruction it is possible to reduce this cumulative rounding error below its present level. The instruction in location 0068 now reads: 30 0003 0129. It should be changed to read 20 0069 0172.

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>1-13</th>
<th>14-28</th>
<th>29-37</th>
<th>38-64</th>
<th>65-69</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>30 0003 0129</td>
<td>20 0069 0172</td>
<td>Leave</td>
<td>Leave</td>
<td>Leave</td>
</tr>
</tbody>
</table>

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. Scaling, 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."

ERRATA

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.002

LINEAR PROGRAMMING

L. S. Woo
IBM, Los Angeles

March 23, 1956

a) Solves a linear programming problem.

b) A maximum of 97 equations, not including the objective functions, is possible. The number of variables is unlimited. Input is 10 digit fixed-point numbers which are converted to double precision floating-point numbers for the calculations.

c) Method is Recursive Generation of Vectors for the Modified Simplex Method as described by Kurt Eisemann.

d) The entire drum is used. Timing varies from 4 minutes per iteration for the first 10 up to 13 minutes per iteration for the 31st through 40th.

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

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

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.003

TRANSPORTATION PROBLEM

S. Poley
IBM, New York

May 17, 1956

a) Solves the transportation problem, i.e., given the requirements at m destinations, and amounts available at n origins, and the cost of shipment from any origin to any destination the program will determine the minimal mode of transportation of a homogeneous product.

(Continued on next column)

b) All input data are restricted to a maximum size of five digits and all operations are in fixed-point. An approximation to the maximum number of destinations, m, and origins, n, is 5m+5n<300 with n<100.

c) Method is essentially the same as the iterative method proposed by A. Charnes and W. W. Cooper in "Management Science," October, 1954.

d) The entire drum is used. Time estimates not given.

e) Provision is made for alternate solutions which yield the same minimum total cost. A SOAP symbolic deck listing with a sample absolute deck listing is included.

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

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.004

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 Optima) of the Transportation Problem furnished by the 650 Program Library prior to February 28, 1958, contain several erroneous cards. The corrections are too numerous to list here; 650 users who expect to run this part of the program may obtain corrected copies of the deck from the library in the usual manner.

The program listing contained in the detailed write-up is correct as issued.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.005

LINEAR PROGRAMMING

J. W. Davis and D. H. Brown
Esso Standard Oil, Baton Rouge, Louisiana

March 29, 1956

a) Solves a linear programming problem.

b) Fixed decimal arithmetic of the form xxxxx.xxxxx is used. Up to 40 equations and any number of variables may be handled.

c) The modified simplex method is used.

d) The program is divided into four parts. Storage required is approximately 211, 87, 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 10.1.006

LINEAR PROGRAMMING

R. L. Graves
Standard Oil, Indiana

May 10, 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.

(Continued on next column)
c) The dual and direct forms of the revised simplex method are used.
d) The entire drum is required. About 26 minutes are required for a 22 x 46
system.
e) A modified Trimble-Kubie interpretive system is used for the floating-point
arithmetic, see Technical Newsletter No. 8.
f) Minimum 650

IBM 650 Library Program Abstracts

LINEAR PROGRAMMING CODE FOR THE AUGMENTED IBM 650

O. R. Perry
IBM, Los Angeles, California

a. **Purpose:** This routine provides a method to find optimal solutions for
relatively large linear programming problems with flexibility of input and
detailed results, while maintaining simplicity and speed in operation.
b. **Range:** The size of the problem which can be handled is restricted by the
following relationships:

\[(M + 1)(N - M + 1) \leq 1900, \quad M \leq 17, \quad N = 57, \quad \text{speed} \approx 20 \text{ seconds per iteration.}\]
c. **Mathematical Method:** Composite Algorithm; reverts 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:** Accuracy: Single precision,
Floating/Fixed: Floating decimal arithmetic used.
IBM 650 System: One 533, automatic floating decimal arithmetic feature,
I/A, and indexing registers.

IBM 650 Library Program Abstracts

RENT OR BUY ANALYSIS

L. Quint
IBM, White Plains, New York

a. **Purpose:** This program is designed to assist management in making a rent or buy
decision on 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 to
industrial corporations this program will make special evaluations for utilities,
banks, 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 use 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.

IBM 650 Library Program Abstracts

THE SYMMETRIC METHOD OF LINEAR PROGRAMMING

E. W. Weiner
Marquette University
Milwaukee, Wisconsin

a. **Purpose:** This routine solves a linear programming problem using the
Symmetric Method which eliminates slack and artificial vectors.
b. **Range:** The size of the problem which can be handled is restricted by the
following relationship:

\[\left( M + 1 \right) \left( N + 1 \right) \leq 1400, \quad M \leq 50, \quad N \leq 50\]
c. **Mathematical Method:** Symmetric Method Algorithm.
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. **Accuracy:** Single precision,
Floating/Fixed: The Wisconsin Floating Decimal routine is used.
IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

LINEAR PROGRAMMING FORCED INVERSION

F. F. Fisher
Western Region Programming System
3424 Wilshire Blvd.
Los Angeles, California

a. **Purpose:** This program is designed for use with the Linear Programming
CODE for the Augmented 650. It has the following features as compared to the existing
routine in IBM: (1) Allows the analyst to pre-select the
final basis variables; (2) A proper selection is made, the number of
iterations required to obtain an optimal solution may be greatly reduced.
As a result, time of solution due to round off may also be improved.
(3) It is completely compatible with the existing version of the Linear
Programming Code for the Augmented 650.
b. **Accuracy:** Single precision floating point.
c. **Method:** Selected variables are forced into the final basis by a modified
simplex procedure. If optimality has not been achieved, the composite
algorithm is utilized to complete the solution.
d. **Storage Requirements:** The entire storage will ordinarily be required.

IBM 650 Library Program Abstracts

LINEAR PROGRAMMING CODE FOR THE AUGMENTED IBM 650

B. 650

Floating/Fixed: Not given.
c. **Mathematical Method:** Symmetric Method Algorithm.
d. **Storage Required:** Not given.
e. **Speed:** Not given.
f. **Relocatability:** Not given.
g. **Remarks:** None.
h. **IBM 650 System:** One 533 required.
Special Devices: None.

IBM 650 Library Program Abstracts
LINEAR DECISION RULE FOR PRODUCTION AND EMPLOYMENT SCHEDULING

W. Folsom
C. C. Holt
Bell Laboratories Interpretive System
Carnegie Institute of Technology
Pittsburgh, Pa.

a) Calculates optimal linear rules for making decisions on aggregate production and employment utilizing quadratic cost functions.

b) Floating decimal point.

c) The mathematical methods are described in papers appearing in "Management Science" Volume 2, No. 1 and 2, October 1955, January 1956.

d) The program requires the following decks:

(1) The Wolontis System* deck
(2) Complex Operations deck
(3) Arcsin Reallocated deck (decks 2 and 3 developed by Dr. P. Marcus, C. L. Tj)
(4) The Linear Decision Rule Program deck

These programs are not relocatable. All four decks are supplied in a single package.

e) Standard Wolontis* 503 and 602 boards are used.

f) Minimum 650.

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

IBM 650 Library Program Abstracts

PRODUCTION LINE BALANCING

T. E. Daum
Westinghouse Electric Corp.
Mansfield, Ohio

a) Purpose: Given the times and precedence relationships between basic jobs on a zoned assembly line, and given the production rate desired, this routine assigns jobs to operators in such a manner as to minimize the total number of operators required.

b) Range: Maximum of 99 "can do" jobs per line. Maximum of 50 jobs per line. Maximum of 24 jobs per operator.
IBM 650 Library Program Abstracts

2DT: A TWO-DIMENSIONAL TRIM ROUTINE

J. W. Burgesson
C. Kenney
IBM, Akron, Ohio

a. Purpose: This program assigns to any given rectangular "stock" piece a layout pattern for smaller rectangular pieces to be cut.
b. Range: The program can handle only one stock piece at a time, but up to 590 unique sizes of pieces to be cut, up to 999 of each. On sample programs the routine has given patterns with as little waste as 1.4%. The program does well with as few choices as 50 pieces of five unique sizes.

c. Mathematical Method: Table lookup method is used.
d. Storage Required: The entire drum is used.
e. Remarks: None.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

PRODUCTION DAY CALENDAR

R. L. Freeman
Portsmouth Naval Shipyard
Portsmouth, New Hampshire

a. Purpose: This program is written to be used as a subroutine for scheduling events which are based upon normal productive working days.
b. Range: The sample calendar is for a five-year period beginning January 1963 and ending December 1964.
c. Mathematical Method: Table lookup method is used.
d. Storage Required: The calendar requires 242 storage locations, and the program requires 255 locations.
e. Remarks: The program is built around two features of the IBM 650: TLU and D sacked 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.

b. Accuracy: Does not apply.

Floating/Fixed: Not given.
c. Mathematical Method: An approximation method is employed, which may not give a minimum figure in all cases. The exact method of computation has been programmed but is prohibitively long in machine time. The 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 operator time in 99% of the cases run to date.
d. Storage Required: The routine takes up the entire drum and IAS.
e. Remarks: Is using the program, the production line is divided into physical "zones." An operator will not be assigned to jobs in more than one zone. Jobs are subdivided into two types, "must do" and "can do." A "must do" job can be performed in only one particular zone, while a "can do" job might be performed in one of several zones. The routine decides the best zone for each "can do" job.
f. IBM 650 System: One 533, indexing registers, and IAS.

IBM 650 Library Program Abstracts

MIN - SCHEDULING

H. S. Peak
Texas Division
The Dow Chemical Company
Freeport, Texas

a. Purpose: The "LESS" program assumes that the only restriction on starting a job is that every job that precedes it in the activity diagram has been completed. "Man - scheduling" adds a further restriction that the total usage of manpower of all jobs in process at any one time cannot exceed specified maximum limits. Limits on 10 classes, or crafts, can be specified.
b. Range: Does not apply.
c. Accuracy: Does not apply.
d. Arithmetic: Fixed point.
e. Mathematical Method: The program is a continuous updating of job priorities and rearrangement of the order of waiting jobs in progress.
f. Remarks: None.

a. Purpose: The program does well with as few choices as 50 pieces of five unique sizes.
b. Range: The sample calendar is for a five-year period beginning January 1958 and ending December 1962.
c. Mathematical Method: An approximation method is employed, which may not give a minimum figure in all cases. The exact method of computation has been programmed but is prohibitively long in machine time. The 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 operator time in 99% of the cases run to date.
d. Storage Required: The routine takes up the entire drum and IAS.
e. Remarks: Is using the program, the production line is divided into physical "zones." An operator will not be assigned to jobs in more than one zone. Jobs are subdivided into two types, "must do" and "can do." A "must do" job can be performed in only one particular zone, while a "can do" job might be performed in one of several zones. The routine decides the best zone for each "can do" job.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

LESS - Phase IIb - Node-Numbering

Frederick Becker, Jr.
IBM
Dallas, Texas

a. Purpose: The "LESS" program assigns to any given rectangular "stock" piece a layout pattern for smaller rectangular pieces to be cut, up to 999 of each. On sample programs the routine has given patterns with as little waste as 1.4%. The program does well with as few choices as 50 pieces of five unique sizes.

b. Accuracy: Does not apply.

Floating/Fixed: Fixed decimal.
c. Mathematical Method: N/A

d. Storage Required: The second and third tables can occupy 500 positions of memory locations 001-1300, 1301-1800 respectively.
e. Remarks: None.

f. IBM 650 System: One 533 is required.

IBM 650 Library Program Abstracts

CONTINUED ON NEXT PAGE
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.

Accuracy: Exact, using the conventions explained in the write-up when applying the formulas.

Floating/Fixed: Fixed point.

Mathematical Method: Formulas are used rather than tables.

Storage Required: Approximately 300 storage locations.

Speed: Read-punch speed.

Relocatability: Relocatable.

Remarks: The conventions used in applying the formulas are explained in the program write-up.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

GENERAL PURPOSE CALENDAR PROGRAM

N. Jasper
National League for Nursing, Inc.
New York 17, 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.

Accuracy: Exact, using the conventions explained in the write-up when applying the formulas.

Floating/Fixed: Fixed point.

Mathematical Method: Formulas are used rather than tables.

Storage Required: Approximately 300 storage locations.

Speed: Read-punch speed.

Relocatability: Relocatable.

Remarks: The conventions used in applying the formulas are explained in the program write-up.

f. IBM 650 System: One 533 required.
COMPUTER AUTOMATED MUSIC
Norman V. Plyler
University of Rochester Computing Center
Rochester, N. Y.

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 console. 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. Percussion effects, such as 407 type-bars slamming to simulate drum beats or cymbal crashes can be incorporated into the selection to enhance the musical effect.

Range: About one and a half octaves are available from high C through middle C down to G and any score in this range or which may be transposed into this range is applicable. The musical score may contain up to 200 notes.

Mathematical Method: Length of time to complete multiply operation determines spacing of pulses to Data Address Light.

Storage Required: Entire Drum, IAS, Index Registers.

Equipment: IBM 650 System including IAS and Index Registers, Digital-to-Audio Converter (Heathkit).

DEBUGGING PROGRAMS
A. M. Petrasanta
IBM, New York

This paper describes a complete, automatic debugging procedure designed to provide the maximum amount of information about a malfunctioning program in the minimum amount of programmer and machine time. The following routines are used in the debugging procedure and complete information about them is given: Flow Tracer, Snapshot Tracer, Symbolic Seven-Per-Card Punch, all by S. Poley; Symbolic Tracing Routine by W. P. Helsing and S. Poley; and Step Codes by F. J. Charnko.

The above routines, except the last one, are written in SOAP symbolic form, and are designed to be used by the SOAP programmer most effectively. The routines, however, can be used by the non-SOAP, or absolute, programmer, but a rudimentary knowledge of the SOAP system is necessary.

FLOW DIAGRAMMING FOR THE IBM 650
B. Diamond
A. E. Charrow
I. M. Sobol
Service Bureau Corporation
Los Angeles, California

This paper describes a flow diagramming technique for the IBM 650. The method is an adaptation of the von Neumann-Goldstine system, and is designed primarily for mathematical and scientific problems.
EAM 0704 PROGRAM LIBRARY ABSTRACT

0704 055LAIWI AVAILABLE PRIOR TO JANUARY 1962
MATRIX INVERSION
Evaluates a matrix stored in core storage, uses an elimination method, the largest element in the column, but the columns are used in order from left to right. The original matrix is destroyed, and its inverse is stored in memory by the inverse. The routine requires 171 cells plus 2N& COMMON. A 1 by 2 matrix can be inverted in a 4076 word machine in about 100 seconds.

0704 065LIPS6 AVAILABLE PRIOR TO JANUARY 1962
Floating exponential
Evaluates floating X to floating X for X greater than zero, accurate to 0 or -3 in eighth decimal digit. Maximum time about 2.27 milliseconds. Uses 63 storage cells 65 common.

0704 065LIPS20 AVAILABLE PRIOR TO JANUARY 1962
Floating natural logarithm
Computes floating natural log of floating X for X greater than zero. TSES sequence with error return for an X of zero or less. Accurate to 0 or -3 in eighth significant decimal digit. Maximum time about 0.17 milliseconds. Uses 69 storage cells 65 common. FORD-- 171.

0704 075UACSH2 AVAILABLE PRIOR TO JANUARY 1962
Reads bcd tape or online card reader. Read either bcd tape with redundancy checking or Hollerith tape. Information is stored in core in BCD form. Routine requires 167 cells plus 9 common.

0704 075UADD5I AVAILABLE PRIOR TO JANUARY 1962
Decimal, octal, bcd loader
Uses with UA TSM 2 or UA CSM 2. Controls tape program UA TSM 2 or tape or card program UA CSM 2 to read information into core. Converts this information to binary - fixed or floating decimal numbers being converted to fixed or floating binary numbers, and decimal or octal integers being converted to binary integers. Reads and stores Hollerith labels. Conversion in this case format is variable. Routine requires 66 storages plus 5 common. Be controlled by transfer cards. Routine requires 172 cells plus 24 common.

0704 075ULM4I AVAILABLE PRIOR TO JANUARY 1962
Interpretation matrix abstraction
Interprets matrix pseudo-instructions and transfers to connect subroutine. Reads from drum to c.s. if necessary. Requires 64 storages plus 2 common if read drum, 24 storages if drum not read.

0704 085CLM1 AVAILABLE PRIOR TO JANUARY 1962
Linear programming system
Uses modified Simplex method with product form of inverse, will solve problems having 297 equations and any number of available. OOD is complete with 210 routines to all complicated backups. Special features include Paremetric linear prec. multiple optimizing forms, Sundry partitioning and restart devices. I/O is fixed pt. calc is DBL prec fl pt. Standard share boards are used. IO on binary cards is indicative of function and is not aps1. Corr.-- 361.253. 306. 328. 340. 080. 064.

0704 105LGRP AVAILABLE PRIOR TO JANUARY 1962
Determination evaluation
Evaluates by gauss elimination method the determinant of a real or complex matrix of order N in single or double precision. Designed for use with GL OPAL1 normal tsx sequence. Uses 191 storages.

0704 105LDP1 AVAILABLE PRIOR TO JANUARY 1962
Floating point double precision abstraction
Allows a set of 20 machine language operations which can be executed in single precision with negligible loss of time or in a double precision mode under control of sense switch 1. Normal tsx sequence. Uses 275 storages.

0704 085CLM1 AVAILABLE PRIOR TO JANUARY 1962
Neumann's method for finding roots of polynomials
Computes roots of a real or complex polynomial of order 6 in single or double precision. Designed for use with GL OPAL1 calling sequence specifies convergence factor. Uses 575 storages plus kfp common for single precision or dualkfp common for double precision.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 116CLS01 AVAILABLE PRIOR TO JANUARY 1962

LEAST SQUARES POLYNOMIAL FIT

A FIT POLYNOMIAL OF ORDER ONE THROUGH SEVEN TO N GIVEN POINTS

OF EQUAL INTERVAL SPACING, ORDERS 12 STORAGES PLUS 3 COMMON.

0704 116CLS02 AVAILABLE PRIOR TO JANUARY 1962

LEAST SQUARES POLYNOMIAL SOLUTION

OF SIMULTANEOUS EQUATIONS

SOLVE P SIMULTANEOUS EQUATIONS IN N UNKNOWNS. SOLUTION IS

REPRESENTED AS A POINT IN FLOATING POINT. REQUIRE 26 STORAGES

PLUS 5 COMMON.

0704 116CLEL AVAILABLE PRIOR TO JANUARY 1962

RELATIVEIZE SYMBOILIC DECK

CONSISTS OF TWO DECKS DESIGNATED AS REL1 AND REL2. REPRODUCE

SYMBOILIC DECK WITH LOCATION SYMBOLS RELATIVE TO FIRST.

OUTPUT IS TO TAPE FOR OFF-LINE PRINTING ONLY, SIMILAR TO

SAME IN MANY RESPECTS, USES CORE AND TAPES 1 AND 2, AND

TAPE 4 IF INPUT FROM TAPE. REVISED DIST. 236

0704 116CLSPM AVAILABLE PRIOR TO JANUARY 1962

SIMULTANEOUS REAL EQUATIONS

DETERMINANT AND EIGENVECTOR FOR REAL MATRIX

REQUIRES 26 STORAGES PLUS 1 COMMON.

0704 116CLSPM1 AVAILABLE PRIOR TO JANUARY 1962

SIMULTANEOUS REAL EQUATIONS

DETERMINANT AND EIGENVECTOR FOR REAL MATRIX

REQUIRES 26 STORAGES PLUS 1 COMMON.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 116CLSPM3 AVAILABLE PRIOR TO JANUARY 1962

SIMULTANEOUS REAL EQUATIONS

X VECTOR SOLUTIONS OF N SIMULTANEOUS EQUATIONS. REQUIRES 324

STORAGES PLUS 7 COMMON.

0704 116CLT1 AVAILABLE PRIOR TO JANUARY 1962

TAN

TAN X FOR X IN RADIANS. REQUIRES 63 STORAGES PLUS 1 COMMON.

0704 121DHAN1 AVAILABLE PRIOR TO JANUARY 1962

HARMONIC ANALYSIS SUBROUTINE

GIVEN A TABLE OF Y IN AN INTERVAL, WHERE Y = F(X),

 WHICH CORRESPONDS TO A SET OF EQUALLY SPACED VALUES OF X.

HAS THE COEFFICIENTS OF A TRIGONOMETRIC SERIES.

0704 121HAN2 AVAILABLE PRIOR TO JANUARY 1962

AITKEN INTERPOLATION FOR N EQUAL INTERVALS

A FLOATING POINT INTERPOLATION ROUTINE USING AITKEN

METHOD FOR EQUAL INTERVALS OF THE ARGUMENT. MAY BE USED

FOR ANY ORDER OF INTERPOLATION. AITKEN'S METHOD AFFORDS A

MORE CONCISE FORMULATION THAN OTHER EQUIVALENT POLYNOMIAL

METHODS.

0704 121HAN3 AVAILABLE PRIOR TO JANUARY 1962

AITKEN INTERPOLATION FOR N EQUAL INTERVALS

A FLOATING POINT INTERPOLATION ROUTINE USING AITKEN

METHOD FOR EQUAL INTERVALS OF THE ARGUMENT. MAY BE USED

FOR ANY ORDER OF INTERPOLATION. AITKEN'S METHOD AFFORDS A

MORE CONCISE FORMULATION THAN OTHER EQUIVALENT POLYNOMIAL

METHODS.

0704 121IPNORM AVAILABLE PRIOR TO JANUARY 1962

RANDOM NUMBER GENERATOR

CALCULATES A RANDOM NUMBER. REQUIRES 20 STORAGES.

0704 1411AS88 AVAILABLE PRIOR TO JANUARY 1962

SOLUTION OF GENERAL MATRIX EQUATION Ax = b.

GIVEN AN ARRAY OF P COLUMNS AND N ROWS, M GREATER THAN N,

OF ELEMTENTS STORED ROW-WISE AT L WHERE A IS N X N AND B IS N X M.

S 885 FINDS THE SOLUTION MATRIX X = A-1 OF DIMENSION N X M.

THE SOLUTION MATRIX IS STORED ROW-WISE AT L. THE PROGRAM IS

GENERALY MOST USEFUL WHEN A IS A COLUMN MATRIX SO THAT X IS

THE SOLUTION TO A SYSTEM OF N LINEAR EQUATIONS IN N UNKNOWNs,

OR WHEN B IS THE IDENTIY MATRIX SO THAT X IS THE INVERSE

OF A, OR TO SET BOTH THE SOLUTION AND THE INVERSE. S 885 USES

293 CELLS AND 1 COMMON.
DIFFERS FROM GL OUTZ IN FOLLOWING WAYS--
USES SENSE SWITCH 2 FOR CARO OR TAPE INPUT OPTION.
FOR !', AND STORED IN CORE STORAGE.
BLOCKS OF FLOATING POINT
DECIMAL C&RACTE c-Mtt.N&O&-NOO N V&-&&&-E -t &
STORAGE CELLS & 114 OP CODE TABLE & 40 ERASABLE
OATA1 FIXED POINT OATAr BINARY COUEO DECIMAL DATA, ANO/OR
0704 204GS INOZ AVAILABLE PRIOR TO JANUARY 1962
ACTUAL DECIMAL INSTRUCTIONS HAVE TO BE READ.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 204GSOUTX AVAILABLE PRIOR TO JANUARY 1962
65 REVISION OF GL OUTZ IN FOLLOWING WAYS——
TAPE OR PRINTER OUTPUT CONTROLLED BY SENSE SWITCH 3,
NO CD-OUTPRINT, LESS FLEXIBLE SPACE CONTROL, PRINTER OUT
PRINTS FLOATING POINT OUTPUT WITH EXPONENT FOLLOWING NUMBER.
400 CELLS OF STORAGE 51 ERASABLE.

0704 206NYIP1 AVAILABLE PRIOR TO JANUARY 1962
INPUT PROGRAM UNDER SENSE SWITCH CONTROL
READS DECIMAL, OCTAL OR BCD INFORMATION FROM A BCD TAPE OR
PUNCHED CARDS, CONVERTS TO BINARY, AND STORES THE RESULTS IN
CORE STORAGE. THIS IS A PACKAGED PROGRAM INCORPORATING UARDI
AND UACR+-. IT USES 972 LOCATIONS.

0704 206NYIP2 AVAILABLE PRIOR TO JANUARY 1962
INPUT PROGRAM UNDER SENSE LIGHT CONTROL
READS DECIMAL, OCTAL OR BCD INFORMATION FROM A BCD TAPE OR
PUNCHED CARDS, CONVERTS TO BINARY AND STORES THE RESULTS IN
CORE STORAGE. THIS IS A PACKAGED PROGRAM INCORPORATING UARDI
AND UACR+-. IT USES 570 LOCATIONS.

0704 206NYOUT2 AVAILABLE PRIOR TO JANUARY 1962
DECIMAL OUTPUT PROGRAM UNDER SENSE LIGHT CONTROL
CONVERTS BINARY NUMBERS TO DECIMAL NUMBERS IN BINARY CODED
DECIMAL FORM AND WRITE THESE ON TAPE 2 AND OR PRINT THEM ON
THE OFF-LINE PRINTER. PROGRAM INCORPORATES UA BADI AND UA SPH.
OCCUPIES 611 LOCATIONS OF WHICH THE LAST 94 ARE ERASABLE.

0704 213NYLOAD AVAILABLE PRIOR TO JANUARY 1962
A VARIABLE FIELD PERIPHERAL INPUT
THIS ROUTINE WILL READ A TAPE PREPARED BY THE PERIPHERAL CARD
READER AND PRINT THE RESULTS IN CORE STORAGE. THIS ROUTINE IS SELF-
CONTAINED. IT IS CALLED FROM WITHIN THE PROGRAM.
IT USES 134 LOCATIONS.

0704 213NYBP5 AVAILABLE PRIOR TO JANUARY 1962
BINARY PUNCH PROGRAM
IF BLOCKS WILL PUNCH A BLOCK OF N WORDS FROM MAGNETIC CORE
STORAGE INTO ABSOLUTE BINARY CARDS. THIS ROUTINE IS SELF-
CONTAINED. IT IS CALLED FROM WITHIN THE PROGRAM.
IT USES 145 LOCATIONS.

0704 213NYPTO AVAILABLE PRIOR TO JANUARY 1962
BINARY TAPE OR DRUM PUNCH
READS ONE RECORD FROM TAPE OR DRUM, OR WRITES ONE RECORD INTO
TAPE OR DRUM. REPLACES NVIDEO AND NVIDO SHARE DISTRIBUTION
75.

0704 215NYBOL AVAILABLE PRIOR TO JANUARY 1962
BINARY OCTAL LOADER
LOADS ABSOLUTE BINARY CARDS AND/OR OCTAL CARDS INTO MAGNETIC
CORE STORAGE, AND WILL EXIT ON A BINARY TRANSFER CARD. OCCUPIES
LOCATIONS 5-117 OCTAL

0704 216NYPLS AVAILABLE PRIOR TO JANUARY 1962
NY DOLL TRANSITION
INTERRUPTS CARD LOADING BY NY DOLL AND SIMULATES PUSHERING THE
LOAD CARDS BUTTON

0704 223CLHIV3 AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION SUM OR PRODUCT OF TWO DOUBLE PRECISION
FLOATING NUMBERS. REQUIRES 26 STORAGEs, NO COMMON.

0704 223CLHIV2 AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV1 AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV0 AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV- AVAILABLE PRIOR TO JANUARY 1962
OEONGE KF OF THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.

0704 223CLHIV AVAILABLE PRIOR TO JANUARY 1962
OBTAINS THE DOUBLE PRECISION COMPLEX SUM OR PRODUCT OF TWO DOUBLE PRECISION
COMPLEX NUMBERS. MAY ALSO BE USED FOR DOUBLE PRECISION REAL OR FPD.
REQUIRES 15 STORAGES, NO COMMON.
VECTOR DOT PRODUCT
COMPUTES THE SCALAR PRODUCT OF TWO N TH ORDER REAL OR
REQUIRES 280 STORAGE CELLS PLUS 3N CELLS DETERMINED BY THE
EIGENVECTORS OF A REAL NXN MATRIX USING A POWER METHOD.

HYPERBOLIC SINE-COSINE, FLOATING
ROUNDED TO SINGLE PRECISION. PRINTING OF COEFFICIENTS AND
REQUIRES 580 STORAGES PLUS COMMON

DIFFERENTIATE 4RE PROVIDED. THE DATA POINTS MUST BE IN

ARGUMENT. COSH IN MQ ON EXIT. SINH IS ACCURATE TO 2 BITS LESS
THAN THE NUMBER OF FRACT lONAL BITS IN THE ARGUMENT 1 BUT NO

ONE POINT BOUNDRY CONDITION DIFFERENTIAL EQU. SOLVER

INCOMPLETE ELLIPTIC INTEGRALS
IS A SUBROUTINE WHICH EVALUATES THE INCOMPLETE ELLIPTIC
INTEGRALS OF THE FIRST AND SECOND KIND FROM A KNOWN PH&
AND X. AUSSIAN INTERGRATION DEFINED BY THE LEGENDRE POLYNOMIAL
IS EMPLOYED.

VECTOR DOT PRODUCT
COMPUTES THE SCALAR PRODUCT OF TWO N TH ORDER REAL OR
REQUIRES 280 STORAGE CELLS PLUS 3N CELLS DETERMINED BY THE
EIGENVECTORS OF A REAL NXN MATRIX USING A POWER METHOD.

HYPERBOLIC SINE-COSINE, FLOATING
ROUNDED TO SINGLE PRECISION. PRINTING OF COEFFICIENTS AND
REQUIRES 580 STORAGES PLUS COMMON

DIFFERENTIATE 4RE PROVIDED. THE DATA POINTS MUST BE IN

ARGUMENT. COSH IN MQ ON EXIT. SINH IS ACCURATE TO 2 BITS LESS
THAN THE NUMBER OF FRACT lONAL BITS IN THE ARGUMENT 1 BUT NO

ONE POINT BOUNDRY CONDITION DIFFERENTIAL EQU. SOLVER

INCOMPLETE ELLIPTIC INTEGRALS
IS A SUBROUTINE WHICH EVALUATES THE INCOMPLETE ELLIPTIC
INTEGRALS OF THE FIRST AND SECOND KIND FROM A KNOWN PH&
AND X. AUSSIAN INTERGRATION DEFINED BY THE LEGENDRE POLYNOMIAL
IS EMPLOYED.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 2400550 AVAILABLE PRIOR TO JANUARY 1962
SIMULTANEOUS MULTIPLE INTEGRATION, FLOATING POINT.
CARRIES OUT SIMULTANEOUSLY N MULTIPLE IF DESIRED INTEGRATIONS
BETWEEN SAME LIMITS. FLOATING POINT. MODIFIED SIMPSON RULE WITH INTERVALS AUTOMATICALLY ADJUSTING TO MEET ERROR SPECIFICATIONS. FOR MULTIPLE INTEGRATION, SUBROUTINE NEED BE ENTERED IN MEMORY ONLY ONCE. REQUIRES 243 WORDS STORAGE PLUS COMMON THROUGH COMMON E. 4.

0704 25441355 AVAILABLE PRIOR TO JANUARY 1962
SINE - COSINE FUNCTION TO COMPUTE THE SINE OR COSINE OF A FLOATING POINT NUMBER.

0704 2411610 AVAILABLE PRIOR TO JANUARY 1962
DIFFERENTIAL EQUATIONS ROUTINE AN OPEN SUBROUTINE TO SOLVE A SET OF N SIMULTANEOUS FIRST ORDER DIFFERENTIAL EQUATIONS. REQUIRES 285 ± 20 STORAGE.

0704 2411701 AVAILABLE PRIOR TO JANUARY 1962
OVERFLOW UNDERFLOW, AND DIVIDE CHECK TESTS DIVIDE THE AMOUNT OF OVERFLOW UNDERFLOW AND DIVIDE CHECK INDICATOR. REQUIRES 34 STORAGES.

0704 2411802 AVAILABLE PRIOR TO JANUARY 1962
REVERSE PARADOXICAL INTERPOLATION INTERPOLATES A FUNCTION, Z-F/X,Y/, GIVEN N VALUES OF X, N VALUES OF Y, AND THE CORRESPONDING Z-F/X,Y/. REQUIRES 156 STORAGES PLUS 20 COMMON.

0704 2411901 AVAILABLE PRIOR TO JANUARY 1962

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 2412661 AVAILABLE PRIOR TO JANUARY 1962
THERMAL ANALYSIS THIS IS A COMPUTER-TYPE PROGRAM TO SOLVE TRANSIENT AND STEADY-STATE THERMAL PROBLEMS WHICH CAN BE REPRESENTED BY A SIMPLE ELECTRICAL NETWORK. USES TAPES 3, 4, 5 AND 6.

0704 2500651 AVAILABLE PRIOR TO JANUARY 1962
FIXED POINT FOURIER COEFFICIENTS COMPUTES FOURIER COEFFICIENTS FOR A GIVEN FIXED POINT, SINGLE PRECISION FUNCTION, GIVING EITHER COMPLETE FOURIER SERIES, SINE SERIES, OR COSINE SERIES.

0704 2511501 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511502 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511503 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511510 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511511 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511512 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511513 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511520 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

0704 2511521 AVAILABLE PRIOR TO JANUARY 1962
MURA INTEGER DUMP PRINTS THE CONTENTS OF A BLOCK OF CORE STORAGE AS FIXED POINT INTEGERS. LOCATIONS 0-102/FOCAL// ARE OVERWRITTEN. Printer OPERATES AT FULL SPEED.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 2532001 AVAILABLE PRIOR TO JANUARY 1962
MURA EIGENVALUE FOR SYMMETRIC MATRICES IN FLOATING POINT NUMBERS, THE ROUTH-05 OCCUPIES 364 LOCATIONS WITH TEM PLATES.

0704 2532002 AVAILABLE PRIOR TO JANUARY 1962
MURA EIGENVALUE FOR SYMMETRIC MATRICES IN FLOATING POINT NUMBERS, THE ROUTH-05 OCCUPIES 364 LOCATIONS WITH TEM PLATES.

0704 2532003 AVAILABLE PRIOR TO JANUARY 1962
MURA EIGENVALUE FOR SYMMETRIC MATRICES IN FLOATING POINT NUMBERS, THE ROUTH-05 OCCUPIES 364 LOCATIONS WITH TEM PLATES.

0704 2532004 AVAILABLE PRIOR TO JANUARY 1962
MURA EIGENVALUE FOR SYMMETRIC MATRICES IN FLOATING POINT NUMBERS, THE ROUTH-05 OCCUPIES 364 LOCATIONS WITH TEM PLATES.
DEBUGGING ROUTINES ANO JOB ACCTG. REQUIRES
0704 262NYPCV, AVAILABLE PRIOR TO JANUARY 1962
OFF LINE OPERATION OF THE 704 OPERATES IN THREE PHASES
0704 261GMJOS AVAILABLE PRIOR TO JANUARY 1962
VERIFIES AN EXECUTIVE ROUTINE WHICH CONTROLS MULTIJOB NON-STOP
AN EXECUTIVE ROUTINE WHICH CONTROLS MULTIJOB NON-STOP
LOADS ABSOLUTE BINARY CARDS WITH Z4 WORDS PER CARD. A
24 HORD PER CARD BINARY LOADER
READS OCTAL ADDRESSES AND WORDS FROM CARDS, CONVERTS TO
READS ONE OR TWO DECIMAL INTEGERS FROM A CARD AND PLACES
MURA READ DECIMAL INPUTS ROUTINE
READS ONE OR TWO DECIMAL INTEGERS FROM A CARD AND PLACES
MURA READ OCTAL NUMBER ROUTINE
READS OCTAL ADDRESSES AND WORDS FROM CARDS, CONVERTS TO
MURA READ OCTAL NUMBER ROUTINE
READS OCTAL ADDRESSES AND WORDS FROM CARDS, CONVERTS TO
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
MURA FIXED POINT ARCTANGENT ROUTINE
COMPUTES ARCTANGENT OF A FIXED POINT FRACTION, REQUIRES 27 WORDS PLUS 2 COMMON.
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE ODTO ABSOLUTE
MURA BINARY PUNCH ROUTINE
PIECES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE
MURA DUMP ROUTINE
COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION
PRINTS ONLINE IN OCTAL THE CONTENTS OF ANY SPECIFIED CORE LOCATIONS ALONG WITH ANY DESIRED BCD INFORMATION. THIS SUBROUTINE MAY BE USED TO MONITOR PROGRAMS, E.G., TO PRINT OUT THE CONTENTS OF A VARIABLE CONTROL WORD UPON ENCOUNTERING AN ERROR. 

MONITOR SUBROUTINE

MONITOR SUBROUTINE AND OUTPUT PROGRAM

IBM 0704 PROGRAM LIBRARY ABSTRACT

PRINTS ONLINE IN OCTAL THE CONTENTS OF ANY SPECIFIED CORE LOCATIONS ALONG WITH ANY DESIRED BCD INFORMATION. THIS SUBROUTINE MAY BE USED TO MONITOR PROGRAMS, E.G., TO PRINT OUT THE CONTENTS OF A VARIABLE CONTROL WORD UPON ENCOUNTERING AN ERROR. 

MONITOR SUBROUTINE

MONITOR SUBROUTINE MAY BE USED TO MONITOR PROGRAMS, E.G., IBM 0704 PROGRAM LIBRARY ABSTRACT

RANDOi.i NUMBER GENERATOR

GENERATES FIXED OR FLOATING POINT UNIFORM RANDOM NUMBERS

SUBROUTINE MAY BE USED TO MONITOR PROGRAMS, E.G., IBM 0704 PROGRAM LIBRARY ABSTRACT

COMPUTATION OF THE TRANSCENDENTAL FUNCTIONS MU AND NU ARE COMPUTED BY POLYNOMIAL APPROXIMATION. REQUIREMENTS BASED ON UNCERTAINTY WITH AN ERROR RETURN. 107 CELLS & 11 COMMON WORDS.

MURA FILTER POINT CUBE ROOT 

COMPUTES THE CUBE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 29 WORDS PROGRAM PLUS 3 TEMPORARY, TAKING IS 1.2 MILLISECONDS PER ITERATION.

MURA SIX COLUMN FRACTION CATHODE RAY TUBE DISPLAY SCANS DATA LOCATIONS LOCATED IN SUCCESSIVE CORE MEMORY LOCATIONS AS THE LINE. 93 PROGRAM PLUS 7 COMMON WORDS. TIMES 550 MS PER LINE.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.

MURA CATHODE RAY TUBE POINT PLOTTER

DISPLAYS A SEQUENCE OF POINTS ON THE CRT. POINTS ARE PLACED AT REGULAR INTERVALS ALONG THE X AXIS. 75 WORDS PROGRAM, AVERAGE TIME PER POINT PLOTTED IS 1.15MS. ON SUBSEQUENT ENTRY.

MURA FLOATING DECIMAL DUMP

PRINTS A SPECIFIED BLOCK OF NUMBERS FROM STORAGE IN FLOATING POINT FORMAT. MURA PRINTER BOARD 1 IS REQUIRED. THE LOCATIONS PEP 0 THROUGH 26 ARE USED BY THIS ROUTINE AND WORDS IN THEM ARE DESTROYED.
REDUCES THE SIZE OF A RETLOCATABLE BINARY DECK OR AN ABSOLUTE BINARY DECK CONTAINING PATCH CARDS BY PUNCHING ANOTHER ABSOLUTE DECK." USES CELLS 0-35.

DOUBLE PRECISION INPUT SCALING FRANK MAJDALI SCALES A GIVEN DOUBLE PRECISION BINARY NUMBER." USES CELLS 0-35.

DOUBLE PRECISION OUTPUT SCALING FRANK MAJDALI SCALES A GIVEN DOUBLE PRECISION BINARY NUMBER TO A DOUBLE PRECISION INTEGER FOR OUTPUT. USES CELLS 0-35.

MOVING AVERAGE OF TIME-SERIES DATA TO ANALYZE A SET OF NON-STATIONARY TIME-SERIES DATA FOR TREND AND/OR PERIODIC COMPONENTS. MOVING AVERAGES ARE USED TO MEASURE THE TREND OR NON-STATIONARY COMPONENTS, WHEREAS THE DEVIATIONS OF THE ORIGINAL X FROM THE MOVING AVERAGES INDICATE SHORTER FLUCTUATIONS. PERIODIC AVERAGES OF THE DEVIATIONS GIVE AN ESTIMATE OF THE PERIODIC COMPONENTS IN THE ORIGINAL DATA. THE OUTPUT OF MOVING AVERAGES AND DEVIATIONS MAY BE USED DIRECTLY AS INPUT TO OTHER ROUTINES. USES CELLS 0-35.

EIGENVALUE SOLUTION. REAL. TO FIND THE HIGHEST EIGENVALUE AND CORRESPONDING EIGENVECTOR OF THE MATRIX EQUATION. USES CELLS 0-35.

ATMOSPHERIC DATA SUBROUTINE THIS SUBROUTINE EFFECTIVELY REPRODUCES PORTIONS OF THE ATMOSPHERIC DATA BASED ON THE ARDC MODEL ATMOSPHERE FOR UP TO 53 KILOMETERS. GIVEN ALTITUDE, FIND CORRESPONDING TEMPERATURE IN DEGREES RANKINE, PRESSURE RATIO, DENSITY RATIO AND VELOCITY OF SOUND IN FT PER SEC. USES CELLS 0-35.

