1620 USERS GROUP
WESTERN REGION
MINUTES OF THE MEETING
JUNE 17-19, 1964
DENVER, COLORADO

ROBERT R. WHITE
WESTERN REGION SECRETARY

CONTENTS

20. A Load-and-Go SPS with Monitor Control; Kenneth M. Lochner and Glenn R. Ingram

21. Examples of 1620 Use in College Administration; Noel T. Smith

22. Automatic Processing of Autospot and Automap Programs with the 1620-1311 Disk System; Jack T. Dunn and Ernie G. Moore

23. General Ray Trace Program; D. H. O'Herren (presented at Tempe, Arizona, December 1963)

1620 USERS GROUP WESTERN REGION
SUMMER MEETING
JULY 17.18.19. 1964 DENVER. COLORADO

ROSTER OF ATTENDEES

1032 MRS. BETTY CILSICK
GEORGIA STATE COLL.
ATLANTA.GA.

1084 LONA HECKART
AEROSPACE CORP.
PATRICK AFB..FLA.

1118 NANCY PAQUIN
PUBLIC HEALTH SERV.
ROCKVILLE.MD.

1118 JAMES E. DALY
PUBLIC HEALTH SERV.
ROCKVILLE.MD.

1170 LANNY L. HOFFMAN
GUGGENHEIM LABS
PRINCETON.N.J.

1177 LEON P. GOLDBERG
PRINCTON U. PROJ ACC
PRINCETON.N.J.

1238 RENE SEUIGNY JR.
HAYES INTERNATIONAL
HUNTSVILLE.ALA.

1238 W. S. HAMER
HAYES INTERNATIONAL
HUNTSVILLE.ALA.

1258 FRANCES K. DURKAN
U.S.A.E.C.
NEW YORK.N.Y.

1273 JUDITH KOERNER
ARGONNE NAT. LAB.
IDAHO FALLS.IDAHO

1290 ROBERT D. WEEMS
N.C. STATE COLLEGE
RALEIGH.N.C.

1352 WALKER R. HURD
DPI.BOARD OF EDUC.
MEMPHIS.TENN.

3016 THOMAS P. SODANO
ITT FEDERAL LABS
FORT WAYNE.IND.

3041 ARTHUR P. WOODS JR.
ARMCO STEEL CORP
MIDDLETOWN.OHIO

3053 EUGENE C. EWING
NAT COOP REFINERY AS
MC PHERSON.KAN.

3055 BARNEY T. WATSON
VA HOSPITAL
OMAHA.NEB.

3059 J. RICHARD BURROWS
H.D.+R..ENGINEERS
OMAHA.NEB.

3082 PAUL A. BICKFORD
OU MED RES COMP CNTR
OKLAHOMA CITY.OKLA.

3082 E. N. BRANDT JR.M.D.
OU MED RES COMP CNTR
OKLAHOMA CITY.OKLA.

3082 LEE HENDERSON
OU MED RES COMP CNTR
OKLAHOMA CITY.OKLA.

3089 CLARENCE B. GERMAIN
CARTE CORP
ST. PAUL.MINN.

3096 FLOYD M. CORE
SEISMOGRAPH SERVICE
TULSA.OKLA.

3107 S. THOMAS PARKER
KANSAS STATE UNIV.
MANHATTAN.KAN.

3108 LEORA THOMAS
PIONEER HI-BRED CORN
DES MOINES.IOWA

3118 J. D. PENDERGRASS
SOUTHWESTER POWER AD
TULSA.OKLA.

3136 EVERETT L. COOK
UNIV. OF WICHITA
WICHITA.KAN.

3206 LOLAFAYE COYNE
MENNINGER FOUNDATION
TOPEKA.KAN.

3216 J. ROBERTS BRITTON
HAWAIIAN ELEC. CO.
HONOLULU.HAWAII

3220 LOUIS BOVITZ
HIBBING AREA TECH.
HIBBING,MINN.

3227 JEROME F. FOECKE
K.C. BOARD OF EDUC.
KANSAS CITY,MO.

3242 RICHARD V. ANDREE
UNIV. OF OKLAHOMA
NORMAN,OKLA.

3242 JACK L. MORRISON
UNIV. OF OKLAHOMA
NORMAN,OKLA.

3273 CHARLES WEISS
USAF CHART + INFO
ST.LOUIS,MO.

3277 ROGER GUDOBBA
LAFAYETTE CLINIC
DETROIT,MICH.

3299 LEO DOUGLAS
ARGONNE NAT. LAB.
ARGONNE,ILL.

3304 CAROL J. BILLINGHAM
LEONARD REFINERIES
ALMA,MICH.

3326 JACK T. DUNN
AVCO CORPORATION
HUNTSVILLE,ALA.

5001 W. W. JONES
ACF INDUSTRIES
ALBUQUERQUE,N.M.

5018 JAMES W. HUNTER
LA COUNTY ENGINEERS
LOS ANGELES,CAL.

5020 DR FRANKLIN GRAYBILL
COLORADO STATE UNIV.
FORT COLLINS,COLO.

5021 JAMES C. IRVIN
US ARMY,CORPS ENGR.
ALBUQUERQUE,N.M.

5027 MARVIN RUBENSTEIN
ELECTRO OPTICAL SYS
PASADENA,CAL.

5028 D. I. CRANICHER
STEARN-ROGER CORP.
DENVER,COLO.

3220 WILLIAM J. MCGRAW
HIBBING AREA TECH.
HIBBING,MINN.

3238 WALTER G. ELWELL
NEB. WESLEYAN UNIV.
LINCOLN,NEB.

3242 GEORGE A. SPRADLING
UNIV. OF OKLAHOMA
NORMAN,OKLA.

3260 JOHN C. MARVEY
IBM
ROCHESTER,MINN.

3277 JAMES L. GRISELL PHD
LAFAYETTE CLINIC
DETROIT,MICH.

3283 JOE D. PEGRAM
MEMPHIS STATE UNIV.
MEMPHIS,TENN.

3302 ARNE GARNESS
CONCORDIA COLLEGE
MOORHEAD,MINN.

3319 RAYMOND T. MCNAMARA
DEFENSE SUB SUPPLY
CHICAGO,ILL.

3337 G. ENYEDY
DIAMOND ALKALI CO.
PAINESVILLE,OHIO

5014 DAVE NIELSON
GD/ASTRONAUTICS
SAN DIEGO,CAL.

5020 MARILYN DOIG
COLORADO STATE UNIV.
FORT COLLINS,COLO.

5020 LARRY JACKSON
COLORADO STATE UNIV.
FORT COLLINS,COLO.

5026 MARVIN J. CARR
DOUGLAS AIRCRAFT CO.
SANTA MONICA,CAL.

5028 B. P. DUNCAN
STEARN-ROGER CORP.
DENVER,COLO.

5028 ED JURACEK
STEARN-ROGER CORP.
DENVER,COLO.

5032 BOB N. MANNING
GOODYEAR AEROSPACE
LITCHFIELD PARK,ARIZ

5032 D. H. O'HERREN
GOODYEAR AEROSPACE
LITCHFIELD PARK,ARIZ

5032 MRS. N. A. KUFFEL
GOODYEAR AEROSPACE
LITCHFIELD PARK,ARIZ

5033 JOHN A. FERLING
CLAREMONT MEN'S COLL
CLAREMONT,CAL.

5041 RONALD E. WILDER
MOTOROLA SEMICONDUCT
PHOENIX,ARIZ.

5043 GLENN R. INGRAM
MONTANA STATE COLL.
BOZEMAN,MONT.

5058 L. L. KOPPIN
SUNDSTRAND AVIATION
DENVER,COLO.

5058 TED J. MCKENNA
SUNDSTRAND AVIATION
DENVER,COLO.

5060 CHARLES A. BETTINGER
UNIV. OF TEXAS
AUSTIN,TEX.

5073 PAUL KLEEWEIN
CAMPUS HIGH SCHOOL
WICHITA,KAN.

5075 WENDELL T. BEYER
UNIV. OF OREGON
EUGENE,ORE.

5076 JAMES N. BOLES
U.C. BERKELEY
BERKELEY,CAL.

5077 PAUL BROWNE
UNION OIL CO.
BREA,CAL.

5086 HARRY CASTLE JR.
PIONEER NATURAL GAS
AMARILLO,TEX.

5086 B. G. GRANT
PIONEER NATURAL GAS
AMARILLO,TEX.

5089 DAVID BAER
U.S.P.H.S.
LAS VEGAS,NEV.

5095 LOWELL A. RASMUSSEN
WEYERHAUSER CO.
TACOMA,WASH.

5096 BOYD C. NORRIS
BUREAU OF REC.
SACRAMENTO,CAL.

5098 BURTON L. WILLIAMS
WHITE SANDS M.R.
WHITE SANDS,N.M.

5104 DR. J.R. GUINN
TEXAS COLL. OF A+I
KINGSVILLE,TEX.

5108 ELTON W. CHASE JR.
CLARK COLLEGE
VANCOUVER,WASH.

5117 ASTRIK DEIRMENDJIAN
NAT CTR ATMOS RES
BOULDER,COLO.

5126 WENDELL L. POPE
UTAH STATE UNIV.
LOGAN,UTAH

5130 A.A.J. HOFFMAN
TEXAS CHRISTIAN UNIV
FORT WORTH,TEX.

5131 ROGER HOFFMAN
MARTIN CO.
DENVER,COLO.

5131 E.E. EVANS
MARTIN CO.
DENVER,COLO.

5131 A.G. BISENIWS
MARTIN CO.
DENVER,COLO.

5133 RALPH D. PERRINE
MASON + HANGER
AMARILLO,TEX.

5133 LEWIS R. WOMBLE
MASON + HANGER
AMARILLO,TEX.

5143 GERALD W. LOCKE
TEXAS TECH. COLLEGE
LUBBOCK,TEX.

5144 JOSEPH A. STRAHL
U.S. WEATHER BUREAU
SACRAMENTO,CAL.

5145 ROBERT C. STEINBACH
GROSSMONT COLLEGE
SPRING VALLEY,CAL.

5147 RICHARD ROSANOFF
NORTH AMERICAN AVIA.
DOWNEY,CAL.

5150 JOHN M. GOODE
HALLIBURTON CO.
DUNCAN,OKLA.

5150 EUGENE BAKER
HALLIBURTON CO.
DUNCAN,OKLA.

5150 GEORGE V. COPLAND
HALLIBURTON CO.
DUNCAN,OKLA.

5150 GEORGE A. LARCADE
HALLIBURTON CO.
DUNCAN,OKLA.

5152 CARL J. REICH
MONTEREY PEN. COLL.
MONTEREY,CAL.

5162 JAMES W. BRUCE
SAN DIEGO CO RD DEPT
SAN DIEGO,CAL.

5164 KENNETH W. JONES
COLO DEPT HIGHWAYS
DENVER,COLO.

5166 JAMES M. ANDERSON
LOCKHEED PROPULSION
REDLANDS,CAL.

5168 WARD CROWLEY
UNIV. OF IDAHO
MOSCOW,IDAHO

5169 JOHN LAFLER
SURVEY RES CTR U.C.
BERKELEY,CAL.

5171 OLIVER BENSON
UNIV. OF OKLAHOMA
NORMAN,OKLA.

5171 CHARLES MAUDLIN JR.
UNIV. OF OKLAHOMA
NORMAN,OKLA.

5171 DAVE ASHBAUCHER JR.
UNIV. OF OKLAHOMA
NORMAN,OKLA.

5172 BERNARD BURGER
HYDRA ELECTRIC CO.
BURBANK,CAL.

5173 RICHARD G. SCHERER
CENTRALIA COLLEGE
CENTRALIA,WASH.

5174 L. PIERCE LOBERG
COMPUTERMAT INC.
LOS ANGELES,CAL.

5177 R.G. WAEDEMON
TEXACO INC.
PORT ARTHUR,TEX.

5181 ROBERT R. WHITE
LA DEPT WATER POWER
LOS ANGELES,CAL.

5189 L.E. HARVEY
FOOTHILL COLLEGE
LOS ALTOS HILLS,CAL.

5192 NOEL T. SMITH
INDIANA STATE COLL.
TERRE HAUTE,IND.

5200 AGNES W. PRUSZKA
USAA DEFENSE BD
FT. BLISS,TEX.

5200 LT. JACK KEEN
USAA DEFENSE BD
FT. BLISS,TEX.

5201 JOHN F. PEARSON
SPRINGFIELD SCH DIST
SPRINGFIELD,MO.

5202 ALFRED L. TAYLOR
N.E. OKLA. A+M COLL.
MIAMI,OKLA.

5202 WALTER E. MOORE
N.E. OKLA. A+M COLL.
MIAMI,OKLA.

5205 M.E. HETHERINGTON
WESTERN STATE COLL.
GUNNISON,COLO.

5214 WILLIAM C. PIQUETTE
OTERO JUNIOR COLLEGE
LA JUNTA,COLO.

5215 ROSEMARY PETERSEN
WDPC-UCLA
LOS ANGELES.CAL.

5217 MAURICE H. WITTEN
FT HAYS KANS ST COLL
HAYS.KAN.

5219 GEORGE G. TOWN
SEATTLE UNIVERSITY
SEATTLE.WASH.

5223 STANLEY WISNIEWSKI
NORTHROP CORP.
BEVERLY HILLS.CAL.

5226 B.S. SHANNON JR
WADLEY RESEARCH INST
DALLAS.TEX.

7021 P.J. REDBERGER
RES COUN OF ALBERTA
EDMONTON.ALBERTA

PAUL S. CHAN
RIVERSIDE CITY COLL.
RIVERSIDE.CAL.

A.H. BEST
GLIDDEN CO.
JACKSONVILLE.FLA.

RUTH C. MOSSMAN
IBM
DENVER.COLO.

DAVID R. DYE
IBM
WHITE PLAINS.N.Y.

RONALD R. LAKE
IBM
GLENDALE.CAL.

W.B. SCOTT
IBM
SAN JOSE.CAL.

DICK WILLIAMS
IBM
LOS ANGELES.CAL.

AUBREY D. WOOD
IBM
OKLAHOMA CITY.OKLA.

5215 JOHN W. RETTENMAYER
WDPC-UCLA
LOS ANGELES.CAL.

5217 JOHN B. O.LOUGHLIN
FT HAYS KANS ST COLL
HAYS.KAN.

5222 ROBIN YOUNG
UNIV. OF NEW MEXICO
ALBUQUERQUE.N.M.

5225 VICTOR W. HOFFMAN
US WEATHER BUREAU
FORT WORTH.TEX.

5226 J.M. HILL M.D.
WADLEY RESEARCH INST
DALLAS.TEX.

PAUL L. TUAN
UNIVERSITY OF UTAH
SALT LAKE CITY.UTAH

REX TURCO
US BUREAU OF RECL.
BILLINGS.MONT.

GERALD W. CALL
IBM
DENVER.COLO.

J. MICHAEL HUHTA JR.
IBM
DENVER.COLO.

STEVE LOPEZ
IBM
WHITE PLAINS.N.Y.

GORDON W. GOESCH
IBM
SAN JOSE.CAL.

DOROTHY MCGOWAN
IBM
LOS ANGELES.CAL.

CHARLES E. BERRY
IBM
LOS ANGELES.CAL.

MINUTES OF THE MEETING - JUNE 17-19, 1964

The general meeting was opened on Wednesday, June 17, 1964, at the Brown Palace Hotel in Denver, Colorado, by the Western Region President, Paul Bickford. After opening remarks by Paul and the IBM representatives, the meeting was turned over to presentation of papers and workshops in parallel technical sessions. New users were welcomed to the group in a session held Wednesday evening. This was followed by the regular sound-off session which is reported on elsewhere in the minutes.

The technical sessions were held as scheduled in the agenda with the exception of changes in rooms. Copies of those papers available are included with these minutes.

The keynote speaker at the luncheon on Thursday, June 18, was Dr. E. N. Brandt, Jr., M.D., Ph.D. The subject of the very well-received talk was, "The Future of Computers in Medicine."

At the last general session on June 19, Dick Williams, Dave Dye, and Chuck Berry of IBM responded to the sound-off session. Paul Bickford announced that the next meeting will be a joint meeting of the Western and Mid-western regions and will be held on the campus of the University of Oklahoma in Norman, Oklahoma, on November 9, 10, and 11, 1964. The meeting was then adjourned.

ROBERT R. WHITE
Western Region Secretary

SOUND-OFF SESSION

The sound-off session was held Wednesday, June 17, 1964, with Chuck Maudlin as Chairman. The following is a report on most of the points made and the response from IBM representatives on Friday, June 19.

The question was raised about the speed of the IBM supplied FORTRAN compilers. It was stated that the old Bell Interpretive System for the 650 was faster than current compiler systems for the 1620. IBM's reply was that the compilers are relatively slow because they incorporate extensive source language error checking at compilation. The FORTRAN/FORMAT compiler is the fastest of the current IBM Fortrans for the 1620 because most of the error checking is done by the pre-compiler. Changes have been made to some of the FORTRAN compilers, notably II-D, Version 2, to provide for in-line compilation and allow error checking of mathematical operations at object time.