TABLE SEARCH ROUTINE ROUTINE USES BINARY SEARCH TECHNIQUE TO FIND AN ENTRY IN AN INDEXED TABLE. CENTRAL SEARCH LOOP CONSUMES NINE CYCLES FOR EACH ENTRY EXAMINED. TABLE LENGTH MAY RANGE FROM ONE WORD TO ALL OF STORAGE. MEAN SEARCH TIME FOR A 1000 WORD TABLE IS 0.56 MSEC. USES CELLS 0-35.

THREE SUBROUTINES SAVE THE CONSOLE FOR CORE, TAPE CHECK, AND SIMULTANEOUS EQUATIONS SUBROUTINE. USES CELLS 0-35.

DETERMINANT EVALUATING SUBROUTINE GIVEN AN ARBITRARY SQUARE MATRIX A AND SOME FLOATING POINT VARIABLE X, THIS SUBROUTINE WILL EVALUATE THE EXPRESSION. USES CELLS 0-35.

A SINGLE-VALUED ARCTANGENT ROUTINE FOR THE ARGUMENTS WITH PROPER QUADRANT ASSIGNMENTS. DIVISION IS CHECKED. USES CELLS 0-35.

ORTHOGONAL POLYNOMIALS SUBROUTINE USES CELLS 0-35.

ITERATION SUBROUTINE GIVEN A RELATIVE ERROR, THIS TECHNIQUE ACCELERATES THE RATE OF CONVERGENCE IF THE ITERATION CONVERGES AND INDUCES CONVERGENCE IF THE ITERATION DIVERGES.

SIMULTANEOUS EQUATIONS SUBROUTINE SOLVES AN EQUATION WHERE A IS A AND X ARE MATRICES OF N BY N-TYPE. USES CELLS 0-35.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 370S0135 AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED MULT.-EXTD RANGE FLOATING BINARY ARITH.
TO MULTIPLY TWO NUMBERS EXPRESSED IN EXTENDED RANGE
FLOATING BINARY, EACH NUMBER OCCUPIES 2 MEMORY CELLS,
35 BIT FRACTION AND 35 BIT EXPONENT. 27 CELLS X 2 CELLS
OF COMMON.

0704 370S0132 AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED DIV.-EXTD RANGE FLOATING BINARY ARITH.
TO DIVIDE TWO NUMBERS EXPRESSED IN EXTENDED RANGE
FLOATING BINARY, EACH NUMBER OCCUPIES 2 MEMORY CELLS,
35 BIT FRACTION AND 35 BIT EXPONENT. PROVIDES FOR ERROR RETURN IN
CASE OF A DIVIDE CHECK. 29 CELLS X 2 CELLS OF COMMON.

0704 370S0139 AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED LOG-EXTD RANGE FLOATING BINARY ARITH.
TO EVALUATE THE NATURAL LOGARITHM OF A NUMBER EXPRESSED IN
EXTENDED RANGE FLOATING BINARY. NUMBER OCCUPIES 2 MEMORY CELLS,
35 BIT FRACTION AND 35 BIT EXPONENT. ERROR RETURN
PROVIDED. 30 CELLS MUST BE IN MEMORY. 151 CELLS X 6 CELLS
OF COMMON.

0704 370S0134 AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED EXP-EXTD RANGE FLOATING BINARY ARITH.
TO EVALUATE THE EXPONENTIAL OF A NUMBER EXPRESSED IN EXTENDED
RANGE FLOATING BINARY. NUMBER OCCUPIES 2 MEMORY CELLS,
35 BIT FRACTION AND 35 BIT EXPONENT. ERROR RETURN WHEN
OUT OF RANGE. 150 CELLS X 9 CELLS OF COMMON.

0704 370S0135 AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED ARC-TAN-EXTD RANGE FLOATING BINARY ARITH.
TO EVALUATE THE ARC-TANGENT OR A NUMBER EXPRESSED IN EXTENDED
RANGE FLOATING BINARY. NUMBER OCCUPIES 2 MEMORY CELLS,
35 BIT FRACTION AND 35 BIT EXPONENT. ERROR RETURN. 350 CELLS MUST BE IN MEMORY.

189 IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 370S0136 AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED SQ-ROOT-EXTD RANGE FLOATING BINARY ARITH.
TO EVALUATE THE SQUARE ROOT OF A NUMBER EXPRESSED IN EXTENDED
RANGE FLOATING BINARY. NUMBER OCCUPIES 2 MEMORY CELLS,
35 BIT FRACTION AND 35 BIT EXPONENT. 69 CELLS X 5 CELLS
OF COMMON.

0704 370S0139 AVAILABLE PRIOR TO JANUARY 1962
DECIMAL PRINT-EXTD RANGE FLOATING BINARY ARITH.
TO PRINT ON-DECK UP TO 6 NUMBERS PER LINE. NUMBERS IN MEMORY
AS EXTENDED RANGE FLOATING BINARY. A 10 DIGIT FRACTION PLUS SIGN
AND A 3 DIGIT EXPONENT PLUS SIGN IS PRINTED. PROVIDES
EQUIPMENT, COMPUTING BETWEEN LINES, AND
OUTPUT CHECKEN WITH OVERPRINT ON FAILING LINES. 356 CELLS X 46 CELLS OF COMMON.

0704 370S0148 AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT FIXED POINT DECK INPUT.
READS UP TO FOUR DECK NUMBERS PER CARD AND STORES THEM IN
COMMON STORAGE AS EITHER NORMALIZED FLOATING POINT OR FIXED
POINT BINARY NUMBERS. PROVIDES FOR COMPUTING BETWEEN CARD IF
DESIRED AND FOR ALTERING THE EFFECTIVE STORAGE LOCATION.
NORMAL TX SEQUENCE WITH ONE CONTROL WORD, ERROR RETURN, AND
TWO NORMAL RETURNS DEPENDING UPON WHETHER THERE IS COPIING
BETWEEN CARDS. USES 750 STORAGE CELLS X 41 COMMON.

0704 370S0148 AVAILABLE PRIOR TO JANUARY 1962
CONSOLE, AUTOMATIC OPERATOR SYSTEM.
READS SYMBOLIC CONTROLS CODES ON TAPE, AUTOMATICALLY
CHRISSES CODES ON TAPE AND CORRECTS THEM OR RINGS THEM.
PRINTS MONOTONICALLY, BUT NO LISTING. LIBRARY OF SUBROUTINES IS AVAILABLE ON TAPE.
INCLUDES ACP-PROTOCOLS, BUT NO PERIPHERAL TAPE-EQUIPMENT IS USED.
USEFUL FOR REMOTE USE OF COMPUTER BY PROGRAMMERS.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 370C0002 AVAILABLE PRIOR TO JANUARY 1962
TRIPLE PRECISION ARITHMETIC PACKAGE
PERFORMS BASIC ARITHMETIC OPERATIONS ON TRIPLE PRECISION
FLOATING POINT NUMBERS. EACH NUMBER REPRESENTED AS A SIGNED
30 BIT FRACTION AND 30 BIT EXPONENT. USES 372 CELLS.

0704 370C0026 AVAILABLE PRIOR TO JANUARY 1962
TRIPLE PRECISION OUTPUT
CONVERTS 4 TRIPLE PRECISION FLOATING BINARY NUMBERS TO
120 LINE 100 FORM WITH 3 FLOATING DECIMAL NUMBERS PER LINE. PROGRAM MUST PROVIDE OWN
FRAMEWORK FOR EXTRACTING THEauxILIARY DATA. 91 CELLS X 30 COMMON.

0704 370C0025 AVAILABLE PRIOR TO JANUARY 1962
TRIPLE PRECISION OUTPUT
CONVERTS 4 TRIPLE PRECISION FLOATING BINARY NUMBERS TO
120 LINE 100 FORM WITH 3 FLOATING DECIMAL NUMBERS PER LINE. PROGRAM MUST PROVIDE OWN
FRAMEWORK FOR EXTRACTING THEauxILIARY DATA. 91 CELLS X 30 COMMON.

0704 370C0050 AVAILABLE PRIOR TO JANUARY 1962
TWO CUM binary AND OCTAL LOADER
LOADS ABSOLUTE BINARY AND OCTAL CARDS IN ANY ORDER, EXECUTES
TRANSFER COMMANDS. THE PUNCH TO CONSIDER BINARY CHECK SUM IS
25 RECOGNIZED. UP TO FOUR OCTAL CUMS, WITH THEIR LOCATIONS, PER CARD.

0704 370C0053 AVAILABLE PRIOR TO JANUARY 1962
WIRELINE FIXED FORMAT CARD READ
READS CARDS WITH FORMAT AND LOCATIONS FIXED BY THE CALLING
SEQUENCE, AS FULL CARD RENDER SPEED. FIXED OUTPUT, FLOATING
DECIMAL, AND MILLISECOND WILL BE CONSIDERED CORR. / 437

0704 370C0050 AVAILABLE PRIOR TO JANUARY 1962
TAPE OPERATOR PROGRAM TOP
TOP IS A SELF-CONTAINED PROGRAM THAT AUTOMATICALLY SEQUENCES
A SET OF COMPLETELY INDEPENDENT CALCULATIONS. THE PROGRAMS
NECESSARY FOR THESE CALCULATIONS ARE SELF-CONTAINED AND SELF-
LOADED FROM PROGRAM FILE TAPeS, EACH TOP WHICH CONTAINS MANY
PROGRAMS, FROM BINARY CARDS. THE INTRINSIC BINARY-TAPe
INSTRUCTION SET FOR THE CALCULATIONS AND THE COMBINED BINARY-
PROGRAMS IF THE OPERATOR ENTERS ON THE INPUT-TAPE TOP LEVEL,
THE PROGRAM IS OPERATIONAL. TOP IS A SELF-CONTAINED PROGRAM PRODUCED
THAT PROGRAMS INTO A SELF-LOADING SEQUENCE FOR THE PROGRAM
AND INITIATES THE SELF-LOADING SEQUENCE FOR THE PROGRAM

189
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 405PFMICL  AVAILABLE PRIOR TO JANUARY 1962
HEAD-WRITE TAPE CONTROL PROGRAM
FOUR ROUTINES FORM A PACKAGE WHICH, WHEN USED WITH A UNIT PRIMARY READ-WRITE TAPE PROGRAM, OSL. NO. 1207, ENABLES THE USER TO READ AND WRITE ON ANY OF THE TEN TAPE UNITS ATTACHED TO THE 765 WITH CONTROL AND WITH A MINIMUM OF TAPE MOVEMENT.

0704 405PFTGSC  AVAILABLE PRIOR TO JANUARY 1962
SORT GENERATOR
PRODUCES A SORT PROGRAM WHICH WILL SEQUENCE DATA AND ARRANGE INPUT IN ASCENDING ORDER.

0704 405PFTGLO  AVAILABLE PRIOR TO JANUARY 1962
SELF-LOADING PROGRAMS.CARDS ABSOLUTE BINARY CARDS. OCCUPIES 16 FIRST STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
CORRELATIONAL KISSUE COMPUTATION
RESIDUAL DEVIATION BETWEEN OBSERVED VALUES AND POINTS OF THE REGRESSION LINE. INPUT BY CARDS/OUTPUT ON DCA TAPE.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
ALPHANUMERICAL READING AND BCD CONVERSION
SAME TASK AS PFOCDZ BUT ALSO SUBSTITUTES A VALID CODE TO DOUBLE PUNCHES. OCCUPIES 15 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
MATRIX INVERSION.
FLOATING POINT MATRIX INVERSION AND SOLUTION OF MATRICEAL EQUATIONS INPUT 50 CARDS OR 50 BCD TAPES ON OFF-LINE PRINTING.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
EIGENVALUE COMPUTATION
DETERMINATION OF THE LARGEST EIGENVALUES OF AN SQUARE MATRIX AND OF THE CORRESPONDING EIGENVECTORS. ITERATIVE METHOD. OCCUPIES 793 CELLS/SVARIABLE BLOCK.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
DIARY PUNCH PROGRAM
PUNCHING INTO ABSOLUTE BINARY CARDS THE CONTENTS OF SEVERAL STORAGE BLOCKS.SELF-LOADING. OCCUPIES 350 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
CHECKSUM CORRECTOR
SELF-LOADING ONE-CARD PUNCHING PROGRAM.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
ZEROS OF A COMPLEX POLYNOMIAL
SINGLE PRECISION FLOATING POINT COMPUTATION OF A POLYNOMIAL WITH COMPLEX COEFFICIENTS. OCCUPIES 676 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
ZEROS OF A REAL POLYNOMIAL
SINGLE PRECISION FLOATING POINT COMPUTATION OF A POLYNOMIAL WITH REAL COEFFICIENTS OCCUPIES 760 STORAGE CELLS.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 405PFTGSC  AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION MATRIX INVERSION
FLOATING POINT INVERSION AND SOLUTION OF LINEAR SYSTEMS. INPUT/OUTPUT BY TAPE. THE ODE OF THE MATEX IS ALLOTTED THE ROUTINE WORKS ALSO IN SINGLE PRECISION. OCCUPIES 311 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
EIGENVALUE COMPUTATION
DETERMINATION OF THE LARGEST EIGENVALUES OF AN SQUARE MATRIX AND OF THE CORRESPONDING EIGENVECTORS. ITERATIVE METHOD. OCCUPIES 793 CELLS/SVARIABLE BLOCK.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
DIARY PUNCH PROGRAM
PUNCHING INTO ABSOLUTE BINARY CARDS THE CONTENTS OF SEVERAL STORAGE BLOCKS.SELF-LOADING. OCCUPIES 350 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
CHECKSUM CORRECTOR
SELF-LOADING ONE-CARD PUNCHING PROGRAM.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
ZEROS OF A COMPLEX POLYNOMIAL
SINGLE PRECISION FLOATING POINT COMPUTATION OF A POLYNOMIAL WITH COMPLEX COEFFICIENTS. OCCUPIES 676 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
ZEROS OF A REAL POLYNOMIAL
SINGLE PRECISION FLOATING POINT COMPUTATION OF A POLYNOMIAL WITH REAL COEFFICIENTS OCCUPIES 760 STORAGE CELLS.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
A MORE ACCURATE RUNGUE-KUTTA
A DIFFERENTIAL EQUATIONS ROUTINE UTILIZING THE METHOD OF RUNGUE-KUTTA-CELL TO SOLVE A SET OF N SIMULTANEOUS FIRST ORDER DIFFERENTIAL EQUATIONS. USES DOUBLE-PRECISION FLOATING POINT ARITHMETIC THROUGHOUT.LARGELY ELIMINATING THE EFFECT OF ROUND-OFF ERROR. REQUIRES THE USE OF SHARE ROUTINE GL DPCA. HAS AN OPTION FOR THE USER TO COMPUTE THE DERIVATIVES IN DOUBLE-PRECISION. PROGRAM OCCUPIES TOTAL OF 499 6 STORAGE/INCLUDING 313 FOR GL DPCA/ CONJ. 419

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
BESSEL FUNCTIONS
BESSEL FUNCTIONS COMPUTES ALL ORDERS OF THE MODIFIED NEUMANN FUNCTIONS OF LARGE ARGUMENTS
THIS ROUTINE WILL COMPUTE THE NEUMANN FUNCTION YIN/L FOR ALL INTEGER ORDERS FROM O TO N, IN LARGER THAN 1/4, FOR LARGE REAL VALUES OF r, OR WILL COMPUTE ONLY YIN.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
MULTIPLE CORRELATIONS AND REGRESSIONS ANALYSIS
ANALYSIS OF LINEAR REGRESSIONS AND CORRELATIONS OF M OBSERVATIONS AND P INDEPENDENT VARIABLES.
SINGLE OR DOUBLE PRECISION. ESTIMATION OF STANDARD DEVIATION AND MEAN VALUE INPUT BY CARDS OR BY BCD TAPE. OUTPUT BY ON-LINE OR OFF-LINE PRINTING. 4 TAPES MIN. REQUIRED. SELF-LOADING PROGRAM. CONJ.479.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
HYPERBOLIC SINE AND COSINE.
FLOATING POINT.
OCCUPIES 77 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
ALPHANUMERICAL READING AND BCD CONVERSION
ALPHANUMERICAL READING AND BCD CONVERSION
MAINLY IDENTICAL SIGNS TO 2 PORTIONS OF A FLOATING POINT DOUBLE PRECISION NUMER. OCCUPIES 47 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT COMPLEX ARITHMETICS.
EXECUTION OF MACHINE OPERATIONS ON COMPLEX NUMBERS BY A PROGRAM WRITTEN IN ORDINARY LANGUAGE. OCCUPIES 320 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT DOUBLE PRECISION ARITHMETICS.
EXECUTION OF MACHINE OPERATIONS ON DOUBLE PRECISION NUMBERS BY A PROGRAM WRITTEN IN ORDINARY LANGUAGE. OCCUPIES 960 STORAGE CELLS.

0704 405PFPCDZ  AVAILABLE PRIOR TO JANUARY 1962
GENERAL POLYNOMIAL PROGRAM
COMPUTATION OF ZEROS OF A POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS. SELF-LOADING. METHOD OF NEWTON.
P704 420CSDS01 AVAILABLE PRIOR TO JANUARY 1962
DUMP STORAGE, CORE, DRUM, AND TAPE
THIS SUBROUTINE IS A MODIFICATION OF A CS-DS SUBROUTINE WHICH DUMP CORES, DRUMS, AND TAPE. TAPE IS NOT REQUIRED USING THE LOGICAL DUMP FOR SAVING THE FIRST 320 WORDS OF CORE MEMORY. A MAGNETIC TAPE IS NOT USED FOR SAVING ENTRIES. THE SAME SENSE OPTION AS WAS USED TO SELECT THE TAPE. WITH 1550 IT IS POSSIBLE TO DUMP ALL OF CORE AND ALL OF DRUM MEMORY WITH ONE PASS ON THE MACHINE. SELF-LABELED GENIC TAPE, REQUIRES MINIMUM TO 731 CARD READERS, 727 TAPE AND 726 PRINTER ON AN ADDITIONAL 727 TAPE. SUPERSEDED BY CS-052 DIST. 494.

P704 421AAWA AVAILABLE PRIOR TO JANUARY 1962
ANALYSIS OF VARIANCE
COMPUTES MEANS, DEGREES OF FREEDOM AND F
AND TAPES, REQUIRING THE USE OF FACTORS FOR UP TO 13 WAY ANALYSIS. ANY NUMBER OF VARIABLES PER WHY AND ANY AMOUNT OF DATA MAY BE USED.

0704 422Hopout AVAILABLE PRIOR TO JANUARY 1962
POPG---A GENERAL PURPOSE PRINT AND PUNCH SUBROUTINE
THIS SUBROUTINE IS A MODIFICATION OF GLUIT-2 CAPABLE OF PRINTING RESPONSES OF UNRESTRICTED RELATIONSHIPS DIFFERENT OPTIONS WITH DIFFERENT SEQUENCE OPTIONS. THE PUNCHING IS NOT DONE. TAPE WRITING IS NOT CHECKED OUT READING.
3. LOCATION OF CALL SEQUENCE ERRORS ARE NOT PRINTED.
4. THE 00-O-0-P-0-T-0-E TEST IS MADE.
The SUBROUTINE USES 147 INSTRUCTION CELLS & 51 ERASABLE CELLS.

P704 423S2AV AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION RACIAN TANGENT SUBROUTINE
COMPUTES DOUBLE PRECISION ARC TANGENT OF A DOUBLE PRECISION ARGUMENT AS DESCRIBED IN 65 INTP.
REQUIRES 65 INTP AND 29 COMMON STORAGE.
85 INTP REQUIRES 75 STORAGE LOCATIONS.

P704 427BChi AVAILABLE PRIOR TO JANUARY 1962
COORDINATION OF UNRESTRICTED INTEGRATIONS
CONVERTS A BIG INTEGER OF 6 OR 12 CHARACTERS TO A BIG INTEGER OF 12 CHARACTERS WHICH CAN BE READ IN FIRST BIT POSITION OR OVERPUNCH OVER LEFTMOST POSITION.
RANGE IS +34,379,738,367 TO 34,379,738,367. USES 63 STORAGE CELLS PLUS 4 COMMON.

P704 429Min AVAILABLE PRIOR TO JANUARY 1962
BINARY TO BCD CONVERSION OF UNRESTRICTED INTEGERS.
CONVERTS A BINARY INTEGER OF 6 OR 12 CHARACTERS TO A BCD INTEGER. APPEAR AS LEFT MOST CHARACTER.
A 3-CHARACTER BCD INTEGER CAN BE PACKED TO A 3 CHAR.
LOGICAL WITH SIGN BIT TREATED AS MAJOR SORTING BIT IN WORD.

P704 430Dpssi AVAILABLE PRIOR TO JANUARY 1962
INTEGRATION BY GAUSSIAN QUADRATURE
INTEGRATES FROM MINUS INFINITY TO PLUS INFINITY BY

P704 431SSE1 AVAILABLE PRIOR TO JANUARY 1962
INTEGRATION BY HERMIT QUADRATURE
INTEGRATES FROM MINUS INFINITY TO PLUS INFINITY BY

192
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 435M14MT 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 IT IN THE SAME LOCATION AS THE ORIGINAL MATRIX. PROGRAM REQUIRES 32 WORDS PLUS 2 TEMPORARY, AN NO BY NO MATRIX IS TRANSPOSED IN LESS THAN 800 MICROSECONDS, CORR. 472

0704 435M24NS AVAILABLE PRIOR TO JANUARY 1962
M-cell upper relocatable binary loader /one card/. LOADS STANDARD RELOCATABLE BINARY CARDS WITHOUT ALTERATION OF LOADING ADDRESS. EXECUTES TRANSFER CARDS. OCCUPIES LAST 22 WORDS OF MEMORY. SELF LOADING.

0704 435M24TU AVAILABLE PRIOR TO JANUARY 1962
M-cell reflected 706 causes the 706 to behave like a 707 in its role as a reader and printer of cards. 50 WORDS PROGRAM PLUS 25 WORDS FOR LOWER BINARY LOADERS. READER AND PRINTER OPERATE AT FULL SPEED. SUPERSEDS MU 704X DIST. 1962.

0704 435M34SDO AVAILABLE PRIOR TO JANUARY 1962
SCOPE GRID PLOTTER TO DISPLAY ON THE 704 OUTPUT RECORDER A GRID OF HORIZONTAL AND VERTICAL LINES. PROVISION IS MADE FOR PLOTTING CERTAIN SPECIFIED LINES HEAVIER THAN OTHERS. PROGRAM REQUIRES 53 WORDS STORAGE PLUS 2 TEMPORARY.

0704 435M47E4 Available PRIOR TO JANUARY 1962
INTERPOLATION, ONE OR TWO VARIABLES GIVEN F(X,Y), Y=F(X), TO FIND A VALUE FOR X AND Y WITHIN A GIVEN EPSILON OF RELATIVE ERROR. REQUIRES 289 WORDS PLUS 16 ERASABLE STORAGE, CORR. 472.

0704 435M54ACQ AVAILABLE PRIOR TO JANUARY 1962
DETERMINANT EXPANSION OF SQUARE MATRICES. PROGRAMMED IN FORTRAN II. PROGRAM REQUIRES 56 WORDS OF STORAGE, 3 COMMON THRU COMMON 6 ON 6 9 WHERE N-ORDER OF THE MATRIX CORR. 102.

0704 435M64M Available PRIOR TO JANUARY 1962
MATRIX MULTIPLICATION. MULTIPLIES TWO MATRICES OF THE FORM A X B = C IN FLOATING POINT ARITHMETIC. REQUIRES 77 WORDS OF STORAGE.

0704 435M74PM AVAILABLE PRIOR TO JANUARY 1962
POLYNOMIAL EXPANSION COMPUTES THE POLYNOMIAL RESULTING FROM THE MULTIPLICATION OF LINEAR AND QUADRATIC FACTORS. REQUIRES 139 WORDS OF STORAGE PLUS 52 WORDS OF COMMON STORAGE.

0704 435N44AH AVAILABLE PRIOR TO JANUARY 1962
ATMOSPHERIC DATA SUBROUTINE. GIVEN A GEOMETRIC ALTITUDE H IN THE RANGE 0 TO 299,000 FEET, COMPUTE THE FOLLOWING IDENTITIES - 1. TEMPERATURE DEGREES KELVIN, 2. DENSITY RATIO, 3. PRESSURE RATIO, 4. VELOCITY OF SOUND /FT/SEC., ROUTINE REQUIRES 193 CELLS PLUS COMMON STORAGE AS NEEDED FOR 5-AT SUBROUTINE.

0704 435N54A290 AVAILABLE PRIOR TO JANUARY 1962
GENERAL CAYLEY HAMILTON DOUBLE SUBROUTINE. THIS SUBROUTINE WILL DRAW A SUB-DIVIDED GRID, WRITE A TITLE, A TOP OF GRID, WRITE A LABEL AND APPROPRIATE SCALE LANDMARKS, PL OF POINTS OR SYMBOLS FOR POINTS ON THE 704 CRT OUTPUT RECORD.

0704 441 CSTY AVAILABLE PRIOR TO JANUARY 1962
IBM 441 PSEUDO COMPUTER SIMULATOR. THIS COMPUTER IS DESCRIBED IN THE BOOK DIGITAL COMPUTER PROGRAMMING BY G. F. MC CACKEN.
ADDS ROW-TO-COLUMN BINARY CARD PUNCH

BINARY CARD-TO-CARD SIMULATOR

BINARY CARD-TO-CARD SIMULATOR

LOGICAL MEMORY SORT

REAL SYMMETRIC MATRIX TO TRIANGULAR FORM.

TRANSFORMING THE MATRIX TO A DIAGONAL BASIS,

STEPWISE MULTIPLE REGRESSION PROCEDURE

STEPWISE MULTIPLE REGRESSION PROCEDURE

STEPWISE MULTIPLE REGRESSION PROCEDURE

STEPWISE MULTIPLE REGRESSION PROCEDURE
LEAST MAXIMUM ABSOLUTE ERROR POLYNOMIAL FIT ON LINE OCTAL DUMP

INTERN 0704 PROGRAM LIBRARY ABSTRACT

MIZES THE MAXIMUM ABSOLUTE ERROR AT A GIVEN SET OF K DATA POINTS. THIS ROUTINE CONFORMS WITH FLOATING POINT ARITHMETIC, REQUIRES K CELLS, FLOATING POINT ARITHMETIC, USES 100 CELLS PLUS 31 COMMON CELLS FOR DATA.

IBM 0704 PROGRAM LIBRARY ABSTRACT

196 0704 PROGRAM LIBRARY ABSTRACT

0704 51200701 AVAILABLE PRIOR TO JANUARY 1962

BCD TO MODIFIED BCD CONVERSION ROUTINE TO CONVERT A SERIES OF BCD WORDS TO MODIFIED BCD.

0704 51200702 AVAILABLE PRIOR TO JANUARY 1962

DATA PROCESSING OUTPUT ROUTINE TO SET UP AND PRINT ONE LINE OF OUTPUT ON AN ON-LINE PRINTER IF SW = 0 ON OR OFF-LINE ON TAPE 1 IF SW = 2 OFF. THIS ROUTINE CONVERTS BOTH FLOATING AND FIXED POINT BINARY NUMBERS TO FIXED POINT OUTPUT AND PRINTS HOLLERITH AND MODIFIED HOLLERITH INFORMATION.

0704 51200802 AVAILABLE PRIOR TO JANUARY 1962

GENERAL PRINTED OUTPUT ROUTINE TO SET UP THE IMAGE OF ONE CARD ON TAPE 3 TO BE PRINTED ON OFF-LINE PUNCH OR TO BE SET UP TO CARD IMAGE IN CORE. THIS ROUTINE CONVERTS BOTH FLOATING AND FIXED POINT BINARY NUMBERS TO FIXED POINT OUTPUT AND PRINTS HOLLERITH AND MODIFIED HOLLERITH INFORMATION.

0704 51200914 AVAILABLE PRIOR TO JANUARY 1962

INTERPRETER FOR 650 PROGRAMS INTERPRETS 650 PROGRAMS WRITTEN ACCORDING TO IBM TECHNICAL NEWSLETTER NO. 11. ACCEPTS EXISTING PROGRAM DUCES WITH MINOR MODIFICATION. PRODUCES THE SAME OUTPUT CARD AFTER TAPE PROGRAMS. PROVIDES UP TO 10 TO 1 SPEED INCREASE OVER 650.

0704 51200941 AVAILABLE PRIOR TO JANUARY 1962

MAKE SAP OCTAL WHEN LOADED USING THE SAP 3-P Fowler PE PSEUDO-OPERATION, PRINTS THE SEGMENT-BINARY INTEGER CONVERSION ROUTINE ON THE BINARY TAPE. THIS ROUTINE CONVERTS BOTH FLOATING AND FIXED POINT BINARY NUMBERS TO FIXED POINT OUTPUT AND PRINTS HOLLERITH AND MODIFIED HOLLERITH INFORMATION.

0704 51400029 AVAILABLE PRIOR TO JANUARY 1962

DETERMINANT EVALUATION AND ROOT EXTRACTION P. C. BAUER, W. D. MARQUARDT, W. L. MEARS. THIS ROUTINE EVALUATES A DETERMINANT WITH POLYNOMIAL ELEMENTS AND EXTRACTS THE ROOTS OF 755 RESULTING POLYNOMIALS. THE ORDER OF THE DETERMINANT MAY VARY FROM 2 TO 25, AND THE DEGREE OF THE ELEMENTS MAY VARY FROM 2 TO 120. THE ROOT EXTRACTION PART HANDLES UP TO 60TH DEGREE POLYNOMIALS IN ADDITION TO THE ROUTINE MAY BE USED TO EVALUATE THE DETERMINANT ONLY FOR EXTRACT THE ROOTS OF A POLYNOMIAL ONLY.

0704 51400582 AVAILABLE PRIOR TO JANUARY 1962

INCOMPLETE GAMMA FUNCTION, GIVEN X AND X, THIS SUBROUTINE WILL COMPUTE THE INCOMPLETE GAMMA FUNCTION DEFINED AS THE INTEGRAL FROM X TO INFINITY OF EXPM-1 U TO THE X POWER.

0704 52100603 AVAILABLE PRIOR TO JANUARY 1962

FACTOR ANALYSIS EUGENE K. DURRANT. ANALYSIS OF A CORRELATION MATRIX, EXTRACTION OF SUCCESSIVE FACTORS AND COMPUTATION OF CUMULATIVE CONTRIBUTIONS. THE ORDER OF MATRIX IS 40, HIGHEST NUMBER OF MEMBERS 15. OFF-LINE OUTPUT. JORDAN'S METHOD.

0704 52100651 AVAILABLE PRIOR TO JANUARY 1962

MUSH DATA ASSEMBLER AND PRINT ROUTINE PROVIDES INPUT AND OUTPUT FOR SC-PUSH. USES A SLIGHTLY MODIFIED RAND LP INPUT TAPE FOR DECAY OUTPUT FORMAT SIMILAR TO THAT OF RAND.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 5258CMEN Available Prior to January 1962

LINEAR PROGRAMMING SUBROUTINE
SOLVES PROBLEM WITH UP TO 55 EQUATIONS BY MODIFIED SIMPLEX
METHOD. MAXIMUM NUMBER OF VARIABLES DEPENDS ON SIZE OF CORE
FOR WHICH ASSEMBLED. SINGLE PRECISION ARITHMETIC USED THROUGH
OUT. ROUND-OFF ERROR IN INVERSE CAN BE REDUCED BY PERIODIC
USE OF A MODIFIED SIMPLEX (M.S.) METHOD. VARIOUS RESTARTS PROVIDED.

0704 525PM8CD Available Prior to January 1962

BINARY TO 8 DIGIT SUBROUTINE
CONVERTS A POSITIVE BINARY INTEGER TO 8 10-DECODED DIGITS
AND ADDS LEADING ZEROS TO ODDS. 37 CELLS AND 3
COMMON.

0704 525PM6CH Available Prior to January 1962

FLOATING-POINT DOUBLE-PRECISION CUBE ROUTINE
REQUIRES 3 DATA CARDS. 35 CELLS. TIMING 0.246 MS.

0704 525PM6CL Available Prior to January 1962

PK CLEAR AND ADD
LEADS INTO MEMORY INFORMATION FROM ABSOLUTE AND
RELOCATABLE BINARY DATA CARDS, CORRECTION-TRANSFER CARDS,
AND CHECK SUM CARDS. ONLY THE DATA CARDS WILL BE CHECKED
PROPERLY SEARCHED WITH NON-LOCATION, ONLY THE LOCATION
OF WHICH 20 ARE PERFORMED IN THEIR DOUBLE-PRECISION FLOATING-POINT
NUMBER, NORMAL TSX SEQUENCE. 52 BIT ACCURACY. REQUIRES 56
STORAGE CELLS PLUS 7 COMMON. TIMING 0.054 MS.

0704 525PM6CM Available Prior to January 1962

RELATIONAL BINARY LOADS AND CHECKS BINARY DATA CARDS AND ABSOLUTE BINARY
CORRECTION-TRANSFER CARDS USING SHARE FORMAT. UPDATING OF
LOCATIONS DONE ON CFT CARDS. PUSHES LOADS-CARDS FOR CARD WITH
A 9 NEW COLUMN PUNCH. OCCUPIES 0.151. AND 77672-77777 OCTAL.

0704 525PM6CN Available Prior to January 1962

HOLLETHER CARD TO TAPE
A SELF-LOADING PROGRAM TO WRITE INFORMATION FROM A HOLLERTHER
CARD ON A 2000-TAB UNIT SPECIFIED ON THE CARD, TERMINATES BY
INSERTING LOAD CARDS SEQUENCE ON A TRAIL CARDS, WHICH IS
USER-ENTERED FROM A PROGRAM TO READ SUCCESSING CARDS. 130
CELLS.

0704 525PM6CU Available Prior to January 1962

DUAL-PRECISION FLOATING-POINT ARITHMETIC PACKAGE
PERFORMS DOUBLE-PRECISION FLOATING-POINT ARITHMETIC OPERA-
TIONS. RESULTS ARE STORED IN PROPER DOUBLE-PRECISION FLOATING-
POINT SUBROUTINE PK ADP MAY BE USED ALONG AS WELL AS WITH
PK INT. REQUIRES 276 STORAGE CELLS.

0704 525PM6D Available Prior to January 1962

FLOATING POINT INTEGER FORMULATION
N IS AN INTEGER LESS THAN -47. METHOD IS ITERATED SINGLE
PRECISION FLOATING MULTIPLICATION APPROXIMATELY 44 N MS.
31 CELLS, 56,296.

0704 525PM6D9 Available Prior to January 1962

DOUBLE-PRECISION FLOATING-POINT INTEGRAL PACKAGE
READS AND EXECUTES CONSECUTIVE MACHINE LANGUAGE INSTRUCTIONS
OF WHICH 20 ARE PERFORMED IN THEIR DOUBLE-PRECISION FLOATING
POINT ANALOG. PACKAGE IS COMPOSED OF PK INT, PK INTD, AND
PRODU. REQUIRES 549 STORAGE CELLS.

0704 525PM6E Available Prior to January 1962

INTERPRETIVE DOUBLE-PRECISION FLOATING-POINT ARITHMETIC
SUBROUTINE.
READS AND EXECUTES CONSECUTIVE MACHINE LANGUAGE INSTRUCTIONS
OF WHICH 20 ARE PERFORMED IN THEIR DOUBLE-PRECISION FLOATING
POINT ANALOG. PRINCIPAL PART OF INTERPRETIVE PACKAGE
PK INOP. PK DOUF MUST BE INCLUDED IN THE ASSEMBLY. REQUIRES
249 STORAGE CELLS PLUS THOSE REQUIRED BY PK DOUP.

0704 525PM6E1 Available Prior to January 1962

INTERPRETIVE DOUBLE-PRECISION FLOATING-POINT ARITHMETIC
SUBROUTINE.
WITH ERROR RETURN AND /2,4/ NORMAL RETURN EXCEPT FOR ERRORS DUE TO ROUND-OFF
AND F/X/ EVALUATION, RESULT IS EXACT IF F/X/ IS EXPRESABLE AS A POLYNOMIAL OF
Degree 25 OR LESS. 37 CELLS AND F/X/ SUBROUTINE.

0704 525PM6E2 Available Prior to January 1962

INTERPRETIVE DOUBLE-PRECISION FLOATING-POINT ARITHMETIC
SUBROUTINE.
EVALUATES NTH ROOT OF A POSITIVE FLOATING POINT NUMBER
WHERE N IS A POSITIVE OR NEGATIVE INTEGER. ACCURATE TO
THE GAMMA FUNCTION.
REQUIRES NATURAL LOGARITHM SUBROUTINE /LN/ WITH SHARE
FLOATING POINT SUBROUTINE FOR NATURAL LOGARITHM FOR-
WARD ERRORS DUE TO ROUND-OFF AND F/X/ EVALUATION, RESULT IS
EXACT IF F/X/ IS EXPRESABLE AS A POLYNOMIAL OF
Degree 29 OR LESS. 67 CELLS AND F/X/ SUBROUTINE.

0704 525PM6ED Available Prior to January 1962

INTERPRETIVE DOUBLE-PRECISION FLOATING-POINT ARITHMETIC
SUBROUTINE.
EVALUATES NTH ROOT OF A POSITIVE FLOATING POINT NUMBER
WHERE N IS A POSITIVE OR NEGATIVE INTEGER. ACCURATE TO
RETURN EXCEPT FOR ROUND-OFF AND ERRORS DUE TO FIX/
STANDARD INPUT-OUTPUT, 1114/ ERROR RETURN ANO /2,4/ NORMAL
RETURN EXCEPT FOR ERRORS DUE TO ROUND-OFF AND ERRORS DUE TO
THE GAMMA FUNCTION.
REQUIRES NATURAL LOGARITHM SUBROUTINE /LN/ WITH SHARE
FLOATING POINT SUBROUTINE FOR NATURAL LOGARITHM FOR-
WARD ERRORS DUE TO ROUND-OFF AND F/X/ EVALUATION, RESULT IS
EXACT IF F/X/ IS EXPRESABLE AS A POLYNOMIAL OF
Degree 29 OR LESS. 67 CELLS AND F/X/ SUBROUTINE.

0704 525PM6F Available Prior to January 1962

FLOATING POINT FLOATING-POINT INTEGER FORMULATION
15-POINT LEGENDRE-GAUSS QUADRATURE INTEGRATION SUBROUTINE
A SHARE TYPE SUBROUTINE FOR EVALUATION OF F/X/ FOR 15 VALUES
OF X IN THE INTERVAL OF INTEGRATION MUST BE PROVIDED. EXCEPT
FOR ERRORS DUE TO ROUND-OFF AND F/X/ EVALUATION, RESULT IS
EXACT IF F/X/ IS EXPRESABLE AS A POLYNOMIAL OF
Degree 25 OR LESS. 79 CELLS AND F/X/ SUBROUTINE.

0704 525PM6F1 Available Prior to January 1962

FLOATING POINT ORDINARY DIFFERENTIAL EQUATIONS SUBROUTINE
Solves A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6F2 Available Prior to January 1962

FLOATING POINT ORDINARY DIFFERENTIAL EQUATIONS SUBROUTINE
Solves A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6G Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6H Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6I Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6J Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6K Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6L Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6M Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.

0704 525PM6N Available Prior to January 1962

FLOATING POINT ORTHOGONAL DIFFERENTIAL EQUATIONS SYSTEM
Solves A SYSTEM OF ORTHOGONAL DIFFERENTIAL EQUATIONS OF ANY
NUMBER, ANY ORDER, LINEAR OR NON-LINEAR. THE SYSTEM IS
RESTRICTED TO ONE INDEPENDENT VARIABLE AND THE BOUNDARY CONDITIONS
ARE GIVEN IN TERMS OF INITIAL CONDITIONS. NUMERICAL
INTEGRATION BY ADAMS FORMULA. 517 CELLS.
GOTO AVAILABLE PRIOR TO JANUARY 1962
PERFORM ANOTHER ROUTINE REQUIRES 15
TRANSFER FUNCTION
OR OTHER, PROGRAM EVEN WHEN THE FORTRAN OBJECT PROGRAM USES
TO COMPUTE ONE POINT OF EITHER THE AUTO- OR CROSS-CORRELATION
TO ACCOMPLISH A TRANSFER FROM A FORTRAN PROGRAM TO A SHARE,
POINTS. 29 LDC. &
FUNCTION,. GIVEN A SET OF TIME-SERIES DATA FOR EQUALLY-SPACED
LCCATIONS, NO COMMON.
EXTENDED TRANSFER FUNCTION
FORTRAN PROGRAM WHENEVER AN END OF FILE IS ENCOUNTERED WHILE
READING A BINARY TAPE. REQUIRES 192 CELLS, NO COMMON.
END OF FILE FUNCTION
REQUIRED 190 CELLS & 295 COMMON.
SWITCH, FOR SINGLE RECORDS TWO CROSSCORRELATIONS, IN-PHASE CO-SPECTRUM, OUT-OF-
WAVE RECORD ANALYSIS OF TWO SIMULTANEOUS RECORDS OF A-
VARIABLE FORMAT FORM DIRECTLY FROM CORE STORAGE WITH AUTO-
SORT, ALGEBRAIC. KEY AND ITEM LENGTH - 1 WORD. OPEN.
ITEMS A POWER OF 2. 1 WORD CLUES /WHICH GIVE LOC. OF KEYS/
AND/OR PRINTS CARO IMAGES. PRINTS DECIMAL DATA IN
ABSOLUTE! AND CORRECTION CARO LOADER
FROM A PERSONNEL MASTER FILE THE DATA NECESSARY TO RUN A
RECORDS AS DESIRED. WILL BYPASS THOSE WORDS ON THE INPUT.
READ TAP:E DATA.
RANDOM SEQUENCE OF PSEUDO-NORMALLY DISTRIBUTED NUMBERS WITH
EACH ENTRANCE PRODUCES THE NEXT NUMBER /IN FLOATING PT/ IN A
LOC.&4 ERASABLE. INCLUDES SN2F /SINE-COSINE/ SUBROUTINE.
NORMALLY DISTRIBUTED PSEUDO-RANDOM NUMBERS.
ORDINARY DISTRIBUTION OF TWO SIMULTANEOUS RECORDS OF A-
VARIABLE FORMAT FORM DIRECTLY FROM CORE STORAGE WITH AUTO-
SORT, ALGEBRAIC. KEY AND ITEM LENGTH - 1 WORD. OPEN.
ITEMS A POWER OF 2. 1 WORD CLUES /WHICH GIVE LOC. OF KEYS/
AND/OR PRINTS CARO IMAGES. PRINTS DECIMAL DATA IN
ABSOLUTE! AND CORRECTION CARO LOADER
FROM A PERSONNEL MASTER FILE THE DATA NECESSARY TO RUN A
RECORDS AS DESIRED. WILL BYPASS THOSE WORDS ON THE INPUT.
READ TAP:E DATA.
0704 63SRWODN AVAILABLE PRIOR TO JANUARY 1962

Determinant Evaluation of Nearly Triangular Matrices
This Fortran Subprogram Evaluates the Determinant of a Matrix A-Alpha Times A 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/ Which Can Be Changed According to Needs of the Program. The Input Matrix A Is Not Destroyed By the Program. 216 Cells Excluding Arrays A and B Are Required.

0704 63SRWODX AVAILABLE PRIOR TO JANUARY 1962

Real Eigenvalues of Real Matrices
This Fortran Subprogram Determines All Real Eigenvalues of a Real Matrix A. It Has a Dimension Statement of A/50,50/ Which Changes According to Needs of the Program. The Common Region Input Matrix A Is Destroyed By the Computation. The Program Requires 3 Subsidiary Subroutines in Addition to the Programs Which Write Output on Tape. The Program Needs For EIG Already Includes the 3 Subsidiaries. Corr. 794.

0704 63SRWOKS AVAILABLE PRIOR TO JANUARY 1962

General Least Squares Fortran Subprogram

0704 63SRWOGN AVAILABLE PRIOR TO JANUARY 1962

General Root Finder Fortran Subroutine
This Fortran Subprogram Finds the Real Roots of Any Analytic Function FF(X). It Has a Dimension Statement of A/50,50/ and B/50/ Which Can Be Changed According to Needs of the Program. The Input Matrix A Requires 655 Cells Excluding the Array A. The Output Subroutines, the Square Root Routine and the Auxiliary Program.

0704 63SRWOKT AVAILABLE PRIOR TO JANUARY 1962

Linear Matrix Equation Solver

0704 63SRWOKT2 AVAILABLE PRIOR TO JANUARY 1962

Nearly Triangularization of a Matrix Subroutine
This For an Subprogram Transforms a Real Matrix A Into a Nearly Triangular Matrix A Plus A Simplicity Transformations. It Has a Dimension Statement of A/50,50/ and B/50/ Which Can Be Changed According to Needs of the Programmer. The Input Matrix A Is Destroyed During Computation. 359 Cells Required Excluding Arrays A and B.

0704 63SRWOKT3 AVAILABLE PRIOR TO JANUARY 1962

Eigenvalue Determinant Subroutine
Given an Real Eigenvalue Alpha Of A Matrix A Of Order N Times N. This Fortran Subprogram Determines the Corresponding Real Eigenvalues A. It Has a Dimension Statement of A/50,50/ and B/50/ According to Needs of the Programmer. The Input Matrix A Is Destroyed During Computation. 359 Cells Required Excluding Arrays A and B. Corr./816

0704 63SRWOKT2 AVAILABLE PRIOR TO JANUARY 1962

Singular Value Decomposition Subroutine
This Fortran Subprogram Decomposes A Matrix A Into the Product of Three Matrices A Equals U Times Sigma Times V Times T. The Matrix Sigma Is a Diagonal Matrix Of Order M Times N. It Has a Dimension Statement of A/50,50/ and B/50/ Which Can Be Changed According to Needs of the Programmer. The Input Matrix A Is Destroyed During Computation. 359 Cells Required Excluding Arrays A and B.

0704 63SRWOKT2 AVAILABLE PRIOR TO JANUARY 1962

Recompute Functions of Order Zero.
Computes a Zero and N Zero Of A From Asymptotic Formulas. Requires 232 Cells Plus 10 Common. 5% Square Root and Log Routines Included.
FLOATING POINT NUMBER. USES 22 CELLS. AT LEAST 6 USUALLY AVAILABLE PRIOR TO JANUARY 1962.

DOUBLE PRECISION FLOATING POINT NUMBER. USES 25 CELLS.

DOUBLE PRECISION FLOATING POINT MULTIPLICATION

DOUBLE PRECISION FLOATING POINT ADDITION

PARTIAL DOUBLE PRECISION FLOATING POINT ADDITION

FLOATING POINT

DOUBLE PRECISION FLOATING POINT ADDITION

DOUBLE PRECISION FLOATING POINT ADDITION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION

DOUBLE PRECISION FLOATING POINT DIVISION
LINEAR EQUATION SOLVER

Solves the vector \( \mathbf{v} \) in least squares sense. Requires 757 cells of program and constants (includes \( \mathbf{A} \), \( \mathbf{v} \), and \( \mathbf{d} \) plus 466 cells of common.

MULTIPLE REGRESSION BACK SOLUTION PROGRAM.

Calculates least squares solutions for the results of the multiple regression code scrap.
THIS 704 PROGRAM USES FORTRAN TO CALCULATE FUNCTION VARIABLES
ESTIMATES, MEANS, STANDARD DEVIATIONS, REGRESSION COEFFICIENTS
INPUT ERROR FOR MULTIPLE REGRESSION CODE SCRAP.
FLOATING-POINT OVERFLOW/UNDERFLOW ADJUSTMENT DURING THE
DIAGNOSTIC OUTPUT ON A BCD OUTPUT TAPE OR ERROR CONTROL, AND
FORTRAN ERROR PACKAGE
PROVIDES MULTIPLE CORRELATION COEFFICIENTS, STANDARD ERROR OF
HUL TLE REGRESS 1 ON & CORRELATION ANAL VS IS PROGRAM.
WITH l DRUM AND AT LEAST 4 TAPES, CORR/944
AND T-TABLE ENTRIES FOR UP TO '39 INDEPENDENT VARIABLES WITH
DESCRIPTION AND A SUBROUTINE NAME-STATEMENT NUMBER TRACE BACK
SUBROUTINE ON SAP SYMBOLIC CARDS. REQUIRES 192&19 COMMON
USES 325 CORE LOCATIONS •.
STORAGE CELLS PLUS LOG AND EXP SUBROUTINES. ALSO EXISTS AS
GENERATE A FORTRAN II PROGRAM TAPE OR ABSOLUTE BINARY
RECORD WITH A BOOTSTRAP PREFACE, OR PUNCH OUT THE PROGRAM
TECHNIQUES. REQUIRES 666 WORDS OF CORE. ALL TEMPORARY
IBM 0704 PROGRAM LIBRARY ABSTRACT
CONTENTS t COUNT-DOWN ON NUMBER Of TRANSFERS TO OR FROM SOME
GIVEN M l'ABLES VK-F/X/ WHERE X IS EQUALLY SPACED KTH ORDER
FORMULA tS P.ERFORMEO ON ALL TABLES.TABLES MUST All BE OF SAME
DIFFERENTIAL EQUATION . .
323 IF SENSE SWITCH. USES CORE STORAGE LOCATION, OR MANUALLY BY THE SETTING OF A SENSE
SWITCH ALLOWS FOR CHANGES IN VALUES WHICH WERE NOT ORIGINALLY
SUCH THAT THE FUNCT10N Y AND ITS FIRST AND SECOND DERIVATIVES
ARE CONTINUOUS IN THE RANGE OF X IN THE TABLE WRITTEN AS 2
SUM/S/ Of SQUARES OF RESIDUALS, ANO THE ERROR MATRIX ARE ALSO
TO COMPUTE ANO PRINT ALL SUMS OF SQUARES ASSOCIATED
WITH FACTORIAL EXPERIMENTATION. ALL SUMS OF OBSERVATIONS
OF SECOND-ORDER EQUATIONS .. SOLVES A SET OF N SIMULTANEOUS
FIRST DERIVATIVES MAY OR HAY NOT APPEAR.
SECOND-ORDER ORDINARY DIFFERENTIAL EQUATIONS, IN WHICH
GENERAL ANAL VS IS OF VAR l ANCE
GENERAL LEAST SQUARE CURVE FITTING ROUTINE
GIVEN AN N X M MATRIX X AN M DIMENSIONAL ROW VECTOR B
AND AN N X N DIAGONAL MATRIX S STORED AS A ROW/ THIS
ROUTINE FIND AN N X M DIMENSIONAL ROW VECTOR V. IF THE USER
SETS ALL 3 = 0 SOLVES V IN THE LEAST SQUARES SENSE.
GENERAL LEAST SQUARE CURVE FITTING ROUTINE
GIVEN AN N X M MATRIX AN M DIMENSIONAL ROW VECTOR B
AND AN N X N DIAGONAL MATRIX S STORED AS A RAW/ THIS
ROUTINE FIND AN N X M DIMENSIONAL ROW VECTOR V. IF THE USER
SETS ALL 3 = 0 SOLVES V IN THE LEAST SQUARES SENSE.
GENERAL LEAST SQUARE CURVE FITTING ROUTINE
GIVEN AN N X M MATRIX AN M DIMENSIONAL ROW VECTOR B
AND AN N X N DIAGONAL MATRIX S STORED AS A RAW/ THIS
ROUTINE FIND AN N X M DIMENSIONAL ROW VECTOR V. IF THE USER
SETS ALL 3 = 0 SOLVES V IN THE LEAST SQUARES SENSE.
CORRELATION AND REGRESSION ANALYSIS,

COLUMN BINARY DISASSEMBLY PROGRAM
ONTO A MASTER PROGRAM TAPE TO BE USED WITH THE DECK
COMPUTED FOR EACH RECORD
ONTO A MASTER PROGRAM TAPE TO BE USED WITH ALL BUT THE
OVER EACH LEVEL OF EACH FACTOR ARE ALSO PRINTED.

AT IN SQUARES ANALYSIS OF VARIANCE

SELF LOADING TAPE WRITING ROUTINE
TO LOAD THE INFORMATION FROM A FORTRAN OBJECT PROGRAM
INTO A MASTER PROGRAM TAPE TO BE USED WITH ALL BUT THE
DECK WHICH MAKES UP THE FINAL RECORD, A CHECK SUM IS
COMPUTED FOR EACH RECORD

SELF LOADING TAPE WRITING ROUTINE
TO LOAD THE INFORMATION FROM A FORTRAN OBJECT PROGRAM
INTO A MASTER PROGRAM TAPE TO BE USED WITH ALL BUT THE
DECK WHICH MAKES UP THE FINAL RECORD.

CORRELATION AND REGRESSION ANALYSIS,
CALCULATIONS ARE PERFORMED AS SPECIFIED BY A CONTROL CARD.
ADDITIONAL OUTPUT FORMATS, PROVISIONS ARE MADE FOR PROGRAM INTUITION
AND RESTART. ADDITIONAL COMPUTATION MAY BE INTRODUCED.
MAXIMUM NUMBER OF VARIABLES IS 110 /SINGLE PREC/ OR 80
DOUBLE PREC/ NUMBER OF OBSERVATIONS IS 2 ± 28 - 1.

COLUMN BINARY DISASSEMBLY PROGRAM
THIS PROGRAM WILL READ A COLUMN BINARY ABSOLUTE OR RELOCATABLE DECK AND TRANSLATE THE INFORMATION
BACK TO SYMBOLIC FORM. SEE GE KOSI

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 784IBIFS1 AVAILABLE PRIOR TO JANUARY 1962

EVALUATES A FOURIER SERIES, FOR GIVEN NUMERICAL VALUES OF ITS INDEPENDENT VARIABLES, THE SERIES TO BE EVALUATED MUST BE GIVEN IN EXPANDED REPRESENTATION AS DEFINED ON THE WRITE UP FOR DFPS1. TIMING USN 6 ± 10 Cycles WHERE K = THE NUMBER OF INDICES PER TERM, AND T THE NUMBER OF TERMS IN THE SERIES TO BE EVALUATED.


INTERPRETIVE ROUTINE, WHICH FACILITATES THE EXECUTION OF A SEQUENCE OF FOURIER SERIES OPERATIONS.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 784ILABF AVAILABLE PRIOR TO JANUARY 1962

DETERMINES THE STANDARD ERROR OF COEFFICIENTS, THE LARGE COEFFICIENT FIRST THE OUTPUT IS 12 BCD WORDS.

IN CANONICAL REPRESENTATION WITH RESPECT TO ANY VARIABLE, IT ALSO REQUIRES SIX TAPES AND FOUR LOGICAL DRUMS. CORRY 884

IN CANONICAL REPRESENTATION OBTAINING AS THE RESULT A SERIES IN CANONICAL REPRESENTATION.

MULTIPLIES TWO FOURIER SERIES.
IN CANONICAL REPRESENTATION OBTAINING AS THE RESULT A THIRD
SERIES IN CANONICAL REPRESENTATION, REQUIRES THE SUBROUTINE APF6.

COMPUTES THE PARTIAL DERIVATIVE OF A FOURIER SERIES, IN CANONICAL REPRESENTATION WITH RESPECT TO ANY VARIABLE, OBTAINING AS A RESULT A SERIES IN CANONICAL REPRESENTATION. TIMING 2.040 ± 0.567 MILLICYCLES MAXIMUM.

PUNCHES A FOURIER SERIES ORNTO BINARY RELOCATABLE CARDS, CANONICAL REPRESENTATION IS USED, BUT NO RESTRICTIONS ARE IMPROVED ON THE INDEX VECTORS. TIMING 100 CYCLES PER MINUTE MAXIMUM.

READS, WITH CHECKING, A FOURIER SERIES FROM BINARY TAPE INTO CORE STORAGE, IN CANONICAL REPRESENTATION.
ARCTANGENT, FLOATING POINT-QUADRANT ALLOCATION

PROGRAM CELLS, NO COMMON.

0704 8ZSJPDEQ AVAILABLE PRIOR TO JANUARY 1962

0104 B25JPATNC AVAILABLE PRIOR TO JANUARY 1962

PLUS 6 WORDS OF COMMON.

KUTT A. STORAGE REQUIREMENTS: ARE 452 WORDS FOR PROGRAM,

DIFFERENTIAL EQUATIONS SOLVER

SOLVES SIMULTANEOUS DIFFERENTIAL EQUATIONS WITH

SEVEN SIGNIFICANT DECIMAL DIGITS ACCURACY. PROGRAM REQUIRES

GENERAL INTEGRAL EVALUATOR

THE INDEPENDENT VARIABLES. METHOD USED IS A FOURTH ORDER RUNGE-

KUTT A. VECTOR VALUED INTEGRAL FROM

INTEGRAL EXPRESSION, WHETHER ITERATED INTEGRAL, MULTIPLE

RECURSIVE FORMULA APPROXIMANTS FOR ANY TYPE OF

INTEGRAL EXPRESSION, WHETHER ITERATED INTEGRAL, MULTIPLE

INTEGRAL OR VECTOR VALUED INTEGRAL FROM A VECTOR VALUED

FUNCTION, OR THE INTEGRAL OF A FUNCTION OF OTHER INTEGRALS.

REQUIRES 425 WORDS PLUS 6 WORDS OF COMMON.

0704 825PJMT AVAILABLE PRIOR TO JANUARY 1962

GENERAL INTEGRAL EVALUATOR

GENERATES THE SIMPSON RULE APPROXIMANTS FOR ANY TYPE OF

INTEGRAL EXPRESSION, WHETHER ITERATED INTEGRAL, MULTIPLE

INTEGRAL OR VECTOR VALUED INTEGRAL FROM A VECTOR VALUED

FUNCTION, OR THE INTEGRAL OF A FUNCTION OF OTHER INTEGRALS.

REQUIRES 2372 WORDS PLUS 5 WORDS OF COMMON.

0704 B30MINTO AVAILABLE PRIOR TO JANUARY 1962

PRINT 655 LOADER DIAGNOSTICS

PRINTS A 704 SAP-CODED FORTRAN II SUBPROGRAM TO

SUPPLY-ON-LINE DIAGNOSTIC COMMENTS ON THE

ACTIVATED ERROR STOPS OF MIPSS2 LOADER.

0704 B30MINTC AVAILABLE PRIOR TO JANUARY 1962

OCTAL CORRECTION CARD READER

_CONNECTIONS A 704 SAP-CODED FORTRAN II SUBPROGRAM TO

LOADEE RELOCATABLE IN ABSOLUTE OCTAL CORRECTION

CARDS AND COMMENT CARDS. CORRECTIONS AND

COMMENTS MAY BE LOGGED ON OUTPUT TAPE 2.

0704 830MINTL AVAILABLE PRIOR TO JANUARY 1962

FOREST OVERLAYER SUBPROGRAM

MELLP-A 704 SAP-CODED SUBPROGRAM THAT ACTS AS AN

OVERLAYER FOR RUNNING PROGRAMS THAT EXCEED CORE

MEMORY SIZE. CORR/ DIST. 866

0704 830MINTP AVAILABLE PRIOR TO JANUARY 1962

WHITE 655 LOADER STORAGE MAP

MELPP-A 704 SAP-CODED FORTRAN II SUBPROGRAM

THAT PRINTS ON TAPE A THE CORE MEMORY STORAGE MAP

FORMED BY THE MIPSS2 LOADER.

0704 830MINTM AVAILABLE PRIOR TO JANUARY 1962

WHITE 655 LOADER STORAGE MAP

MELPM-A 704 SAP-CODED FORTRAN II SUBPROGRAM

THAT PRINTS ON-TAPE THE CORE MEMORY STORAGE MAP

FORMED BY THE MIPSS2 LOADER.

0704 830MINTF AVAILABLE PRIOR TO JANUARY 1962

WHITE 655 LOADER STORAGE MAP

MELPF-A 704 SAP-CODED FORTRAN II SUBPROGRAM

THAT WRITES THE CONTENTS OF CORE MEMORY AS

A SINGLE SELF-LOADING RECORD ON TAPE A.

0704 832MINTC AVAILABLE PRIOR TO JANUARY 1962

COMPLEX NUMBER INTERPRETIVE SYSTEM FLOATING POINT

A TWO-ADDRESS COMPLEX NUMBER INTERPRETIVE SYSTEM DESIGNED TO

WORK WITHIN SAP PROGRAMS. IT OFFERS A TOTAL OF TWELVE

ALGEBRAIC OPERATIONS, FOUR CONTROL OPERATIONS AND THREE

TRIGONOMETRIC OPERATIONS. INDEXING IS AVAILABLE BUT IS LIMITED

TO ONE INDEX REGISTER.

0704 833MINTQ AVAILABLE PRIOR TO JANUARY 1962

HESSEL FUNCTIONS J0/X, Y0/X, J1/X, Y1/X FOR X > 0

REQUIRES 375 CELLS.*

0704 833MNHTQ AVAILABLE PRIOR TO JANUARY 1962

HESSEL FUNCTION J1/X AND Y1/X

REQUIRES 270 CELLS.

0704 8370FRLM AVAILABLE PRIOR TO JANUARY 1962

HESSEL FUNCTION OF THE FIRST KIND FOR NLLS.

OR NLLS MUST BE USED. MODIFIED VERSION OF CS RSL2 USES 88

LOCATIONS IN LOWER MEMORY. CORR/ 896

0704 8370FRLN AVAILABLE PRIOR TO JANUARY 1962

NON-LINEAR LEAST SQUARES.

ITERATES FOR THE LEAST SQUARES ESTIMATES OF PARAMETERS WHEN

DATA ARE BEING FITTED WITH NON-LINEAR FUNCTIONS. THE USER

PROVIDES A PROGRAM TO EVALUATE THE FUNCTION AND ITS DERIVA-

TIVES. THE VARIANCE OF ANY FUNCTION OF THE PARAMETERS CAN BE

ESTIMATED.

0704 8370FRLU AVAILABLE PRIOR TO JANUARY 1962

FLOATING-POINT OVERFLOW/UNDERFLOW ROUTINE FOR NLLS.

O R NLLS MUST BE USED. PRINTS ON-LINE THE LOCATION OF THE ORDER

CAUSING FLOATING-POINT OVERFLOW OR UNDERFLOW SETS OVERFLOWED

REGISTERS TO 32 BINARY ONES WITH THE CORRECT SIGN AND UNDER-

FLOWED REGISTERS TO CONCLUDES 60 LOCATIONS.

0704 8370FRLC AVAILABLE PRIOR TO JANUARY 1962

SINE AND COSINE FUNCTIONS FOR NLLS.

O R NLLS MUST BE USED. MODIFIED VERSION OF IB SINL USES 104

LOCATIONS IN LOWER MEMORY. CORR/ 896

0704 8449SLPL AVAILABLE PRIOR TO JANUARY 1962

STUDENT'S T AT .05 LEVEL

COMPUTES STUDENTS T AT THE .05 LEVEL FOR A FIXED OR FLOATING

POINT ARGUMENT. TIMING - 1.6 MS. USES 75 LOCATIONS IN LOWER

MEMORY.

0704 8430RCLK AVAILABLE PRIOR TO JANUARY 1962

EXPONENTIAL/3/ROUTINE FOR NLLS.

OR NLLS MUST BE USED. PRINTS ON-LINE THE LOCATION OF THE ORDER

CAUSING FLOATING-POINT OVERFLOW OR UNDERFLOW SETS OVERFLOWED

REGISTERS TO 32 BINARY ONES WITH THE CORRECT SIGN AND UNDER-

FLOWED REGISTERS TO CONCLUDES 60 LOCATIONS.

0704 8430RCLN AVAILABLE PRIOR TO JANUARY 1962

STUDENT'S T AT .05 LEVEL

COMPUTES STUDENTS T AT THE .05 LEVEL FOR A FIXED OR FLOATING

POINT ARGUMENT. TIMING - 1.6 MS. USES 75 LOCATIONS IN LOWER

MEMORY.

0704 8430RCLX AVAILABLE PRIOR TO JANUARY 1962

EXPONENTIAL/3/ROUTINE FOR NLLS.

OR NLLS MUST BE USED. PRINTS ON-LINE THE LOCATION OF THE ORDER

CAUSING FLOATING-POINT OVERFLOW OR UNDERFLOW SETS OVERFLOWED

REGISTERS TO 32 BINARY ONES WITH THE CORRECT SIGN AND UNDER-

FLOWED REGISTERS TO CONCLUDES 60 LOCATIONS.

0704 8430RCLX AVAILABLE PRIOR TO JANUARY 1962

EXPONENTIAL/3/ROUTINE FOR NLLS.

OR NLLS MUST BE USED. PRINTS ON-LINE THE LOCATION OF THE ORDER

CAUSING FLOATING-POINT OVERFLOW OR UNDERFLOW SETS OVERFLOWED

REGISTERS TO 32 BINARY ONES WITH THE CORRECT SIGN AND UNDER-

FLOWED REGISTERS TO CONCLUDES 60 LOCATIONS.

0704 8430RCLX AVAILABLE PRIOR TO JANUARY 1962

EXPONENTIAL/3/ROUTINE FOR NLLS.

OR NLLS MUST BE USED. PRINTS ON-LINE THE LOCATION OF THE ORDER

CAUSING FLOATING-POINT OVERFLOW OR UNDERFLOW SETS OVERFLOWED

REGISTERS TO 32 BINARY ONES WITH THE CORRECT SIGN AND UNDER-

FLOWED REGISTERS TO CONCLUDES 60 LOCATIONS.

0704 8430RCLX AVAILABLE PRIOR TO JANUARY 1962

EXPONENTIAL/3/ROUTINE FOR NLLS.

OR NLLS MUST BE USED. PRINTS ON-LINE THE LOCATION OF THE ORDER

CAUSING FLOATING-POINT OVERFLOW OR UNDERFLOW SETS OVERFLOWED

REGISTERS TO 32 BINARY ONES WITH THE CORRECT SIGN AND UNDER-

FLOWED REGISTERS TO CONCLUDES 60 LOCATIONS.

0704 8430RCLX AVAILABLE PRIOR TO JANUARY 1962

EXPONENTIAL/3/ROUTINE FOR NLLS.

OR NLLS MUST BE USED. PRINTS ON-LINE THE LOCATION OF THE ORDER

CAUSING FLOATING-POINT OVERFLOW OR UNDERFLOW SETS OVERFLOWED

REGISTERS TO 32 BINARY ONES WITH THE CORRECT SIGN AND UNDER-

FLOWED REGISTERS TO CONCLUDES 60 LOCATIONS.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 009MPDUMP AVAILABLE PRIOR TO JANUARY 1962
FORTRAN DUMP PROGRAM
THESE SUBROUTINES PRINT ON OR OFF-LINE DESIGNATED VARIABLE
THE NAME OF THE PROGRAM CALLING DUMP AND THE FORMULA
NUMBERS.

0704 009MPEND AVAILABLE PRIOR TO JANUARY 1962
FORTRAN END CARD SEARCH.
END SEARCHES A FORTRAN SOURCE PROGRAM TAPE AND STOPS
WHEN IT DISCOVERS AN END CARD.

0704 009MPFOT AVAILABLE PRIOR TO JANUARY 1962
FORTRAN TAPE WRITE PROGRAM.
FOR WRITES A TAPE FROM A FORTRAN BINARY DECK WHICH CAN
BE LOADED BY THE USE OF MPFOT, THE FORTRAN LIBRARY LOADER.

0704 009MP.OUT AVAILABLE PRIOR TO JANUARY 1962
SELF LOADING TAPE WRITE PROGRAM.
THIS IS A 2 CARD MODIFICATION TO PEPL/GELE, THE GENERAL
PROGRAM LOADER, TO FACILITATE SEGREGATION OF SELF-LOADING
PROGRAM TAPES. USES 2 OCTAL LOCATIONS DIRECTLY REFL MPFOT.

0704 009MPWRED AVAILABLE PRIOR TO JANUARY 1962
RATIONAL NUMBER ARITHMETIC
A RATIONAL NUMBER A/B HAS AN EXACT REPRESENTATION
AN INTERNAL FORMULA NUMBERS AND THE CURRENT VALUES
OF A/B.

0704 90704BACK AVAILABLE PRIOR TO JANUARY 1962
OR TAPE, MAKES POSSIBLE REREADING FROM STORAGE
BACK, GIVING THE NAMES OF ALL SUBROUTINES UNDER THE EXTERNAL
NAME, PROGRAM DOWN TO THE POINT WHERE CONTROL WENT TO
BCD TAPE-CARD READING FOR MULTIPLE SCAN.

0704 90704NBACK AVAILABLE PRIOR TO JANUARY 1962
THE LOADER, PROVIDES OPTIONS FOR THE FOLLOWING

FCRTAN TAPE WRITE PROGRAM.
FOR WRITES A TAPE FROM A FORTRAN BINARY DECK WHICH CAN
BE LOADED BY THE USE OF MPFOT, THE FORTRAN LIBRARY LOADER.

0704 90704NULU AVAILABLE PRIOR TO JANUARY 1962
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 90704PMP AVAILABLE PRIOR TO JANUARY 1962
FORTRAN MAP AND MISSING SUBROUTINE PRINT-OUT PROGRAM
PRINTS A LINE A MAP OF SUBROUTINE CALLS FOR EACH
OCTAL ADDRESSES OR Prints OUT MISSING SUBROUTINE NAMES.

0704 90704NULU AVAILABLE PRIOR TO JANUARY 1962
TO WRITE 2 DIMENSIONAL ARRAY BINARY INFO ON TAPE
TO WRITE 2-DIMENSIONAL ARRAY OF BINARY INFORMATION ON
TAPE, PROCEEDED BY 2 INTEGERS GIVING THE NUMBER OF ROWS
AND COLUMNS AND FOLLOWED BY A CHECK SUM. A COMPANION
PROGRAM NU NTR READS THE BINARY TAPE AND CHECKS THE SUM.

0704 90704NULU AVAILABLE PRIOR TO JANUARY 1962
TO READ AND CHECK NU WTR-WRITTEN RECORDS ON TAPE
TO READ AND CHECK RECORDS OF INFORMATION WHICH HAVE
BEEN WRITTEN BY NU WTR. ALSO DETECTS END-OF-FILE.

0704 90704NULU AVAILABLE PRIOR TO JANUARY 1962
THE TAPE IS IN THE SAME FORMAT AS THE INPUT TAPE.
DATA LISTING ARE COMPUTED
AND PRINTED, OPTIOAL FEATURE INCLUDE USE OF SYMTHETIC
RESERVATIONS AND ALSO RE-EVALUATION OF ANY NUMBER OF ANY
COMBINATION OF VARIABLES. CORR/917

0704 90704NULU AVAILABLE PRIOR TO JANUARY 1962
MULTIPLE REGRESSION: COMPREHENSIVE ANALYSIS
INCLDES ALL NORMAL PHASES OF STATISTICAL REGRESSION
ANALYSIS, STARTING WITH DATA LISTING ALL VARIABLES.
COMPUTATION PROCEEDS THRU LEAST SQUARES FITTING, STANDARD
STATISTICAL COEFFICIENTS, STANDARD ERRORS, SUMS OF SQUARES,
AND AVERAGES ARE COMPUTED AND PRINTED. PREDICTIONS AND
RESIDUALS FOR EACH DATA IN ITEM LISTING ARE COMPUTED
AND PRINTED. OPTIONAL FEATURES INCLUDE USE OF SYMTHETIC
RESERVATIONS AND ALSO RE-EVALUATION OF ANY NUMBER OF ANY
COMBINATION OF VARIABLES. CORR/917

0704 90704NULU AVAILABLE PRIOR TO JANUARY 1962
FORTRAN II BINOMIAL COEFFICIENT SUBROUTINE
FOR NON-NEGATIVE, INTEGRAL NUMBERS LESS THAN 13, COMPUTES A
SET OF BINOMIAL COEFFICIENTS BY ADDITION IN THE FORTRAN
SINGLE-PRECISION FLOATING-POINT MODE AND STORES THEM IN A ONE
DIMENSIONAL ARRAY, MAXIMUM ACCURACY IS MAINTAINED DURING THE
COMPUTATION, WITH INCLUDED BINOMIAL CORRECTION CARD, MINIMUM
LOUP IS 13 CYCLES FOR 70V/ AND IS EXECUTED N/M-1/2 TIMES.
MAX CHOP CYCLES.

0704 90704NULU AVAILABLE PRIOR TO JANUARY 1962
FORTRAN II BINOMIAL COEFFICIENT FUNCTION SUBPROGRAM
FOR NON-NEGATIVE, INTEGRAL NUMBERS LESS THAN 25, COMPUTES ANY
BINOMIAL COEFFICIENT BY ADDITION IN THE FORTRAN SINGLE-
PRECISION FLOATING-POINT MODE AND STORES THEM IN A ONE
DIMENSIONAL ARRAY, MAXIMUM ACCURACY IS MAINTAINED DURING THE
COMPUTATION, WITH INCLUDED BINOMIAL CORRECTION CARD, MINIMUM
LOUP IS 27 CYCLES FOR 70V/ AND IS EXECUTED N/M-1/2 TIMES. 7413A COM
CONF/900

OBTAINS THE N-STRIP TRAPEZOIDAL RULE INTEGRATION/EQUAL INTERVALS/ETC. REQUIRES 46 LOCATIONS IN FULL VERSION, 42 IN STRIPPED

PSEUDO-ISBN SUBROUTINE

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 9997P 81
Available prior to January 1962

INTEGRAL PROGRAMMING 1
AN INTEGRAL VERSION OF PK 81, HANDLES PROBLEMS WITH ONE
OBJECTIVE FUNCTION, UP TO 35 VARS., AND AT MOST 75-N
CONSTRAINTS, WHERE N IS THE NUMBER OF
VARIABLES AND 0104. 9b9PK 81 P81 AVAILABLE PRIOR TO JANUARY 1962
ON DEGENERATE PROBLEMS. REQUIRES 32K MEMORY,
OBJECTIVE FUNCTION, UP TO 35 VARIABLES, AND AT MOST
0704 9997P 82 AVAILABLE PRIOR TO JANUARY 1962
INDEPENDENT FORTRAN PROG. FOR SOLVING INTEGER PROGRAMMING
PROBLEMS. METHOD USED IS BASIC 2, THE ALL-INTEGER ALGORITHM
EMPIRED IN PK 81, BUT CONTAINS MODIFICATIONS WHICH PERMIT
SOLVING SOME PROBLEMS. INTEGRATION TIME DEPENDS ON THE
NUMBER OF ITERATIONS WHICH IS GENERALLY REDUCED, WITH THE RESULT THAT
THE CODE IS FASTER FOR DIFFICULT PROBLEMS, SLOWS ONLY ON SIMPLE PROBLEMS.
MACHINE AND PROBLEM RESTRICTIONS ARE SAME FOR PK 81; 1237

0704 9997P 83 AVAILABLE PRIOR TO JANUARY 1962
INDEPENDENT FORTRAN PROG. FOR SOLVING INTEGER PROGRAMMING
PROBLEMS SIMULATING IP 83. EFFECTIVE PROBLEMS SIMULATING IP 83 EXCEPT
ON DEGENERATE PROBLEMS, REQUIRES 12K MEMORY, 1 TAPE.
TAPE-FOR ID STP placeholders. N MAY NOT EXCEED 100 AND
TOTAL NUMBER OF OBJECTIVE FUNCTIONS AND CONSTRAINTS HAS A
LIMIT OF 100-N. EMPLOYS VARIOUS METHODS OF INTEGER
ACCU'ACY OR NUMBER OF COEFFICIENTS SPECIFIED IN
CALL STATEMENT, RESULTS CAN BE TESTED AT UP TO 105 POINTS
0704 9997P 84 AVAILABLE PRIOR TO JANUARY 1962
INDEPENDENT FORTRAN PROG. FOR SOLVING INTEGER PROGRAMMING
PROBLEMS SIMULATING IP 84 EXCEPT ON DEGENERATE PROBLEMS, REQUIRES 12K MEMORY, 1 TAPE.
TAPE-FOR ID STP placeholders. N MAY NOT EXCEED 100 AND
TOTAL NUMBER OF OBJECTIVE FUNCTIONS AND CONSTRAINTS IS A
LIMIT OF 100-N. EMPLOYS VARIOUS METHODS OF INTEGER
ACCU'ACY OR NUMBER OF COEFFICIENTS SPECIFIED IN
CALL STATEMENT, RESULTS CAN BE TESTED AT UP TO 105 POINTS
0704 9997P 85 AVAILABLE PRIOR TO JANUARY 1962
INDEPENDENT FORTRAN PROG. FOR SOLVING INTEGER PROGRAMMING
PROBLEMS SIMULATING IP 85. EFFECTIVE PROBLEMS SIMULATING IP 85 EXCEPT
ON DEGENERATE PROBLEMS, REQUIRES 12K MEMORY, 1 TAPE.
TAPE-FOR ID STP placeholders. N MAY NOT EXCEED 100 AND
TOTAL NUMBER OF OBJECTIVE FUNCTIONS AND CONSTRAINTS IS A
LIMIT OF 100-N. EMPLOYS VARIOUS METHODS OF INTEGER
ACCU'ACY OR NUMBER OF COEFFICIENTS SPECIFIED IN
CALL STATEMENT, RESULTS CAN BE TESTED AT UP TO 105 POINTS

0704 9997P 86 AVAILABLE PRIOR TO JANUARY 1962
LINEAR PROGRAMMING WITH UPPER BOUNDS ON VARIABLES
THIS LINEAR PROGRAMMING SUBROUTINE WILL SOLVE PROBLEMS THAT HAVE
UPPER BOUND RESTRICTIONS ON SOME OR ALL THE VARIABLES. THE
FUNCTION DEPENDS QUADRATICALLY ON THE VARIABLES.
FUNCTION OF THE FIRST AND SECOND KIND, GIVEN N.
EXPLICIT SOLUTION OF THE GENERAL CUBIC EQUATION
ROOTS ARE OBTAINED BY METHOD OF DEL FERRO. 289 LOCATIONS PLUS
PACKED PROBLEM SIZE IS 256 DOUBLE AND 17232 VARIABLES. CODE DOES
A PRESENTATION OF A TAPCHED JANG REMEERING. JOB CAN BE INTERRUPTED.
10000000000 08359714 PRESENTATION OF A TAPCHED JANG REMEERING. JOB CAN BE INTERRUPTED.

0704 9997P 87 AVAILABLE PRIOR TO JANUARY 1962
BESSEL FUNCTION OF COMPLEX ARGUMENT AND ORDER.
VALUES OF JN(X) AND YN(X), WHERE N IS ANY INTEGER NOT
NEGATIVE AND X IS ANY REAL NUMBER.

0704 9997P 88 AVAILABLE PRIOR TO JANUARY 1962
VARIABLE METRIC MINIMIZATION
A FORTRAN ROUTINE DETERMINES LOCAL MINIMA OF DIFFERENTIABLE FUNCTIONS OF N VARIABLES, THE PROGRAM
EMPLOYS VARIABLE METRIC METHOD FOR MINIMIZATION IN THE
PROCESS OF LOCATING EACH MINIMUM A MATRIX IN WHICH TRIANSITERS THE
FUNCTION DEPENDS QUADRATICALLY ON THE VARIABLES AND
MORE THAN N ITERATIONS ARE REQUIRED. ROUTINE REQUIRES
6,137 STORAGE. VU00 BY 20 AN013 S10 1117

0704 9997P 89 AVAILABLE PRIOR TO JANUARY 1962
GENERALIZED OUTPUT SUBROUTINE
THIS SUBROUTINE IS A ROUTINE TO OUTPUT A TWO-DIMENSIONAL
ARRAY IN A FIXED GENERAL FORMAT.

0704 9997P 90 AVAILABLE PRIOR TO JANUARY 1962
SACKSPACE FILE-FORWARD SPACE FILE. TO MOVE A BINARY
OR DECIMAL TAPE FORWARD OR BACKWARD A SPECIFIED NUMBER OF
RECORDS. AT THE COMPLETION OF THIS SUBROUTINE, THE TAPE WILL
BE POSITIONED READY TO READ OR WRITE THE FIRST RECORD OF
THE FILE REQUESTED.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 1000001
Available prior to January 1962

PUNCH ABSOLUTE COLUMBINARY.
PUNCHES ON-LINE ABSOLUTE COLUMN BINARY CARDS IN THE
STANDARD SHARE FORMAT SO THAT THEY MAY BE LOADED BY THE
FORTRAN II ISD LOADER. ALTHOUGH THE CARDS PUNCHES ARE
ABSOLUTE CARDS, THE LOADING ADDRESSES MAY BE THE SAME AS
OR DIFFERENT THAN THE LOCATIONS FROM WHICH THE DATA IS
PUNCHING

0704 1000005
Available prior to January 1962

INFORMATION PROCESSING LANGUAGE V INTERPRETIVE SYSTEM
INTERPRETS AND EXECUTES PROGRAMS WRITTEN IN ILPS, A
SUBLANGUAGE, AS DESCRIBED IN "INFORMATION PROCESSING
LANGUAGES V MANUAL" SECTIONS I AND II

0704 1004007
Available prior to January 1962

CHEBYSHEV TRUNCATION SYSTEM 1
COMPUTES POLYNOMIAL RATIONAL AND CONTINUED FRACTIONS
APPROXIMATIONS TO ANALYTIC FUNCTIONS. DOUBLE PRECISION
ACCU'ACY, INPUT-DERIVATIONS OF THE CODE SHOULD BE
ACCURACY OF NUMBER OF COEFFICIENTS SPECIFIED IN
CALL STATEMENT, RESULTS CAN BE TESTED AT UP TO 105 POINTS
0704 1004008
Available prior to January 1962

ON-LINE LOADER FOR EDL, BNL, AB, AND TSG, CARDS
UPPER, LOWER VERSIONS OF IS CARDS WITH PROVISIONS FOR TSG PCH.
0704 1004009
Available prior to January 1962

CAR TO TAPE SIMULATOR AND ROW TO COLUMN CONVERTER.
72184 AND 70/B4 SIMULATION OF HOLLERITH AND COLUMN BINARY
7F4; ALSO ROW TO COLUMN CONVERSION. 10519

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 1004010
Available prior to January 1962

NUMERICAL INTEGRATION BY MIDPOINT PROCEDURE-
WITH PREDEFINED INTERVAL PLACEMENT.
FORTRAN II FUNCTION SUBROUTINE EVALUATES THE INTEGRAL OF A
FUNCTION UPON INPUT, WHICH IS THE SUM OF SUMS OF PRODUCTS
FROM THE SECOND DERIVATIVE OF THE FUNCTION. ITERATIONS NOT USED.
INTEGRATION IS DONE IN ONE STEP. ONE DIMENSIONAL, PROGRAM
USES 266 LOCATIONS, NO COMMON STORAGE USED.