A request was made for an advanced monitor workshop which would supply enough information so that users can modify the Monitor program. Dick Williams stated that it might be possible to hold such a class for advanced systems programmers on a limited enrollment basis. Coincident with this request was a strong request for much more complete documentation on the software systems. Dick Williams stated that an operator's guide for changing Monitor will probably be available from his office and suggested that the Users Group bring more pressure to bear on IBM, through the Users Group Executive Council, to supply better documentation.

The complaint was made that some Systems Engineers are not filling out APAR forms when requested. It was stated that this is part of their job and if it is not done, the user should contact his branch manager.

Inquiries were made regarding possible future software systems and the answers were as follows:

FORTRAN IV - not being planned.
Report Program Generator - not being planned.
A non-disk loader for Monitor programs - not likely
  because of complexity.
A disk-independent SPS with all disk and printer
  op codes - not being planned.
Network Analysis - near future release.
Linear Programming Package - release in July.
Non-variable subroutines for FORTRAN II - now available
  on FTN II-D, Version II.

Requests for hardware additions were made and the answers to these were as follows:

Reset on Typewriter and 1622 - RPQ
Punch and Read start on console - RPQ
Programmable time clock - RPQ
Tele-processing - RPQ 834308; approximate price - $600/mo. including a 1026 adapter which can handle up to 26 - 1050 consoles on line.

There was some comment about the 1620 failing to operate properly outside of a temperature range of 70-80 degrees F. IBM representatives requested that users notify their C.E. so that corrective action can be taken.

In answer to a request for a flexible rental schedule for the first year of use, the reply from IBM was that no changes from present pricing policies are planned.

It was explained that the machine characteristic of not destroying the contents of the read buffer on a reset operation was designed on purpose. It was felt that the best method was to require positive action to destroy this information.

Inquiries about excessive down-time on the 1443 or the punch-read feed feature were answered by other users. Apparently this is not a general problem.

IBM announced the availability of COGO-ID. It is described in Bulletin H 20-4219-0. The program number is 1620-UG-05X and it requires a 20K 1620 with one 1311 disk pack.

This concluded the sound-off session.

WEDNESDAY - JUNE 17, 1964

8:00 Registration Outside Central City Room

8:30 Welcome and Opening Remarks. . . Paul Bickford, Regional President
IBM Announcements. . . . . Dick Williams, Chuck Berry, Angelo Arena
R. P. Paterniti
Index Registers. . . . . Miss Dorothy McGowan, IBM

10:00 Coffee

	Session A Central City Room	Session B Leadville & Silver Plume Rooms
10:30	A Least Squares Solution for A Range Measuring Instrumentation System -- Oliver Lee Kingsley and Burton L. Williams (5098)	IBM Monitor I Workshop
11:00	Boundary Value Problems in Ordinary Differential Equations with Constant Coefficients -- Richard Rosanoff and Gordon Mah (5157)	IBM Monitor I Workshop (Continued)
11:30	(1) A program to check elementary machine language laboratory exercises (2) A 519 Simulator -- R. C. Steinbach (5145)	IBM Monitor I Workshop (Continued)
12:00	Lunch ---On your own	
1:15	Introduction to Matrices -- C. E. Maudlin, Jr. (5171)	IBM Monitor I Workshop
3:00	Coffee	
3:30	Simultaneous Linear Equations with Complex Coefficients -- Nancy C. Kuffel (5032)	IBM Monitor I Workshop (Continued)
4:00	Applications of Numerical Filters in the Power Spectral Analysis of Stationary Time Series -- A. J. Hoffman (5130)	IBM Monitor I Workshop (Continued)
4:30	The Acquisition and Utilization of an IBM 1620 Computer in the U. S. Public Health Service -- James A. Daly and Nancy A. Paquin (1118)	IBM Monitor I Workshop (Continued)

5:00 Adjournment of Day's Sessions

7:30 New Users Session. . . . . .Paul Bickford. . . . . Stratton Room &)
8:15 Sound-off Session. . . . . .C. E. Maudlin. . . . . Tabor Room)

THURSDAY - JUNE 18, 1964

	Session A Ballroom B	Session B Tabor & Stratton Rooms
8:00	Late Registration	Promenade Area
8:30	How the IBM 1620 Assists Student Counselors at Junior College -- Paul S. Chan, IBM	Generalized Filter Network A/C Steady Analysis Program -- Davis H. O'Herren (5032)
9:00	(Continued)	FORTRAN II - Debugging Techniques and Aids -- Leon P. Goldberg (1177)
9:15	AD-APT -- Bill Rogers, IBM	FORTRAN II (Continued)
10:00	Coffee	
10:30	1620 Computer Utilization in a Wind Tunnel Data Acquisition System -- Stanley Wisniewski (5223)	FORTRAN II (Continued)
10:50	Linear Programming -- Dr. S. T. Parker (3107)	FORTRAN II (Continued)
12:00	Luncheon . . . . .Ballroom	
	Keynote Address: Future of Computers in Medicine. . .Dr. E. N. Brandt, Jr., M. D., Ph. D.	
1:45	IPL V -- Wendell T. Beyer (5075)	IBM Disk File Applications
3:00	Coffee	
3:30	Petroleum Exploration and Production Applications for the IBM 1620 and Plotter -- Jack L. Morrison (5171)	FORTRAN II and The 1443 -- L. Hoffman (1177)
4:00	A Control Systems Approach to Automatic Jet Engine Testing -- Aubrey D. Wood, IBM	FORTRAN II and The 1443 (Continued)
5:00	Adjournment of Day's Sessions	

FRIDAY - JUNE 19, 1964

8:00 Registration

	Session A Ballroom B	Session B Onyx Room
8:30	SPS Tutorial Workshop -- Clarence B. Germain	Panel on Education: Chairman: Richard V. Andree, University of Oklahoma Members: Charles A. Bettinger, University of Texas Noel T. Smith, Indiana State College Donald L. Ferguson, Campus High School
10:00	Coffee	
10:30	SPS Tutorial (Continued)	Automatic Processing of AUTOSPOT and AUTOMAP Programs with the 1620-131 Disc System -- Jack T. Dunn (3326)
12:00	Lunch. . .On your own	
1:00	FORTRAN Teaching Techniques -- Wendell L. Pope	Workshop on 10:30 Paper
1:20	A Load and Go SPS with Monitor Control -- Kenneth M. Lochner and Glenn R. Ingram (5043)	Workshop on 10:30 Paper
1:55	How The 1620 is Used at Colorado State University -- Franklin A. Graybill	Workshop on 10:30 Paper

2:30 IBM Reply From Sound-Off Session. . . . . .Ballroom B

3:15 Meeting Adjournment

"A Least Squares Solution for a Range Measuring Instrumentation System"

by

Oliver Lee Kingsley
and
Burton L. Williams

Range Instrumentation Systems Office
White Sands Missile Range
New Mexico

June 17, 1964

A Least Squares Solution
for a
Range Measuring Instrumentation System

ABSTRACT

A brief development of the least squares equations for an instrumentation system that measures a set of ranges to obtain an estimate of the space position of a space vehicle. In addition, a method for obtaining the precision of that space position is included.

I INTRODUCTION

At White Sands Missile Range, the instrument that measures the radial range to an object in space is called a Distance Measuring Equipment or simply DME. An instrumentation system capable of giving Euclidean three space position estimates is termed as a DME/DME system. A typical DME/DME system usually consists of three non-colinear instrument or equipment sites used to measure range.

The typical solution equations are the classical deterministic set that rejects the minor image solution. A four or more DME system presents a problem because a slight error in any range measurement will not produce a set of homogeneous space position estimates by the classical approach. There is a need for a good method to combine the set of overdetermined measurements into a single space point estimate. The least squares method will provide the required space point estimate if the set of measurements are unbiased.

II LEAST SQUARES DATA REDUCTION EQUATIONS

The least squares equations developed minimize the sums of squares of the error set of range measurements.

The observational equation from which the error equation is derived is written:

(1) $R_{mi} = R_i + E_{mi}$

where R_{mi} denotes the measured range from the i-th DME to the tracked vehicle.

R_i denotes the true range from the i-th DME to the tracked vehicle.

E_{mi} denotes the measurement error associated with the observation taken from the i-th DME.