0704 1004012
Available prior to January 1962

EIGENVALUES AND EIGENVECTORS OF REAL SYMMETRIC MATRICES
A GENERAL PROGRAM BUILT AROUND SUBROUTINE ANF402
D I S T. 66 WHICH USES GASS-JORDAN ELIMINATION
COMPILTED WITH DIMENSION 98 BUT CAN BE RECOMPILED WITH
DIMENSION 16 TO RUN ON 44 PCH. GENERAL INPUT PUNCH-OUT AND CHECKS OF
VALUES AND VECTORS BY SUBSTITUTION INTO MATRIX EQUATION

0704 1004013
Available prior to January 1962

MATRIX INVERSION AND LINEAR EQUATIONS A GENERAL PROGRAM BUILT AROUND SUBROUTINE ANF402
DIST. 66 WHICH USES GASS-JORDAN ELIMINATION
COMPILTED WITH DIMENSION 20 BUT CAN BE RECOMPILED WITH
DIMENSION 16 TO RUN ON 44 PCH. GENERAL INPUT PUNCH-OUT AND CHECKS OF
VALUES AND VECTORS BY SUBSTITUTION INTO MATRIX EQUATION

0704 1004014
Available prior to January 1962

LAGRANGE INTERPOLATION USES 7 POINTS, THREE PRECEDING AND THREE AFTER VALUE -LIMIT
OF 256 POINTS IN TABLE

0704 1004015
Available prior to January 1962

ASSOCIATED LEGENDRE FUNCTIONS 1
THIS PROGRAM COMPUTES THE ASSOCIATED LEGENDRE FUNCTIONS
POLY, WHERE N IS LESS THAN OR EQUAL TO M. THE PROGRAM
REQUIRES THAT UNITED AIRCRAFT UA SQR-4 BE ASSEMBLED WITH IT.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 1061 JSZM
AVAILABLE PRIOR TO JANUARY 1962
ZERO, MINIMUM SOLVER
Solves the class of problems which can be stated as
\[ f_1(x) = \ldots = f_n(x) = 0 / \min \quad x_i \]
where the functions, \( f_i \), may have zeros and/or local minima.

0704 1042 JPHICO
AVAILABLE PRIOR TO JANUARY 1962
RING COMPONENTS GENERATOR
Computes \( f(x) = H_{-1} f(x) - \ldots - H_{-n} f(x) \), where \( f(x) \) is a vector function, \( H_i \) is a matrix of order \( i \), and \( \eta \) is a constant scalar.

0704 1041 JSZIPGR
AVAILABLE PRIOR TO JANUARY 1962
SIMULATED PARTIAL DIFFERENTIAL EQUATIONS SOLVER
Solves the problem \( A \phi = \beta \), \( \phi \in \mathbb{R}^n \), \( A \in \mathbb{R}^{n \times n} \), \( \beta \in \mathbb{R}^n \),

0704 1040 JPHIN
AVAILABLE PRIOR TO JANUARY 1962
GAUSS APPROXIMANT GENERATOR
This subroutine can be used for generating the Gauss approximant for any type of integral expression, whether it be an iterated integral, vector-valued integral, or integral of a function of other integrals, or any combination of these. It uses 227 locations.

0704 1039 JSQP1
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 300. The program will operate on a 700 with a minimum of 16, 64 drums, and 1900 cards. The code, with the addition of two cards, can run on a 7090 with compatibility.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 1070 RMELFK
AVAILABLE PRIOR TO JANUARY 1962
COMPLETE ELLIPTIC INTEGRALS OF THE FIRST KIND
This subroutine evaluates the complete elliptic integrals of the first kind for different values of the modulus \( k \). It uses natural log subroutine LASS20 or the equivalent that uses common through common 2.

0704 1079 NOTIA
AVAILABLE PRIOR TO JANUARY 1962
SOLUTION OF RATIONAL SCHRODINGER EQUATION
This is a Fortran program to calculate the eigenvalues and eigenfunctions of the rational Schrodinger equation.
FIXED STEP SIZES MAY BE USED. A MODIFICATION FLOATING POINT OPTIMIZED RUNGE KUTTA DEPENDENT ON A VARIABLE ERROR CONTROL. FIXED STEP SIZES MAY BE USED. A MODIFICATION FLOATING POINT OPTIMIZED RUNGE KUTTA DEPENDENT ON A VARIABLE ERROR CONTROL.

PSEUDORANDOM NUMBER GENERATOR GIVES A NORMALIZED FLOATING POINT number Z-SUBI BETWEEN -1 AND 1. THE NUMBER Z-SUBI/RLED IS PRODUCED, WHERE Z-SUBI IS A SEQUENCE OF UNIFORMLY DISTRIBUTED PSUEDORANDOM NUMBERS ON THE INTERVAL /-1,1/.

MULTIVARIATE TECHNIQUE OF COMPONENT ANALYSIS. METHOD DEPLOYS MULTIPLE REGRESSION ON THAT COEFFICIENTS WHICH ARE DERIVED REPRESENT ORTHOGONAL CONTRIBUTIONS OF RESPECTIVE TYPES OF EQU. THIS SUPPRESSING EFFECTS OF CORRELATIONS AMONG INDEPENDENT VARIABLES, AN EIGENVALUE-EIGENVECTOR ANALYSIS OF C0-ASSOCIATED MATRIX EXPRESSES RELATIONSHIP BETWEEN INDEPENDENT VARIABLES AND ORTHOGONAL COMPONENTS, ADAPTATION OF CA 0054 USED AS SUBROUTINE.

MONITOR SYSTEM. AUTOMATIC CODE LANGUAGE LIKE FORTRAN, WITH RESTRICTION TO SINGLE SUBSCRIPTS. HANDLES MIXED ARITHMETIC.

MONITOR SYSTEM. AUTOMATIC CODE LANGUATE LIKE FORTRAN, WITH RESTRICTION TO SINGLE SUBSCRIPTS. HANDLES MIXED ARITHMETIC.

SHARE CATALOG UPDATE, LISTER, 1403 PROGRAM, REQUIRES A 1403 WITH ADV. OPT. OF 4K 1401 WITH ADV. PROG., INCLUDES SUBROUTINES ON SYS AND ON SYRINGE, INCLUDES SUBROUTINES ON SYRINGE. AUTOMATIC CODER, COMPATIBLE WITH SAP AUTOMATIC CODING SYSTEM WHOSE SOURCE LANGUAGE INCLUDES SAP CODING AS WELL AS STATEMENTS IN MATHEMATICAL LANGUAGE AND ENGLISH, TRANSLATES AUTOMATIC CODE TO SAP CODE, WHICH IS THEN ASSEMBLED USING OS, INCLUDES IN SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS.

SHARE CATALOG UPDATE, LISTER, 1403 PROGRAM, REQUIRES A 1403 WITH ADV. OPT. OF 4K 1401 WITH ADV. PROG., INCLUDES SUBROUTINES ON SYS AND ON SYRINGE, INCLUDES SUBROUTINES ON SYRINGE. AUTOMATIC CODER, COMPATIBLE WITH SAP AUTOMATIC CODING SYSTEM WHOSE SOURCE LANGUAGE INCLUDES SAP CODING AS WELL AS STATEMENTS IN MATHEMATICAL LANGUAGE AND ENGLISH, TRANSLATES AUTOMATIC CODE TO SAP CODE, WHICH IS THEN ASSEMBLED USING OS, INCLUDES IN SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS.

AUTOMATIC CODE, COMPATIBLE WITH SAP AUTOMATIC CODING SYSTEM WHOSE SOURCE LANGUAGE INCLUDES SAP CODING AS WELL AS STATEMENTS IN MATHEMATICAL LANGUAGE AND ENGLISH, TRANSLATES AUTOMATIC CODE TO SAP CODE, WHICH IS THEN ASSEMBLED USING OS, INCLUDES IN SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS.

SHARE CATALOG UPDATE, LISTER, 1403 PROGRAM, REQUIRES A 1403 WITH ADV. OPT. OF 4K 1401 WITH ADV. PROG., INCLUDES SUBROUTINES ON SYS AND ON SYRINGE, INCLUDES SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS. AUTOMATIC CODER, COMPATIBLE WITH SAP AUTOMATIC CODING SYSTEM WHOSE SOURCE LANGUAGE INCLUDES SAP CODING AS WELL AS STATEMENTS IN MATHEMATICAL LANGUAGE AND ENGLISH, TRANSLATES AUTOMATIC CODE TO SAP CODE, WHICH IS THEN ASSEMBLED USING OS, INCLUDES IN SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS, INCLUDES SUBROUTINES ON SYSTEMS.

MULTICOMPONENT DISTILLATION PROGRAM. SOLVES N+1 TO PLANT-COMMON DISTILLATION, BUBBLE, GLOW AND PLANT POINT PROBLEMS FOR up TO 23 COMPONENTS ON 1K MACHINE.
0704 1231TVPRP

AVAILABLE PRIOR TO JANUARY 1962

A PROGRAM TO GENERATE 1401 T/P PROGRAM ON OUTPUT TAPE TO MINIMIZE OPERATOR ATTENTION IN 1401 PRINT OPERATION FROM THE OUTPUT TAPE THROUGH PROGRAMMED 1401 INSTRUCTIONS WRITTEN ON THE TAPE AT THE TIME OF 704 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 1232AAICE4

AVAILABLE PRIOR TO JANUARY 1962

INTEGRATION WITH CONTROLLED ERROR

AAICE4 IS DESIGNED TO BE USED IN CONJUNCTION WITH AN INTEGRATION SUBROUTINE/INT1 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 NUMERICAL PROCESS IS CONTROLLED BY ADJUSTING THE INTEGRATION STEP SIZE BASED ON THE RELATIVE ERROR AS ESTIMATED BY EXTRAPOLATION TO ZERO STEP SIZE.

0704 1233AAINT1

AVAILABLE PRIOR TO JANUARY 1962

SECOND, THIRD, AND FOURTH ORDER RUNGE-KUTTA INTEGRATION

AAINT1 IS A FORTRAN II SUBROUTINE DESIGNED TO BE USED IN CONJUNCTION WITH AA 1234 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 1234AAWEG2

AVAILABLE PRIOR TO JANUARY 1962

NEUMANN ITERATION

GIVEN AN IMPPLICIT EQUATION OF THE FORM F(X-F/X)=AA WEG2 WILL FIND A VALUE FOR X WHICH WILL PROVIDE A SPECIFIED ACCURACY IN EITHER A RELATIVE OR ABSOLUTE SENSE.

0704 1244ANC001

AVAILABLE PRIOR TO JANUARY 1962

A GENERAL PROGRAM FOR SYSTEMS EVALUATION

**ABRAC - 01**

**704 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 parallelepips is 2885 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 16 x 16 x 26 mesh (two group), the running time might be from 1.5 to 2.0 hrs. per iteration. Three to four iterations may be required.

(6) References:

**APC0I**

**704 Nuclear Code**

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:
The APC0I code processes the flux tapes from a PDQ02 problem and its adjoint. The integrals

\[ \int \beta_i \beta_j \, dA \]

are obtained in an x-z 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 PDQ02 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:

**ART - 04**

**704 Nuclear Code**

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:
Replaces ABRAC - See Page II.3 for details

(4) References:
Letter, 7-31-58.

**ATBAC**

**704 Nuclear Code**

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:
Obtains detailed information concerning thermal conditions within a reactor core during transient operation. The method used applies particularly to plate type pressurized water reactors. The model used is that of a hot channel in a parallel flow path with the normal channels. A single normal channel is analyzed for heat transfer with pressure drop, with flow characteristics in the channel being determined a priori by the loop containing the reactor, heat exchangers, and pumps. The pressure drop across the normal channel thus determines the flow conditions in the hot channel, in conjunction with the hot channel heat transfer. In this way it is possible to simulate such varied transients as complete and staggered loss of flow, cold water accident, and rod pump accident.

(4) Restrictions or Limitations:
In normal usage the code is limited to a two-pass core with a maximum of 20 points per pass. Great caution must be used in selecting a value of \( \Delta t \) so that no instability is introduced into either the heat transfer or kinetics equations. The IBM equipment includes an 8K core, two tape units, and one drum unit.

(5) Approximate Performance:
A typical 20-point, 3-second transient with no scram will run about 15-20 minutes.

(6) References:

**BINTO**

**704 Nuclear Code**

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:
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) Restrictions or Limitations:
In normal usage the code is limited to a two-pass core with a maximum of 20 points per pass. Great caution must be used in selecting a value of \( \Delta t \) so that no instability is introduced into either the heat transfer or kinetics equations. The IBM equipment includes an 8K core, two tape units, and one drum unit.

(6) References:
4. NCO Newsletter No. 5, 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 homogenous 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 xenon poisoning calculation is optional at each time step.

(4) Restrictions or Limitations: Max of 35 regions and 150 mesh intervals with either two or four groups. At most 10 time steps can be done automatically. Only the uranium, plutonium, and fission product chains along with two burnable poisons are considered time dependent with a maximum of 30 elements in all. Code requires 8K core, four tape units, and one drum unit.

(5) Approximate Performance: From 15 min. to 4 hrs. Average of 30 min.

(6) References:

CETEB

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, mono-energetic P3 approximation to the transport equation in cylindrical geometry. The cylinder is assumed to be infinitely long and symmetric with respect to rotations about the Z axis. The external boundary condition may be specified as reflecting or vacuum, or as a special type of cell condition. Any material 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 350 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:
CEND MPC-20.

COFIT

704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: Fits by least squares the curve \( y = A \cos(Bx - C) \) to from 4 to 500 points of observed data, computing the parameters \( A, B, C, \) and the standard deviations of the estimates of \( A, B, C \ldots \). It is also possible to investigate the errors in a region about the final values of \( A, B, C \), by computing the sums of the squares of the residuals at a series of points in the neighborhood.

(5) Approximate Performance: 300 point problem \( y \) in 2 min.

(6) References:
arbitrary function times the vector flux density. Spatial moments of all density functions are furnished directly. The neutron source may be monenergetic with either isotropic or directional angular distributions, or else the source may be that from deuterons bombarding deuterons.

(4) Restrictions or Limitations:
A 32K core memory is required, and 5 tape units are required.

(5) References:

FLEER

(1) Code Originated by:
OE-Keolis Atomic Power Lab.

(2) Computer:
704

(3) Description of Code:
FLEER will solve the three-group, two-dimensional neutron diffusion equation in a triangular coordinate system. Up to 14,500 mesh points are allowed. The outer boundary of the point mesh must be a parallelogram. A special 100 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.

(4) Restrictions or Limitations:
A 32K memory is required, as well as 7 tapes and 4 drums.

(5) Approximate Performance:
Approximately running time for a problem is about 40 minutes per 1000 points.

(Continued on next column)

References:

FLIP

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
FLIP was developed specifically for the calculation of flow transients occurring in a multi-loop flow system closed by a common flow path. The program is based on a multi-loop model of up to three inertially symmetric flow loops with one common motor, variable frequency, induction motor driven pumps per loop having a separate motor power supply.

(4) Restrictions or Limitations:
An 8K memory is required.

(5) Approximate Performance:
The problem should run between 0.5 h and 1 h for any accident with final time of 6.0 seconds and just transient output.

(6) References:

(7) Material Available:
1. KAPL-2086.
2. Binary deck.

Note: The information above was abstracted from KAPL-2086.
Approximate Performance:
For a sample problem, the 704 running time was 3.3 minutes for the calculation and normal point-out of the 3.3 minutes of running time, approximately 1.5 minutes were used in writing the output on tape.

References:

HECTIC

(1) Code Originated by: Aerojet-General Nuclear
(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 "modal" or "lumped parameter" type calculation.

(4) Limitations or Restrictions: An 8K memory is required.

(5) Approximate Performance: A full-size run requires approximately 15 minutes.

(6) References:

HEAT

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: HEAT is a code which finds a one-dimensional solution to the general heat transfer equation. Specifically written for applications in reactor fuel rod design, the code requires cylindrical geometry conditions and input parameters of surface temperature and power density. The conductivity may be assumed to be a function of temperature.

(4) Restrictions or Limitations: The maximum number of points for which temperature values may be distributed is a maximum of 25 regimes. An 8K core memory is required.

(5) Approximate Performance: The approximate running time for a typical problem varies from 1.0 to 1.0 minutes.

(6) References:
MUFT-4: 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 factors 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:

PDQ-2: 704 Nuclear Code

1. Code Originated by: Westinghouse - Bettis Plant
2. Computer: 704
3. Description of Code: Similar to PDQ - 2 except that a single-line over-relaxation is used.
4. Restrictions or Limitations: Requires 32K core memory.
5. References:

PECAN: 704 Nuclear Code

1. Code Originated by: Aerojet - General Nucleonics
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 specific power plant design to a major requirement such as weight, economy, or output.
4. Restrictions or Limitations: The code is restricted to the use of a gaseous working fluid within a temperature range of 300°C to 3200°C, but in otherwise general.
5. References:

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, monodirectional, 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.
4. Restrictions or Limitations: Requires 32K core memory.
5. References:
PROP and JET  
704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code: These programs form the power distribution for a reactor core in three dimensions from previously determined one and two-dimensional power shapes. Thermal data are calculated for various axial traverses, and the results can be sorted to determine the worst areas for further study. PROP, the first of the two codes, operates on the nuclear data determined by TURBO. It combines the (x,y) radial power shapes from several time steps in each of several TURBO problems on a single tape in a convenient form for further calculations. JET then combines any selected group of these radial power shapes with a single axial power shape which has been previously determined by a one-dimensional axial study. The JET code also performs thermal criteria and power sharing calculations.

(4) Restrictions or Limitations: This program requires about a 16K or 32K core memory. The core to be studied may have as many as 100 axial mesh intervals and 25 axial regions. It may have up to 55 radial regions, and, depending on machine size, up to 3750 or 6000 interior radial mesh points. As many as 50 or 6000 of the radial mesh rectangles may contain fuel.

(5) Approximate Performance: The running time for a problem having 1512 fueled rectangles, 35 axial intervals, 8, radial fuel regions, and 10 axial regions is less than 1 hr. total.


RANCH  
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, while the CURE code uses only horizontal and vertical mesh sides. A typical problem is run on the SET 02 code much faster than on the CURE code. On the other hand, the SET 03 is subject to round off errors when the mesh is sufficiently refined, while the method used in the SET 03 code is inherently "stables". A 32K core memory is required as well as 2 tapes. No drums are required.

(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 10 angles.

(5) Approximate Performance: About 5 minutes.


Note: The information above was abstracted from WAPD-TM-268.
SIMPL - 1

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704

(3) Description of Code:
A neutron diffusion code for one-group or two-group problems. Provides the reactivity, neutron spectrum, and fission rate in the reactor core. The code is capable of handling reactors with axial and radial symmetry.

(4) Restrictions or Limitations:
A maximum of 100 regions and 1000 mesh intervals are permitted.

(5) Approximate Performance:
Typical computing time for a 20-field point problem is 5 minutes.

(6) References:

SIMPL - 2

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704

(3) Description of Code:
A specific neutron diffusion code for one-group or two-group problems. Provides the reactivity, neutron spectrum, and fission rate in the reactor core. The code is capable of handling reactors with axial and radial symmetry.

(4) Restrictions or Limitations:
A maximum of 100 regions and 1000 mesh intervals are permitted.

(5) Approximate Performance:
Typical computing time for a 20-field point problem is 5 minutes.

(6) References:

SOFOCATE

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704

(3) Description of Code:
A neutron diffusion code for one-group or two-group problems. Provides the reactivity, neutron spectrum, and fission rate in the reactor core. The code is capable of handling reactors with axial and radial symmetry.

(4) Restrictions or Limitations:
A maximum of 100 regions and 1000 mesh intervals are permitted.

(5) Approximate Performance:
Typical computing time for a 20-field point problem is 5 minutes.

(6) References:

SPAN - 2

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704

(3) Description of Code:
A neutron diffusion code for one-group or two-group problems. Provides the reactivity, neutron spectrum, and fission rate in the reactor core. The code is capable of handling reactors with axial and radial symmetry.

(4) Restrictions or Limitations:
A maximum of 100 regions and 1000 mesh intervals are permitted.

(5) Approximate Performance:
Typical computing time for a 20-field point problem is 5 minutes.

(6) References:

SPIC - 1

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704

(3) Description of Code:
A neutron diffusion code for one-group or two-group problems. Provides the reactivity, neutron spectrum, and fission rate in the reactor core. The code is capable of handling reactors with axial and radial symmetry.

(4) Restrictions or Limitations:
A maximum of 100 regions and 1000 mesh intervals are permitted.

(5) Approximate Performance:
Typical computing time for a 20-field point problem is 5 minutes.

(6) References:

(Continued on next page)
(3) Description of Code:
The SPIC - I code calculates the fast-neutron dose rate at the thermal neutron flux at a point outside a right circular cylindrical source 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 16K core memory is required. Other limitations are those of the SPAN - 2 code.

(5) Approximate Performance:
Typical computing and coding time for a 20-field-point-problem, in which there are 10 side and 10 top shields, is 6.5 minutes.

(6) References:

STDY-3

704 Nuclear Code

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
STDY-3 is a computer program designed for the thermal analysis of a pressurized water nuclear 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 6.75 minutes.

(6) References:
2. WAPD-TM-213.

Note: The information given above was abstracted from Reference 1.

TEMP - 2

704 Nuclear Code

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
The TEMP - 2 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 regionwise application of the analytical stress expressions of Reference 6 (2) below.

(4) Restrictions or Limitations:
The size of the core memory required is not given in Reference 6 (1), but it is believed to be 32K. The program provides for minimum of 3 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 11-point problem requires about 5 seconds of computer time per time step.

(6) References:

TRIP - 1

704 Nuclear Code

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
The TRIP - 1 program is designed to solve the P3 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 rigorously (within the limits of a P3 approximation). Simultaneous line over-relaxation is used to solve the difference equations.

(4) Restrictions or Limitations:
A 32K core memory is required. Nine tape units are required. No more than 750 interior mesh points are allowed.

(5) References:
TURBO

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code:
Two space dimensions and time version of CANDILE for x-y (TURBO 1 and 3), and r-z (TURBO 2 and 4) geometry. Otherwise same as CANDILE except that the 9DG spatial calculation is used. Maximum amount calculation is TURBO-3 for x-y or TURBO-4 for r-z.
(4) Restrictions or Limitations:
Mix of 35 compositions. Number of mesh points limited by size of core according to the number pairs 8K-3300, 16K-3750, 32K-6500; with a minimum of 812 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:

WANDA 2

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code:
Transient temperature and stresses in axially symmetric solid or hollow bodies. Derivatives of these quantities are also given.
(4) Restrictions or Limitations:
Requires an 8K core memory, 1 drum unit, and 1 tape unit.
(5) Approximate Performance:
1-15 minutes, average 3 minutes.
(6) References:

TUT - TS

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code:
The TUT - TS code provides, for a one-energy model, a means of calculating a representative 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 content; 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:

UFO

(2) Computer: 704
(3) Description of Code:
Solves the few-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 eigenvalue 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:
1-15 minutes, average 3 minutes.
(6) References:
WB TSG - 1  

704 Nuclear Code

(1) Code Originated by: 
Westinghouse - Bettis Plant

(2) Computer: 
704

(3) Description of Code: 
Computes in one-dimensional form the tangential, axial, and radial thermal stresses for cylinders with internal heat generation.

(5) Approximate Performance: 
20 minutes.

(6) References: 
3. NCG Newsletter No. 5, p. 5.

ZOOM  

704 Nuclear Code

(1) Code Originated by: 
University of California, Radiation Lab.

(2) Computer: 
704

(3) Description of Code: 
Solves the one-dimensional multigroup neutron diffusion equation for slabs, cylinders or spheres. A 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.

(5) Approximate Performance: 
10 minutes.

(6) References: 
UGRL NTS-1-Preliminary (UGRL 5293 available in about 1 month), September 1956.

JOXY  

704 Nuclear Code

(1) Code Originated by: 
Aerojet-General Nuclear

(2) Computer: 
704, (FLOCO-II-2.0)

(3) Description of Code: 
The JOXY program solves the homogeneous or inhomogeneous multi-group transport equation in x-y co-ordinate. Vacuum, surface source, or reflecting boundary conditions are available as options. In the homogeneous case the user may request the computation of reactivity, period, critical concentrations of some composition or the critical thickness of a zone. The S0 approximation is used.

(4) Restrictions or Limitations: 
Scattering must be isotropic.

(5) Approximate Performance: 
One and one-half hours for 6 group, 1000 mesh points on the 7090 (using the binary editor).

(6) References: 

(Continued on next column)
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AP-001-D

PROGRAM NAME
CHANGE-CARD-LOAD

PURPOSE: To load program cards into 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: 702
Tapes: 705 X Model 1 or II Other
Card Reader: X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:
George Widding
AFASC-3E
Washington 25, D. C.

April 1958, Bulletin 57 - 45

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AP-002-D

PROGRAM NAME
MEMORY SWITCH OUT

PURPOSE: To punch out program decks incorporating change cards to cut down the
size of program decks and serial number cards in the deck. 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: 702
Tapes: 705 X Model 1 or II Other
Card Reader: X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual X

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:
George Widding
Headquarters, USAF
AFASC-3E
Washington 25, D. C.

August 1957, Bulletin 50 - 108

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AP-011-D

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: 702
Tapes: 705 X Model 1 or II Other
Card Reader: X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:
George Widding
AFASC-3E
Washington 25, D. C.

April 1958, Bulletin 57 - 41

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AP-012-D

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**

**PROGRAM NAME**

**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:**

**PROGRAM LANGUAGE:**

**PROGRAM TYPE:**

**CONTRIBUTED BY:**

*George Widdling, AFASC-3E*

**Data Processing Division**

**Headquarters, USAF, Washington 25, DC**

April 1958, Bulletin 57 - 43

---

**GUIDE**

PROGRAM WRITE-UP ABSTRACT

**INDICATIVE CODE**

**PROGRAM NAME**

**PURPOSE:** To function as a debugging aid in cases where memory print fails. The routine lists each PRINT I step executed, along with numerical values of the operands and results, if any.

**MACHINE:**

**PROGRAM LANGUAGE:**

**PROGRAM TYPE:**

**CONTRIBUTED BY:**

*George Widdling, AFASC-3E*

**Data Processing Division**

**Headquarters, USAF, Washington 25, DC**

August 1957, Bulletin 50 - 117

---

**GUIDE**

PROGRAM WRITE-UP ABSTRACT

**INDICATIVE CODE**

**PROGRAM NAME**

**PURPOSE:** To produce the coefficients of the polynomial which fits given 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 the given data when required.

**MACHINE:**

**PROGRAM LANGUAGE:**

**PROGRAM TYPE:**

**CONTRIBUTED BY:**

*Robert P. TaPscott*

**Allison Division, General Motors Corp.**

(Continued on next page)
### PROGRAM WRITE-UP ABSTRACT

**PROGRAM NAME**

**MACHINE**

- 702 __ 705, _ x (Specify)
- One (Specify)
- Model: _
- Other:

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual (Specify)

**PROGRAM TYPE**: Complete Program

- Macro-Instruction
- Subroutine

**CONTRIBUTED BY**: W. R. Brittenham

**INDICATIVE CODE**

A0-004-0

(August 1957, Bulletin 50 - 121)

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**PROGRAM NAME**

**MACHINE**

- 702 __ 705, _ x (Specify)
- One (Specify)
- Model: _
- Other:

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual (Specify)

**PROGRAM TYPE**: Complete Program

- Macro-Instruction
- Subroutine

**CONTRIBUTED BY**: W. R. Brittenham

**INDICATIVE CODE**

A0-005-0

(August 1957, Bulletin 50 - 123)

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**PROGRAM NAME**

**MACHINE**

- 702 __ 705, _ x (Specify)
- One (Specify)
- Model: _
- Other:

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual (Specify)

**PROGRAM TYPE**: Complete Program

- Macro-Instruction
- Subroutine

**CONTRIBUTED BY**: W. R. Brittenham and G. W. Kuss

**INDICATIVE CODE**

A0-010-0

**PROGRAM NAME**

**MACHINE**

- 702 __ 705, _ x (Specify)
- One (Specify)
- Model: _
- Other:

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual (Specify)

**PROGRAM TYPE**: Complete Program

- Macro-Instruction
- Subroutine

**CONTRIBUTED BY**: W. R. Brittenham and G. W. Kuss

**INDICATIVE CODE**

A0-014-0

**PROGRAM NAME**

**MACHINE**

- 702 __ 705, _ x (Specify)
- One (Specify)
- Model: _
- Other:

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual (Specify)

**PROGRAM TYPE**: Complete Program

- Macro-Instruction
- Subroutine

**CONTRIBUTED BY**: W. R. Brittenham and G. W. Kuss

April 1958, Bulletin 57 - 47

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**PROGRAM NAME**

**MACHINE**

- 702 __ 705, _ x (Specify)
- One (Specify)
- Model: _
- Other:

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual (Specify)

**PROGRAM TYPE**: Complete Program

- Macro-Instruction
- Subroutine

**CONTRIBUTED BY**: W. R. Brittenham

**INDICATIVE CODE**

A0-016-0

**PROGRAM NAME**

**MACHINE**

- 702 __ 705, _ x (Specify)
- One (Specify)
- Model: _
- Other:

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual (Specify)

**PROGRAM TYPE**: Complete Program

- Macro-Instruction
- Subroutine

**CONTRIBUTED BY**: W. R. Brittenham and G. W. Kuss
**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**

Search Master Program Type

**PROGRAM NAME**

A0-011-0 Search Master Program

**PURPOSE:**

To search a master program tape on 02.01 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:**

702 705 X Model I Other (Specify)

Card Reader 760 Other

**PROGRAM LANGUAGE:**

Autocoder Symbolic Actual (Specify)

**PROGRAM TYPE:**

Complete Program X

Macro-Instruction X Label

Subroutine

**CONTRIBUTED BY:**

W. R. Brittenham and G. W. Kuss
A. G. Smith Corporation

Distribution No. 5

---

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**

BW - 002 - O

**PROGRAM NAME**

Address Modification

**PURPOSE:**

To provide a common set of address modification macro instructions for 705 Model II and 805 Model III. This version contains revisions to the macro instruction MOVEA of contribution BW - 001 - 0. The macro instructions included are:

- Macro Name
  - Address and Move
  - Subtract Address and Move
  - Increment Address
  - Decrement Address
  - Calculate Address
  - Initialize Address
  - Move Address
  - Unconditional Transfer

**MACHINE:**

702 705 X Model II and III Other (Specify)

Card Reader 760 Other

**PROGRAM LANGUAGE:**

Autocoder Symbolic Actual (Specify)

**PROGRAM TYPE:**

Complete Program X

Macro-Instruction X Label

Subroutine

**CONTRIBUTED BY:**

Boeing Airplane Company
Wichita Division

Distribution No. 8

---

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**

BW - 002 - O

**PROGRAM NAME**

Miscellaneous General Purpose

**PURPOSE:**

Move Data

Digit Selection

Fixed Memory Counter

Linkage to Subroutine

Option Hall

Sequence Check

Sign a Field

Strip Field

Variable Memory Counter

**MACHINE:**

702 705 X Model II and III Other (Specify)

Card Reader 760 Other

**PROGRAM LANGUAGE:**

Autocoder Symbolic Actual (Specify)

**PROGRAM TYPE:**

Complete Program

Macro-Instruction X Label

Subroutine

**CONTRIBUTED BY:**

The Curtis Publishing Company
Independence Square
Philadelphia 5, Pennsylvania

Written by: William Anderson
IBM Corporation

Distribution No. 5

---
GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
40,000 position 705 with 1 tape unit

PURPOSE:
To calculate seasonal adjustment factors for series of any length between five and twelve years.

GENERAL DESCRIPTION:
The program is an adaptation of "Census Method II" for calculating seasonal adjustment factors. The steps involved in this method are described in detail in the Census releases, "Seasonal Variations in the Labor Force, Employment, and Unemployment" Series P-50, No. 82, April, 1958, and also in Technical Paper No. 12, "Seasonal Adjustments by Electronic Computer Methods" by Julius Shiskin and Harry Eisenpress, published by the National Bureau of Economic Research.

REQUIREMENTS AND RESTRICTIONS:
This program is written for a 12-digit mantissa Print I system for TRC's. However, it may be used by any Model II system after it is pre-edited by that particular 12-digit mantissa system.

CONTRIBUTED BY:
Charles E. Gudew, E. J. duPont de Nemours
Nancy K. Brower, IBM, Wilmington, Delaware

 ребантото содержание: 40,000 позиции 705 с 1 ленточным устройством

ПОЗНАНИЕ:
Для расчета сезонных коэффициентов для серии любой длины между пятью и двенадцатью годами.

ОБЩЕЕ СООТНОШЕНИЕ:
Программа является адаптацией метода "Севис Метод II" для расчета сезонных коэффициентов. Шаги, вовлекаемые в этот метод, описаны в деталях в релизах Сезонных колебаний в трудовой силе, занятости и безработицы, Серия P-50, No. 82, April, 1958, а также в технической работе No. 12, "Сезонные корректировки с помощью Электронных компьютеров" по Julius Shiskin и Harry Eisenpress, выпущенной National Bureau of Economic Research.

ПОТРЕБНОСТИ И ОГРАНИЧЕНИЯ:
Программа написана для системы сдельного жесткого диска TRC. Однако, она может быть использована с любой системой Model II после предварительной редактировки этой системы с 12-разрядным мантиссой.

ПРЕДОСТАВЛЕНА:
Charles E. Gudew, E. J. duPont de Nemours
Nancy K. Brower, IBM, Wilmington, Delaware
2. Tracing may be discontinued at any time during a run by turning off TRC. 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
EK-0004-0

PROGRAM NAME
EKACTO - 10 DIGIT CONVERSION

PURPOSE
Enable programmer to write and store in actual as 10 digits (Ex: RAD 02 25519). The routine processes cards punched in 10 digit form, checks instructions for validity, giving listings and condensed cards as output.

MACHINE
J702

Ifapes__Model...Other __

Printer TRC Drum

Card Reader (Optional) Punch (Optional)

PROGRAM LANGUAGE: Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction Label

Subroutine

CONTRIBUTED BY:
Earl Althoff
Eastman Kodak Company

January 1958, Bulletin 55 - 71

705 CUSTOMER CONTRIBUTION

Program Write-up Abstract

INDICATIVE CODE
EK-0004-0

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 Test 5; namely, to provide a means of checking 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 last 240 digits. Only 244 positions of accumulator 00 are available to 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 last 240 digits. The program occupies 643 memory positions.

CONTRIBUTED BY:
E. Althoff, Eastman Kodak
Rochester, New York

GUIDE PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EK-0004-0

PROGRAM NAME
EKACTO - 10 DIGIT CONVERSION

PURPOSE
Enable programmer to write and store in actual as 10 digits (Ex: RAD 02 25519). The routine processes cards punched in 10 digit form, checks instructions for validity, giving listings and condensed cards as output.

MACHINE
J702

Ifapes__Model...Other __

Printer TRC Drum

Card Reader (Optional) Punch (Optional)

PROGRAM LANGUAGE: Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction Label

Subroutine

CONTRIBUTED BY:
Earl Althoff
Eastman Kodak Company

January 1958, Bulletin 55 - 71

705 CUSTOMER CONTRIBUTION

Program Write-up Abstract

INDICATIVE CODE
EK-0004-0

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 Test 5; namely, to provide a means of checking 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 last 240 digits. Only 244 positions of accumulator 00 are available to 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 last 240 digits. The program occupies 643 memory positions.

CONTRIBUTED BY:
E. Althoff, Eastman Kodak
Rochester, New York

GUIDE PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EK-0004-0

PROGRAM NAME
EKACTO - 10 DIGIT CONVERSION

PURPOSE
Enable programmer to write and store in actual as 10 digits (Ex: RAD 02 25519). The routine processes cards punched in 10 digit form, checks instructions for validity, giving listings and condensed cards as output.

MACHINE
J702

Ifapes__Model...Other __

Printer TRC Drum

Card Reader (Optional) Punch (Optional)

PROGRAM LANGUAGE: Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction Label

Subroutine

CONTRIBUTED BY:
Earl Althoff
Eastman Kodak Company

January 1958, Bulletin 55 - 71

705 CUSTOMER CONTRIBUTION

Program Write-up Abstract

INDICATIVE CODE
EK-0004-0

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 Test 5; namely, to provide a means of checking 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 last 240 digits. Only 244 positions of accumulator 00 are available to 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 last 240 digits. The program occupies 643 memory positions.

CONTRIBUTED BY:
E. Althoff, Eastman Kodak
Rochester, New York

GUIDE PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EK-0004-0

PROGRAM NAME
EKACTO - 10 DIGIT CONVERSION

PURPOSE
Enable programmer to write and store in actual as 10 digits (Ex: RAD 02 25519). The routine processes cards punched in 10 digit form, checks instructions for validity, giving listings and condensed cards as output.

MACHINE
J702

Ifapes__Model...Other __

Printer TRC Drum

Card Reader (Optional) Punch (Optional)

PROGRAM LANGUAGE: Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction Label

Subroutine

CONTRIBUTED BY:
Earl Althoff
Eastman Kodak Company

January 1958, Bulletin 55 - 71

705 CUSTOMER CONTRIBUTION

Program Write-up Abstract

INDICATIVE CODE
EK-0004-0

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 Test 5; namely, to provide a means of checking 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 last 240 digits. Only 244 positions of accumulator 00 are available to 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 last 240 digits. The program occupies 643 memory positions.
GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EQ-005-0

PROGRAM NAME
ALTERED MEMORY PRINT

PURPOSE: To print out, in indexed form, the contents of memory which have been changed since the initial loading of a given program.

MACHINE: 702 705 X Model 1 or II Other (Specify)
Printer X TRC X Drum
Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual (Specify)
Other

PROGRAM TYPE: Complete Program X
Macro-Instruction Label
Subroutine Label

CONTRIBUTED BY:
Arthur Rosenzweig
James M. Kappos
Equitable Life Assurance Society of the U.S.
393 Seventh Avenue
New York 1, New York

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EQ-006-0

PROGRAM NAME
SELECTIVE TAPE PRINT

PURPOSE: To print directly, or to write on a tape for subsequent printing, all or selected records of specified tapes.

MACHINE: 702 705 X Model 1 or II Other (Specify)
Printer X TRC X Drum
Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual (Specify)
Other

PROGRAM TYPE: Complete Program X
Macro-Instruction Label
Subroutine Label

CONTRIBUTED BY:
B. Gordon
The Equitable Life Assurance Society of the United States
393 Seventh Avenue
New York 1, N. Y.

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EQ-007-0

PROGRAM NAME
SEQUENCE CHECK

PURPOSE: Sequence-check a file of variable-length tape records and/or delete records which exceed a given length.

MACHINE: 702 705 X Model 1 or II Other (Specify)
Printer X TRC X Drum
Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual (Specify)
Other

PROGRAM TYPE: Complete Program X
Macro-Instruction Label
Subroutine Label

CONTRIBUTED BY:
Milton P. Persily
Equitable Life Assurance Society of the United States
393 Seventh Avenue
New York 1, N. Y.

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EQ-009-0

PROGRAM NAME
Tic-Tac-Toe

PURPOSE: Demonstration of logical ability and speed of the 705.

MACHINE: 702 705 X Model 1 or II Other (Specify)
Printer X TRC X Drum
Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual X Other (Specify)

PROGRAM TYPE: Complete Program X
Macro-Instruction Label
Subroutine Label

CONTRIBUTED BY:
Milton P. Persily
The Equitable Life Assurance Society of the United States
393 Seventh Avenue
New York 1, N. Y.

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EL-002-0

PROGRAM NAME
Time Series Routine

PURPOSE: To calculate statistical indices of average, variance, and standard deviation on time series data. A visual interpretation of the data is provided by plotting each point sequentially as a plus or minus deviation from the average. A cell count is shown to indicate the distribution profile.

(Continued on next page)
MACHINE: 702 705 1 Other (Specify)
Printer 1 TRC Drum 1
Card Reader 1 700 Other

PROGRAM LANGUAGE: Autocoder Symbolic Actual (Specify)
Other Autocoder A

PROGRAM TYPE: Complete Program X

Macro-instruction Label
Subroutine Label

CONTRIBUTED BY:
F. R. Pfaff
Esso Standard Oil Company
Linden, N. J.

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EZ-003-0

PROGRAM NAME
Stepwise Regression

PURPOSE:
To develop an equation expressing a dependent variable, Y, as a function of as many as 50 independent variables, multiply regression analysis.

MACHINE: 702 705 1 Other (Specify)
Printer 1-717 TRC Drum 1
Card Reader 1 700 Other

PROGRAM LANGUAGE: Autocoder Symbolic Actual (Specify)
Other Autocoder A

PROGRAM TYPE: Complete Program X

Macro-instruction Label
Subroutine Label

CONTRIBUTED BY:
W. E. Clayton
D. M. Smith
Esso Standard Oil Company
Linden, New Jersey

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EZ-005-0

PROGRAM NAME
Product Inverse Linear Programming

PURPOSE:
To calculate optimum solutions for problems involving up to 99 linear constraints and 120 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 1 Other (Specify)
Printer 1-717 TRC Drum 1
Card Reader 1 700 Other

PROGRAM LANGUAGE: Autocoder Symbolic Actual (Specify)
Other Autocoder A

PROGRAM TYPE: Complete Program X

Macro-instruction Label
Subroutine Label

CONTRIBUTED BY:
R. W. Schrage
D. M. Smith
W. E. Zieman
Esso Standard Oil Company
Linden, New Jersey

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EZ-004-0

PROGRAM NAME
Matrix Inversion

PURPOSE:
To invert a matrix and/or to solve simultaneous linear equations.

MACHINE: 702 705 1 Other (Specify)
Printer 1-717 TRC Drum 1
Card Reader 1 700 Other

PROGRAM LANGUAGE: Autocoder Symbolic Actual (Specify)
Other Autocoder A

PROGRAM TYPE: Complete Program X

Macro-instruction Label
Subroutine X Label

CONTRIBUTED BY:
Ecco Standard Oil Company - M. H. Grosz
15 West 51 St., N. Y. C.
International Business Machines Corp. - B. P. Daughoux
New York City

(August 1957, Bulletin 00 - 129)
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
HB-001-0

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: 705

705 X

Model 1 or 2

Other

Card Reader

700

Other

Programmer

和其他

PROGRAM LANGUAGE: Autocoder

Symbolic

Actual

Other

PROGRAM TYPE: Complete Program

Macro-Instruction

Label

Subroutine

Label

CONTRIBUTED BY:
W. M. Harp
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
IB 0002

PROGRAM NAME
Card Image

MACHINE SPECIFICATIONS:
20,000 or 40,000 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 13, 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 13, 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
IB 0003

PROGRAM NAME
Flow Chart Listing From Assembly Program

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.

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. 260 may connect one location on a page to a lower location on the same page (backward transfers).
3. 999 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
IB 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, d(M), and the number of equations to be solved, N (550).

On line print-out of solutions is provided and optional print-out of inverse matrix.

CONTRIBUTED BY:
D. Loprer, IBM, Birmingham

705 CUSTOMER CONTRIBUTION

Program Write-Up Abstract

INDICATIVE CODE
IB 0007

PROGRAM NAME
Tape Duplication

MACHINE SPECIFICATIONS:
20,000 or 40,000 Position 705

714 Tape Control Unit

PURPOSE:
To provide exact duplication of one tape from another.

RESTRICTIONS:
1. Record length may not exceed 19,785 characters for a 20,000 position 705, nor may it exceed 39,785 characters for a 40,000 position 705.
2. Records to be duplicated must not contain the following sequence of five characters: END0 which is used in determining end of record. If this sequence appears in records, any desired five characters may be substituted for it.
GENERAL DESCRIPTION:

The input tape for this program is mounted on tape unit OZOO; output is written on tape OZO 1. 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: IB 0009

PROGRAM NAME: 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, 0001 and December 31, 9999. This program will also compute the given date for the following holidays, both fixed and variable,

- New Years Day
- Lincoln's Birthday
- St. Valentine's Day
- Washington's Birthday
- April Fools Day
- Memorial Day
- Independence Day
- Columbus Day
- Halloween
- Veteran's Day
- Christmas Day

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, preceding the year that Lincoln or Washington was born, in how many 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 Raiffa

705 CUSTOMER CONTRIBUTION

Program Write-up Abstract

INDICATIVE CODE: IB 0010

PROGRAM NAME: 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:

\[ b + I = (b + I) \times 100 \]

and

\[ b \times I = 99 \]

where \( b \) is number of matrix rows and \( I \) is number of column vectors.

CONTRIBUTED BY:

Imperial Oil Limited
Toronto, Canada

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE: IB 0011

PROGRAM NAME: 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 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. Bemer, W. M. Selden and A. S. Petroulakis, IBM, WHQ

Distribution No. 8
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE  PROGRAM NAME
001-0 Address Listing

PURPOSE:
To produce an actual address listing following a 705 assembly of programs written in either Autocoder, Print l, or Symbolic language. The program reads the listing tape produced by the assembler and prepares a sorted table of address-location references - which is written out on the listing tape following the tape marks.

MACHINE: 705

PROGRAM LANGUAGE: Autocoder

CONTRIBUTED BY:
Lockheed Aircraft Corporation
California Division
Burbank, California

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE  PROGRAM NAME
PG-001 Simulation of the IBM 650

MACHINE SPECIFICATIONS:
40K IBM 705 with card reader & card punch. (Simple additional modifications permit tape input and output).

PURPOSE:
To modify the program for simulating the IBM 650 on the IBM 705 to handle any 650 program written for the basic card 650 with alpha device. The write-up and program deck for the original simulator (reference #1) so as to take advantage of the expanded memory of the 40K version of the 705 and thus gain an increase in speed.

REstrictions:
Will handle any 650 program written for the basic card 650 with alpha device. The write-up and program deck cover only the modifications.

GENERAL DESCRIPTION:
A program already exists (reference #1) which simulates the IBM 650 on the 20K 705. Since the 650 Magnetic Drum storage contains 2019 digits, each 10-digit 650 word had to be converted to a packed 7-digit 705 word to allow space for the simulation program itself. This modification was written to simulate the 650 drum in the 20K upper memory of a 40K IBM 705. Elimination of the PAC & UNPAC routines formerly necessary has increased the speed of the simulation of the 650 run at speeds approximately the same as for the 650 itself.

CONTRIBUTED BY:
Procter & Gamble and the IBM, Cincinnati Office

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE  PROGRAM NAME
PG-003 Simulation of the IBM 650

PURPOSE:
To look for, diagnose, and correct where possible 0901, 0902, and 0903 errors. Handles end of file conditions in a specified manner as outlined in the program description. Includes flip-flop of tapes. Can be used with some or all of the following on line: Any number of 754 tapes, drums, 217 printer, punch, and card reader. Takes care of RD, RD 01, WR, WR 01, WTM, and RWW, but not WRB.

MACHINE: 705

PROGRAM LANGUAGE: Autocoder

CONTRIBUTED BY:
The Northwestern Mutual Life Insurance Company
720 East Wisconsin Avenue
Milwaukee 2, Wisconsin

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE  PROGRAM NAME
PG-004 Generalized Transfer Any Routine

PURPOSE:
To look for, diagnose, and correct where possible 0001, 0002, and 0003 errors. Handles end of file conditions in a specified manner as outlined in the program description. Includes flip-flopping of tapes. Can be used with some or all of the following on line: Any number of 754 tapes, drums, 217 printer, punch, and card reader. Takes care of RD, RD 01, WR, WR 01, WTM, and RWW, but not WRB.

MACHINE: 705

PROGRAM LANGUAGE: Autocoder

CONTRIBUTED BY:
Richard Bullis, IBM
Northwestern Mutual Life Insurance Company
720 East Wisconsin Avenue
Milwaukee 2, Wisconsin

Distribution No. 6

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE  PROGRAM NAME
PG-005 Simulation of the IBM 650

MACHINE SPECIFICATIONS:
40K IBM 705 with card reader & card punch. (Simple additional modifications permit tape input and output).

PURPOSE:
To modify the program for simulating the IBM 650 on the IBM 705 (reference #3) so as to take advantage of the expanded memory of the 40K version of the 705 and thus gain an increase in speed.

REstrictions:
Will handle any 650 program written for the basic card 650 with alpha device. The write-up and program deck cover only the modifications.

GENERAL DESCRIPTION:
A program already exists (reference #1) which simulates the IBM 650 on the 20K 705. Since the 650 Magnetic Drum storage contains 2019 digits, each 10-digit 650 word had to be converted to a packed 7-digit 705 word to allow space for the simulation program itself. This modification was written to simulate the 650 drum in the 20K upper memory of a 40K IBM 705. Elimination of the PAC & UNPAC routines formerly necessary has increased the speed of the simulation of the 650 run at speeds approximately the same as for the 650 itself.

CONTRIBUTED BY:
Procter & Gamble and the IBM, Cincinnati Office

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE  PROGRAM NAME
PG-006 Generalized Transfer Any Routine

PURPOSE:
To look for, diagnose, and correct where possible 0001, 0002, and 0003 errors. Handles end of file conditions in a specified manner as outlined in the program description. Includes flip-flopping of tapes. Can be used with some or all of the following on line: Any number of 754 tapes, drums, 217 printer, punch, and card reader. Takes care of RD, RD 01, WR, WR 01, WTM, and RWW, but not WRB.

MACHINE: 705

PROGRAM LANGUAGE: Autocoder

CONTRIBUTED BY:
The Northwestern Mutual Life Insurance Company
720 East Wisconsin Avenue
Milwaukee 2, Wisconsin

Distribution No. 6

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE  PROGRAM NAME
PG-007 Simulation of the IBM 650

MACHINE SPECIFICATIONS:
40K IBM 705 with card reader & card punch. (Simple additional modifications permit tape input and output).

PURPOSE:
To modify the program for simulating the IBM 650 on the IBM 705 (reference #3) so as to take advantage of the expanded memory of the 40K version of the 705 and thus gain an increase in speed.

REstrictions:
Will handle any 650 program written for the basic card 650 with alpha device. The write-up and program deck cover only the modifications.

GENERAL DESCRIPTION:
A program already exists (reference #1) which simulates the IBM 650 on the 20K 705. Since the 650 Magnetic Drum storage contains 2019 digits, each 10-digit 650 word had to be converted to a packed 7-digit 705 word to allow space for the simulation program itself. This modification was written to simulate the 650 drum in the 20K upper memory of a 40K IBM 705. Elimination of the PAC & UNPAC routines formerly necessary has increased the speed of the simulation of the 650 run at speeds approximately the same as for the 650 itself.

CONTRIBUTED BY:
Procter & Gamble and the IBM, Cincinnati Office
GUIDE PROGRAM WRITE-UP ABSTRACT

PROGRAM TYPE: Complete Program

Program Write-Up Abstract

INDICATIVE CODE
PG:004-0

PROGRAM NAME
CHECK TAPE SETTINGS

PURPOSE:
To check that one and only one tape unit is dialed to the unit position of each designated input and output tape. Types "Check Tape Settings" and halts in case of duplicate settings; stops at I/O No Response if no tape is dialed to one of the designated tape addresses.

MACHINE:
705 X Model: I or II Other (Specify)

Card Reader 760 Other Punch (optional)

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual

CONTRIBUTED BY:
Edward B. Berninger Procter & Gamble

January 1958, Bulletin 55 - 91

GUIDE PROGRAM WRITE-UP ABSTRACT

PROGRAM TYPE: Complete Program

Program Write-Up Abstract

INDICATIVE CODE
PG:005-0

PROGRAM NAME
TRANSPORTATION PROBLEM

PURPOSE:
To solve the "Transportation Problem", a special case of linear programming. The program can accommodate matrices with M x N  700, 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 56 minutes.

MACHINE:
705 X Model: I or II Other (Specify)

Card Reader 760 Other Punch (optional)

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual

CONTRIBUTED BY:
S. Hickenlooper, D. W. Grace, E. B. Berninger Procter & Gamble

January 1958, Bulletin 58 - 59

GUIDE PROGRAM WRITE-UP ABSTRACT

PROGRAM TYPE: Complete Program

Program Write-Up Abstract

INDICATIVE CODE
PG:006-0

PROGRAM NAME
BINARY TABLE SEARCH

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 BRSCH macro in a program and stored in an in-line record area. Arguments can be up to 79 characters long and functions up to 245, 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:
705 X Model: I or II Other (Specify)

Card Reader 760 Other Punch (optional)

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual

CONTRIBUTED BY:
Richard B. Thomas, Procter & Gamble

April 1958, Bulletin 57 - 53
GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-008-0

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: 705
705 X Model I or II Other

Printer
TRC
Drum

Card Reader
760 Other

PROGRAM LANGUAGE:
Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

CONTRIBUTED BY:
Richard B. Thoman
Procter and Gamble

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-009-0

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 255 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: 705
705 X Model I or II Other

Printer
TRC
Drum

Card Reader
760 Other

PROGRAM LANGUAGE:
Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

CONTRIBUTED BY:
William H. Grover
Procter and Gamble

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-010-0

PROGRAM NAME
Move Variable, Grouped Fields

PURPOSE:
To move a group of fields which are set up for high-speed transmission. The number of fields can vary from group to group and the size of each field can be variable. The method used is described on p. 3-4 of 702/705 Bulletin 20, Dec. 1956.

MACHINE: 705
705 X Model I or II Other

Printer
TRC
Drum

Card Reader
760 Other

PROGRAM LANGUAGE:
Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

CONTRIBUTED BY:
The Procter & Gamble Company

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-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 90 macros assembled. The change requires only three patch cards which overlay part of the present routine.

MACHINE: 705
705 X Model I, II Other

Printer
TRC
Drum

Card Reader
760 Other

PROGRAM LANGUAGE:
Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

CONTRIBUTED BY:
William F. Reiland
Procter and Gamble

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SB-001-0

PROGRAM NAME
SORT 58 SB-001-0

PURPOSE:
To sort fixed or variable length records via TCU.

MACHINE: 705
705 X Model I, II Other

Printer
TRC
Drum

Card Reader
760 Other

PROGRAM LANGUAGE:
Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

CONTRIBUTED BY:
The Procter & Gamble Company

* patches for existing program

Distribution No. S

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SB-001-0

PROGRAM NAME
SORT 58

PURPOSE:
To sort fixed or variable length records via TCU.

MACHINE: 705
705 X Model I, II Other

Printer
TRC
Drum

Card Reader
760 Other

PROGRAM LANGUAGE:
Autocoder Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

CONTRIBUTED BY:
Procter and Gamble

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-018-0

PROGRAM NAME
More Variable, Grouped Fields

PURPOSE:
To move a group of fields which are set up for high-speed transmission. The number of fields can vary from group to group and the size of each field can be variable. The method used is described on p. 3-4 of 702/705 Bulletin 20, Dec. 1956.
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SB-002-0

PROGRAM NAME
Analyzer

PURPOSE:
To produce an edited listing in several optional sequences, cross referencing
the data available in an Autocoder Assembly Listing Tape.

MACHINE: 702 ___ _ 705 X Model II Other (Specify)

Card Reader 760 ___ Other

PROGRAM LANGUAGE: Autocoder ___ Symbolic ___ Actual ___

Other (Specify)

PROGRAM TYPE: Complete Program __

Macro-Instruction ___ Label ___

Subroutine ___ Label ___

CONTRIBUTED BY:
Directorate of Ballistic Missiles
San Bernardino Air Materiel Area
San Bernardino, California

Written by: Faye Redus

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SB-005-0

PROGRAM NAME
Tape Input/Output

PURPOSE:
To present a complete set of operations for all functions involving on line
705 tape units controlled by TRC and TCU. Macro instructions and sub­
routines are available for tape read, write, read-while-write, control
operations, housekeeping, label treatment, blocking/deblocking of grouped
records, and transfer - any analysis. A utility routine provides for restart if the checkpoint options are used.

MACHINE: 702 ___ _ 705 X Model II Other (Specify)

Card Reader 760 ___ Other

PROGRAM LANGUAGE: Autocoder ___ Symbolic ___ Actual ___

Other (Specify)

PROGRAM TYPE: Complete Program __

Macro-Instruction ___ Label ___

Subroutine ___ Label ___

CONTRIBUTED BY:
Directorate of Ballistic Missiles
San Bernardino Air Materiel Area
San Bernardino, California

Written by: K. Lantz, L. Cohn, T. Carstens, C. Buss, O. Evans, D. Fisher

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SB-006-0

PROGRAM NAME
Mem Print Analyzer

PURPOSE:
Rearranges instruction data extracted from the MEM PRINT 75 output tape
and produces a listing showing all instruction addresses cross referenced to
memory locations.

MACHINE: 702 ___ _ 705 X Model II Other (Specify)

Card Reader 760 ___ Other

PROGRAM LANGUAGE: Autocoder ___ Symbolic ___ Actual ___

Other (Specify)

PROGRAM TYPE: Complete Program __

Macro-Instruction ___ Label ___

Subroutine ___ Label ___

CONTRIBUTED BY:
Directorate of Ballistic Missiles
San Bernardino Air Materiel Area
San Bernardino, California

Written by: C. Kubik

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SL-001-9

PROGRAM NAME
SQCOTT Tape Test System

PURPOSE:
To reduce machine time required for testing, and produce test output shortly
after each testing session.

MACHINE: 702 ___ _ 705 X Model II Other (Specify)

Card Reader 760 ___ Other

PROGRAM LANGUAGE: Autocoder ___ Symbolic ___ Actual ___

Other (Specify)

PROGRAM TYPE: Complete Program __

Macro-Instruction ___ Label ___

Subroutine ___ Label ___

CONTRIBUTED BY:
Standard Oil Company (Indiana)
Chicago, Illinois

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SP-001-0

PROGRAM NAME
Tape Characteristics

PURPOSE:
To prepare a listing of tape capacity, and passing speed in minutes, for various
record lengths, and for 727, 729.2 and 729.4 tape drives, with both high and low
recording density for 729 units.

(Continued on next page)
The generalized routines use three tapes. All other tape requirements depend upon the running program.

CONTRIBUTED BY:
H. E. Peabody, IBM, Atlanta, Georgia
Assignee to Southern Railway

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SR-005-0

PROGRAM NAME
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 typewriter sheet.

Routine is set up for program input on card reader but is easily modified for program input on tape.

MACHINE: 702 __ 705 X Model II Other (Specify)

Tapes: 10 Printer, 1:73:0 TRC Drum

Card Reader: 760 Other

PROGRAM NAME: Tape Operation, Tape Label and Trailer Checking

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
SR-006-0

PROGRAM NAME
Generalized T65 ROUTINE

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 II Other (Specify)

Tapes: * Printer, 1:73:0 TRC Drum (Specify)

Card Reader: 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual X (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction X Label _ See write-up

Subroutine X Label SCRAPS, LABTR

CONTRIBUTED BY:
Southern Railway Company - F. P. Ludlow, Jr.
15th and F 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 C28-5031, is assumed.

**MACHINE**: 702

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual

**CONTRIBUTED BY**: A. F. Rundquist
Department of the Army
TAGOC, Data Processing Branch
Washington, D.C.

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**
XE - 001 - 0

**PROGRAM NAME**
Sort 54 Technique of Modification of Phase III

**PURPOSE**
To change the assignment routine of Sort 54 to use the file size on a control card as a factor in creating the fastest possible sort and to automatically set up over maximum sorts.

**MACHINE**: 702

**PROGRAM LANGUAGE**: Autocoder, Symbolic, Actual

**CONTRIBUTED BY**: A. F. Rundquist
Department of the Army
TAGOC, Data Processing Branch
Washington, D.C.
IBM 0709 LIBRARY ABSTRACT

0709 368501109 AVAILABLE PRIOR TO JANUARY 1962

ASSEMBLY PROGRAM FOR T&E IBM 709.

0709 423427509 AVAILABLE PRIOR TO JANUARY 1962

THE CONTROL RECORD FOR THE FIRST PASS.

0709 445516709 AVAILABLE PRIOR TO JANUARY 1962

THE CONTROL RECORD FOR THE SECOND PASS.

0709 505037009 AVAILABLE PRIOR TO JANUARY 1962

TAPE COMPARE FOR THE 709.

0709 505037009 AVAILABLE PRIOR TO JANUARY 1962

TAPE CURRENCY FOR THE 709/0CTAL PRINT.

0709 505037009 AVAILABLE PRIOR TO JANUARY 1962

PRINTS RECORDS OR FILES, ON LINE OR WRITES TAPE AS FOR OFF
LINE PRINT. BINARY CONTROL CARD, WILL READ MORE THAN ONE
CONTROL CARD, WILL PRINT A SELECTED SEQUENCE OF WORDS FROM
EACH RECORD.

0709 507690509 AVAILABLE PRIOR TO JANUARY 1962

FLOATING POINT ARCCOSINE SUBROUTINE

MUST BE FOLLOWED BY 10 MSG, TIMING 4.0 MS, 9 LOC.
CORR.S/1499., ADEN/1369.

0709 507690509 AVAILABLE PRIOR TO JANUARY 1962

FLOATING POINT ARCCOSINE SUBROUTINE

BASED ON 704 PROGRAM L.A 925, TIMING ABOUT 2.0 MS ERROR ...
AT MOST 190-6, ABSOLUTE FOR LOG SMALLER THAN 1, RELATIVE
OTHERWISE.

0709 510598019 AVAILABLE PRIOR TO JANUARY 1962

COMPACT ATTACHED PROGRAM, J709 PROGRAM.
PRINTS ONE TO TWELVE RED WORDS IN ONE LINE, TAKES 65 CELLS
PLUS 27 OF COMPROM, DELAYS UNTIL PRINTING IS COMPLETED.

0709 510598019 AVAILABLE PRIOR TO JANUARY 1962

FLOATING POINT NATURAL LOGARITHM

PROVIDES COMMUNICATION BETWEEN THE OPERATOR, THE PROGRAM
AND THE MACHINE FOR CONNECTING, DISCONNECTING, ASSIGNING
AND DISASSIGNING MAGNETIC TAPE.

0709 510598019 AVAILABLE PRIOR TO JANUARY 1962

ASSEMBLY PROGRAM FOR THE IBM 709
THE TAPE WRITING ROUTINE
THE CONTROL RECORD FOR THE FIRST PASS
THE FIRST PASS
THE CONTROL RECORD FOR THE SECOND PASS
THE SECOND PASS
THE CALL CARD FOR THE ASSEMBLER.
FLOW CHART ANALYSIS BY BOOLEAN MATRIX MANIPULATION

DETECTS ERRORS IN CONNECFIVITY OF FLOW CHARTS UP TO 500 BOXES
BY TREATING A FLOW CHART AS A BOOLEAN MATRIX. WILL ALSO DETECT
ERRORS IN SUBPROGRAMS IN THE FLOW CHART IF INFORMATION ABOUT
LOCATION BEGIN AND RETURN MACROS IS PROVIDED. SUPSOLVER, A ROUTINE
TO COMPUTE THE NATURAL LOGARITHM OF A NORMALIZED FLOATING-
POINT NUMBER, WILL BE REQUIRED. FOR USE IN THIS ROUTINE, 102 LOCATIONS
ARE REQUIRED, IF USE IS MADE OF REPLACEMENT OF SQUAD.
EQUATOR-ECLIPTIC ROTATION

FORTRAN WRITE-UP OF RW ROBL. ROTATE A GIVEN VECTOR ABOUT DECKS OR IN PREVIOUSLY COMPILED OBJ. DECKS. OTHER FEATURES DENT VARIABLES FOR 5000 OBSERVATIONS. CORR/ 1015, 1106 EQUATIONS BY LIMITED-INFORMATION, TWO-STAGE LEAST-SQUARES1 ESTIMATES THE COEFFICIENTS OF A SYS. OF LINEAR STOCHASTIC

PEST IS AN ASSEMBLY ROUTINE FOR USE ON THE IBM 709

ECUATOR-ECLIPTIC ROTATION

UP TO 70 ECUATS. IN 70 DEPENDENT VARIAALES ANO 70 INDEPEN­

MAY BE R:EAO FROM READER OR TAPE. MAXIMUM PERIOD- 50

IBM 0709 FORT.

IBM 0709 978WOI0F AVAILABLE PRIOR TO JANUARY 1962

IBM 956LCPSN AVAILABLE PRIOR TO JANUARY 1962

IBM 954RWFODL AVAILABLE PRIOR TO JANUARY 1962

IBM 961PPPEST AVAILABLE PRIOR TO JANUARY 1962

IBM 9.97MLCVRT AVAILABLE PRIOR TO JANUARY 1962

IBM 995FDEDIT AVAILABLE PRIOR TO JANUARY 1962

IBM 997MDRTF AVAILABLE PRIOR TO JANUARY 1962

IBM 990RWLE4F AVAILABLE PRIOR TO JANUARY 1962

IBM 985RWBF8F AVAILABLE PRIOR TO JANUARY 1962

IBM 990WD093 AVAILABLE PRIOR TO JANUARY 1962

IBM 999LO039 AVAILABLE PRIOR TO JANUARY 1962

IBM 1000960 AVAILABLE PRIOR TO JANUARY 1962

IBM 1001NA8610 AVAILABLE PRIOR TO JANUARY 1962

IBM 1001N4600 AVAILABLE PRIOR TO JANUARY 1962

IBM 1001N46010 AVAILABLE PRIOR TO JANUARY 1962
RECORDS, 1-6 SCATTERED CONTROL FIELDS, INTERRUPT FEATURES

CONTROL CARDS ARE USED TO SPECIFY ALL SORT PARAMETERS.

RECORDS ARE SUMMARIZED OUT.

BINARY INPUT OF N Results, VARIABLE OR FIXED LENGTH BLOCKED

709/7090 GENERALIZED VARIABLE LENGTH RECORD SORT

OPTIONAL INPUT AND OUTPUT LABELING. MINIMUM MACHINE REQUIREMENTS ARE REQUIRED.

RATIONAL APPROXIMATION METHOD INPUT IN AC-HQ OR FROM CORE,

DESIGNED TO ASSIST IN CHECKOUT OF PROGRAMS USING SUBROUTINES

FORTRAN-LOADING TO COPY MEMORY ON TO TAPE.

MODE Tapes ON ANY PORTIONS OF THE ITEMS. OPTIONAL CHECKSUM

FROM MO BW VIPP. AN ILLEGAL CALL WILL CAUSE ON-LINE INDICATION.

SCATTER EQUIVALENCE DECK TO BE ASSEMBLED WITH SCAT ROUTINES

SCAT EQUIVALENCE DECK TO BE ASSEMBLED WITH SCAT ROUTINES

TRANSMISSION.

FORTRAN MULTIPLE CORRELATION ANALYSIS PROGRAM

This program is for the statistical analysis of a set of

MOCES R1, R2, ..., RM WHERE Pi - R1, R2, ..., RM WHERE K IS THE

DEPENDENT VARIABLE, y1/2, y1/3, ..., y1/N ARE INDEPENDENT

VARIABLE FUNCTIONS, AND THE R VALUES ARE TO BE STATISTICALLY

ESTIMATED FROM THE DATA.

709 1120AFLDC

AVAILABLE PRIOR TO JANUARY 1962

ADDRESS LOCATION SUBROUTINE.

FINDS THE LOCATION OF ANY CONSTANT OR VARIABLE IN THE PROGRAM

WHERE MAY BE FIXED OR FLOATING, SUBSCRIPTED OR NOT.

SUBSCRIPTS MAY BE EXPRESSIONS OF STANDARD FORTRAN FORM.

FORTRAN MULTIPLE CORRELATION ANALYSIS PROGRAM

This program is for the statistical analysis of a set of

MOCES R1, R2, ..., RM WHERE Pi - R1, R2, ..., RM WHERE K IS THE

DEPENDENT VARIABLE, y1/2, y1/3, ..., y1/N ARE INDEPENDENT

VARIABLE FUNCTIONS, AND THE R VALUES ARE TO BE STATISTICALLY

ESTIMATED FROM THE DATA.

709 1121VNC

AVAILABLE PRIOR TO JANUARY 1962

FORTRAN MULTIPLE CORRELATION ANALYSIS PROGRAM

This program is for the statistical analysis of a set of

MOCES R1, R2, ..., RM WHERE Pi - R1, R2, ..., RM WHERE K IS THE

DEPENDENT VARIABLE, y1/2, y1/3, ..., y1/N ARE INDEPENDENT

VARIABLE FUNCTIONS, AND THE R VALUES ARE TO BE STATISTICALLY

ESTIMATED FROM THE DATA.

709 1135NWIPP

AVAILABLE PRIOR TO JANUARY 1962

VARmABLE INFORMATION PROCESSING PACKAGE

709-600 VIPP, LIKE 709UIPP, IS A COLLECTION OF SUBROUTINES

DESIGNED TO SERVE AS AN EFFICIENT GENERAL PURPOSE

DATA PROCESSING PACKAGE CORR.1178

709 1130B2WPM

AVAILABLE PRIOR TO JANUARY 1962

VIPP MERGER

SECOND PHASE OF A GENERAL PURPOSE SORTER. FIRST PHASE IS PI ON BKP. K2 MERGE VARIOUS LENGTH ITEMS ON ANY PORTIONS OF THE ITEMS, OPTIONAL CHECKSUM CONTROL TO GUARD AGAINST THE SORT, RECOVERY PROCEDURE, TAPE COUTNS FOR TAPE ERROR DIAGNOSIS. FAVORABLE TIMING. MAY BE RUN AS A SINGLE PHASE MERGER TO MERGE 2, 3 OR 4 SORTED FILES.

709 1137NWRUG

AVAILABLE PRIOR TO JANUARY 1962

VIPP BUG TRAP

DESIRED TO ASSIST IN CHECKOUT OF PROGRAMS USING SUBROUTINES FROM MO BW VIPP, AN ILEGAL CALL WILL CAUSE ON-LINE INDICATION OF THE CALL AND BUG LOCATIONS.

709 1137NWSYN

AVAILABLE PRIOR TO JANUARY 1962

VIPP SYNONYM DECK

SCAT COVERAGE DECK TO BE ASSEMBLED WITH SCAT ROUTINES USING ON VIPP.

709 1144NPDAT

AVAILABLE PRIOR TO JANUARY 1962

DOUBLE PRECISION FLOATING POINT ANCECIPATION SUBROUTINE

RATIONAL APPROXIMATION METHOD INPUT IN AC-HQ OR FROM CORE,

OUTPUT IN RADIAN, EITHER PRECISION VALUE OR CORRECTED FOR

QUADRATURE, DEPENDING ON OPTION CHosen. 256 LOCATIONS & 14 COM-

MUN'S ARE REQUIRED. SUCH AS NON IEPAD.

709 1159N4GOT

AVAILABLE PRIOR TO JANUARY 1962

709/7090 GENERALIZED VARIABLE LENGTH RECORD SORT

This generalized sort routine provides a 2-5-WAY MERGE, BOC

ON INPUT OF N DECS, VARIABLE OR FIXED LENGTH BLOCCED

RECORDS, 1-SCATTERED CONTROL FIELDS, INTERRUPT FEATURES, OPTIONAL INPUT AND OUTPUT LABELING. MINIMUM MACHINE REQUIREMENTS ARE REQUIRED.

MINIMUM MACHINE REQUIREMENTS ARE REQUIRED.

709 1160M5SHT

AVAILABLE PRIOR TO JANUARY 1962

RESTART PROGRAM FOR MOV SORT

USED TO RESTART A SORT AT THE BEGINNING OF ANY PHASE OR MERGE

PASS, RELOADS CHECKPOINT TAPE INTO CORE AND CHECKS THE TAPE

TRANSMISSION.
APWRC-SYNFAR

709 Nuclear Code

(1) Code Originated by:
The Martin Co. (Baltimore)

(2) Computer:
109 (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
grid. 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 lethargy
dependent bucklings.

(4) Restrictions or Limitations:
A 32K memory with ten tape units. For transport calculations, two or
three groups may be used, and P1, P2, P3, P9, and P16 calculations
may be made. The P16 calculation may not be done in cylindrical
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 Pl or SN Diff
   Flux or Adjoint in Slab, Cylinder, or Spherical Geometry",
   MND-C-2460, 1961.

(7) Material Available:
1. SYNFA-01 Binary Deck.
2. SYNFA-01 Tape (2 files).
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

FILE NUMBER L1.001

MASCOT (Modified Assembly System Converted to Tape)
Aaron C. Williams
IBM Corporation
340 Market Street
San Francisco II, California

Purpose: This program is a variation of the 1401 SPS - I system that uses magnetic tape to store intermediate results rather than punched cards.


Restrictions / Range: Reiteration is possible with MASCOT, and is necessary if the program to be assembled has over 260 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

FILE NUMBER L1.002

CARAT I
Aaron C. Williams & Jackson McElmell

Direct Inquiries to: Mr. Aaron C. Williams
IBM Corporation
340 Market Street
San Francisco II, 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 160 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 1401 with High - Low - Equal Compare, Register Feature, and High-Low-Equal compare, 3 Model 729 or 7330 Tape Drives.

IBM 1401 PROGRAM LIBRARY ABSTRACT

FILE NUMBER 1.1.004

CARAT II
Aaron C. Williams & Margery C. Rendahl

Direct Inquiries to: Aaron C. Williams & Margery C. Rendahl
IBM Corporation
340 Market Street
San Francisco II, 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 program, 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 65%. 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 List-Punch routines 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)

Additional Remarks: A companion program, CALL (Carat Assembled Logical Loader), is available for use with CARAT II. This program allows the user to load assembled programs directly from the CARAT output tape (7095). This makes it unnecessary to punch the object program until it is completely debugged. The CALL program also provides for patching.

IBM 1401 PROGRAM LIBRARY ABSTRACT

FILE NUMBER L1.005

MAST (Minneapolis Assembly of SPS Two)
Richard T. Firtko

Direct Inquiries to: Mr. Richard T. Firtko
Test Center Coordinator
IBM 1401 Test Center
200 Foshay Tower
Minneapolis 12, Minnesota

Purpose / Description: This program is a variation of the 1401 SPS II Assembly Program to use magnetic tape to store partly assembled output of PASS I rather than on punched cards. Punching will occur at the end of each iteration.

Method: Source language 1401 SPS

Restrictions / Range: Reiteration is possible with MAST, and necessary if program to be assembled has over 254 labels.

Storage Requirements: 4K minimum

Equipment Specifications: 4K Model C 1401 with no special devices, one tape unit. Additional core will allow faster assembly due to more labels processed per iteration.

IBM 1401 PROGRAM LIBRARY ABSTRACT

FILE NUMBER L1.006

FULL MAST (Full Minneapolis Assembly of SPS Two)
Richard T. Firtko

Direct Inquiries to: Mr. Richard T. Firtko
Test Center Coordinator
IBM 1401 Test Center
200 Foshay Tower
Minneapolis 12, Minnesota

Purpose / Description: This program is a variation of the 1401 SPS II Assembly program. It is completely automatic from input, through post list, and punching. Any reiterations will be performed automatically.

Method: Source language 1401 SPS

Restrictions / Range:
1. Will handle multiple programs stacked in reader for assembly.
2. Allows reassembly of previously assembled programs.
3. Sense switch selection of one per card or condensed output.

Storage Requirements: 4K minimum

Equipment Specifications: 4K or larger Model C 1401 with sense switches, read release, and 3 tape units. Writeup includes indication of minor changes that can be made to run without sense switches and read releases.

IBM 1401 PROGRAM LIBRARY ABSTRACT

FILE NUMBER L1.007

704 ASSEMBLY OF 1401 SPS PROGRAMS
R. Nelson

Direct Inquiries to: R. Nelson
IBM Applied Science
Albuquerque, New Mexico

Purpose / Description: To use the 704 to assemble 1401 SPS programs which include special features and revised mnemonic operating codes. Also, to be able to assemble 1401 programs before 1401 delivery.

Method: N/A

Restrictions / Range: No limit to the number of cards per program. There is a maximum of 200 symbols per program.

Storage Requirements: 8K or 32K

(Continued on next page)
IBM 1401 PROGRAM LIBRARY ABSTRACT

IBM 1401 Generalized Merge Program for Unblocked Records

J. E. Carreras & P. MacGregor

Purpose/Description: The purpose of this program is to lessen machine time required for generation and assembly of a program generated by the standard TRPG 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

CARD REPORT PROGRAM GENERATOR AND AUTOCODER ASSEMBLY

J. L. Dorsey

Direct Inquiries to: Mr. J. L. Dorsey
IBM Corporation
1071 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 TRPG 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.
1. Additional Remarks:

2. Session because


4. such as SPS type cards,

5. instruction cards will contain the length of the instruction, location to be

6. may be actual machine language or 4 digit addresses.

7. IBM

8. This area is cleared upon reading an end card. The correction loader may be

9. IBM 1401 PROGRAM LIBRARY ABSTRACT

10. F. E. Johnston

11. Albuquerque, New Mexico

12. Purpose:

13. Method:

14. Restrictions, Range:

15. Equipment Specifications:

16. No special features needed.

17. IBM 1401 PROGRAM LIBRARY ABSTRACT

18. CALL

19. Robert W. Heald

20. IBM

21. 140 Market Street

22. San Francisco 11, California

23. Purpose:

24. Method:

25. Restrictions, Range:

26. Equipment Specifications:

27. IBM 1401 PROGRAM LIBRARY ABSTRACT

28. TRICOM II

29. Dick Nichols


31. Dept. 72, Building 6

32. 4300 East 5th Avenue

33. Columbus 16, Ohio

34. Purpose:

35. Method:

36. Restrictions, Range:

37. Equipment Specifications:

38. IBM 1401 PROGRAM LIBRARY ABSTRACT

39. CARD REPRODUCING AND/OR LISTING PROGRAM FOR THE IBM 1401

40. (Continued on next column)
IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.006

THOMII

Purpose: FACTOR I is a program testing routine, which makes possible
controlled testing of any number of assembled card system 1401 Object Programs.

Method: All test output is identified by test program title on the printer, and in the
punch chucks. Stacker identification cards also indicate the number of the stacker
selected (NP, 4, 8/2). At the end of each program test an automatic storage
printout is obtained. Word marks, in 100 position increments is provided.

Restrictions, Range: Card programs only, with total memory not exceeding 3700
positions.

Storage Requirements: Factor is stored in the upper 300 positions of 4K 1401.

Equipment Specifications: 1401 4K, 1402, 1403

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.008

BINARY TAPE DUMP

F. J. X. Berckman
Westinghouse Electric Corporation
Steam Division, B. Plant, Room 410
Los Angeles, California

Purpose: This program provides the ability to dump a binary tape in octal
equivalents. The printed result is in word blocks with eight blocks to a line.

Method: Does not apply.

Restrictions, Range:

a) Variable length records acceptable. Maximum length recorded is
2250 characters or 366 words.
b) Single or double spacing available (SSB).

Storage Requirements: Not given.

Equipment Specifications: 1401 Standard Model C3, Two Tapes, column binary,
advanced programming package, High-Low-Equal Compare, Sense Switches (optional).

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.009

ZIP (Instant Printing)

Keith Swan
Southern California
Los Angeles 57, California

Purpose/Description: A utility load and go program for listing cards at a rate
of 650 lines per minute.

Method: Source language SPS

Restrictions/Range: 10 fields of any size can be listed. Field 10 can be accu­
ulated up to 12 positions and edited. Without control cards, an 80-80 list is
obtained. Card count, limited page headings, and page numbering are included,

Storage Requirements: N/A

Equipment Specifications: Read release and print buffer required for any 2K or
larger (401).

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.010

ESCAPE (Effortless System of Calculating and Printing Everything)

W, J. Teagarden

Direct Inquiries to: Southern Permanente Services
143 South Alvarado Street
Los Angeles 57, California

Purpose/Description: A utility program which provides rapid conversion of
SPS, 442, and 554 jobs to the 1401. This load and go program also may be used
to reproduce cards as well as gang punching, selective reproducing, sequence
numbering, listing or combinations of these operations. Combines the functions
of the previously published Card Reproducing and/or Listing Program (1,4.003)
and BANG I and II (16,1.002) without the restrictions of BANG I and II.

Method: Source language SPS

Restrictions/Range: Three separate routines (or two card routines and end-of­
file routines) may be developed. The effective working storage of the object pro­
gram is comprised of 20 counters and 20 storage units of ten positions each.
Multiplication and division can be executed only from counters.

Storage Requirements: Approximately 1800 positions of core are available to
build the three routines of 1,000 positions, 500 positions and 300 positions.

Equipment Specifications: 4K 1401. Punch feed read, multiply-divide and
high-low-equal compare features are required if program is completely used.
IBM 1401 PROGRAM LIBRARY ABSTRACT

FITS (Fourteen-O-one Input-output Tape-control System)
Robert E. Engelson
IBM Corporation
655 North Street
New Haven, Connecticut

Purpose/Description: This program supplies Open, Close, Get, and Put subroutines to users awaiting the full 1401 package for 1401 Autocoder. In addition, it supplies the advantages of an ICG compatible package to users who are unable to assemble Autocoder due to their systems configuration (less than 4 tape drives).

Method: FITS has been written in two source languages, aimed at the two groups mentioned in the "Purpose" paragraph. FITS I is written in 1401 Autocoder. FITS 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 is required.
2. Writing is in the Move Mode only.
3. Header and trailer labels are always written on the output files.
4. Input files are acceptable with or without header labels.
5. The FITS subroutines provide the following:
   A. Open:
      1. Input File: Checks file ID number and reel number.
      2. Output File: Checks creation date and retention cycle.
      3. Write Output header label.
   B. Get
      Places the next record in a work area for use by the program. All tape reading, deblacking, error routines and end of reel conditions are taken care of by the subroutine.
   C. Put
      Moves each record sequentially from a work area to a blocking area, automatically writing to 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 reel condition occurs.
   D. Close:
      Processes the end of file trailer label and removes the tape from use.

Storage Requirements: Approximately 1700 positions.

Equipment Specifications: High-Low-Equal Compare

IBM 1401 PROGRAM LIBRARY ABSTRACT

SCOOP I and II
Robert E. Engelson & Louis P. Poulin
IBM Corporation
1115 - 15th Street
Sacramento, California

Purpose/Description: To provide a simple method of converting 90 column input cards to 80 column output cards in ascending sequence.

Method: The user of SCOOP specifies in Column Control Cards each column to be translated FROM and TO. A Translation Table control card permits complete control over character translation. The user must program his own output routine and assemble it with SCOOP. Program Exit and Entry points have been provided for this purpose.

Restrictions/Range: Field tests and actual customer conversion usage have proven that undelimited 90 column round hole cards can be accurately read in a 1402 read feed when it is properly adjusted for normal 80 column card reading. Vertical 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 be included as part of the output routine.

Storage Requirements: 4,000 positions of storage

(Continued on next column)
IBM 1401 PROGRAM LIBRARY ABSTRACT

1401 CORE PRINTOUT ROUTINE - VARIABLE

Purpose/Description: To print the contents of core storage in a format useful for debugging. This program performs the following operations:

1. Prints the contents of the print band.
2. Prints the contents of the index registers.
3. Prints the contents of core storage beginning with location 3000.
4. Prints the contents of the print band.

Prints the contents of core storage to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Method: N/A

Restrictions/Range: By rotating the control card you designate the amount of core to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Storage Requirements: N/A

Equipment Specifications: IBM 1401 Model D, E, or F; Advanced Programming Package.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.018

File Number 1.4.016

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.017

File Number 1.4.019

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.020

 Ibrahim Corporation
414 Grand Avenue
Oakland 10, California

Purpose/Description: To re-read or re-write tape records when errors occur using a minimum amount of storage (576 positions).

Method: Source language 1401.SPS (Continued on next column)

Restictions/Range: No more than 100 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: 3000 positions.

Equipment Specifications:
- 4000 core-storage positions
- IBM 1401 Model D, E, or F
- Advanced Programming Package
- High-Low-Equal Compare
- Sense Switches

Additional Remarks: The speed is 100 cards per minute for unmatched cards, 400 cards per minute for matched cards.