The error equation is easily obtained from the observational equation, thus:

(2) $E_{mi} = R_{mi} - R_i$

The true value needs to be replaced by some suitable approximation. Later, the true value will be estimated by the final solution or reduction equations. The assumption is made that the true value can be represented by the linear terms of a Taylor series about some nearby point R_o, where:

$$(3) \quad R_o = R_o\,(X_o, Y_o, Z_o)$$

From the i-th DME, the range to the point (X_o, Y_o, Z_o) is written:

$$(4a) \quad R_{oi} = \sqrt{(X_o - X_i)^2 + (Y_o - Y_i)^2 + (Z_o - Z_i)^2}$$

For any space position (X, Y, Z), the equation becomes:

$$(4b) \quad R_i = \sqrt{(X - X_i)^2 + (Y - Y_i)^2 + (Z - Z_i)^2}$$

The linear Taylor series representation is written:

$$(5) \quad R_i = R_{oi} + \left.\frac{\partial Ri}{\partial x}\right|_o (X - X_o) + \left.\frac{\partial Ri}{\partial y}\right|_o (Y - Y_o) + \left.\frac{\partial Ri}{\partial z}\right|_o (Z - Z_o)$$

where

$$\left.\frac{\partial Ri}{\partial x}\right|_o = (X_o - X_i)/R_{oi}$$

$$\left.\frac{\partial Ri}{\partial y}\right|_o = (Y_o - Y_i)/R_{oi}$$

$$\left.\frac{\partial Ri}{\partial z}\right|_o = (Z_o - Z_o)/R_{oi}$$

The weighted error sums of squares for k measurements from k DME sites is written:

$$(6a) \quad \sum_{i-1}^{k} E_{mi}^2 = \sum_{i-1}^{k} Wi \left[R_{mi} - R_i\right]^2$$

where Wi is the weight given to each measurement. The errors sums of squares for equally weighted measurements is written:

(6b) $$\sum_{i=1}^{k} E_{mi}^{2} = \sum_{i=1}^{k} \left[R_{mi} - R_{i}\right]^{2}$$

Generally, the components of the instrumentation system are near enough alike in performance and behavior that equation (6b) is applicable. For a system that consists of heterogeneous distance measuring equipment (system components), each weight that would be inversely proportional to the equipments range variance would be appropriate.

The error sums of squares are minimized with respect to the range measurement parameter which is a function of the three orthogonal components X, Y and Z. The three resulting equations thus formed are called the normal equations from which the estimates $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ are obtained:

(7a) $$\frac{\partial}{\partial X}\left[\sum_{i=1}^{k} (R_{mi} - R_{i})^{2}\right] =0$$

(7b) $$\frac{\partial}{\partial Y}\left[\sum_{i=1}^{k} (R_{mi} - R_{i})^{2}\right] = 0$$

(7c) $$\frac{\partial}{\partial Z}\left[\sum_{i=1}^{k} (R_{mi} - R_{i})^{2}\right] = 0$$

The constant terms not involving $(X-X_o)$, $(Y-Y_o)$ and $(Z-Z_o)$ are placed on the right hand side of the equation:

$$(8a)\quad \sum \left\{ \frac{1}{R_{oi}^{2}} \left[(X_o-X_i)(X_o-X_i)\Delta X + (Y_o-Y_i)(X_o-X_i)\,\Delta Y + (Z_o-Z_i)(X_o-X_i)\Delta Z \right] \right\} = C_1$$

$$(8b)\quad \sum \left\{ \frac{1}{R_{oi}^{2}} \left[(X_o-X_i)(Y_o-Y_i)\Delta X + (Y_o-Y_i)(Y_o-Y_i)\,\Delta Y + (Z_o-Z_i)(Y_o-Y_i)\Delta Z \right] \right\} = C_2$$

$$(8c)\quad \sum \left\{ \frac{1}{R_{oi}^{2}} \left[(X_o-X_i)(Z_o-Z_i)\Delta X + (Y_o-Y_i)(Z_o-Z_i)\,\Delta Y + (Z_o-Z_i)(Z_o-Z_i)\Delta Z \right] \right\} = C_3$$

where

$$C_1 = \sum_{i=1}^{k} \left[(R_{mi}-R_{oi})(X_o-X_i)/R_{oi} \right]$$

$$C_2 = \sum_{i=1}^{k} \left[(R_{mi}-R_{oi})(Y_o-Y_i)/R_{oi} \right]$$

$$C_3 = \sum_{i=1}^{k} \left[(R_{mi}-R_{oi})(Z_o-Z_i)/R_{oi} \right]$$

$$\Delta X = \hat{X} - X_o$$

$$\Delta Y = \hat{Y} - Y_o$$

$$\Delta Z = \hat{Z} - Z_o$$

The equations can be written in compact form by matrix notation:

$$(9a)\qquad A\begin{pmatrix}\Delta X\\ \Delta Y\\ \Delta Z\end{pmatrix} = \begin{pmatrix}C_1\\ C_2\\ C_3\end{pmatrix}$$

or

$$(9b)\qquad A \quad \Delta = C$$

Solving for from equation (9b):

$$(10)\qquad \Delta = A^{-1} C$$

The final solution for X, Y, and Z can be written:

$$(11)\qquad \begin{pmatrix}X\\ Y\\ Z\end{pmatrix} = \begin{pmatrix}X_o\\ Y_o\\ Z_o\end{pmatrix} \quad A^{-1} C$$

The necessary start point $(X_o, Y_o, Z_o,)$ can be obtained from a deterministic solution for three range measurements*. The region for convergent solutions has not been fully explored at this time.

---

* Armijo, Larry , "Determination of Trajectories Using Range Data from Three Non-Colinear Radar Stations", Technical Memorandum 766, USASMSA, Sept. 1960, WSMR,N.M.

III A METHOD FOR ESTIMATING INSTRUMENTATION SYSTEM PRECISION

The term precision estimate refers to the standard deviation estimates for the coordinate data X, Y and Z from the instrumentation system. If there exists a common range measurement variance (σ_m^2), then by use of the relative variance-covariance matrix, A^{-1}, from equation (10) estimates of the component variances can be obtained. The diagonal elements from the inverse matrix A^{-1} are used to estimate the component variance:

$$(12) \quad \begin{pmatrix} \hat{\sigma}_x^2 \\ \hat{\sigma}_y^2 \\ \hat{\sigma}_z^2 \end{pmatrix} = \begin{pmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{pmatrix} \hat{\sigma}_m^2$$

where A_{11}, A_{22} and A_{33} are from the matrix A^{-1}.

The total variance estimate is defined by the equation:

$$(13) \quad \hat{\sigma}_T^2 = \hat{\sigma}_X^2 + \hat{\sigma}_Y^2 + \hat{\sigma}_Z^2$$

In terms of the matrix A, the total variance estimate is written:

$$(14) \quad \hat{\sigma}_T^2 = \hat{\sigma}_m^2 \quad \mathrm{Tr} \quad A^{-1}$$

where Tr denotes the trace of the inverse of the matrix A.

THE BOUNDARY VALUE PROBLEM IN ORDINARY DIFFERENTIAL EQUATIONS WITH CONSTANT COEFFICIENTS

By Richard Rosanoff and Gordon Mah*

INTRODUCTION

In the process of solving seventh and eleventh order boundary value problems in closed form, the authors experienced an improvement in their understanding of the requirements of good formulation for a digital computer. Better control of numerical error and easier programming and debugging were the products of improved formulation.

The specific problems programmed were boundary value problems in ordinary differential equations with constant coefficients. The general solution of the ordinary differential equations with constant coefficients was known in 1739. It is mathematically simple. Its nature has been thoroughly explored. Perhaps just because of this, it provides an excellent vehicle for the study of the consequences of restrictions imposed by automatic digital computation.

To put together a good computer program, one must consider several ideas and points of view. Thus, although we shall be concerned with simple material, the structure of our paper is mildly complex. We include, therefore, the following outline of the material to be covered:

1. Mathematical preliminaries
    a. The finite nature of the computer number system
    b. The implications of the finite number system
    c. A definition of linear dependence
    d. Linear dependence in a finite number system
2. The computationally significant features of the physical problems used as illustrations. The equations and boundary conditions to be solved are also presented.

---

*Senior Engineers, Apollo Structures Analysis, Space and Information Systems Division, North American Aviation, Inc., Downey, California.

3. Characteristic polynomials and their roots

4. Basis functions and boundary conditions

    a. Definition of a basis and the mathematical requirements for their selection

    b. The Wronskian

    c. Boundary conditions in terms of the Wronskian

5. Selecting a basis

    a. A group of bases

    b. Physical criteria

    c. Linear independence

        (1) Wronskian

        (2) Gramian

    d. The effect on conditioning number of a dual coordinate system

6. Remarks and conclusions

I. MATHEMATICAL PRELIMINARIES

The set of numbers which can be represented in a digital computer is not the set of real numbers. It is not even the set of rational numbers. In fact, it is a finite set. It is bounded from above and below, and it is nowhere dense on the real number system. It is not a field. For brevity, let us speak of the computer number set to mean the set of numbers available on a particular computer. For the real number system, it is true that if:

$$A \times B = C$$

and if A and B belong to the real number system, so does C. This is the property of closure. Closure does not hold for the computer number set.

$$10^{51} \times 10^{51} = 10^{102}$$

Both factors on the left are members of the set of numbers with which the IBM 1620 Computer operates. Their product is not a member of the set.

The equation:

$$A + 0 = A$$

Defines a unique zero in the real number system. But notice with eight significant figure arithmetic defining the computer number set:

$$e^{9.3} + e^{-9.3} = 10938.019 + .000091424231 = 10938.019 = e^{9.3}$$

yet $e^{-9.3}$ is not zero because

$$e^{9.3} \times e^{-9.3} = .99999997$$

It will be seen that this difference in the number systems lies at the root of many numerical difficulties. The properties of functions must be reviewed in light of the number system available for the calculation.

One of the properties of sets of functions which is important to our discussion is the property of linear independence. By way of review we state that N functions ϕ_i i = 1, n are said to be linearly dependent if there exists a set of N constants C_i i = 1, n not all zero such that:

$$\sum_{i=1}^{n} C_i \phi_i = 0$$

If no such set of C_i exist the functions ϕ_i are said to be linearly independent.

As an example, we might consider three vectors in a physical space of three dimensions. If the vectors are not coplanar, then they are linearly independent. There are, at most, three linearly independent vectors in the space. Therefore, any fourth vector in the space can be completely described in terms of three linearly independent bases or coordinate vectors in the space. If, by some accident, a coplanar set of basis vectors were taken, we should be unable to completely represent a general vector in the space. We have a deficiency perpendicular to the plane of our base vectors.

The geometrical ideas are carried beyond a physical three space to N dimensional spaces in a very useful manner (Ref's. 5, 17). Further, a function itself, defined in some region, may be considered as a vector in a space of infinite dimensions, i.e., a function space. The directions in the space are associated with the real numbers in the region (Ref's. 18, 19).

Linear independence is essentially the same idea in these extensions to higher spaces. The complete description of some desired function as a linear combination of the members of a set of functions requires that these functions "span" the space. If any one of them can be represented by a finite linear combination of the others, it adds nothing to the description of the space.

On the other hand, linear independence over the real number system <u>does not</u> insure linear independence in a finite number system. For example:

$$C_1 \sinh(X) + C_2 \cosh(X) = \delta$$

δ may not be zero in the real number system, but $\delta = 0$ on a computer, if X is large enough, and

$$C_1 = - C_2 \neq 0$$

II. THE PHYSICAL PROBLEM

We have seen that the number system available to us for computation is limited and lacks some of the characteristics of the real number system in which we think. Later, we shall be faced with a choice of basis functions between bases which are completely equivalent in the mathematics of the set of real and complex numbers. Although we shall find adequate numerical arguments to justify a choice, there is also a physical argument. It is our point of view that the formulation which represents the physical problem most directly will probably be the best formulation from every point of view.

With this objective in mind, we need not spend the time required for the derivation of equations. We have a seventh order problem and an eleventh order problem which have been solved on our 1620. The seventh order problem arose in the calculation of stresses developed in the glue line of a lap joint such as is shown in Figure 1.

Two metal plates, A and B, (see Figure 1) of unequal thickness are joined by gluing in a lap joint of Length L. If the plates are pulled by forces at some distance from the joint the load must be transmitted from one plate to the other by shear stresses in the glue. Further, the eccentricity of the joint will introduce some bending, and the resulting rotation of the joint will result in a transverse component of the force at the edge of the joint. If the plates were completely rigid, one would expect the loads to be uniformly distributed along the glue line. In fact, however, the plates are not rigid but will elongate and bend in response to the loads they are carrying. Here lies the crux of the problem. At Point 1, Plate A is elongated. Plate B is not. We have almost a discontinuity in deformation. This requires a relative motion which is resisted by the glue. The resistance in the glue results in the transfer of load from Plate A to Plate B. The transfer of load eventually brings the plates to the same extension at some interior point. The situation at Point 2 is identical to that at 1 except that the roles of the plates are interchanged.

The important thing to remember, as we construct the solution, is that stresses arise at the boundaries due to the elastic deformation of the plates. These stresses are damped out as one moves to the interior. Once the two plates have developed equal extensions, the solution should be very similar to the solution for infinitely rigid plates.

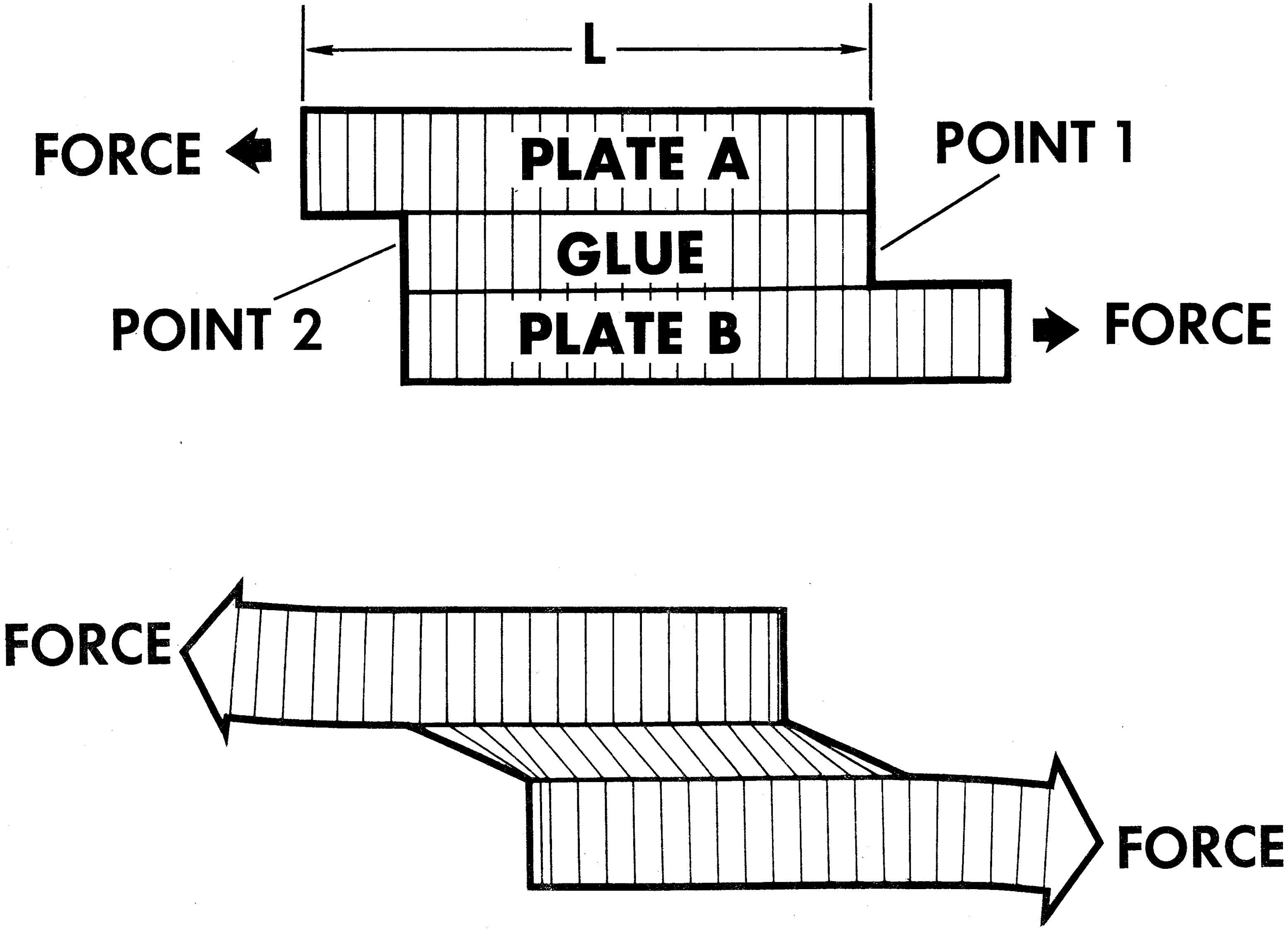


Figure 1. Physical Problem Represented by Seventh Order Equation

The shear stress in the glue is labeled τ. The extension stress, transverse to the joint is labeled σ.