UC TDOP, TAPE TO PRINTER OR PUNCH

Direct Inquiries to: Art Christopher

Purpose/Description: To print the contents of core storage beginning with location 3000. Prints the contents of the index registers.

Method: Source language 1401.SPS (Continued on next column)

Restrictions/Range: Noise records are not tested. The only alternatives are re-writing and re-reading.

Storage Requirements: 576 positions.

Equipment Specifications:
- IBM 1401 Model D, E, or F
- Advanced Programming Package

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.019

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.017

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.018

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.016

SD 1403 Search Program - Card Version

Direct Inquiries to: Fred G. Stockton

Purpose/Description: This program searches a deck of IBM cards (library deck), the cards within meet any one of a number of criteria. The criteria are specified in a simple code on set-up cards prefixed to the library deck. Matched cards are counted for the criterion 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 report of the card count for each criterion is printed.

Method: This program searches and prints records maintaining vertical alignment of equivalent fields from record to record and block to block which avoids the staggered print pattern associated with most tape print programs. This program handles both variable and constant length, single and block records whose length may be determined on tape. No parameters are required as the program is completely generative. A count representing the actual position of the last character of each line printed is maintained on the right margin - print positions 177-182. 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 multiple reels, and to emulate card-to-card 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.

Cleveland Datacenter
2424 Euclid Avenue
Cleveland 13, Ohio

Purpose/Description: To obtain printed or punched output from a file of tape records.

Method: N/A

Restrictions/Range: Requires Advanced Programming, Column Binary, (if Direct Inquiries to: Paul Tani

Purpose/Description: To print the contents of core storage in a format useful for debugging. This program performs the following operations:

1. Prints the contents of the print band.
2. Prints the contents of the index registers.
3. Prints the contents of core storage beginning with location 3000.
4. Prints the contents of the print band.

Prints the contents of core storage to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Method: N/A

Restrictions/Range: By rotating the control card you designate the amount of core to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Storage Requirements: N/A

Equipment Specifications: IBM 1401 Model D, E, or F; Advanced Programming Package.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.018

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.017

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.019

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.020

 Ibrahim Corporation
414 Grand Avenue
Oakland 10, California

Purpose/Description: To print the contents of core storage in a format useful for debugging. This program performs the following operations:

1. Prints the contents of the print band.
2. Prints the contents of the index registers.
3. Prints the contents of core storage beginning with location 3000.
4. Prints the contents of the print band.

Prints the contents of core storage to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Method: N/A

Restrictions/Range: By rotating the control card you designate the amount of core to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Storage Requirements: N/A

Equipment Specifications: IBM 1401 Model D, E, or F; Advanced Programming Package.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.018

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.017

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.019

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.020

 Ibrahim Corporation
414 Grand Avenue
Oakland 10, California

Purpose/Description: To print the contents of core storage in a format useful for debugging. This program performs the following operations:

1. Prints the contents of the print band.
2. Prints the contents of the index registers.
3. Prints the contents of core storage beginning with location 3000.
4. Prints the contents of the print band.

Prints the contents of core storage to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Method: N/A

Restrictions/Range: By rotating the control card you designate the amount of core to be printed. Any bands which are blanks without wordmarks are automatically skipped. On the printed form a groupmark will print as an asterisk. No distinction is possible between the two.

Storage Requirements: N/A

Equipment Specifications: IBM 1401 Model D, E, or F; Advanced Programming Package.
IBM 1401 PROGRAM LIBRARY ABSTRACT

1401 TAPE LIBRARY CONTROL SYSTEM

Robert W. Heald

Direct Inquiries to: Mr. Robert W. Heald
IBM Corporation
1215 10th Street
Sacramento, CA

Purpose/Description: To insure the proper mounting of magnetic tapes for each job run and to facilitate the maintenance of the library. To eliminate the necessity for coding error routines. To provide end of reel and end of library logic in a routine manner.

Method: N/A

Restrictions/Range: N/A

Storage Requirements: Approximately 2000 storage positions.

Equipment Specifications: 1401 - Model C, D, E. F 13-16 or F 23-26, Advanced Programming Package, High-Low-equal Compare

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.011

Purpose: To provide a source program at the SPS level and create the required linkage for the floating point operations written in macro format.

Method: Source Language SPS

Restrictions, Range: Total package
4-digit positions 0133 thru 1140
10-digit positions 0333 thru 1177
14-digit positions 0533 thru 1244

Storage Requirements: Approximately 2400 storage positions.

Equipment Specifications: 1401 - any model, no special features required, 3400 positions of storage, Model C3 or E3 equipped with multiply and divide, Advanced Programming Feature, and two magnetic tape units.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.012

9 x 9 TEN MILLIONSECOND MULTIPLY SUBROUTINE

Mr. Richard B. Feaster & Mr. William H. Post
IBM
340 Market Street
San Francisco, CA

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 ms. per multiplication.

Storage Requirements: 340 Positions.

Equipment Specifications: IBM - any model, no special features required.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.013

SCION (Scientific 1401 Programming with Floating Point)

John Discala
IBM
9250 Wilshire Blvd.
Beverly Hills, CA

(Continued on next column)

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.013

Purpose: Provides a source program at the SPS level and creates the required linkage for the floating point operations written in macro format.

Method: Source Language SPS

Restrictions, Range: Total package
4-digit positions 0133 thru 1140
10-digit positions 0333 thru 1177
14-digit positions 0533 thru 1244

Storage Requirements: Approximately 2400 storage positions.

Equipment Specifications: 1401 - any model, no special features required, 3400 positions of storage, Model C3 or E3 equipped with multiply and divide, Advanced Programming Feature, and two magnetic tape units.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.013

9 x 9 TEN MILLIONSECOND MULTIPLY SUBROUTINE

Mr. Richard B. Feaster & Mr. William H. Post
IBM
340 Market Street
San Francisco, CA

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 ms. per multiplication.

Storage Requirements: 340 Positions.

Equipment Specifications: IBM - any model, no special features required.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.013

SCION (Scientific 1401 Programming with Floating Point)

John Discala
IBM
9250 Wilshire Blvd.
Beverly Hills, CA

(Continued on next column)

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.013

Purpose: Provides a source program at the SPS level and creates the required linkage for the floating point operations written in macro format.

Method: Source Language SPS

Restrictions, Range: Total package
4-digit positions 0133 thru 1140
10-digit positions 0333 thru 1177
14-digit positions 0533 thru 1244

Storage Requirements: Approximately 2400 storage positions.

Equipment Specifications: 1401 - any model, no special features required, 3400 positions of storage, Model C3 or E3 equipped with multiply and divide, Advanced Programming Feature, and two magnetic tape units.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.013

9 x 9 TEN MILLIONSECOND MULTIPLY SUBROUTINE

Mr. Richard B. Feaster & Mr. William H. Post
IBM
340 Market Street
San Francisco, CA

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 ms. per multiplication.

Storage Requirements: 340 Positions.

Equipment Specifications: IBM - any model, no special features required.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 5.0.013

SCION (Scientific 1401 Programming with Floating Point)

John Discala
IBM
9250 Wilshire Blvd.
Beverly Hills, CA

(Continued on next column)
**IBM 1401 PROGRAM LIBRARY ABSTRACT**

### 1401 LINEAR PROGRAM

**Harm K. Schreur**

IBM

**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)

### IBM 1401 PROGRAM LIBRARY ABSTRACT

**File Number:** 10.1.005

### 1401 LINEAR PROGRAM (Continued)

...(Continued as in image)
IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 11,0,003

Numerical Solution of Legendre's Differential Equation on the IBM 1401

Curt Kamlin

Purpose/Description: A demonstration program which computes and tabulates the Legendre functions P_n^{m}(x) in the interval -1 < x < 1.

Mathematical Method: Numerical integration of Legendre's differential equation

Where

\( (x^2-1)P_n^{m} = n(n+1)P_{n-2}^{m} \)

and for

\( \frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \)

by the Range-Kutta 2nd order method according to the scheme in figure 1. Integration step: 0.01.

Restrictions/Range: N/A

Storage Requirements: 2,800 positions

Equipment Specifications: IBM 1401 with 4000 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 using 2,800 storage positions computes and tabulates the Legendre functions P_1^{m}(x) in 6.8 minutes by numerical solution of Legendre's differential equation.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 11,0,004

A Program for Solving Systems of Linear Equations on the IBM 1401

Soren Nordin

Purpose/Description: A program for solving linear equation systems. It is also well suited as a demonstration program.

Mathematical Method: The system of equations is solved using the elimination method. All arithmetic operations are performed in floating point numbers.

(Continued on next page)
IBM 1401 PROGRAM LIBRARY ABSTRACT  File Number 13.1.000

1441 Card-to-Tape Program

C. R. Mayo, T. S. Schurman (IBM), R. F. Vorwald
McDonnell Automation Center
P. O. Box 516
St. Louis 66, Missouri

Purpose: The program was written specifically to replace the SHARE 80 x 84 board of the IBM 709 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 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 13.1.002

1451 Card-to-Tape Program

R. F. Vorwald
McDonnell Automation Center
P. O. Box 516
St. Louis 66, Missouri

Purpose: The program was written to punch, in the first 50 columns of a card, the corresponding positions of any tape (binary or BCD). Stops are provided at an end of file and at persistent tape read errors.

Method: Each record is read and tested for error. If in error, the mode is switched. This process is repeated until 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 50 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

ACT - Automatic Checkout Technique
Robert Kaneharu

Direct Inquiries to: Lloyd W. Green
North American Aviation, Inc.
Programming Dept. 692
Los Angeles 45, California
Spring 5-2011, Ext. 3054

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. Input 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

File Number: 13.1.007

DUMP 01

Deek Nichols

Direct Inquiries to: Dick Nichols
North American Aviation, Inc.
Dept. 92, Building 6
4600 East 5th Avenue
Columbus 16, Ohio

Purpose/Description: The purpose of this 1401 Utility is to have the facility of "Dumping" the contents of magnetic tapes, whether in BCD, or Octal 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 its 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: BCD-2534 characters in Binary 42.2+Words.
2. Records longer than the maximum will be truncated and treated as though they were exactly 2534 characters. No indication of the truncation will be given.
3. When sense switches D through G are down, switches B and C are not active.

Equipment Specifications:
1401 Model C-3, advanced programming package, high-low equal, Column Binary and One (1) tape drive.

Storage Requirements: Memory 4K. All programs are written in SPS.

IBM 1401 PROGRAM TAPE WRITER

File Number: 13.1.006

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 198 and memory from 001 to the first group-mark word-mark is written with word-marks on logical tape 1. If sense switch D 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 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 is replaced by a group-mark without word-mark. Thus if group-mark word-marks are required they should be constructed during execution.

Equipment Specifications:
1401 machine

Storage Requirements: 40KC

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.009

NCGP - REPRODUCE, GANG-PUNCH, COUNT & PRINT

B. J. Manring

Direct Inquiries to: B. J. Manring
8621 Georgia Avenue
Silver Spring, Maryland

Purpose/Description: To reproduce cards, performing operations which would otherwise entail wiring a separate reproducer board, to list cards where a reformatting of the card image is desired, and to serially number cards and/or lines on a page.

(Continued on next page)
**IBM 1401 SIMULTANEOUS CARD-TO-TAPE AND/OR TAPE-TO-PRINTER**

**Direct Inquiries to:** J. Oldenburg
Republic Aviation Corporation
Farmingdale
Long Island, New York

**Equipment Specifications:** Requires approximately 3900 memory locations of 4K 1401. Error conditions are handled by the program. As far as practical, IBM Applied Programming tape error philosophy is employed.

**Storage Requirements:** Requires approximately 3900 memory locations of 4K 1401.

**Purpose:** To efficiently simulate all phases of peripheral equipment operation on the IBM 1401 at maximum (13600) operating speeds, e.g.: card-to-tape (columns binary and/or Hollerith) - 600 CPM tape-to-card (binary and/or decimal) - 250 CPM tape-to-printer (single space or program control with buffered output option) - 400 LPM.

Each I/O subroutine is interchangeable and may be restarted with minimum operator action.

**Restrictions, Range:**

- To be used primarily for supplying input to and developing output from 709/7090 computers. Requires a MOD C 1401 with advanced program-ming package, two tape units, high-low-equal compare, print storage, and column binary feature. Should not be used for making 704 input tapes without minor modifications to card-to-tape subroutine. Requires 1405 memory to be cleared prior to loading (IBM two-card clear memory routine is attached to front of object deck).

**Method:** Under normal operating conditions, SCRAMBLE performs I/O functions at maximum 1401 operating speeds, e.g.; card-to-tape (columns binary and/or Hollerith) - 600 CPM tape-to-card (binary and/or decimal) - 250 CPM tape-to-printer (single space or program control with buffered output option) - 400 LPM.

Each I/O subroutine is interchangeable and may be restarted with minimum operator action.

- Requires approximately 3900 memory locations of 4K 1401.
- Requires approximately 3900 memory locations of 4K 1401.

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**File Number:** 13,3,000

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**File Number:** 13,2,000

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**File Number:** 14,0,000

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**File Number:** 12,5,000

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**File Number:** 12,3,000
IBM 1620 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 312 symbols allowed by SPS II for assemblers on a 20K 1620. LAMP allows 470 symbols and will reduce assembly time by up to 55%.

Mathematical Method: Does not apply

Restrictions, Range:
1. LAMP will accept any SPS II statement with the following exceptions:
   a. DAS, DBH, DNM, DN, DMTY, and BNG
   b. RN must specify BNG, KNV, RAGC, RAY
   c. BP, BH, RN, BNP, BNN, and BNZ (User must use instead RN, RN, RN, BNN, BNL, BNN)
   d. SV, BDN, BC1, BCI, BC2, BC4 and BNC1, BNC2, BNC3, BNC4 (User must use RN and BNN)
   e. The TDM instruction will be assembled with a flag in position 7.
   f. The input for both pass 1 and pass 2 must be from card.
   g. All references to subroutines have been eliminated.
   h. Error 1 and Error 3 have been eliminated.
   i. Checking for record marks in label and op. code fields has been eliminated.

Storage Requirements: Processor occupies all of memory.

Equipment Specifications: IBM 1620, 1622.

Purpose / Description: Program Loader for the IBM 1620 with card input, designed specifically for commercial applications requiring more than the 312 symbols allowed by SPS II for assemblers on a 20K 1620. LAMP allows 470 symbols and will reduce assembly time by up to 55%. Mathematical Method: Does not apply.

Restrictions, Range: Processor occupies all of memory.

Equipment Specifications: IBM 1620, 1622.

Purpose / Description: Output of one card per instruction executed showing instruction, its address, and F. Q. and general products field (where applicable).

Method: Not applicable

Restrictions / Range: Not available

Storage Requirements: Not available

Equipment Specifications: Not applicable

IBM 1620 PROGRAM LIBRARY ABSTRACT

RELOCATING LOADER (Tape)
W. J. Richards

Direct Inquiries to: W. J. Richards
Phillips Engineering Co., Inc.
4145 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 P or Q fields, have constants exactly 12 digits in length, and change in required in the SPS processor.

Storage Requirements: Locations (1980 - 00399)

Equipment Specifications: Paper tape, Memory 20K, and no other special features required.

IBM 1620 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. Indirect addresses are completely traced. The mnemonics for the commands are printed. The programmer specifies, by two number 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 #14 cannot be interrogated by the traced program without special (but trivial) modification of the program.

Storage Requirements: 2166 locations.

Equipment Specifications: Any 1620 with indirect addressing.

Additional Remarks: Of the 2366 locations, all but one are completely relocatable. The digit 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 as produced by SPS, which may be compressed if desired. It is possible to include optional instructions such as FIT-MP, etc., without difficulty. The program has been written for card I/O only.

IBM 1620 PROGRAM LIBRARY ABSTRACT

TRACING PROGRAM FOR THE IBM 1620 WITH CARD INPUT/OUTPUT (CARD)
Ralph L. Miller

Direct Inquiries to: Ralph L. Miller
IBM Corporation
1418 S. Michigan Avenue
Chicago 5, Illinois

Purpose / Description: Output of one card per instruction executed showing instruction, its address, and F. Q. and general products field (where applicable).

Method: Not applicable

Restrictions / Range: Not available

Storage Requirements: 1139 card locations -- relocatable SPS

Equipment Specifications: Memory 2OK, and no other special features required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 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 control of address step, full or branch trace, elimination of BT subroutines, and typed or card output.

Mathematical Method: Each traced instruction executes its own output format.
Restrictions, Range: Will not properly handle more than 5 digits in an immediate
command. A digit change encountered in instruction or data will result in short
line if typed. No such restriction in card mode.
Storage Requirements: N/A
Equipment Specifications: N/A

STROBIC - Skelly Trace Routine with Option on Branch and transmit and
Direct Inquiries to: G. R. Berry
Indirect address Conversion (Card)
K. R. Tieman
Skelly Oil Company
Accounting Department - Computer Programming Unit
P. O. Box 1650
Tulsa, Oklahoma
Liber 4-2311, Extension 634
Purpose/Description: STROBIC is a full trace routine for the 1620 computer
computer equipped with a 16ZO card read/punch unit and the indirect addressing special feature.
STROBIC will trace the automatic divide, the indirect address feature, and the
transfer numeric strip/move flag/transfer numeric fill 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.
Additional Remarks: Language--1620 S.P.S., Entry: Console, Output: Punched
cards, one card for each traced instruction.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 1.6.005

TRACE AND IA SIMULATOR (Tape)
Charles E. Berry

Direct Inquiries to: Charles E. Berry
IBM Corporation
1211 S. W. 6th Avenue
Portland, Oregon
CA-8-6623
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 opera-
tion code - may be digit, field, or record. User may execute portions of pro-
gram at full speed or show trace at a predetermined instruction.
Method: Not applicable
Restrictions/Range: Cannot re-enter trace made from automatic mode internal
in a BT-IB pair.
Storage Requirements: 2613 plus 26 at the end of memory
Equipment Specifications: Memory 20K, 40K, 60K, Automatic Divide and Paper
Tape: No other special features required.
Additional Remarks: Relocatable. Immediate fields may be 12 digits long.
Record marks internal to fields or in instructions are acceptable. Typewriter con-
tral 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 1.6.006

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 in 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 BT subroutine, 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: N/A positions.
Equipment Specifications: 20K tape 16ZO 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--SPS - completely relocatable.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 1.5.001

FORTRAN SOURCE TAPE CORRECTOR (Tape)
D. S. Gardner

Direct Inquiries to: D. S. 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 + 1/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
349 S. Broadway
Akron 8, 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 is 1620 FORTRAN statements, unaltered with respect to "card columns".
Output is a tidied-up statement, C (if present) in position 1, statement number
(if present) in positions 5-8, statement itself in positions 7-72. Excessively
long statements are edited by elimination of blanks to fit in positions 7-72 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 047.
Method: N/A
Restrictions/Range: N/A
Storage Requirements: N/A
Equipment Specifications: Basic paper tape 1620
Additional Remarks: The language is SPS.
IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.5.003

TAPE EDIT (Tape)

Additional Remarks: Language is SPS.

Restrictions/Range: No special features are required.

Storage Requirements: N/A

Equipment Specifications: 1620 paper tape.

Purpose/Description: To dump portions of memory in data or instruction for.

Direct Inquiries to: W. T. Gault

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.6.002

1620 I. D.A. Edit Subroutine (Tape)

Neil Lewis

Direct Inquiries to: Neil Lewis

System Engineer-Scientific (766441)

IBM Corporation

Honolulu, Hawaii

Purpose/Description: This routine is an indirect addressing version of the 1620 I.D.A. 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: 160 positions

Equipment Specifications: Tape 1620, Memory 20K, 40K, 60K with indirect addressing, no other features required.

Additional Remarks: Language relocatable, SPS, Symbolic

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.6.004

1620 AUTOPLOTTER (Tape)

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 manual.

Restrictions, Range: Graph paper sizes up to 20 inches high and 100 inches wide.

Accuracy plus or minus 0.010 inches on all points plotted. Graphs include automatic generation of all scales and labels.

Speed: Main frame time 5 to 6 minutes, off-line time 5 to 10 minutes.

Method: An original scanning and curve-fitting technique is used.

Storage Requirements: 20K, no special features. Modified 870 system used as plotter. See preliminary manual.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.6.003

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 0.010 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 tape, no special features. Modified 870 system used as plotter. See preliminary manual.
IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 1.6.005

1620 I D A Edit Subroutine (Card)

Neil Lewis

Direct Inquiries to: Neil Lewis
Systems Engineer-Scientific (756641)
IBM Corporation
Honolulu, Hawaii

Purpose/Description: This routine is an indirect addressing version of the
1620 Edit Subroutine 1.6.010.

Restrictions, Range: Does not apply.

Storage Requirements: 106 positions.

Equipment Specifications: Card 1620, memory 20K, 60K, 60K with
Direct Addressing, all other features required.

Additional Remarks: Language-Non-relocatable (Relativised) Symbolic
Numerical Method: Does not apply.

Storage Requirements: 306 positions.

Additional Remarks: Programs to be edited must be written in SPS and must follow
a few additional rules itemized in the write-up of the edit program.

Restrictions, Range: Does not apply.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 1.6.006

1620 FORCOM (Card)

Bob Lowen

IBM Detroit North
7120 Second Boulevard
Detroit 2, Michigan

Purpose: To provide alphameric 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: 990 digits.

Equipment Specifications: IBM 1620 card, any core size. No special features
required.

Additional Remarks: 1620 SAY is a FORTRAN Subroutine or Independent. It is

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 1.6.007

SPS - To - FORTRAN SUBROUTINE EDIT (Tape)

C. I. Johnson

IBM Corporation
1730 Cambridge Street
Cambridge, Massachusetts

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 FOR-
TRAN subroutine.

Restrictions, Range: A maximum of nine 40-character records may be stored
in core at one time.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 1.6.008

1620 FORTRAN Input-Output Routine Using Format Control. (Card)

Donald C. Willan

Direct Inquiries to: Donald C. Willan
7120 Second Boulevard
Detroit 2, Michigan

Purpose/Description: To give greater flexibility and control to Fortran Output
on cards and typewriter. It is now possible to leave off insignificant digits,
leave control of the decimal point, and have control of the number of words per
line with no sacrifice of storage area.

Mathematical Method: Does not apply.

Restrictions/Range: The output numbers are limited to $10^9$ and $10^{-9}$.
Four formats are available and up to 11 numbers per line can be specified in
each format. Up to 23 words per line can be specified if the next format is
not used. (See miscellaneous notes in writeup.)

Storage Requirements: The program occupies location 4342 to 7492.

Equipment Specifications: Card 1620, memory 20K, Indirect Addressing,
and other special features required T76, THF, MP.

Additional Remarks: To use this program a modified subroutine deck must be
used when processing a Fortran program. No changes need be made to the
program, so that the unmodified subroutine deck can be used if desired. The
language used in SPS and is non-relocatable. It will handle both fixed and floating
point numbers on input and output.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 1.6.009

SPS - To - FORTRAN Subroutine Edit (Revision) (Tape)

C. I. Johnson

Direct Inquiries: C. I. Johnson
IBM Corporation
1730 Cambridge Street
Cambridge, Massachusetts

Purpose/Description: 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 FOR-
TRAN subroutine.

Mathematical Method: Does not apply.

Restrictions/Range: Does not apply.

Storage Requirements: Program is always loaded between 00402 and 04419.

Equipment Specifications: Memory 20K, Paper Tape Machine. No other special
features required.

Additional Remarks: Edit Routine Written In: SPS Language (1) Programs to be
edited must be written in SPS and must follow a few additional rules itemized in
the write-up of the edit program. (2) The edit routine produces the SPS object
program automatically. An optional feature to the ability to list the tape in FORTRAN subroutine form. Also optional is the ability to insert up to 1,000
digits of remarks on the listing in addition to a heading including the name, date,
and number of the subroutine. This version replaces the original #1.6.007.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 1.6.010

1620 EDIT SUBROUTINE (Tape)

Neil Lewis

Direct Inquiries to: Neil Lewis
Systems Engineer - Scientific (756641)
IBM Corporation
Honolulu, Hawaii

Purpose/Description: This routine inserts a continuous series of numeric data
into an alphametric record as specified by the programmer, leaving it
ready for printing or punching. Automatic zero suppression and the ability
to handle all alphametric characters are standard features. All data following a
decimal point is printed. When room is provided ahead of a decimal point, the
routine forces that at least one figure or zero precedes the decimal point.

(Continued on next page)
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 16ZO, memory 20K, 40K, 60K. No other special features required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)

File Number 1.6.011

Direct Inquiries to: Neil Lewis
Systems Engineer—Scientific (756641)
IBM Corporation
Honolulu, Hawaii

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 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: Card 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Subroutine is applicable to either a tape or card oriented 16ZO, also requires a three instruction linkage (Macro form) to the dump routine.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 FLOAT SUBROUTINE (Tape)

File Number 1.6.012

Direct Inquiries to: Henry L. Schmitz, Jr.
Systems Engineer—Scientific
IBM Corporation
Springfield, Massachusetts

Purpose/Description: To translate data from fixed point form to the internal floating point form required by the floating point subroutines of the Symbolic Programming System.

Mathematical Method: N/A
Restrictions/Range: Numbers from -9,999,999 to +9,999,999 can be handled. The user cannot specify a power of ten to be added to the computed characteristic.
Storage Requirements: 889 positions
Equipment Specifications: Base 1620
Additional Remarks: Subroutine is applicable to either a tape or card oriented 16ZO.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 FIX SUBROUTINE (Tape)

File Number 1.6.013

Direct Inquiries to: Henry L. Schmitz, Jr.
Systems Engineer—Scientific
IBM Corporation
Springfield, Massachusetts

Purpose/Description: To translate data from the internal floating point form required by the floating point subroutine and functional subroutines to a fixed point form more readily understood.

(Continued on next column)
Statement B: FORM: FAT (4Z1) Notes

may be easily accommodated to 46ZO Fortran for tape I/O. Example Fortran

Additional Remarks: These subroutines were written for 16ZO card Fortran but

~N/A

~N/A

Storage Requirements: FAT and CLE are relocatable subroutines for 16ZO card

are no other restrictions placed on the use of this subroutine,

Equipment Specifications: Memory 20K, Indirect Addressing.

Additional Remarks: These subroutines were written for 16ZO card Fortran but

may be easily accommodated to 16ZO Fortran for tape I/O. Example Fortran Statements: FORM * FAT (421) Notes: (1) "Form" could be any unused symbol

(2) 4 of (421) specifies digits before decimal

(3) 1 of (421) specifies digits after decimal

(4) 1 of (421) specifies space between words

(5) Decimal in (421) is essential to make a floating point number.

(6) Sign is in addition to spaces

(7) If number to be printed is too large or too small the exponent of ten is specified

CLEAR * CLE (400)

Notes: (1) This statement restores normal Fortran format

(2) Any fixed point number in parenthesis will achieve same result.

IBM 1420 PROGRAM LIBRARY ABSTRACT

File Number 1, 6, 017

GOHOT (Generator Of Hermaphroditic Object Tapes) (Tape)

Direct Inquiries to: Dick Conner

IBM Corporation

462 Seventh Avenue

Pittsburgh 15, Pennsylvania

Purpose/Description: Gohot punches a program in self-loading, self-reproducing form. This tape, and any of its descendants, loads itself or reproduce 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 lie entirely within cells 00401-19999 and must use decimal arithmetic. Record marks throughout the program do not constitute an obstacle to Gohot.

Storage Requirements: 00000-00299 (tables are restored at end)

Equipment Specifications: Memory 200, 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 1420 PROGRAM LIBRARY ABSTRACT

File Number 1, 6, 019

FORTRAN II DIAGNOSTICIAN (CARDS)

James Stoiber, Charles Snyder, & Jack Burgeson

Direct Inquiries to: Jack Burgeson

IBM Akron

(Continued on next column)
IBM 650 Simulator Program (Card)  
F. C. Toscano

Direct Inquiries to: F. C. Toscano
IBM Corporation
125 South Flower Street
Los Angeles 17, California

Purpose/Description: Simulation of the IBM 650 on the IBM 1620. It allows execution of 650 language programs in a 1620 without reprogramming.

Method: N/A

Restrictions/Range: N/A

Storage Requirements: N/A

Equipment Specifications: Tape 1620, 20K; no other special features required.

Additional Remarks: Program is written to be compiled with SPS two pass compiler and subroutines. It uses floating point arithmetics and numbers must be entered in standard 60 character floating point notation. System used successfully on approximately 10 programs to date. (7/23/61)

IBM 650 Simulator Program (Card)  
F. C. Toscano

Direct Inquiries to: F. C. Toscano
IBM Corporation
125 South Flower Street
Los Angeles 17, California

Purpose/Description: Simulation of the IBM 650 on the IBM 1620. It allows execution of 650 language programs in a 1620 without reprogramming.

Method: N/A

Restrictions/Range: N/A

Storage Requirements: N/A

Equipment Specifications: Tape 1620, 20K; no other special features required.

Additional Remarks: Program is written to be compiled with SPS two pass compiler and subroutines. It uses floating point arithmetics and numbers must be entered in standard 60 character floating point notation. System used successfully on approximately 10 programs to date. (7/23/61)
Mathematical Method: Odd Integer

Restrictions, Range: The argument of the subroutine must be exactly 2 "L", digits in length. The argument will be destroyed in the course of the calculation. The "L" low order digits of the argument will be replaced by the result. The minimum value that "L" may assume is 2. The only upper bound upon "L" is the amount of storage available.

Storage Requirements: 630 locations+1,"02 locations for Odd Integer field.

Additional Remarks: The subroutine is supplied in symbolic form, on cards, for assembly with the user's program. It is completely relocatable. It has additional features required.

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 50 floating point.

Method: Variation on the Gaussian elimination technique, known as the product 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.

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 size, 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 1622 attachment. Division feature not required.

Additional Remarks: This program uses SPS Language, and is non-relocatable.

EIGENVALUES OF REAL SYMMETRIC MATRICES ON THE 1620 DATA PROCESSING SYSTEM (Card)

Neil Lewis
IBM
88 Kapahulu Blvd.
Honolulu 13, Hawaii

Purpose: Will solve for the eigenvalues and associated eigenvectors of a real, symmetric matrix to order 50.

Restrictions, Range: The program consists of 3 basic parts.

A) Phase 1: makes leading 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 Gaussian elimination, Jacobi method. Eigenvalues are typed out at the conclusion of phase 2.

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.

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 50 floating point.

Method: Variation on the Gaussian elimination technique, known as the product 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.

EIGENVALUES OF REAL SYMMETRIC MATRICES ON THE 1620 DATA PROCESSING SYSTEM (Card)

Neil Lewis
IBM
88 Kapahulu Blvd.
Honolulu 13, Hawaii

Purpose: Will solve for the eigenvalues and associated eigenvectors of a real, symmetric matrix to order 50.

Restrictions, Range: The program consists of 3 basic parts.

A) Phase 1: makes leading 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 Gaussian elimination, Jacobi method. Eigenvalues are typed out at the conclusion of phase 2.

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.

Equipment Specifications: 1620, paper tape, 20K core. No other devices are necessary.

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 size, 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 1622 attachment. Division feature not required.

Additional Remarks: This program uses SPS Language, and is non-relocatable.

EIGENVALUES OF REAL SYMMETRIC MATRICES ON THE 1620 DATA PROCESSING SYSTEM (Card)

Neil Lewis
IBM
88 Kapahulu Blvd.
Honolulu 13, Hawaii

Purpose: Will solve for the eigenvalues and associated eigenvectors of a real, symmetric matrix to order 50.

Restrictions, Range: The program consists of 3 basic parts.

A) Phase 1: makes leading 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 Gaussian elimination, Jacobi method. Eigenvalues are typed out at the conclusion of phase 2.

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.

Equipment Specifications: 1620, paper tape, 20K core. No other devices are necessary.

MATRICES INVERSION (Tape)

Dale Anderson
IBM
340 S. Broadway
Akron 8, Ohio

Purpose: 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. Output is in Fortran language.
**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number: 5, 0, 007**

**SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS (Cards)**

**Burr Preston**

**Direct Inquiries to:** Burr Preston
IBM Corporation
500 N. Dearborn Street
Chicago 16, Illinois
Whitehall 4-1364

**Purpose/Description:** This program solves sets of homogeneous simultaneous linear equations and provides either printed or punched output with heading. It is designed for ease of use. Operating instructions and error messages are automatically typed. Data values are entered in free form notation as a group of digits with a decimal point. An optional power of ten may be added to each value.

**Method:** The Jordan method of elimination is used.

**Restrictions/Range:** A maximum of 26 equations in 26 unknowns may be solved. A maximum of eight significant digits per matrix element is allowed.

**Storage Requirements:** The entire core for 26 equations.

**Equipment Specifications:** Memory 20K. Card Input-Output and no other special features required.

**Additional Remarks:** The language is Fortran (approximately 80 statements).

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number: 6, 0, 003**

**SCRAP (Sixteen-twenty Card Regression Analysis Program) (Card)**

**D. N. Leeson**

**Direct Inquiries to:** D. N. Leeson
IBM Corporation
425 Park Avenue
New York City, New York
Whitehall 4-1466

**Purpose/Description:** This program performs a complete regression analysis on a maximum of 24 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. Core-Divide not required.

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number: 6, 0, 004**

**STRAP (Stepwise Regression Analysis Program) (Tape)**

**L. S. Holmes & A. R. Calville**

**Direct Inquiries to:** L. S. Holmes
IBM Corporation
Beaumont, Texas

**Purpose/Description:** STRAP is a multiple stepwise regression analysis program containing provisions 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:

\[ \hat{y} = \beta_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i \]

Where \( \hat{y} \) is the dependent variable, \( x_i \) are the independent variables, and \( \hat{\beta}_i \) are the coefficients to be determined.

**Mathematical Method:** N/A

**Restrictions/Range:** N/A

**Storage Requirements:** 20,000 positions

**Equipment Specifications:** Basic 1620, paper tape input/output.

**Additional Remarks:** Floating point arithmetic. Non-relocatable.

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number: 6, 0, 005**

**FREQUALIZER (Tape)**

**Robert Axelrod**

**Direct Inquiries to:** Paul Sanders
Statistical Services
Abbott Laboratories
North Chicago, Illinois

**Purpose:** This program performs a complete regression analysis on a maximum of 24 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. Core-Divide not required.

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number: 6, 0, 006**

**REGRESSION ANALYSIS PROGRAM (Tape)**

**D. N. Leeson**

**IBM Eastern Regional Office**

**Direct Inquiries to:** D. N. Leeson
IBM Corporation
425 Park Avenue
New York City, New York

**Purpose:** This program performs a complete regression analysis on a maximum of 24 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. Core-Divide not required.

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number: 6, 0, 007**

**REGRESSION ANALYSIS PROGRAM (Card)**

**D. N. Leeson**

**IBM Eastern Regional Office**

**Direct Inquiries to:** Paul Sanders
Statistical Services
Abbott Laboratories
North Chicago, Illinois

(Continued on next column)
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 logs, any number of data points.
Storage Requirements: 20,000 digits
Equipment Specifications: Memory 20K, and no other special features required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

STEPWISE MULTIPLE LINEAR REGRESSION (Tape)
R. Bukacek & W. Galle

Direct Inquiries to: W. J. Galle
Armour & Company
Operations Research
601 N. Wabash
Chicago, Illinois

Purpose/Description: Accepts sets of observations and forms linear regressions in a stepwise fashion subject to statistical criterion (F-Tests).
Method: Stepwise linear regression
Restrictions/Range: Maximum number of independent variables = 18, \( 12 \leq N \leq 50 \), Maximum number of dependent variables on input tape = 25 - Nind, \( 12 \leq N \leq 50 \), Maximum number of independent variables which can be Apertor Suppressed from consideration in stepwise process = 15 \( 12 \leq N \leq 50 \).
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 writing for complete description and notes.

IBM 1620 PROGRAM LIBRARY ABSTRACT

ANALYSIS OF VARIANCE (Card)
D. G. Wynman

Direct Inquiries to: D. G. Wynman
IBM Corporation
464 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 consistent. This allows 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. 12 variables are maximum for Phase II, 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 20K. No other special features required.

Additional Remarks: The program has been coded in SPS using SPS floating point subroutines for all of the mathematics. Programs are compiled independently and run by loading and executing the routines in sequence. Operation is continuous. About 1500 instructions are used with an additional 345 for a report generator not including SPS subroutines. Two of the eight routines use most of 20K memory.
The program is being used consistently by two card 1620 installations in the Oakland area. Cards or paper tape may be used as input/output.
POLYNOMIAL CURVE FITTING (Tape)

W. R. Graves
IBM
2640 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 or the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions / Range: Not given.

Mathematical Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations, Experimental data are recorded in standard 1620 FORTRAN format.

Storage Requirements: Not given.

Remarks: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, paper tape reader, paper tape punch. Will run on any 1620 for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

POLYNOMIAL CURVE FITTING (Card)

Direct Inquiries to: W. R. Graves
IBM
2640 Canal Street
New Orleans 19, Louisiana

Purpose/Description: This program generates an approximating polynomial by the least squares technique. The equation as 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 or the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions / Range: Not given.

Mathematical Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations, Experimental data are recorded in standard 1620 FORTRAN format.

Storage Requirements: Not given.

Remarks: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, 1622 card read-punch. Will run on any 1620 for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

POLYNOMIAL CURVE FIT (Tape)

Dale Anderson

Purpose/Description: This program fits an nth degree polynomial to m sets of weighted or unweighted data points (x, y). Provision is made for processing the same set of (x, y) points through polynomials of increasing degree n. A complete evaluation is made of each fit and statistics indicating "goodness of fit" typed out.

Method: Least square solution of simultaneous equations.

Restrictions / Range: n less than or equal to the smaller of (12, M-1); m less than or equal to 120.

Storage Requirements: N/A

Equipment Specifications: Basic tape 1620. Because of the coding language used, it can easily be converted to card 1620 - with I/O modifications to any hardware accepting Fortran coding.

Additional Remarks: The language is Fortran (approximately 160 statements).

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 SUBDIVISION PROGRAM (Tape)

H. W. Van Ness
C. E. Berry
K. J. Love
IBM
1212 S. W. 6th Avenue
Portland 4, Oregon

Purpose: Compute necessary data for the subdivision of land into smaller parcels. 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; 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 16,000 positions except in Pass 1 and the co-ordinate type out.

Equipment Specifications: Minimum 1620, 16,000 positions of core and paper-tape input-output.

IBM 1620 PROGRAM LIBRARY ABSTRACT

CUT AND FILL (Tape)

Ben A. Shaw
IBM
9/3 R. Robert Street
St. Paul 1, Minnesota

Purpose: compute grades, apply typical sections, compute slope intervals, areas, and volumes when given P.V. Stations, Elevations, and Lengths of Vertical Curves, Typical Sections and where they are to be used, Shrinkage Factors, and Preliminary Terrain Gross Sections.

Restrictions, Range: This program does not compute horizontal curve transitions. It is limited to 50 Terrain Points/ Cross Section, Ten Typical Sections, and ten P.V.'s. The horizontal distances are to even feet, and the elevations are to tenths of a foot.

Method: Does not apply.

Storage Requirements: Four program passes are required - utilizing all of the 16,000 positions except in Pass 1 and the co-ordinate type out.

Equipment Specifications: Minimum 1620, 16,000 positions of core and paper-tape input-output.
IBM 1620 TRAVELER ANALYSIS PROGRAM (Card)  
D. T. Mitchell

Direct Inquiries to:  D. T. Mitchell  
IBM Corporation  
Midwestern Regional Office  
618 South Michigan  
Chicago 5, 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 sine and cosine are calculated within $2 \times 10^{-8}$ 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 TRAVELER ANALYSIS PROGRAM (Tape)  
D. T. Mitchell

Direct Inquiries to:  D. T. Mitchell  
IBM Corporation  
Midwestern Regional Office  
618 South Michigan  
Chicago 5, 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 sine and cosine are calculated within $2 \times 10^{-8}$ insuring 3-decimal place accuracy in latitudes and departures.

Storage Requirements: 20K Core is required.

Equipment Specifications: Basic 1620 without any features; paper tape reader.

Additional Remarks: The source language is machine.

IBM 1620 TRAVELER ANALYSIS PROGRAM  
File Number 9.2.007
**MULTICOMPONENT DISTILLATION TOWER DESIGN CALCULATIONS (Tape)**

Ray N. Sauer  
IBM  
2601 South Main Street  
Houston 2, Texas

**Purpose:** To estimate the distillation tower requirements for a given separation, feed rate and thermal condition and set of relative volatilities.

**Restrictions/Range:** 30 components.

**Method:** Short cut methods of Feakle, Underwood, and Gilliland.

**Storage Requirements:** FORTRAN program with SPS patcher that fits within 20K.

**Equipment Specifications:** IBM 1620 with paper tape and 20K memory.

---

**GAS NETWORK ANALYSIS (Card)**

**Direct Inquiries to:** IBM Public Utility Department  
Midwestern Region  
615 South Michigan Avenue  
Chicago 5, Illinois

**Purpose:** With the use of this program, an analysis can be made for as many as 750 pipes in a low and/or medium pressure system with consideration given to modification and/or expansion.

**Restrictions/Range:** See purpose

**Storage Requirements:** 20,000 core locations

**Equipment Specifications:** IBM 1620 with Autodivide

---

**M-100 MOMENT OF INERTIA AND CENTROID CALCULATIONS (Card)**

G. J. Reed

**Direct Inquiries to:** R. C. Wernick  
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 is dependent on the number of grid spaces required to define the body. The time may be approximated by (T + 0.1) NBC 30 seconds, NBC is the number of divisions in the grid system.