The analysis is an extension of a classical paper by Goland and Reissner (Ref. 2). The equations of equilibrium (under some simplifying assumptions) are:

$$\frac{d^3\tau}{dx^3} - \xi_1 \frac{d\tau}{dx} - \xi_2 \sigma = 0$$

$$\frac{d^4\sigma}{dx^4} + \xi_3 \sigma + \xi_4 \frac{d\tau}{dx} = 0 \qquad (2.1)$$

which together give the homogenous differential equation:

$$\frac{d^7\tau}{dx^7} - \xi_1 \frac{d^5\tau}{dx^5} + \xi_3 \frac{d^3\tau}{dx^3} - (\xi_1 \xi_3 - \xi_2 \xi_4) \frac{d\tau}{dx} = 0 \qquad (2.2)$$

The parameters ξ_i are related to the elastic properties of the plates and glue line. We do not wish to burden the discussion with definitions of the notation. Instead, permit us to state the fact about the ξ_i which is significant to the subject at hand. All of the ξ_i represent purely passive elements. They are geometric and elastic quantities.

The boundary conditions are given in the form of six inhomogenous differential equations of equilibrium which must be met, three at each boundary and one integral condition of equilibrium to be met over the domain. These may be written as:

$$\sum b_{i,j} \frac{d^j\tau(x)}{dx^j} = F_i$$

and

$$\int_0^L \tau(x)\, dx = \text{EXTERNAL SHEAR} \qquad (2.3)$$

We shall summarize the 11th order problem more rapidly. The physical problem is once again the determination of stresses in a glued joint. For this problem however, we consider a section of honeycomb sandwich material. It consists of a core with face plates A, shown in Figure 2. The section is assumed spliced to a similar honeycomb section through splice plates equal in thickness to the face plate.

The three equations of equilibrium are:

$$\sigma_c = \frac{2h}{P_1} \frac{d^3\tau}{dx^3} - \frac{4h}{3} \frac{d\tau}{dx}$$

$$\sigma_b = \frac{1}{P_3} \frac{d^4\sigma_c}{dx^4} + \sigma_c + \frac{h}{2} \frac{d\tau}{dx}$$

$$\frac{d^4\sigma_b}{dx^4} + 2P_2\sigma_b - P_2\sigma_c = 0 \qquad (2.4)$$

A consequence of these equations is:

$$\frac{d^{11}\tau}{dx^{11}} - \frac{2P_1}{3} \frac{d^9\tau}{dx^9} + (2P_2 + P_3) \frac{d^7\tau}{dx^7} - \frac{P_1}{3} \left(4P_2 + \frac{5P_3}{4}\right) \frac{d^5\tau}{dx^5}$$

$$+ P_2P_3 \frac{d^3\tau}{dx^3} - \frac{P_1P_2P_3}{6} \frac{d\tau}{dx} = 0 \qquad (2.5)$$

The boundary conditions are similar in form to those of the 7th order problem.

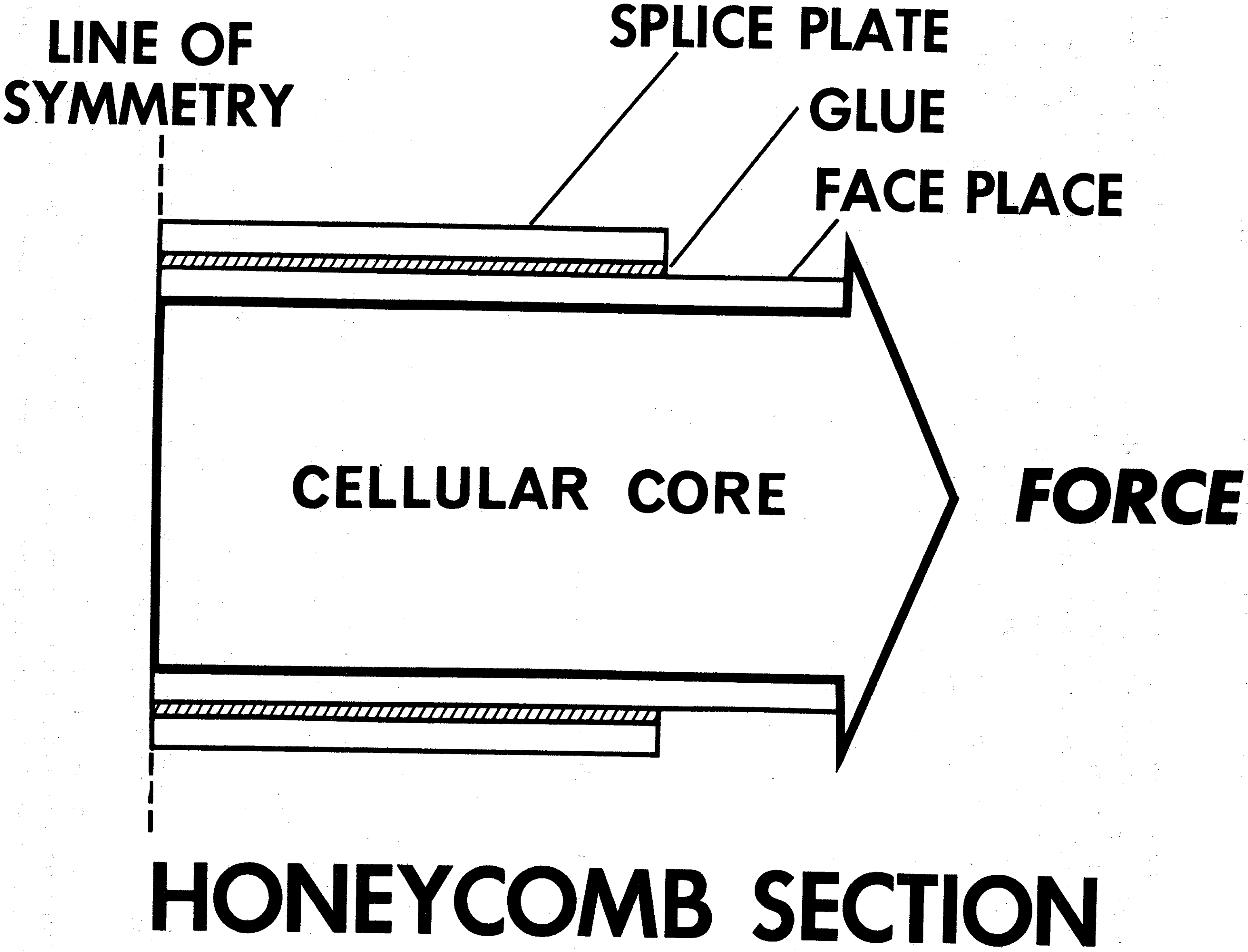


Figure 2. Physical Problem Represented by Eleventh Order Equation

III. THE CHARACTERISTIC POLYNOMIAL

From the differential equations it is possible to write characteristic polynomials of seventh and eleventh order, respectively. As will be seen later, the roots of the characteristic polynomial provide a wealth of information concerning the nature of the solution, the relative importance of the various solution components in the derivative, hence, the boundary conditions. Even measures of the maximum stress and physical realism of the solution are provided in a qualitative way. They provide the best link between the physical parameters which occur in the coefficients of the differential equation and the solution.

Unfortunately, (Ref's. 3, 6 and 11) polynomials are subject to ill-conditioning and their roots may be poorly determined. In Reference 3 it is shown that the relative error in a root α of a polynomial $P(x) = \Sigma a_i X^i$ due to an error δa_s in the coefficient a_s is given by

$$\frac{\delta \alpha}{\alpha} = \frac{\alpha^{s-1} a_s}{P'(\alpha)} \frac{\delta a_s}{a_s}$$

Thus, it is only necessary for

$$\frac{\alpha^{s-1} a_s}{P'(\alpha)}$$

to be large to have a great magnification of relative error in the root as compared to the coefficient. There is a great proliferation of methods for obtaining polynomial roots (Ref's. 3, 4, 6, 7, 11, 12, and 13). The reason for this great number of methods is that none of them are generally satisfactory.

For the two problems solved, use was made of the easily variable precision of the 1620 version of Fortran. In the first place, the polynomials were studied throughout the data range of the problem, and solutions at different levels of machine precision were compared.

The zero root of the two polynomials is, of course, precisely known and removed without any contribution to error. The reduced polynomials both consist of even ordered terms only so that we may solve polynomials of fifth and third rather than tenth and sixth order. The constant coefficient is negative so that the fifth and third order polynomial have at least one positive real root each. Thus, our original equations are known to have a pair of real roots which are equal in magnitude and opposite in sign. Following Lanczos (Ref. 5), we obtain this root, either directly or as a reciprocal, between zero and one. The method used was to repeatedly evaluate the polynomial by synthetic division and test to isolate the root in one of the half intervals.

To minimize contributions to error in subsequent roots due to the removal of the real roots, we requested a special synthetic division subroutine from our systems programmers. We wish to acknowledge the excellent job done by John Sherman of our Division in this respect. Mr. Sherman provided us with a compact subroutine which performs the entire synthetic division iteration at 28 decimal digit floating point. In this routine the numbers stored at 28 digits include only those currently needed in the calculation to provide a full high precision iteration so that any order polynomial may be handled without additional storage. The net storage required was less than required for the Fortran version of the same routine. The routine is now part of our 1620 Fortran library.

Investigation throughout the data range showed the remaining roots to be complex. We obtained the remaining roots of the 11th order polynomial by the use of Muller's Method, which, so far in our experience, has never failed to converge (Ref's. 6 and 7). We would welcome comments based on the experience of the audience in this regard. Because the roots may be obtained as the square roots of roots of a lower ordered polynomial, they are symmetrically distributed about the origin as shown in Figure 3.

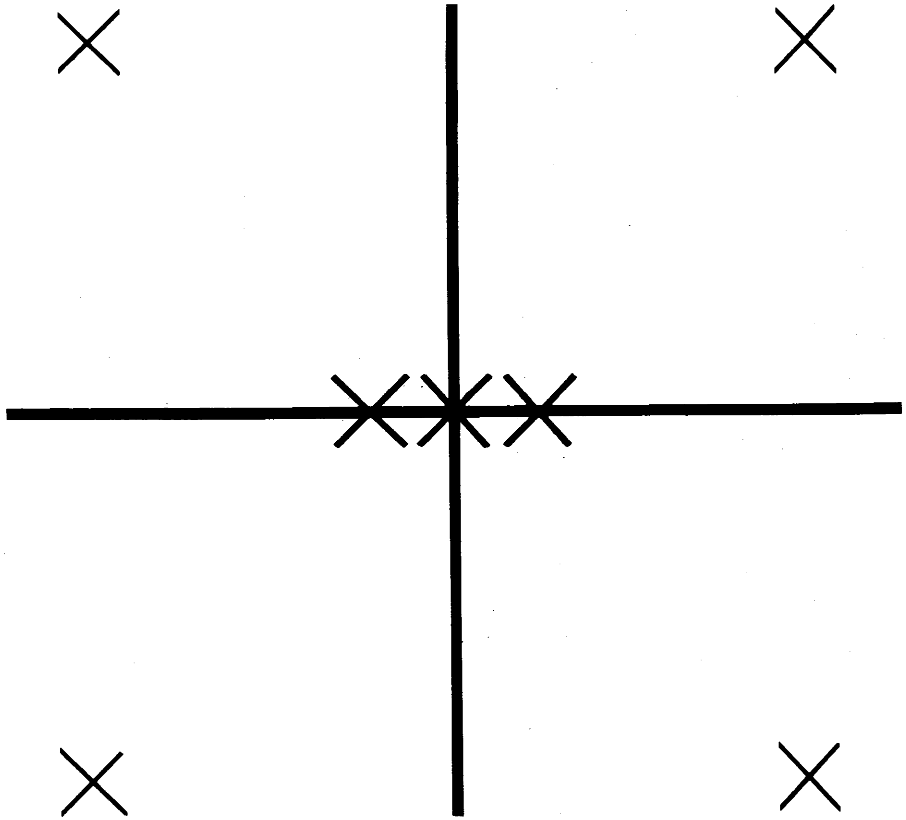


Figure 3. Map of Roots in the Complex Plane - Seventh Order Equation

IV. BASIS FUNCTIONS AND BOUNDARY CONDITIONS

The solution of a homogenous fourth order ordinary differential equation with constant coefficients whose characteristic polynomial has the roots $\pm \alpha \pm i\beta$ can be represented as a linear combination of the functions.

$$\left\{ e^{(\alpha + i\beta)x},\ e^{(\alpha - i\beta)x},\ e^{(-\alpha + i\beta)x},\ e^{(-\alpha - i\beta)x} \right\} \qquad (4.1)$$

Of course, this is not the only set which could have been selected. Such a set is called a system or basis of functions if the member functions are linearly independent. A test for the linear independence is provided by the Wronskian determinant. This is a determinant whose first row is the system itself and whose jth row is made up of J-1st derivative of the function in the corresponding column.

We shall have a great deal more to say about linear independence in the next section of the paper. For the present, we wish to show how the programming and debugging of the problem are simplified by writing the derivative boundary conditions in terms of the Wronskian. Recall the form of the derivative boundary conditions.

$$\sum_{i=1}^{n-1} b_{i,j} \frac{d^{j-1}\tau(x)}{dx^{j-1}} = F_i \qquad (4.2)$$

$$j = 2, n \qquad \text{at } x = 0 \qquad \text{and } x = L$$

We see that

$$\frac{d^{j-1}\tau(x)}{dx^{j-1}} \qquad (4.2a)$$

is a column vector. We also see that recognition of the boundary conditions in the form (4.2) permits us to deal separately with the specification of boundary conditions (b_{ij}) and the determination of the vector (4.2a). For our 11th order problem the matrix ($b_{i,j}$) was computed from a coded input. This provided a flexibility which was most useful when numerical difficulties were seen in the boundary conditions themselves. Such a scheme suggests the possibility of writing a generalized program.

Notice the vector (4.2a) is a function of X and exists in an N space. On the other hand, the matrix $(b_{i,j})$ can be written as N linearly independent conditions at either X = 0 or X = L. Thus, for the two-point boundary value, problem N conditions must be selected from 2N derivative conditions and possibly some integral conditions. This draws attention to the fact the boundary conditions selected must be such as to specify a unique solution. We shall not give adequate coverage to this problem in this paper. Let us now examine the vector (4.2a). It has a physical meaning without regard to the basis or coordinate system in which the solution is written. That is to say it could be written at various values of X as a table of numbers which would be independent of the manner in which it was obtained. To analyze it symbolically, however, we must assume a basis of functions, say $\{\phi_i\}$. The first element of (4.2a)

(j = 1) is the zeroth derivative, or the function

$$\tau(x) = \Sigma \phi_k(x) a_k \tag{4.3}$$

where the a_K are the constants of integration to be determined by the equations 4.2. But:

$$\frac{d^{j-1}\tau(x)}{dx^{j-1}} = \sum_{k=1}^{n} \frac{d^{j-1}\phi_k(x)}{dx^{j-1}} a_k = W(x) a_k \tag{4.4}$$

where W(x) is the matrix of functions from the Wronskian determinant.

The recognition of these matrix products is the key to relieving the program of unmanageable detail. As will be seen later, a factor in the choice of the basis is the ease with which the Wronskian may be developed by a simple set of do loops. The $(b_{i,j})$ matrix may be checked separately. Any linear independence may be displayed in easier-to-recognize form. The program has pattern.

As we have indicated, the specification of boundary conditions which are sufficient and compatible involves more than we can discuss here. The reader is referred to Ref. 19.

V. SELECTION OF THE SYSTEM OR BASIS FUNCTIONS FOR THE SOLUTION

Given an ordinary differential equation whose characteristic polynomial contains the non-repeated roots $\pm \alpha \pm i\beta$ there are many choices of functions for the solution. Consider, for example, these four bases or systems:

$$\left| \; e^{(\alpha + i\beta)x}, \; e^{(\alpha - i\beta)x}, \; e^{(-\alpha + i\beta)x}, \; e^{(-\alpha - i\beta)x} \; \right| \qquad (5.1)$$

$$\left| \; \operatorname{Cosh}(\alpha x) \operatorname{Cos}(\beta x), \; \operatorname{Sinh}(\alpha x) \operatorname{Cos}(\beta x), \; \operatorname{Cosh}(\alpha x) \operatorname{Sin}(\beta x), \; \operatorname{Sinh}(\alpha x) \operatorname{Sin}(\beta x) \; \right| \qquad (5.2)$$

$$\left| \; e^{-\alpha x} \cos(\beta x), \; e^{-\alpha x} \sin(\beta x), \; e^{\alpha x} \cos(\beta x), \; e^{\alpha x} \sin(\beta x) \; \right| \qquad (5.3)$$

and: $Z = L - X$

$$\left| \; e^{-\alpha x} \cos(\beta x), \; e^{-\alpha x} \sin(\beta x), \; e^{-\alpha z} \cos(\beta z), \; e^{-\alpha z} \sin(\beta z) \; \right| \qquad (5.4)$$

The question arises, is there a choice? If these functions are mathematically equivalent, which they are, can one set be superior to another for digital programming? The answer is yes. Let us first dispose of (5.1) on the arbitrary basis that we prefer not to perform complex arithmetic if we can avoid it.