---

**ELECTRIC LOAD FLOW PROGRAM (Tape)**

Frank Maulina  
Systems Engineer  
IBM Corp.  
421 7th Avenue  
Pittsburgh 19, Pa.

**Purpose:** This program is designed to calculate voltages and power flows in a system of a maximum size of 150 buses and 340 lines, and allow changes to be made to the base system and be rerun.

**Restrictions/Range:** All calculations are done in a fixed point.

1. Net load or generation at any bus must be less than 10,000 Megawatts and Megavars.
2. The self impedance of any bus must be less than 1000.000 per unit.
3. The sum of squares of G and B components of any bus must be less than 10,000,000,000 per unit.
4. The accuracy of the results may be predetermined by the operator by specifying tighter tolerances in the iterative solution.

**Speed:** Average time per iteration:

- Time in milliseconds: 600 times No. of buses + 132,8 times No. of lines + 916 times 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 2 PASS.

**Remarks:** This is an independent program and is assembled into fixed locations but is still relocatable unless reassembled.

**Equipment Specifications:** Basic 1630, 20K tape tape system.
Selection of Economic Conductor Size - Specific Case

Direct Inquiries to: R. H. Snow
New England Electric System Program #18 (Card)
worth of total annual costs for any four conductor sizes, and prints these costs as specific values. They are, however, all on separate cards, and can be used all together to give the total annual cost of any combination of conductors. This program is used for the selection of the economic conductor size for each year for a period not exceeding 20 years, on a 1000 wire-foot basis.

Method: This program is based upon the methods presented in the following Electric World Articles, by L. J. Rankine.

Date: October 3, 1955
Title: Three-Thirds Rule Used for Capacitors KVAR
September 26, 1968
Method of Locating Shunt Capacitors Suitable
for Computer Solutions.

Restrictions, Range: Four standard capacitor bank sizes are considered.
Storage Requirements: About 3000 memory locations are used.
Equipment Specifications: IBM 1620 - Tape input/output.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 9.4.003

ELECTRIC LOAD FLOW PROGRAM (Card)
Franz Mosina
Systems Engineer
IBM Corporation
441 South Main Street
Hopkinton, Massachusetts

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 #18 (Card)

R. H. Snow
Direct Inquiries to: R. H. Snow
New England Electric System
245 South Main Street
Hopkinton, Massachusetts
Glenwood 3-0243 Ext. 32

Purpose/Description: Given installed costs, resistances, a load forecast, unit loss costs, and other pertinent data, this program calculates cumulative present worth of total annual costs for any four conductor sizes, and prints these costs for each year for a period not exceeding 20 years, on a 1000 wire-foot basis. Results are presented in tabular form and may easily be transferred to a graph, if desired.


Restrictions, Range: On Page 3 of the write-up, note that the depreciation rate, fixed charge rate, interest rate, and required return are built into the program as specific values. They are, however, all on separate cards, and can be changed according to the accounting practices of the user.

Storage Requirements: About 3000 memory locations are required, exclusive of tapes and subroutines.

Equipment Specifications: IBM 1620 (20 K memory) and 1620 reader.

Additional Remarks: The speed varies with number of years in load forecast, calculations and print-out for a ten-year period required about 2 minutes.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 9.4.015

Economic Conductor Size Selection by Kelvin's Law (Tape)
R. F. Stelness

Direct Inquiries to: R. F. Stelness
IBM Corporation
New York City, New York

Purpose/Description: To choose the conductor size that minimizes the overall cost of material and line losses.

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.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 9.4.006

SHORT CIRCUIT ANALYSIS (Card)
George S. Haralampu

Direct Inquiries to: George S. Haralampu
New England Electric System
441 Stuart Street
Boston 16, Massachusetts
Commonwealth 6-5800, Ext. 372

Purpose/Description: This program is to be used for the determination of current distribution constants, bus voltages, and impedance ratios under faulted conditions. This program is a one pass program, and complex network impedances are used.

Mathematical Method: Gauss-Seidel iterative method

Restrictions, Range: Does not apply

Storage Requirements: 33 buses and 58 lines

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 equivalent, should be avoided.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 9.4.007
Short Circuit Calculations (Card)
G. S. Haralampu

Direct Inquiries to: G. S. Haralampu
441 Stuart Street
Boston 16, Massachusetts
Commonwealth 6-5800

Extension 372

Purpose/Description: This program is to be used for the determination of current distribution constants, bus voltages, X/R ratios, and impedances to the point of fault, under faulted conditions. The program is limited to a 25 by 28 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 25 by 28 B-constant matrix

Storage Requirements: The program uses essentially the entire 20K core. The speed depends on the matrix size -- once the B-constants have been read an average case might take about 2 minutes.

(Continued on next page)
IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 9,4,009

CURVE FITTING - SIMULATED PLANT RECORD METHOD (Card)

William D. Garland

Direct Inquiries to: William D. Garland
New England Electric System
44 Stuart Street
Boston 14, Massachusetts

Purpose/Description: This program is designed to find the best fitting average life within each 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 book balances and the balances simulated for the best fit lives.

Method: A. Formula Terms:
- LC: long life assumed
- LL: shorter life assumed
- LU: best fit for one curve
- BL: balances simulated for LU
- BU: balances simulated for LC
- LL - LU

B. Formula:
- \( (BL - BU) / (BU - BL_0) \)

Note: Result accepted only when
- \( LL - LU - BL \leq 1 \)

Restrictions/Range: N/A

Storage Requirements: About 18,000 features required.

Equipment Specifications: IBM 1620 Computer with a 20K memory card and a 1620 Card Reader-Punch

Additional Remarks: The speed is approximately one second per pipe of Y values equally spaced along the X axis. The program scales them to the range 0-50 and plots them on the 1620 typewriter. Baseline indication is plotted also.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 9,7,001

STRAIN GAGE DATA REDUCTION ON THE IBM 1620 (Card)

R. C. Wenrick

Direct Inquiries to: R. C. Wenrick
ACF Industries
P.O. Box 1666
Albuquerque, New Mexico
CH 7-0361, 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: IBM 1620, Automatic Divide, and no other special features required.

Additional Remarks: The language is SPS. 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.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 9,7,002

GENERALIZED PLOTTER II (Card)

Jack Burgeson

Direct Inquiries to: Jack Burgeson
IBM Corporation
Boston, Massachusetts

Purpose/Description: This program balances the flow of water in a pipe network starting with assumed flows and produces the corrected system flows.

Mathematical Method: Hardy 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.
GENERALIZED PLOTTER (Cards)  
Jack Burgeson

Direct Inquiries to: Jack Burgeson - IBM
340 S. Broadway
Akron 8, Ohio

Purpose/Description: Given up to 400 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 400 points

Storage Requirements: Uses all storage

Equipment Specifications: Uses basic card 1620

Additional Remarks: Language is GTRAN.

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 9.7.004

S-105 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: The discontinuity and membrane effects in a tapered hub, used to connect a flange to a thin shell, are computed.

Method: Approximations as described in ASME "Design Data and Methods;" solution is done in 2-and-8 floating point form.

Restrictions/Range: The tapered hub must be "long" to give accurate results.

Storage Requirements: 18,500 core positions

Equipment Specifications: Memory 25K, Automatic Divide, and no other special features required.

Additional Remarks: Language is SPS. Running time depends on the number of variables requiring the divide hardware.

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 9.7.005

S-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: Timoshenko, "Theory of Plates and Shells" and authors.

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 SPS and utilizes three library subroutines which are the following: 1) L-109 Computation of C and 1st Order Bessel Functions; 2) L-105 Solution of Simultaneous Equations; The running time varies between 4 and 6 minutes depending on the hub dimensions. It has been run 50 times successfully. All subroutines are included in the card deck.

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 10.1.001

LINEAR PROGRAMMING CODE 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. Optimal 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: 

(M = 2) (N = 3)(Memory = 37600)

where M is number of restricting equations, N is number of non-basis variables, Memory is core size in digits.

Speed: Speed of solution is dependent upon the size and density of the matrix being solved. A 30 by 40 matrix which is reasonably block-diagonal will require about 20 seconds per iteration.

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.

Storage Requirements: Storage locations 00012 through 03750 are occupied by sub-programs and limiting 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 1620 paper tape reader.

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 10.1.002

LINEAR PROGRAMMING FOR THE 1620 (Card)  
Katherine Krieger
Ray D. Eke
IBM
31-25 Queens Blvd.
Woodside 17, N. Y.

Purpose: Solves linear programming problems with output of detailed results.

Method: Not given.

Restrictions/Range: The size of the problem is restricted to the following relationship:

(m+n) sq in <= memory \leq 37600

where m is the number of restrictions, n is the number of non-basis independent variables, memory is 20,000, 60,000, or 60,000.

The precise time required per iteration depends on the size and density of the matrix. As an approximation, a problem with 30 equations and 40 non-basis variables requires about 20 seconds per iteration.

All computations are performed in 2-and-8 floating point form. Matrix input can be either fixed point or floating point.

Method: Not given.

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
9250 Wilshire Blvd.
Beverly Hills, California

G. Smith
IBM
3545 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. Other applications include vehicle distribution, production scheduling, transshipment, and personnel assignment.

Restrictions/Range: Input consists of sources (M), destinations (N), and costs for shipping from sources to destinations. These values must be non-negative and positive for each. All calculations are performed in fixed-point arithmetic.

Maximum matrix size:

- For 20 K core: 
  - A 34 x 20 matrix with 44 iterations required four minutes for solution, plus I/O time. A 10 x 8 matrix required ten minutes for solution, plus I/O time.

Storage Requirements: For 20,000 positions of storage, matrices may be stored of the size noted in restrictions above.

Source Language: The program is coded in machine language.

Equipment Specifications: IBM 1620, 20K storage, paper tape reader, paper tape punch.

Purpose/Description: The purpose of this program is to prepare a linear program matrix for the Nichols, Nickel, Davie Card Linear Program. Machine preparation of this matrix has the following advantages:

1. Calculation errors are eliminated.
2. The input data to the MXV has physical meaning and can readily be scanned for errors.

This program performs a matrix by vector multiplication to prepare a linear program input vector. The range of multiplication, vector number assigned to output vector, and ID of output vector are all controlled by control cards which may be interspersed with matrix loading.

Mathematical Method: N/A

Restrictions/Range: The range of both equations and vectors can be specified by the user. A 34 x 20 matrix with 44 iterations required four minutes for solution, plus I/O time. A 10 x 8 matrix required ten minutes for solution, plus I/O time.

Mathematical Method: The method used is that of Ford and Fulkerson (Management Science 3 (1): pp. 24-32, October, 1956). The cost is minimized for shipping a product from a set of sources to a set of destinations, subject to a supply of the product at each source and a demand of the product at each destination. The cost is minimized for shipping a product from a set of sources to a set of destinations. Other applications include vehicle distribution, production scheduling, transshipment, and personnel assignment.

Additional Remarks: The approximate running time is 7 minutes to produce 30 time11 successfully to 6/Z.4/61. SPS two page, Fixed Point calculations, relocateable, Uses modified SPS loader for both data and program.

Linear Programming Code for the Card 1620 with Punched Card Option for Final Output (Card)

Lou Davis and Art Nickel

b. Derivation-Reference: Some (Nichols') notation and techniques were derived from the writeup of the "Linear Programming Code for the Augmented 650" by G. E. Feeny. Reference is also made to C. E. Nichols' writeup for the 1620 paper tape input/output version.

Restrictions/Range:

- Restrictions: a 1620 Card Read-Punch Unit. This program was rewritten for a 286 machine. Certain changes in the program deck are necessary to enable it to run on a 286 or 586 machine. These changes are indicated in Appendix E. The size of the problem which can be handled is restricted by the following relationship:

\[ m + n < 1620 \]

where \( m \) is the number of restrictions, \( n \) is the number of non-basis independent variables, and memory is 286, 40B, or 586.

(Continued on next page)
Sales Forecasting Simulator Using First Order Exponential Smoothing (Card)

Craig L. Johnson

Direct Inquiries to: Craig L. Johnson
IBM Corporation
1730 Cambridge Street
Cambridge, Massachusetts

Purpose/Description:

1. To provide a method for investigating the applicability of the technique of exponential smoothing for forecasting demand for a specific product.
2. To demonstrate the technique of exponential smoothing.

Method: Exponential smoothing

Restrictions/Range: Will analyze demand for twenty-four (24) periods on each run. Restrictions are normal Fortran Input/Output.

Storage Requirements: Approximately 16,500 digits

Equipment Specifications: Memory 20K, and no other special features required.

Additional Remarks: This program was modified from the 650 program written by Welker and Goodfriend and includes such things as Say statements (headings) and cost evaluations.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 10, 2, 004

AN INVENTORY MANAGEMENT SIMULATOR (Card)

C. J. Welker & C. M. Goodfriend

Direct Inquiries to: C. J. Welker
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 card size, cannot be accommodated. 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

Equipment Specifications: This program was written in the 1620 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 technique. 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

THE INVENTORY MANAGEMENT SIMULATOR (Tape)

C. J. Welker & C. M. Goodfriend

Direct Inquiries to: C. J. Welker
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 this program is for the IBM 1620 tape system.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 10, 2, 003

AN INVENTORY MANAGEMENT SIMULATOR (Card)

J. L. Spivack & Cliff Smith

Direct Inquiries to: John L. Spivack
IBM Corporation
1455 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)
CRITICAL PATH SCHEDULING (Card)

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 10.3.005

Direct Inquiries to: Chuck Snyder & Jim Sneedke

IBM Corporation

Akron 8, 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.
Restrictions/Ranges: N/A

Storage Requirements: 2500 Core Positions

Equipment Specifications: Paper Tape 1620, memory 22K and no other special features required.

Additional Remarks: The program has intellectual interest only and serves no useful function other than to demonstrate a reflective binary grey code.

IBM 1620 PROGRAM LIBRARY ABSTRACT File Number 11.0.004

THE EXECUTIVE GAME: (Tape)
E. Jury & 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 decisions and the resulting affects on the market. This program is a translation of the U.C.L.A. game for the IBM 650.

Method: N/A

Restrictions/Ranges: Eight teams

Storage Requirements: Total memory

Equipment Specifications: Memory 22K and no other special features are required.

Additional Remarks: This program is written in I.P.S. The need for an automatic divide feature will be a function of which I.P.S. tape is available. The 1620 User’s Group has permission to publish this program and preliminary writing, but its use should be restricted to members of the Group only. A more complete write-up will be available later. This has been put in this form following many requests from sources.

IBM 1620 PROGRAM LIBRARY ABSTRACT File Number 11.0.005

BLACKJACK GAME: (Tape)
A. J. Lang

Direct Inquiries to: A. J. Lang
Fairchild Camera and Instrument Corporation
Barrington, Illinois

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/Ranges: Does not apply.

Storage Requirements: 4500 core positions.

Equipment Specifications: 1620 with attached 1621. No other special features are required.

IBM 1620 PROGRAM LIBRARY ABSTRACT File Number 11.0.006

1620 BLACKJACK DEMONSTRATION: (Card) Karl E. Holt

Direct Inquiries to: Karl E. Holt
IBM Corporation
3600 Lindell Boulevard
St. Louis, Missouri

Purpose/Description: Demonstration Game of Blackjack 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 clever comments are typed out at end of each hand giving almost human image to 1620.

Method: N/A

(Continued on next column)
IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 SIMULATION OF A ONE-ARMED BANDIT (Card)
Dick Conner

Direct Inquiries to: Dick Conner
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 (S, *• @, 1, =, ¥, 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 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 same switch settings, thus affording the better a choice of different amounts to bet, from five cents to ninety cents. The sixteenth combination of switch settings causes the player's net winnings or losses to be printed in edited format, and the program to reinitialize for another player. The "house" man can at any time cause printing of grand totals of bets, payoffs and net profit for the day.

Restrictions, Range: Not given.

Method: Reusable pseudo-random number generator, partially initialized by player to prevent identical output each time the program is loaded.

Mathematical Method: Fixed point arithmetic.

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**

**7070 - Addition to Basic Fortran**
Russell Ranshaw
Computation and Data Processing Center
University of Pittsburgh
Pittsburgh 15, 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, \( x \), \( y \)  
2. IF SENSE LIGHT \( 0, x \), \( y \)  
3. SENSE LIGHT \( 1 \) ON  
4. ASSIGN \( x, y \), TO \( v \)  
5. GO TO \( v, x, y, \ldots \)

### b. Machine Requirements:
**Processor:** The additions occupy 130 locations; at present they are assembled into 5000-5119. There is room, however, in a 5K machine to make the same additions.

**Object:** Electronic switches 1-9 may be used if SENSE LIGHT instruction for "lights" 1-9 are used.

### c. General Description:
The Machine language Realizations of the above statements are:
1. IF SENSE SWITCH \( 0, x \), \( y \)  
2. IF SENSE LIGHT \( 0, x \), \( y \)  
3. SENSE LIGHT \( 1 \) ON  
4. ASSIGN \( x, y \), TO \( v \)  
5. GO TO \( v, x, y, \ldots \)

### d. Capabilities and Limitations:
- Does not apply.

---

**IBM 7070 Library Program Abstracts**

**7070 - Addition to Basic Fortran**

**Russell Ranshaw**

**7070 - Basic FORTRAN Punch With Carriage Control**

**George Greenacre**
P. O. Box 5561
South Charleston, S. W. Va.

### a. Purpose:
- IF (SENSE SWITCH \( 0, x \), \( y \)  
- IF SENSE LIGHT \( 0, x \), \( y \)  
- SENSE LIGHT \( 1 \) ON  
- ASSIGN \( x, y \), TO \( v \)  
- GO TO \( v, x, y, \ldots \)

### b. General Description:
The routine is supplied in 5/12 relocatable form, and also is available as a routine for linkage with the Basic Fortran Package documentation. Upon entry, the routine prints a record consisting of one word, having control information to cause the 7070 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**

**7070 - Basic FORTRAN Punch With Carriage Control**

**George Greenacre**
P. O. Box 5561
South Charleston, S. W. Va.

### a. Purpose:
- IF (SENSE SWITCH \( 0, x \), \( y \)  
- IF SENSE LIGHT \( 0, x \), \( y \)  
- SENSE LIGHT \( 1 \) ON  
- ASSIGN \( x, y \), TO \( v \)  
- GO TO \( v, x, y, \ldots \)

### b. General Description:
The routine is supplied in 5/12 relocatable form, and also is available as a routine for linkage with the Basic Fortran Package documentation. Upon entry, the routine prints a record consisting of one word, having control information to cause the 7070 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**

**7070 - Basic FORTRAN Punch With Carriage Control**

**George Greenacre**
P. O. Box 5561
South Charleston, S. W. Va.

### a. Purpose:
- IF (SENSE SWITCH \( 0, x \), \( y \)  
- IF SENSE LIGHT \( 0, x \), \( y \)  
- SENSE LIGHT \( 1 \) ON  
- ASSIGN \( x, y \), TO \( v \)  
- GO TO \( v, x, y, \ldots \)

### b. General Description:
The routine is supplied in 5/12 relocatable form, and also is available as a routine for linkage with the Basic Fortran Package documentation. Upon entry, the routine prints a record consisting of one word, having control information to cause the 7070 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**

**7070 - Basic FORTRAN Punch With Carriage Control**

**George Greenacre**
P. O. Box 5561
South Charleston, S. W. Va.

### a. Purpose:
- IF (SENSE SWITCH \( 0, x \), \( y \)  
- IF SENSE LIGHT \( 0, x \), \( y \)  
- SENSE LIGHT \( 1 \) ON  
- ASSIGN \( x, y \), TO \( v \)  
- GO TO \( v, x, y, \ldots \)

### b. General Description:
The routine is supplied in 5/12 relocatable form, and also is available as a routine for linkage with the Basic Fortran Package documentation. Upon entry, the routine prints a record consisting of one word, having control information to cause the 7070 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**

**7070 - Basic FORTRAN Punch With Carriage Control**

**George Greenacre**
P. O. Box 5561
South Charleston, S. W. Va.

### a. Purpose:
- IF (SENSE SWITCH \( 0, x \), \( y \)  
- IF SENSE LIGHT \( 0, x \), \( y \)  
- SENSE LIGHT \( 1 \) ON  
- ASSIGN \( x, y \), TO \( v \)  
- GO TO \( v, x, y, \ldots \)

### b. General Description:
The routine is supplied in 5/12 relocatable form, and also is available as a routine for linkage with the Basic Fortran Package documentation. Upon entry, the routine prints a record consisting of one word, having control information to cause the 7070 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**

**7070 - Basic FORTRAN Punch With Carriage Control**

**George Greenacre**
P. O. Box 5561
South Charleston, S. W. Va.

### a. Purpose:
- IF (SENSE SWITCH \( 0, x \), \( y \)  
- IF SENSE LIGHT \( 0, x \), \( y \)  
- SENSE LIGHT \( 1 \) ON  
- ASSIGN \( x, y \), TO \( v \)  
- GO TO \( v, x, y, \ldots \)

### b. General Description:
The routine is supplied in 5/12 relocatable form, and also is available as a routine for linkage with the Basic Fortran Package documentation. Upon entry, the routine prints a record consisting of one word, having control information to cause the 7070 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

IBM 7070 Tape Records Conversion (XXA15)

August 18, 1960

R. T. Miller, Jr.

Purposes:
1. To convert 650 tape records, written either alpha or numeric,
to 7070 tapes.

Machine Requirements:
1. One 1401 650 core storage tape reader, two (2) 729 II or 729 IV
2. Two (2) 7500 utility panels.

General Description:
The 7500 utility panels are used primarily as an input and output device. The
7070 is used to calculate network data, to merge activity descriptions with
input to the system in cards and the output in 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
the tape. The existence of input errors in a network will not restrict the
successful processing of other valid networks.

Hardware: 24 locations
Index accs. 81 (2, 5), 82 (2, 5)
Access 1, 2

Method: 65 - Modified characteristic * number of decimal places required

Notes: On number of decimal places required in accs. 1, 2.
The subroutine will cater for positive or negative numbers of
decimal places, therefore any modified characteristics can be
converted.

Floating point number is available at GCM at the completion of
the routine.
a. **General Description:** With the Program in storage, a priority branch to copy is issued when channel A is switched on. The routine reads the control card, sets up the tape operation, alters 0941 to enter the second phase of its routine, and returns control to the main Program. Succeeding interrupts read a data card using a 16 word RDW and interrogate the tag of the first word; if not, the output RDW is set to 8 words and a tape record written; if the tag is alphabetic, the card is checked for 16 blanks; if any non-blanks is encountered, a tape record is written; if the card is blank, a segment mark is written. In all cases, a priority response occurs after the tape is written.

b. **Capabilities and Limitations:** Does not apply.

d. **Off-line Equipment:** - THAT equipment necessary to prepare a tape suitable as input to the Condensed Card Load Program (8 word numeric records) and the Modification Routine (16 word records).

e. **General Description:** After being loaded into core storage, the Modification Routine reads an entry. The entry is then examined to see if it is an execute entry. If so, a branch to the first instruction on that entry is effected. If it is not an execute entry, a short edit is performed to insure that the format is correct (any deviation from the prescribed format will cause the entry to be disregarded). If the entry is found to be a 7070 instruction, a proper operation code is extracted if the tag field is the IX. In case a program position marked in the entry are combined and the new instruction is moved into memory as directed. If the entry is found to be a constant, the information contained in the Operand Field is moved into memory as directed.

d. **Capabilities and Limitations:** Any acceptable 7070 instruction, along with the operation follow and constants may be processed.

---

**IBM 7070 Library Program Abstracts**

**IBM 7070 Program Modification Routine**

Contributed By: Maurice K. Morin

**IBM 7070 Program Abstracts**

**IBM 7070 Program Abstracts**

**IBM 7070 Library Program Abstracts**

**IBM 7070 Library Program Abstracts**

**IBM 7070 Library Program Abstracts**

---
IBM 7070 Library Program Abstracts

7070 SIMPLE IOCS

Contributed By: Robert Jurmen
The B. F. Goodrich Company
Akron H, 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:

1) XL
2) P(Tape Command) Any channel-unit, and RDW (if applicable)

E. Capabilities and Limitations:
- In case of an uncorrectable error, priority will be released to the B++. OB operations release priority to the following instruction. This procedure facilitates debugging as priority is released without otherwise affecting machine status.
- Care must be taken to clear all final status words. 4 instructions go into 9150-9155 and 13 locations any other place are used. These can be reduced by standardizing input-output channels and reducing the error messages. All accumulators are used by the package.

IBM 7070 Library Program Abstracts

7070 MATES (MASTER Tape Executive Programs)

Author: Vincent J. Battaglia
Organization: INTERNATIONAL BUSINESS MACHINES
Chicago Downtown, 618 S. Michigan Avenue
Chicago, Illinois

a. Purpose:
The Librarian generates and maintains a master tape. It accepts programs in squeeze deck format and produces a single tape record plus an identification record for each program (or phase of a program). The Librarian obtains programs from a Library tape writer or equivalent control program.

b. Machine Requirements:
STORAGE TAPES READER

LOCATOR: 140 words 7500 or 7501
LIBRARIAN: 236 words 7500 or 7501

C. Capabilities and Limitations:
The card image input to the Librarian must be in numeric right word load format on tape. Tape density on input and output of the Librarian is at the discretion of the user.

IBM 7070 Library Program Abstracts

TAPECHECK SUBROUTINE

Contributed By: H. Byman, Applied Science
IBM Svenska AB
Gaveplan 20
Stockholm 4, SWEDEN

a. Purpose:
A subroutine for checking properly execution of tape reading and writing operations.

b. Machine Requirements:
1 electronic switch, 3 index words, locations # 97, # 98, # 100 and # 120, 50 ordinary storage locations, the priority mask registers and initial and final status words (as required by tape units used).

IBM 7070 Library Program Abstracts

Big File Generator (BFG)

Contributed By: W. J. Walker
IBM Corporation
2 Broadway
New York, N. Y.

a. Purpose:
To Generate data files from card input for use in testing 7070 programs.

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

1. 10,000 words of memory.
2. Card-to-Tape equipment to create an input tape to the BFG.
3. One 727 L, IV or 727 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:
1. The BFG is an extension of the IBM TFG program; records of the TFG type can be generated by the BFG.
2. The BFG is preferable to the regular TFG when many larger records are to be created and only a few fields will be changed from record to record.
3. The BFG program can only be used with the PILOT program Tape System.

IBM 7070 Library Program Abstracts

7070 PAT COMPILER

Contributed By: W. J. Walker
IBM Corporation
N. Y. Financial
2 Broadway
New York, 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 7099 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 channel and unit will also be typed from each TFG control card encountered. Messages may also be typed to identify each Utility Program included in the Pat tape. This typed list will be in the same sequence as the programs on tape and serve as a reference sheet during use. The PAT Compiler Call card defines the beginning of a packet and contains the necessary information for compiling of the packet.

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: Mike Clark
IBM Corporation
Los Angeles Datacenter
4242 Wilsshire 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 7500 or a 1501 Card Reader. The PAT Compiler program is available in two versions, one using the IBM 7070 DCSS system and requiring a 10K core 7070; the other not using DCSS and not requiring the 10K core 1700. Either PAT Compiler may be modified to run on any given input/output 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 card load program. During testing, the PAT Compiler System places no restriction 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.

Procedures are available within the PAT Compiler to add new programs or to delete old programs.

Each PAT Compiler program condensed deck consists of two parts: the PAT Compiler program itself, and the utility programs to be incorporated onto the PAT tape by the PAT Compiler program.

All the utility programs used by the PAT Compiler System are modified versions of the standard utility programs.

IBM 7070 Library Program Abstracts

7070 LORELIZ (Location Reference Listing)
Contributed By: Mike Clark
IBM Corporation
Los Angeles Datacenter
4242 Wilsshire Blvd.
Los Angeles 5, California

A. Purpose: LORELIZ is a program used in conjunction with a modified Fortran compiler, designed to create a cross-reference listing of programs assembled by Autocoder 74.

B. Machine Requirements: STORAGE TAPES
LORELIZ: 5000 words
SORT 9B: 5000 words

C. General Description: The cross-reference of the object program is into these major areas:
1. Listing by address
2. Listing by index 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 Alteration switches.
1. Listing by address
2. Listing by index word usage
3. Listing by Electronic switch usage
4. Listing by Accumulator usage

IBM 7070 Library Program Abstracts

7070 SCAN
Contributed By: Ronald J. Repking
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., 27 D O 's, 150 variables, etc. Subscripts are not checked.

IBM 7070 Library Program Abstracts

7070 — 650 PANEL SIMULATOR
Contributed By: C. W. Kuster & J. W. Lake
Texas Instruments Incorporated

A. Purpose: This program is designed for use in conjunction with the IBM 7070 Program which simulates the 1966 650. This program simulates the 43 panel, thus eliminating the need for wiring 7070 read and punch panels to replace the 553 panel used by the 650 programs.

B. Machine Requirements: Index words 76 through 81, electronic switches 2 through 24, and 1550 instructions and locations that may be assembled anywhere outside of the area required by the IBM 7070 Simulation Program.
The IBM 7090 Simulation Program with the Panel Simulator included can
usually be run on a 650 core machine by removing unused portions of the
program. If the entire system is required, you must have a 10X core
machine. Some of the sections which can be easily removed are: core
segment, O3 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

IBM 7070 - Simulation of Basic 650 on Basic 7070

A. Cooper (Richard King and Jim Lake)
P.O. Box 1249
Houston 1, Texas

Purpose: To simulate a basic 650 program on a basic 7070. The 650
control panel is also simulated.

Machine Requirements: An IBM 7070 with one tape channel and two 729
unit record devices. Additional magnetic tape units may be required.

This program may be used to simulate any standard 650 program on a
basic 7070. The 650 program is simulated as it is run on the 650
machine, including stepping commands, priority processors, and
output devices. The 650 program is simulated with the control cards
used in the test run on the 650 machine.

IBM 7070 Library Program Abstracts

IBM 7070 - GRONK - a 7070 Simulator for the 650

Russell Ranshaw
Computing and Data Processing Center
University of Pittsburgh
Pittsburgh 13, Pennsylvania

Purpose: GRONK is a program for the IBM 650 to simulate an IBM 7070.

Machine Requirements: An IBM 650 with 600 to 900 words of storage
space, including control cards, control panel, and core storage.

General Description: GRONK is a program for the IBM 650 to simulate
an IBM 7070. It is a 7070 simulator for the 650 machine.

IBM 7070 Library Program Abstracts

IBM 7070 SIMULATING THE CARD 650 ON A TAPE ORIENTED 7070

Contributed By: John D. Fehd
IBM Corporation
Oakland, California

Purpose: - - This program is designed to simulate card 650 programs
at speeds ranging from 2 to 3 times faster than the present IBM 650 Simu-
lator 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
prior to each 650 program and the first record must give the con-
trol setting and program number. The 7070 can be halted just prior
to each 650 program if desired (old, 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 7070 with the following results:

IBM 7070 Library Program Abstracts

IBM 7070 SIMULATING THE TAPE 650 ON A CARDSORTER 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 1481-7070 system.

Machine Requirements: Minimum of 729 B or 729 W tape drives for simulating
unit record input and output. Additional tape drives as required for tape input and output.

General Description: This operating technique combines the use of a portion of the 1481 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 Divi-
sion.

Capabilities: A card deck containing the 1481 System, the Simulator Program, and
the 650 Program is developed for each 650 Program to be simulated. A set of decks can be written on a reel of tape using a Type
1481C 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
then 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

IBM 7070 SIMULATING THE TAPE 650 ON A CARDSORTER 7070

Contributed By: John D. Fehd
IBM Corporation
Oakland, California

Purpose: - - This program is designed to simulate card 650 programs
at speeds ranging from 2 to 3 times faster than the present IBM 650 Simu-
lator 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
prior to each 650 program and the first record must give the con-
trol setting and program number. The 7070 can be halted just prior
to each 650 program if desired (old, 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 7070 with the following results:

Limiter
650 Storage
Speed
650 I/O Speed
1. Read Bound
500 Words
5,0 to 1
220 cpm, input
2. Punch Bound
1000 Words
6.4 to 1
100 cpm, output
3. Compute Bound
1000 Words
3.8 to 1
44 cpm, input

This program uses five cards per tape record and the tapes are con-
trolled by the IBM 7070 Input/Output Control System. It will not simu-
late any of the minus operation code instructions and it is restricted
to one type 533.

Each 650 program that is to be simulated will require 1401 programs
for input and output.

An operators manual and technical description will be supplied with
the program.
IBM 7070 Library Program Abstracts

**ABFLOATSIM** – Abbreviated Floating point hardware Simulator

Contributed By:

**Author:** H. Hyman, Applied Science

**Organization:** IBM Svenska AB

*Address:* Göteborgs 23

*Stockholm*, **SE**NDEN

---

**Purpose:** An interpretative subroutine which essentially simulates floating decimal hardware.

**Machine Requirements:** 2 index words and 126 ordinary storage locations.

**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. Average execution times: FA - 1.4 ms; FB, FBV, FBU - 2.0 ms; FM - 2.3 ms; FD - 4.3 ms; FEV, FF - 1.6 ms.

**Capabilities and Limitations:** The normal restrictions on the floating decimal arithmetic (described in the 7070 Reference Manual) must be adhered to. The function of accumulator 1 is not simulated and consequently neither are the double precision floating decimal operations FAD, FADS, FR and FDD.

---

**IBM 7070 Library Program Abstracts**

**IBM 7070 Linear Programming Code S1**

Contributed By:

**Author:** A. E. Speckhard

**Organization:** International Business Machines

**Western Region**

---

**Purpose:** Instrument the original simplex algorithm with variations for the IBM 7070.

**Machine Requirements:** Basic 7070 with 5K memory, on-line card reader, punch, and printer. Modifications to the basic S1 code are available to provide operation on a tape oriented system.

**General Description:** Utilizes the original simplex algorithm with variations and has the following features:

1. Provides options for negative elements in the right hand side, two phase or mixed price solution, reinversion and specification of arbitrary transformations, containing of column vectors, multiple cost rows, and multiple "in" vectors.

2. Accommodates large problems. A realistic limit is approximately 500-600 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 at option of the user. Input data is in single precision fixed point form.

**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 20,000 variables.

---

**IBM 7070 Library Program Abstracts**

**IBM 7070 Linear Programming Code S2**

Contributed By:

**Author:** D. C. Potter & A. E. Speckhard

**Organization:** International Business Machines

**Western Region**

---

**Purpose:** Instrument the revised simplex product form algorithm with variations and options for the IBM 7070.

**Machine Requirements:** Basic 7070 with 10K memory, two tape channels, two tape units per channel, on-line card reader and printer. Modifications to the basic S2 code are available to provide operation on a tape oriented system.

**General Description:** Utilizes the revised simplex product form algorithm with variations and has the following features:

1. Provides options for negative elements in the right hand side, two phase or mixed price solution, reinversion and specification of arbitrary transformations, containing of column vectors, multiple cost rows, and multiple "in" vectors.

2. Accommodates large problems. A realistic limit is approximately 500-600 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 at option of the user. Input data is in single precision fixed point form.

**Capabilities and Limitations:**

- Provide operation on a tape oriented system.
- Modification to the basic S2 code are available to provide operation on a tape oriented system.
- 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 20,000 variables.

---

**IBM 7070 Library Program Abstracts**

**IBM 7070 Linear Programming Code S1**

**IBM 7070 Linear Programming Code S2**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Library Program Abstracts**

---

**IBM 7070 Linear Programming Code S1**

Contributed By:

**Author:** A. E. Speckhard

**Organization:** International Business Machines

**Western Region**

---

**Purpose:** Instrument the original simplex algorithm with variations for the IBM 7070.

**Machine Requirements:** Basic 7070 with 5K memory, on-line card reader, punch, and printer. Modifications to the basic S1 code are available to provide operation on a tape oriented system.

**General Description:** Utilizes the original simplex algorithm with variations and has the following features:

1. Provides options for negative elements in the right hand side, two phase or mixed price solution, solution of arbitrary transformations.

2. Describes the solution completely including cost ranges with upper and lower limiting variables, and activity ranges with upper and lower limiting variables.

3. Computation is in single precision floating point.

**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 100 equations and 1,000 variables.

---

**IBM 7070 Library Program Abstracts**

**IBM 7070 Linear Programming Code S2**

Contributed By:

**Author:** D. C. Potter & A. E. Speckhard

**Organization:** International Business Machines

**Western Region**

---

**Purpose:** Instrument the revised simplex product form algorithm with variations and options for the IBM 7070.

**Machine Requirements:** Basic 7070 with 10K memory, two tape channels, two tape units per channel, on-line card reader and printer. Modifications to the basic S2 code are available to provide operation on a tape oriented system.

**General Description:** Utilizes the revised simplex product form algorithm with variations and has the following features:

1. Provides options for negative elements in the right hand side, two phase or mixed price solution, solution of arbitrary transformations, containing of column vectors, multiple cost rows, and multiple "in" vectors.

2. Accommodates large problems. A realistic limit is approximately 500-600 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 at option of the user. Input data is in single precision fixed point form.

**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 20,000 variables.
IBM 7070 Library Program Abstracts

Subroutine for IBM 7070

Arcsin x

Contributed By: Derby, England

A. Purpose: This program computes \( \sin^{-1} x \) or \( \cos^{-1} x \) for \( x \in [-1, 1] \) 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 method consists of a separation into integral and fractional parts, an evaluation of \( \sin \) and \( \cos \), and an adjustment of sign for quadrant correction. The maximum error is \( 10^{-8} \). Average execute time is 16.8 milliseconds.

D. Capabilities and Limitations: Input must be normalized floating decimal of form \( 10.00000 \times 10^{ \pm n } \) and an adjustment of sign for quadrant correction. The program requires 70 locations (excluding the square root) and can be used on all 7070 configurations. The maximum error is not greater than \( 5 \times 10^{-8} \). Average execute time is 16.8 milliseconds.

Subroutine for IBM 7070

Arcsin x

Contributed By: Derby, England

A. Purpose: This program computes \( \sin^{-1} x \) or \( \cos^{-1} x \) for \( x \in [-1, 1] \) 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 method consists of a separation into integral and fractional parts, an evaluation of \( \sin \) and \( \cos \), and an adjustment of sign for quadrant correction. The maximum error is \( 10^{-8} \). Average execute time is 16.8 milliseconds.

D. Capabilities and Limitations: Input must be normalized floating decimal of form \( 10.00000 \times 10^{ \pm n } \) and an adjustment of sign for quadrant correction. The program requires 70 locations (excluding the square root) and can be used on all 7070 configurations. The maximum error is not greater than \( 5 \times 10^{-8} \). Average execute time is 16.8 milliseconds.

Error: Max. error is 1 in 8th decimal place.
ARCTANGENT SUBROUTINE
Contributed By:
H. Hyman, Applied Science
Organization:
IBM Svenska AB
Grevageatan 20
Stockholm 6, SWEDEN

Purpose: A full precision, fixed point subroutine to compute the inverse tangent function, expressed in radians.

Machine Requirements: All accumulators, the compare indicators, 1 electronic switch, 2 index words and 90 ordinary storage locations.

General Description: The arctangent is approximated by a polynomial of the fourth degree. The constants of the polynomial are stored in a 50 word table. Accuracy: The magnitude of the maximum error is 0.000000003. Average execution time: 5.4 milliseconds.

Capabilities and Limitations: The argument X must satisfy:
-1 ≤ X ≤ 1.

7070 - Degrees to Radians Conversion
M. Roberts
AC Spark Plug Div GMC
Milwaukee, Wisconsin

Purpose: To convert degrees in floating point notation to radians.

Machine Requirements: Floating point hardware, 49 core storage words

General Description: The subroutine will convert one or a table of values.

Capabilities and Limitations: Angles to be converted must not exceed 17.4533 radians.

IBRM 7070 Library Program Abstracts
Available prior to January 1962
a. Purpose: A full precision, fixed point subroutine to compute the hyperbolic tangent.

b. Machine Requirements: All accumulators, the compare indicators, 1 electronic switch, 3 index words and 109 ordinary storage locations.

d. Capabilities and Limitations: The argument X must be of the form $-10 \leq X \leq 10$

---

**IBM 7070 Library Program Abstracts**

**MODULO Z$^{TT}$ CONVERSION SUBROUTINE**

Contributed By:

Author: S. Nordin, Applied Science

Organization: IBM Svenska AB

Gävlegatan 20

Stockholm 6, SWEDEN

a. Purpose: A double-precision, fixed point subroutine to convert numbers modulo $2^{Z}$.

b. Machine Requirements: All accumulators, 2 index words and 25 ordinary storage locations.

c. General Description: If wished, this subroutine may be used to increase the permitted range for the arguments, when using the Sine Cosine Subroutine and the Tangent-Cotangent Subroutine by the same author.

d. Capabilities and Limitations: The argument X must be expressed in radians and satisfy: $-10^{10} < X < 10^{10}$

---

**INVERSE TANGENT/COTANGENT SUBROUTINE**

Contributed By:

Author: G. J. Elliott, Applied Science

Organization: IBM Svenska AB

Gävlegatan 20

Stockholm 6, SWEDEN

a. Purpose: A full precision, fixed point subroutine to compute the principal value (in radians) of the inverse tangent or cotangent function.

b. Machine Requirements: All accumulators, the compare indicators, 2 electronic switches, 2 index words and 57 ordinary storage locations.

c. General Description: The argument is transformed to satisfy $|X| < 10^{10}$, then the Arctangent Subroutine by H. Hyman is used to compute the function. Accuracy: The maximum error is $0.000000005$. Average execution time: 6-7 milliseconds.

d. Capabilities and Limitations: The argument X must be either zero or satisfy:

$$10^{-10} < |X| < 10^{10}$$

---

**SINE AND COSINE SUBROUTINE**

Contributed By:

Author: H. Hyman, Applied Science

Organization: IBM Svenska AB

Gävlegatan 20

Stockholm 6, SWEDEN

a. Purpose: A half-precision, fixed point subroutine to compute the sine or cosine of an angle given in degrees.

b. Machine Requirements: All accumulators, 1 electronic switch, 2 index words and 92 ordinary storage locations.

c. General Description: The sine or cosine function is approximated by a polynomial of the second degree. The choice of constants in this polynomial depends on the argument value. One of 18 sets of constants is used. Accuracy: 5 decimal places. Average execution time: 2.9 milliseconds.

d. Capabilities and Limitations: The argument X must be of the form $\pm 1000000000$ and satisfy:

$$-1000000000 < X < 1000000000$$

---

**TANGENT-COTANGENT SUBROUTINE**

Contributed By:

Author: S. Nordin, Applied Science

Organization: IBM Svenska AB

Gävlegatan 20

Stockholm 6, SWEDEN

(Continued on next column)
b. Machine Requirements: All accumulators, the compare indicators, 5 index words and 113 ordinary storage locations.

c. General Description: For arguments in the interval [0.5, 1.0] the function is approximated by a polynomial of the 5th degree. The argument is divided into five intervals. In each interval the arcsine function is approximated by a polynomial of the fifth degree. In the interval [0.512, 1] the function is approximated by a polynomial of the 7th degree. Accuracy: The magnitude of the maximum error is 1.0 x 10^-9. Average execution time: 6.4 milliseconds.

dl. Capabilities and Limitations: Input must be normalized registers. The program requires 54 locations, the three Accumulators and Index Word 98.

IBM 7070 Library Program Abstracts

b. Purpose: This program computes 10X or eX in floating decimal form. Applied Programming Department IBM

a. Purpose: A full precision, fixed point subroutine to compute the hyperbolic sine, cosine or cotangent of a number.

b. Machine Requirements: All accumulators, the compare indicators, 2 index words and 101 ordinary storage locations.

c. General Description: This subroutine uses an Exponential Subroutine by T. Bath. The hyperbolic cosine and cotangent are also computed in this way for X 0.1 but otherwise X is approximated by a polynomial of the nth degree. Accuracy: The magnitude of the maximum error is 2.10^-9. Average execution time: 14.5 milliseconds.

dl. Capabilities and Limitations: The input must be normalized floating decimal. The program requires 95 locations and will alter Accumulators 1, 2, and 3. Index Word 99 and the high-new-equal indicator.

IBM 7070 Library Program Abstracts

b. Purpose: This program computes log (BASE 10 or BASE e) of X in floating decimal form.

a. Purpose: A full precision, fixed point subroutine to compute the sine or cosine function.

b. Machine Requirements: All accumulators, the compare indicators, 2 index words and 55 ordinary storage locations.

c. 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, 0.5]. The functions are approximated by the polynomials:

\[ \sin X \approx a_1 X + a_2 X^3 + a_3 X^5 + a_4 X^7 \]
\[ \cos X \approx a_2 X^2 + a_4 X^4 + a_6 X^6 + a_8 X^8 \]

Accuracy: The magnitude of the maximum error is 1.0 x 10^-9. Average execution time: 6.4 milliseconds.

dl. Capabilities and Limitations: The argument X must be expressed in radians and satisfy -15.7 X < 15.7.

IBM 7070 Library Program Abstracts

b. Purpose: This program computes 10^X or e^X in floating decimal form. Applied Programming Department IBM

a. Purpose: A full precision, fixed point subroutine to compute the sine or cosine function.

b. Machine Requirements: All accumulators, the compare indicators, 5 index words and 55 ordinary storage locations.

c. General Description: The problem may be reduced to that of calculating the functions with arguments in the interval [0, 0.5]. The functions are approximated by the polynomials:

\[ \sin X \approx a_1 X + a_2 X^3 + a_3 X^5 + a_4 X^7 \]
\[ \cos X \approx a_2 X^2 + a_4 X^4 + a_6 X^6 + a_8 X^8 \]

Accuracy: The magnitude of the maximum error is 1.0 x 10^-9. Average execution time: 6.4 milliseconds.

dl. Capabilities and Limitations: The argument X must be expressed in radians and satisfy -15.7 X < 15.7.
EXEMPLARY SUBROUTINE

Contributed By:

Author: T. Rabe, Applied Science

Organization: IBM Svenska AB

Gävleban 20

Stockholm 6, SWEDEN

a. Purposes: A full precision, fixed point subroutine to compute the exponential function.

b. Machine Requirements: All accumulators, the compare indicators, the overflow indicator for accumulator 2, 3 index words and 102 ordinary storage locations.

c. General Description: The exponential function is approximated by a polynomial of the fourth degree. The maximum error is 3 in the last digit. Average execution time is 8.4 milliseconds.

d. Capabilities and Limitations: The magnitude of the argument must be less than 10^10.

IBM 7070 Library Program Abstracts

No. 2. 007

Available prior to January 1962

NATURAL LOGARITHM SUBROUTINE

Contributed By:

Author: H. Hyman, Applied Science

Organization: IBM Svenska AB

Gävleban 20

Stockholm 6, SWEDEN

a. Purposes: A full precision, fixed point subroutine to compute the logarithm.

b. Machine Requirements: All accumulators, the compare indicators, the overflow indicator for accumulator 2, 3 index words and 102 ordinary storage locations.

c. General Description: The logarithm 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 of the maximum error is 0.000 000 03. Average execution time: 10.3 milliseconds.

d. Capabilities and Limitations: The argument X of lnX must satisfy:

\[ 10^{-14} \leq X \leq 10^{48} \]

IBM 7070 Library Program Abstracts

No. 2. 006

Available prior to January 1962

SQUARE ROOT X

Contributed By:

Author: L. R. Bor, Applied Programming Dept.

IBM

a. Purposes: This program computes the square root of a ± 0 in floating decimal form.

b. Machine Requirements: This program uses only fixed point operation codes, and can be used on all 7070 configurations.

c. General Description: The method consists of a linear approximation followed by two iterations of Newton's formula (modified). The maximum error is ±1 in the eighth place of the digitand. Average execution time: 10.3 milliseconds.

d. Capabilities and Limitations: Input must be normalized floating point numbers. Accumulators B, C, X indicators are not used.
IBM 7070 Library Program Abstracts

Square Root, Topleur Method

Rolls 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.

Accuracy: Method:

Range: (A) if maximum accuracy is required -

Calling sequence: a BLX 51, R3005

a+1 normal return, n+2

(B) if less accuracy is required, enter 00 in L.W., 52 (6, 9), where n is the number of decimal places of accuracy required, R, n £ 5.

Calling sequence: a BLX 51, R3005

a+1 error return, n £ 0

a+2 normal return.

Output: n/2 to 10 decimal places in 9992.

Space: 23 locations. Index words 51 (2, 5), 52 (4, 9).

Method: The Topleur process of successive subtraction of odd numbers. This is based on the fact that, for n ≥ 2,

x = 2^n (X - 1) 1 Z

and is the method normally used in desk machine computation.

Accuracy: When used with maximum accuracy, the maximum error is 5 in the 9th decimal place.

Timing: Average execution time is approx. 7½ 1.3 ms.

For maximum accuracy (n = 6), the time is approx. 11.1 ms.

IBM 7070 Library Program Abstracts

7070 - nth Root of X

Rolls Royce Ltd.
P.O. Box 31
Derby, England

Purpose: This subroutine computes nth root x for a single precision fixed point argument.

Range: 0 ≤ x ≤ 1, n £ 9999.

Usage: Input: x to 10 decimal places in 9992.

Output: x to 10 decimal places in 9992.

Space: 26 locations. Index words 51 (2, 5), 52 (2, 5).

Method: Balley iteration follows by two applications of Newton’s iteration method -

\[
Y_{i+1} = \frac{1}{2} \left( x \cdot \frac{1}{Y_i} + Y_i \right)
\]

Accuracy: Maximum error is 5 in the 16th decimal place.

Timing: Average execution time is approx. 12.7 ms.
Square Root Subroutines

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:

\[ \sqrt{X} = (X + \frac{1}{2} A) \text{ where initial approximation is } X = 0 \]

\[ n = 1 \cdot 10^2 \]

d. Capabilities and Limitations: Input must be normalized floating point. Maximum error is 1 in eighth place.

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 in this polynomial depends on the first two digits in the argument. One of 32 sets of constants is used. Accuracy: 5 digits. Average execution time: 2.15 milliseconds.
d. Capabilities and Limitations: The program will accept any positive argument where the first two digits are not both zeroes. The program will also accept the arguments 90 and -90.

IBM 7070 Library Program Abstracts
Available prior to January 1962

SQUARE ROOT SUBROUTINE
Contributed By:
Author: G. J. Elliott, Applied Science
Organization: IBM Svenska AB
Octagon 20
Stockholm 6, SWEDEN

a. Purpose: A full precision, fixed point subroutine to compute the positive square root of a number.
b. Machine Requirements: All accumulators, the compare indicators, 2 index words and 46 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: 8.9 milliseconds.
d. Capabilities and Limitations: Does not apply.

IBM 7070 Library Program Abstracts
Available prior to January 1962

DOUBLE PRECISION FLOATING ADD
Contributed By:
Author: R. Haertle, M. Roberts
Organization: AC Spark Plug Div GMC
Milwaukee, Wisconsin

a. Purpose: Add two 16 digit floating point numbers.
b. Machine Requirements: Floating hardware, 35 core storage words
c. General Description: The AC Spark Plug double precision add subroutine must be used with this subroutine.

d. Capabilities and Limitations: A 16 digit product is developed. The AC Spark Plug double precision add subroutine must be used with this subroutine.

IBM 7070 Library Program Abstracts
Available prior to January 1962

DOUBLE PRECISION FLOATING MUL
Contributed By:
Author: R. Haertle, M. Roberts
Organization: AC Spark Plug Div GMC
Milwaukee, Wisconsin

a. Purpose: Multiply two 16 digit floating point numbers.
b. Machine Requirements: Floating hardware, 35 core storage words
c. General Description: The AC Spark Plug double precision multiply subroutine must be used with this subroutine.

IBM 7070 Library Program Abstracts
Available prior to January 1962

DOUBLE PRECISION FLOATING DIV
Contributed By:
Author: R. Haertle, M. Roberts
Organization: AC Spark Plug Div GMC
Milwaukee, Wisconsin

a. Purpose: Divide a 16 digit floating point number by a 16 digit floating point number to obtain a 16 digit floating point quotient.
b. Machine Requirements: Floating hardware, 30 core storage words
c. General Description: The AC Spark Plug double precision divide add and multiply routines must be assembled with this routine.

IBM 7070 Library Program Abstracts
Available prior to January 1962

(Continued on next page)
**Interpolation Subroutine**

**Title:** Table Interpolation

**Author:** H. Hyman

**Organization:** IBM Svenska AB

**Purpose:** To find an interpolating using 2, 3 or 4 points.

**Method:**

- **Entry:** \[ x \text{ in } 9992 \text{ with the same alignment as } x_k. \]
- **Y** is a sequential block of floating decimal numbers.
- **Space:** 22 locations and CIBM + 10 index words.
- **Timing:** \( 0.8 \text{ milliseconds per number to be converted.} \)

**IBM 7070 Library Program Abstracts**

**Title:** Table Interpolation

**Author:** H. Hyman

**Organization:** IBM Svenska AB

**Purpose:** To find an interpolating using 2, 3 or 4 points.

**Method:**

- **Entry:** \[ x \text{ in } 9992 \text{ with the same alignment as } x_k. \]
- **Y** is a sequential block of floating decimal numbers.
- **Space:** 22 locations and CIBM + 10 index words.
- **Timing:** \( 0.8 \text{ milliseconds per number to be converted.} \)

**IBM 7070 Library Program Abstracts**

**Title:** Table Interpolation

**Author:** H. Hyman

**Organization:** IBM Svenska AB

**Purpose:** To find an interpolating using 2, 3 or 4 points.

**Method:**

- **Entry:** \[ x \text{ in } 9992 \text{ with the same alignment as } x_k. \]
- **Y** is a sequential block of floating decimal numbers.
- **Space:** 22 locations and CIBM + 10 index words.
- **Timing:** \( 0.8 \text{ milliseconds per number to be converted.} \)

**IBM 7070 Library Program Abstracts**

**Title:** Table Interpolation

**Author:** H. Hyman

**Organization:** IBM Svenska AB

**Purpose:** To find an interpolating using 2, 3 or 4 points.

**Method:**

- **Entry:** \[ x \text{ in } 9992 \text{ with the same alignment as } x_k. \]
- **Y** is a sequential block of floating decimal numbers.
- **Space:** 22 locations and CIBM + 10 index words.
- **Timing:** \( 0.8 \text{ milliseconds per number to be converted.} \)
IBM 7077 Library Program Abstracts

† IBM 7077 Library Program Abstracts

7070 MATRIX INVERSION AND SIMULTANEOUS EQUATIONS

CONTRIBUTED BY: W. W. Marks and Gordon Smith
IBM Corporation
Los Angeles, Wilshire

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 491 locations.

IBM 7077 Library Program Abstracts

† IBM 7077 Library Program Abstracts

SINGLE PRECISION MATRIX INVERSION

Contributed By:

Author: H. Ryman, Applied Science
Organization: IBM Svenska AB

© Olivestad 20
Stockholm 6, SWEDEN

a. Purpose: A single precision, floating point program for a 7070 without floating decimal hardware to invert a matrix and solve systems of linear equations.

b. Machine Requirements: All accumulators, the compare indicators, the priority mask registers, 1 electronic switch, 12 index words, storage locations, a storage area for the augmented matrix and one or two tape units.

c. General Description: The program uses the pivot elimination method of Jordan, and will automatically select a non-zero pivot element. The program may also be used to solve an arbitrary number of systems of equations, where the coefficients of the unknowns are given by the matrix to be inverted. Average execution time is approximately 3.2 * n² milliseconds, where n is the order of the matrix and b is the number of systems.

d. Capabilities and Limitations: Let n be the order of the matrix and b the number of systems of equations. The restrictions are then as follows:

a. 5,000 word machine: n < 57, n(xb) < 4380
b. 10,000 word machine: n < 57, n(xb) < 9730

IBM 7077 Library Program Abstracts

† IBM 7077 Library Program Abstracts

Solution of Simultaneous Linear Equations

M. Roberts
AC Spark Plug Div, GMC
Milwaukee, Wisconsin

a. Purpose: To find x1, x2, ..., xₙ of the following equation set:

\[ a₁x₁ + a₂x₂ + ... + aₙxₙ = b₁ \\
\[ a₂x₁ + a₃x₂ + ... + aₙxₙ = b₂ \\
\[ \vdots \\
\[ aₙx₁ + a₁x₂ + ... + aₙ₋₁xₙ = bₙ \]

b. Machine Requirements: Floating hardware, approximately 200 words plus the matrix area are the storage requirements.

c. General Description: Crout's Reduction

d. Capabilities and Limitations: Input must be in normalized floating form. Accumulators and indicators are not saved.

IBM 7077 Library Program Abstracts

† IBM 7077 Library Program Abstracts

7070 SLEEP, SOLVE SIMULTANEOUS LINEAR EQUATIONS WITH PIVOTING

Contributed By: Robert Judson
The B. F. Goodrich Co.
Akron 18, 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 DFM locations for N equations. Location PV must not be disturbed.

D. Method: Elimination to solution form followed by back solution.

E. Source Language Entry: BLK LINK, SOLVE with equations stored sequentially by rows. a 11 is in location PV1 and n in accumulator No. 1, right justified. Solution will be available in locations original right hand vector, i.e., Xᵢ in PV₁+1H, Xᵢ in PV₂H, etc.

IBM 7077 Library Program Abstracts

† IBM 7077 Library Program Abstracts

7070 STEPWISE MULTIPLE REGRESSION ANALYSIS, MR1

Contributed By:

Author: Gary Lotto
Organization: University of Pittsburgh

Computation and Data Processing Center
University of Pittsburgh
Pittsburgh 12, Pennsylvania

a. Purpose: This program will report the results of a multiple regression analysis for up to 150 variables. Independent variables are introduced one at a time in the order that they contribute to regression on the dependent variable.

b. Machine Requirements: The program is written for 10K machine with floating point hardware. It may be modified for fixed point hardware, a SK or LOK machine, etc. Storage used is a function of the number of variables included. Output is printed or punched. Input is on cards or tape.

c. General Description: During each step, a variable is included or deleted, and the correlation matrix either "reduced" or "increased" from the effects of this operation, in such a way that the same logarithm may be used on one printout as on any other. Independent variables are introduced one at a time in the order that they contribute to regression on the dependent variable.

d. Capabilities and Limitations: This program provides means, standard deviations, correlation coefficients, and simple correlation coefficients for up to 40 variables. The operator may, by manual intervention, prohibit certain independent variables from entering into regression, force inclusion or deletion of certain variables, change the dependent variable, or change the significance levels for inclusion or deletion at any time.

IBM 7077 Library Program Abstracts

† IBM 7077 Library Program Abstracts

7070 MULTIPLE LINEAR REGRESSION BY THE STEPWISE METHOD

CONTRIBUTED BY: R. E. Boss
Systems Engineer
Los Angeles, Wilshire

December, 1960

SPECIFICATIONS: This program provides means, standard deviations and simple correlation coefficients for up to 40 variables. This is the limiting number of this version, however, it can be extended by modifying the FORTRAN dimension statement and recompiling.

(Continued on next page)
Contributed By

Gary Lotto

University of Pittsburgh
Computations and Data Processing Center
University of Pittsburgh
Pittsburgh, Pennsylvania

IBM 7070 Library Program Abstracts

Available prior to January 1962

7070 INTERCORRELATION MATRIX - CORR2 - FOR CARD INPUT

Contributed By

Gary Lotto

University of Pittsburgh
Computations and Data Processing Center
University of Pittsburgh
Pittsburgh, Pennsylvania

Purpose: This program will report the vector of means and standard deviations, the number of cases, and the symmetric matrix of correlations between every variable and every other of a set of up to 130 variables.

Capabilities and Limitations: The program will handle up to 130 variables (up to 95 variables on a SK machine) with the restriction that the maximum sum of squares (treating the data as whole numbers) must be less than 10^10. The matrix is left in storage for further analysis if desired.

IBM 7070 Library Program Abstracts

Available prior to January 1962

7070 - Principal Axis Factor Analysis

Contributed By

A. W. Bendig

University of Pittsburgh

Purpose: To compute the eigenvalues and eigenvectors of a square symmetric matrix of size V.

Capabilities and Limitations: The program will handle up to 130 variables (up to 95 variables on a SK machine) with the restriction that the maximum sum of squares (treating the data as whole numbers) must be less than 10^10. The matrix is left in storage for further analysis if desired.

IBM 7070 Library Program Abstracts

Available prior to January 1962

Sequential Multiple Linear Regression Analysis on the IBM 7070

Contributed By

Donald G. Wyman

IBM Corporation

Purpose: To solve for the coefficients in a regression equation using an analysis of variance to select only the variables which meet a prescribed significance test.

Capabilities and Limitations: The program will handle up to 72 variables. Phase 1 reads and transforms input and forms simple correlation for up to 72 variables. Phase 2 solves for the coefficients, either directly or stepwise, from any system of equations formed as a subset of the 72 variables to a maximum of 55 independent and one dependent.

303
Multiple Correlation and Regression Analysis by the Stepwise Method. 

Contributed By:

Author: R. E. Bosse
Organization: IBM Corporation

Purpose: The program provides means, standard deviations, and simple correlation coefficients for all variables.

Intermediate results include those variables in the regression, and the variable added to the equation to improve the "goodness of fit" of each step.

Other results include the standard error of each regression coefficient and the error of estimate of the dependent variable, a multiple correlation coefficient, and a comparison of actual data and predicted values. Variable transformations are available.

b. Equipment Specifications:
(a) S, 000 or 10, 000 word 7070
(b) On-line card reader
(c) Minimum of three tapes

c. Source Language: FORTRAN

d. Time Required: (2n2logn) additions and multiplications and (n2log2n) divisions

e. Accuracy: Single precision floating point.

IBM 7070 Library Program Abstracts

1.3.007

Random Numbers and Random Normal Deviates Generator

Contributed By:

Author: P. Dickerman
Organization: IBM Corporation

Purpose: The AC Spark Plug log and line routings must be used with this subroutine.

Capabilities and Limitations: Does not apply.

IBM 7070 Library Program Abstracts

1.7.002

The Inventory Management Simulator-7070 Full Fortran Version

Contributed By:

Author: C. Z. Walker
Organization: IBM Corporation

Purpose: This program allows the user to test inventory replenishment rules and demand forecasting techniques; the objective is to prove the validity of methods which can then be installed in the inventory operation system.

b. Machine Requirements: Include machine components, special features, storage requirements, carbon panels—standard or special. 10K core memory, card reader, printer, from one to five tape drives (dependent upon subprogram configuration used).

IBM 7070 Library Program Abstracts

1.7.001

The Inventory Management Simulator-7070 Full Fortran Version

Contributed By:

Author: C. Z. Walker
Organization: IBM Corporation

Purpose: This program allows the user to test inventory replenishment rules and demand forecasting techniques; the objective is to prove the validity of methods which can then be installed in the inventory operation system.

b. Machine Requirements: Include machine components, special features, storage requirements, carbon panels—standard or special. 10K core memory, card reader, printer, from one to five tape drives (dependent upon subprogram configuration used).

c. General Description: Mathematical method, accuracy, speed, and appropriate Mathematical method, simulation accuracy not applicable. Speed: Running times vary considerably depending upon the subprogram configuration used. However, eighty to one-hundred demand transactions per minute can serve as a reasonable estimate.

d. Capabilities and Limitations: The subprogram package allows this program to be adapted to many inventory situations. In addition the program structure is such that the user can readily incorporate his own program variations; thereby tailoring the simulator to meet his requirements.
IBM 7070 Library Program Abstracts

7070 - Transportation Problem (Dennis Technique)
Robert Judson
The B. F. Goodrich Company
Dept. 6973 - 3dgy, 24-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 paths, 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 500 and 750 x 575. Program will be furnished in Symbolic Autocoder form so that it can be readily modified for a 10K or larger memory.
c. Timing: 118 x 12 Approx. 70 seconds with 1/3 costs excluded
12 x 118 Approx. 90

c. 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
(KFG No. 79102) and Main Program (KFG No. 79101)
so as to facilitate adaptation by users with card oriented equipment.

IBM 7070 Library Program Abstracts

7070 Management Decision-Making Exercise
Contributed By:

Author: John A. Flint
Organization: IBM Corporation

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
1-4 729-U or IV Tapes (Channel 1 only)
10 K Storage
Peripheral printer (120 or 1441)

c. General Description: The exercise has been modeled after the business strategy game constructed by Richard Bellman, Franco Rincardi, 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 outcomes 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.
FOURTH ORDER SINGLE OR DOUBLE EQUAL INCREMENT INTERPOLATION

DETERMINESlland REMAINING VARIABLES/QUALITIES SPECIFIC COMPUTES ENTHALPY OF SAT. LIQ. AS FUNCTION OF TEMPERATURE COMPUTES ENTHALPY AND ENTROPY OF COMPRESSED LIQUID AS EXIT PRESSURES AND EITHER INLET TEMPERATURE OR INLET FUNCl'IONS OF PRESSURE AND TEMPERATURE

ENTHALPY OR ENTROPY IN LIQUID SUPERHEAT OR WET REGIONS COMPUTES ENTHALPY ENTROPY SPECIFIC VOLUME OF SATURATED VAPOR ENTROPY SPECIFIC VOLUME OF SUPERHEATED STEAM ENTHALPY OF SATURATED LIQUID

7090 1095WHISL AVAILABLE PRIOR TO JANUARY 1962 VISCOSITY OF LIQUID WATER COMPUTES VISCOSITY OF LIQUID. CORR. 1225

7090 1095WHISV AVAILABLE PRIOR TO JANUARY 1962 VISCOSITY OF STEAM COMPUTES VISCOSITY OF STEAM AS FUNCTION OF PRES., AND TEMP.

7090 1095WHSL AVAILABLE PRIOR TO JANUARY 1962 SPECIFIC VOLUME OF SATURATED LIQUID COMPUTES SPEC. VOL. OF SAT. LIQ. AS FUNCTION OF TEMPERATURE

7090 1095WHSE AVAILABLE PRIOR TO JANUARY 1962 MINIMUM ERROR ROUTINE FOR STEAM TABLE DISTRIBUTION ERROR FACILITY FOR WH STEAM TABLES

7090 1095WHSD AVAILABLE PRIOR TO JANUARY 1962 THERMODYNAMIC PROPERTIES OF WATER AND STEAM A COLLECTION OF FORTRAN TYPE SUBROUTINES TO ALLOW THE COMPUTATION OF VARIOUS THERMODYNAMIC PROPERTIES. ENTHALPY, ENTROPY, TEMPERATURE, PRESSURE SPECIFIC VOLUME, QUALITY, AND VISCOSITY OF WATER AND STEAM ON THE 709 OR 7090.

7090 1113ABMTR AVAILABLE PRIOR TO JANUARY 1962 MULTIPLE TAPE TEST ROUTINE THIS SELF LOADING ROUTINE CAN TEST UP TO 20 BLANK TAPES AT ONE TIME USING EITHER OR BOTH CHANNEL A AND CHANNEL B.

7090 1110MPMDS AVAILABLE PRIOR TO JANUARY 1962 OFFLINE EDIT FOR FORTRAN MONITOR WITH SOURCE LANG DEBUG THIS CORRECTION PROVIDES A NEW OFF LINE EDITOR FOR THE PREVIOUSLY DISTRIBUTED DEBUG PACKAGE OF THE FORTRAN MONITOR. THE EDITOR WAS PREPARED BY REPLACING THE IBM COLUMN EDITOR RECORDS 8,4,7,7,7,4,3,4,3,4,4,1,5,7,1,7/1,7/1/1,7/1/1/1 WITH THE DEBUG PACKAGE CORR. 1245

IBM 7090 PROGRAM LIBRARY ABSTRACT

ONE PHASE MONITOR SYSTEM, A MONITOR PROGRAM COMPOSED OF SIX A/8 MAJOR PROGRAMS.

REQUIRES A TWO CHANNEL 36K MACHINE, 7090 OR 709 WITH DATA CHANNEL TRAPS, NORMAL OPERATION USES NINE TAPES.

SUBMITAL IS CONTAINED ON FIVE A/8 TAPES, A HIGH DENSITY BINARY SYSTEM TAPE, TWO SYMBOLIC TAPES, AND TWO LISTING TAPES CORR 1962

7090 1095WHHL AVAILABLE PRIOR TO JANUARY 1962 ENTHALPY AND ENTROPY OF COMPRESSED LIQUID COMPUTES ENTHALPY AND ENTROPY OF COMPRESSED LIQUID AS FUNCTIONS OF PRESSURE AND TEMPERATURE

7090 1095WHHS AVAILABLE PRIOR TO JANUARY 1962 ENTHALPY OF SATURATED LIQUID COMPUTES ENTHALPY OF SAT. LIQ. AS FUNCTION OF TEMPERATURE

7090 1095WHHS AVAILABLE PRIOR TO JANUARY 1962 ENTHALPY ENTROPY SPECIFIC VOLUME OF SUPERHEATED STEAM COMPUTES ENTHALPY, ENTROPY, AND SPECIFIC VOLUME OF SUPERHEATED STEAM AS FUNCTIONS OF PRESSURE AND TEMP.

7090 1095WHSV AVAILABLE PRIOR TO JANUARY 1962 ENTHALPY ENTROPY SPECIFIC VOLUME OF SATURATED VAPOR COMPUTES ENTHALPY, ENTROPY, AND SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTIONS OF PRESSURE

7090 1095WHsd AVAILABLE PRIOR TO JANUARY 1962 ENTHALPY OF SATURATED LIQUID COMPUTES ENTHALPY OF SAT. LIQ. AS FUNCTION OF TEMPERATURE

IBM 7090 PROGRAM LIBRARY ABSTRACT

ONE PHASE MONITOR SYSTEM, A MONITOR PROGRAM COMPOSED OF SIX A/8 MAJOR PROGRAMS.

REQUIRES A TWO CHANNEL 36K MACHINE, 7090 OR 709 WITH DATA CHANNEL TRAPS, NORMAL OPERATION USES NINE TAPES.

SUBMITAL IS CONTAINED ON FIVE A/8 TAPES, A HIGH DENSITY BINARY SYSTEM TAPE, TWO SYMBOLIC TAPES, AND TWO LISTING TAPES CORR 1962

7090 1095WHHL AVAILABLE PRIOR TO JANUARY 1962 ENTHALPY AND ENTROPY OF COMPRESSED LIQUID COMPUTES ENTHALPY AND ENTROPY OF COMPRESSED LIQUID AS FUNCTIONS OF PRESSURE AND TEMPERATURE

7090 1095WHHS AVAILABLE PRIOR TO JANUARY 1962 ENTHALPY OF SATURATED LIQUID COMPUTES ENTHALPY OF SAT. LIQ. AS FUNCTION OF TEMPERATURE
CRITICAL PATH AND RESOURCE SUMMARY CALCULATION

GENERALIZED INTEGRATION SUBROUTINE
A set of simultaneous ordinary differential equations is solved using either Runge-Kutta or one of several sets of predictor-corrector formulas. Predictor-corrector formulas are started with Runge-Kutta points. A variable integration interval with error control can be used optionally with predictor-corrector formulas. Uses 757 locations.

TIME SERIES DECOMPOSITION AND AJUSTMENT
A program to adjust seasonal and irregular time series to a form that shows primarily the trend-cyclical movements, seasonal factors, irregular fluctuations and many summary measures useful in time series analysis are computed in the process. Basically adaptation of Tennessee Valley Authority program. For each series to be pre-processed, fit adjusting for delivery days and fit fitting least squares trend line as forecasting aid. Corr. 11576.

TAYLOR SERIES RATIONAL FUNCTION CURVE FITTING

LEAST SQUARE CURVE FITTING

TAYLOR SERIES INTEGRATION METHOD
Computes the offset circle probability function. Also extended to generalized circular and elliptical coverage function. AVERAGE TIME - 6 MILLISECONDS PER CASE. Corr. 983.

STEPHEWYN PROGRAMMING SYSTEM - DECISION TO CALL. LP950 is a complete programming and operating system including a complete reference manual. The system contains an assembler, interpreter, compiler, and dump. Use descrip."...-9160. The system contains an assembler, interpreter, compiler, and dump. Use descrip."...-9160. The system contains an assembler, interpreter, compiler, and dump. Use descrip."...-9160. The system contains an assembler, interpreter, compiler, and dump. Use descrip."...

IBM 7090 PROGRAM LIBRARY ABSTRACT

GAMMA, XI-GAMMA, XI, Gamma, Xi, where Gamma, Xi is defined as the integral from 0 to infinity of exp(-u) times u to the Gamma, Xi power du. Subroutine also evaluates the Poisson term and finds the upper limit of the integral. Accuracy of results is 6 decimal places. 130 regression variables. Requires 32K core and 3 tapes.

IBM 7090 PROGRAM LIBRARY ABSTRACT

HILLSEC<10NDS PER CASE. Corr. 983.

ORDINARY DIFFERENTIAL EQUATION SOLUTION - FORM OF SOLUTION

IBM 7090 PROGRAM LIBRARY ABSTRACT

HILLSEC<10NDS PER CASE. Corr. 983.

ORDINARY DIFFERENTIAL EQUATION SOLUTION - FORM OF SOLUTION

IBM 7090 PROGRAM LIBRARY ABSTRACT

HILLSEC<10NDS PER CASE. Corr. 983.

ORDINARY DIFFERENTIAL EQUATION SOLUTION - FORM OF SOLUTION
(1) Code Originated by: Atomics International Division of North American Aviation, Inc.
(2) Computer: 7090 (FORTRAN)
(3) Description of Code: (Indicated status, if known)
·1090 Nuclear Code
To adjust cross-section data based on data from a critical experiment involving fission foils and oscillator measurements. In use, available.
(4) References:
~Nuclear Codes" 7090 Nuclear Code
(1) Code Originated by: Atomics International
(2) Computer: 7090 (FORTRAN)
(3) Description of Code: The basic purpose of this code is to compare the costs of various fuel cycles. AIMFIRE uses non-spatial two-group theory to predict keff as a function of burnup. Options are available by which changes in certain heterogeneous effects with burnup can be taken into account. The code contains a library of fast and thermal microscopic cross-sections, decay constants, and fission yields for 40 isotopes. The present version is designed to investigate uranium fuel systems.
(4) Restrictions or Limitations:
There must be no more than 101 spaces nor more than 18 energy groups. Only downscattering is permitted, but can be from a given group to any lower group.
(5) Approximate Performance:
For a 16 group, 101 mesh point problem, 3 minutes would be a typical time for a single problem, although times may be as low as 30 seconds.
(6) Reference:
(7) Material Available:
1. NAA-SR-6706 Addendum.
2. FORTRAN source deck.
Note: The information given above was abstracted from NAA-SR-6706.

7090 Nuclear Code
AIMFIRE
(1) Code Originated by: Atomics International
(2) Computer: 7090 (FORTRAN, FAP)
(3) Description of Code: 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 ("Some Topics in Reactor Kinetics", Sec. Geneva Conf., p. 629, 1958).
(4) Restrictions or Limitations:
The maximum number of differential equations that can be solved simultaneously is 58. Within this limitation, there may be 1 delayed neutron groups, n < 12, and n other linear feedback equations, n < 54.
(5) Approximate Performance:
About 2 seconds per cycle, each cycle divided into three parts.
(6) References:
(7) Material Available:
1. NAA-SR-MEMO 4980 and Addendum.
2. FORTRAN-FAP source deck.
Note: The information given above was abstracted from NAA-SR-MEMO 4980.

7090 Nuclear Code
AIM-6
(1) Code Originated by: Atomics International
(2) Computer: 7090 (FORTRAN, FAP)
(3) Description of Code: 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 single parent-daughter decay schema or by a Way-Wigner type relationship.
(4) Restrictions or Limitations:
A 32K memory is required.
(5) Approximate Performance:
There must be no more than 101 spaces nor more than 18 energy groups. Only downscattering is permitted, but can be from a given group to any lower group.
(6) References:
(7) Material Available:
1. NAA-SR-MEMO 4822.
2. FORTRAN source deck.

7090 Nuclear Code
AIM-6
7090 Nuclear Code

FOG

(1) Code Originated by: Atomics International

(2) Computer: 7090 (FORTRAN)

(3) Description of Code:
The FOG code is one-dimensional neutron diffusion theory code. The difference equations used are designed in conserve neutrons in cylindrical and spherical geometry. The principal options available include calculation of the adjoint flux, five 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 buckling 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 239 mesh points and 40 regions. Scattering is permitted only to the next lower group.

(5) Approximate Performance:
Varies widely, but execution time may generally be expected to be less than 30 seconds.

(6) References:

(7) Material Available:
1. NAA Program Description.
2. FORTRAN source deck.

FORM

(1) Code Originated by: Atomics International

(2) Computer: 7090 (FORTRAN)

(3) Description of Code:
The FORM, or FORM-MUFT, code is a fourier transform slowing-down code quite similar to the MUFT-4 code, but containing some additional options, including the option of changing cross sections in the F4 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 1.4 seconds.

(6) References:

(7) Material Available:
1. NAA-SR-MEMO 5764.
2. FORM source deck.

Note: The information given above was abstracted from NAA-SR-MEMO 5764.

FORTRAN SNG

(1) Code Originated by: General Dynamics Corporation General Atomic Division

(2) Computer: 7090 (FORTRAN)

(3) Description of Code:
Calculates low- and multi-group cross-sections using the F1 equations. A full scattering matrix is included for both F1 and F2 scattering terms. Resonance absorption is treated by the methods developed by L. W. Nordheim.

(4) Restrictions or Limitations:
A maximal problem requires about 1 minute on the 7090.

(5) References:

(6) Material Available:
1. NAA Program Description.
2. FORTRAN source deck.

GAM-1

(1) Code Originated by: General Dynamics Corporation General Atomic Division

(2) Computer: 7090 (FORTRAN)

(3) Description of Code:
Calculates low- and multi-group cross-sections using the F1 equations. A full scattering matrix is included for both F1 and F2 scattering terms. Resonance absorption is treated by the methods developed by L. W. Nordheim.

(4) References:

(5) Material Available:
1. NAA-SR-MEMO 5764.
2. FORM source deck.

GRACE-I

(1) Code Originated by: Atomics International

(2) Computer: 7090 (FORTRAN)

(3) Description of Code:
GRACE-I is a multigroup, multiregion, gamma-ray attenuation code designed primarily for computing gamma-ray heating and gamma-ray dose rates in multigroup finite or semi-infinite slab shields. A different buildup factor may be specified for each source region considered.

(4) Restrictions or Limitations:
If a 704 is used, at least 8K memory is required. As many as 10 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.

(Continued on next column)
(5) Approximate Performance: A sample problem involving 1 source region, 9 mesh points and 1 energy group required 65 minutes on the 709.

(6) Reference: 

(7) Material Available: 
1. NAA-SR-3719 (a listing of the FORTRAN source program 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 point source or a cylindrical source. The source, which may be located in either the central region of the system or a cylindrical 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 704, at least a 14K memory is required. As many as 20 regions, 10 mesh points per region, 20 gamma-ray energy groups, 20 shield materials, and 20 material buildup factors may be included in a single calculation.

(5) Approximate Performance: A sample problem required 1.64 minutes on the 709.

(6) Reference: 

(7) Material Available: 
1. NAA-SR-Memo 4449. 
2. FORTRAN source deck.

---

#DG 2-90 1990 Nuclear Code

(1) Code Originated by: International Business Machines Corporation

(2) Computer: Language

(3) Description of Code: Description of PG-2 which eliminates need for use of compatibility package. Handles up to 3000-3000 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 FOR Codes. Punchout card output from these codes is used as input to the PERT code. Using cross section data, fluxes, and adjoint fluxes, the relation change in kinf may be calculated. Cross sections may be weighted with the adjoint flux and/or flux. The neutron lifetime for the delay groups may also be calculated.

(4) Restrictions or Limitations: A linear perturbation theory is used for the calculations of the relative change in kinf. (Continued on next column)

---

(5) Approximate Performance: Generally less than 30 seconds for an 18 group problem.

(6) Reference: 

(7) Material Available: 
1. NAA Program Description. 
2. FORTRAN source deck.

---

PREP NORC Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: NORC

(3) Description of Code: Elastic scattering transfer cross-sections are calculated using mass no., 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 jou.


SAIL 7090 Nuclear Code

(1) Code Originated by: Atomics International

(2) Computer: 7090 (FORTRAN)

(3) Description of Code: Limitations: The code is limited to a single energy group, 100 regions, 100 intervals, and planar geometry. The order of approximation must be 2, 4, 8, or 8.

(4) Approximate Performance: The running time is generally less than one minute. A sample S1 problem involving 7 mesh points required 21 seconds, including loading the program into memory.

(5) Reference: 

(7) Material Available: 
1. NAA Program Description. 
2. FORTRAN source deck.

---

SIZZLE 7090 Nuclear Code

(1) Code Originated by: Atomics International

(2) Computer: Language

(3) Description of Code: Description of Code: The code is limited to a single energy group, 100 regions, 100 intervals, and planar geometry. The order of approximation must be 2, 4, 8, or 8.


---

B - 7090 Nuclear
(1) **Code Originated by:**
Atomic International

(3) **Description of Code:**
This code solves the one-dimensional non-energetic Boltzmann equation in cylindrical geometry, using the S4 approximation. In addition to the flux distribution, cell-averaged parameters are computed. An input guess to the flux may be used or a diffusion calculation may be performed 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 100 regions and 400 intervals. With these dimensions, a 32K memory is required.

(5) **Approximate Performance:**
About 15 seconds for a 50 mesh point problem.

(6) **References:**
2. FORTRAN source deck.

---

**NAME:** TEMPEST-II

**Code Originated by:**
Atomic International

**Computer:** IBM 7090

**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 Neutro equation. The code provides microscopic and macroscopic cross-section averages over the thermal neutron spectrum.

**Restrictions or Limitations:**
A 32K memory is required.

**Approximate Performance:**
About 75-120 seconds.

**References:**
2. FORTRAN source deck.

---

**NAME:** TWENTY GRAND

**Code Originated by:**
Aerojet-General Nuclear

**Computer:** IBM 7090

**Description of Code:**
The Twenty Grand program for the IBM 7090 is capable of solving neutron diffusion problems in cylindrical or slab geometry for one to six groups. Up to 5000 mesh points may be used. Neutron transfer from any group to any other group is permitted. Leakage in the third dimension in X-Y geometry may be treated by a buckling which can vary with region and group. Three types of symmetry conditions may be handled automatically. The zero flux, zero derivative, and logarithmic boundary conditions are available.

**Restrictions or Limitations:**
Restrictions must be isotropic.

**Approximate Performance:**
One and one-half hours for 6 group, 1000 mesh points on the 7090 (using the binary editor).

**References:**

---

**NAME:** DABY

**Code Originated by:**
Aerojet-General Nuclear

**Computer:** IBM 7090

**Description of Code:**
The DABY program solves the homogeneous or heterogeneous multi-group transport equation in xy or zr geometry. Vacuum, surface source, or reflecting boundary conditions are available as options. In the homogeneous case the user may request the computation of reactivity, period, critical concentrations of some composition or the critical thickness of a zone. The S3 approximation is used.

**Restrictions or Limitations:**
Restrictions must be isotropic.

**Approximate Performance:**
One and one-half hours for 6 group, 1000 mesh points on the 7090 (using the binary editor).

**References:**