The usual textbook treatment is to point out that the functions must be linearly independent. For the mathematician, this condition is met for all of the bases under discussion. For the digital programmer, however, this situation may be quite different. If the domain L of the solution is large enough, all of the sets of functions expressed in the computer number set become linearly dependent:

$$\lim_{\alpha L \to \infty} \frac{\sinh(\alpha L)}{\cosh(\alpha L)} = 1$$

$$\lim_{\alpha L \to \infty} e^{-\alpha L} \cos(\beta L) = \lim_{\alpha L \to \infty} e^{-\alpha L} \sin(\beta L) = \lim_{\alpha L \to \infty} e^{-\alpha L \pm i\beta L} = 0 \qquad (5.5)$$

In the machine, this breakdown in linear independence becomes exact. That is to say, we may have $\cosh\alpha L = \sinh\alpha L$ to the last digit. Of course one may expect computational difficulties long before the loss of the last tragic digit.

Is there any way out of this dilemma? Again the answer is yes. The moment we realize that the finite number of digits in the calculation limits our ability to produce the "exact" solution we begin to consider analogies with approximate methods. We seek functions to represent our solution which "look like" the solution. Clearly basis (5.4) is greatly superior in this light. Only (5.4) of the bases considered, directly represents a function which arises at disturbances at the boundaries and is damped as it proceeds to the interior of the region.

Comparing basis (5.4) with basis (5.2) it is seen that the functions of (5.2) approach each other in exactly the range in which they become large. The functions of (5.4), on the other hand, approach each other in exactly the range in which they drop out of the solution.

It is not surprising that the basis (5.4) is superior when we take note of the fact that it contains more physical information. Only basis (5.4) is cognizant of the location of the disturbance caused by mismatching strains at X = L.

Let us inquire into the physical significance of positive real parts of characteristic polynomial roots. In the initial value problem, with time as the independent variable, positive real parts have been used as criteria of stability. When space is made the independent variable, and the problem is formulated as an initial value problem, a numerical instability is quickly associated with positive real parts. Consider for example, the differential equation

$$EI \frac{d^4 y}{dx^4} = - ky \tag{5.6}$$

which is the homogenous equation for the deflection of a railroad track. In Ref. 15, the solution is shown in several forms including the method of initial conditions.

$$y(x) = y_o \cosh(\lambda x) \cos(\lambda x)$$

$$+ \frac{\theta_o}{2\lambda} \left| \cosh(\lambda x) \sin(\lambda x) + \sinh(\lambda x) \cos(\lambda x) \right|$$

$$- \frac{M_o}{2\lambda^2 EI} \sinh(\lambda x) \sin(\lambda x) - \frac{Q_o}{4\lambda^3 EI} \left| \cosh(\lambda x) \sin(\lambda x) - \sinh(\lambda x) \cos(\lambda x) \right| \quad (5.7)$$

where

y_o = displacement at origin

θ_o = slope at origin

M_o = bending moment at origin

Q_o = shear at origin

In fact, the solution is damped as one moves away from a local load or disturbance. The method of initial conditions, however, requires that this damped function be composed of a linear combination of rapidly expanding functions.

In the finite arithmetic of the digital computer, this means that if a fly settles on the railroad track in Denver and the initial conditions are determined to the full capacity of the computer number set some place this side of Los Angeles, the railroad tracks will be ripped and torn in the most terrible carnage since World War II.

With time as the independent variable, the model is realistic. Positive exponentials mean positive feedback. The rate of growth of the function is proportional to the function with a positive coefficient of proportionality. "It takes money to make money." "Population growth is explosive in a favorable environment"—"chemical reactions become explosive if the rate of the reaction increases with the reaction products" (including the final energy as a product). But time moves on. Functions do not have causes today and develop into effects yesterday.

Space coordinates, however, are quite arbitrary. One does not need to establish a coordinate system to pull a glued joint. Thus, the physical meaning of the positive exponential in our glue line problem is seen to be a stereotyped pattern of coordinate system specification.

It is well known that the choice of a coordinate system can make the solution of a problem easier or more difficult (Ref. 21). In at least some problems it has great numerical consequences (Ref. 5). Interestingly enough, since the boundary conditions supply the information for the evaluation of the constants of integration, we see that they contain vital information about the origion of the coordinate system. One of the problems in specifying boundary conditions is to make sure they produce a unique solution in a defined coordinate system.

Stress analysts have traditionally handled positive exponentials by testing such a parameter as α over the domain. If αL is large (say $\alpha L > 6$), he uses basis (5. 3) as a special basis. He sets the arbitrary constants associated with positive exponentials equal to zero (when $\alpha L > 6$) and calls his solution the semi-infinite case. We think the choice of basis (5. 4) is superior for several reasons. For one thing, the characteristic polynomial may contain more than one set of complex roots with positive real parts. One set may correspond to a semi-infinite case and another to a short case. This situation actually occurs in our 11th order problem. Additionally, we should like to pick one basis and avoid programming more than one basis of functions.

If our mission in this paper were only to show that the basis (5. 4) is superior for our problem, we could certainly bring this discussion to a close now. Our interest, however, is broader. We wish to explore the relative merits of bases in hopes that we may obtain criteria which help us in the solution of some other problem.

We have identified part of our numerical difficulties as arising from a breakdown in linear independence due to the finite nature of the computer number set. Linear independence is established if there are non-zero values of the Wronskian and Gramian. For basis (5. 3) or (5. 4), we rediscovered an ancient device for writing the Nth derivative which will render our discussion easier. Consider one of the functions, say $e^{-\alpha x}\cos(\beta x)$

$$\phi = e^{-\alpha x}\cos(\beta x)$$

$$\frac{d\phi}{dx} = -\alpha e^{-\alpha x}\cos(\beta x) - \beta e^{-\alpha x}\sin(\beta x) \qquad (5.8)$$

30

But identify

$$\alpha + i\beta = R(\cos(\theta) + i\sin(\theta))$$

$$\frac{d\phi}{dx} = e^{-\alpha x}R\left(-\cos(\theta)\cos(\beta x) - \sin(\theta)\sin(\beta x)\right)$$

$$= -e^{-\alpha x}R\cos(\beta x - \theta) \tag{5.9}$$

It can be shown that:

$$\frac{d^j e^{-\alpha x}\cos(\beta x)}{dx^j} = e^{-\alpha x}(-R)^j\cos(\beta x - j\theta)$$

$$\frac{d^j e^{-\alpha x}\sin(\beta x)}{dx^j} = e^{-\alpha x}(-R)^j\sin(\beta x - j\theta) \tag{5.10}$$

And since

$$Z = L - X \qquad \frac{dz}{dx} = -1$$

$$\frac{d^j e^{-\alpha z}\cos(\beta z)}{dx^j} = e^{-\alpha z}R^j\cos(j\theta + \beta z) \tag{5.11}$$

$$\frac{d^j e^{-\alpha z}\sin(\beta z)}{dx^j} = e^{-\alpha z}R^j\sin(j\theta + \beta z)$$

In the seventh order problem we used the basis

$$\left\{ e^{-\alpha_1 x}\cos(\beta_1 x),\ e^{-\alpha_1 x}\sin(\beta_1 x),\ e^{-\alpha_2 x},\ 1,\right.$$

$$\left. e^{-\alpha_1 z}\cos(\beta_1 z),\ e^{-\alpha_1 z}\sin(\beta_1 z),\ e^{-\alpha_2 z} \right\} \tag{5.12}$$

The jth row of the Wronskian determinant for this basis is:

$$\left\{ e^{-\alpha_1 x}(-R_1)^{j-1}\cos(\beta_1 x - (j-1)\theta_1),\ e^{-\alpha_1 x}(-R_1)^{j-1}\sin(\beta_1 x - (j-1)\,\theta_1),\right.$$

$$(-R_2)^{j-1}e^{-\alpha_2 x},\ \frac{d^{j-1}(1)}{dx^{j-1}},\ e^{-\alpha_1 z}(-R_1)^{j-1}\cos(\beta_1 z + (j-1)\,\theta_1),$$

$$\left. e^{-\alpha_1 z}(-R_1)^{j-1}\sin(\beta_1 x + (j-1)\,\theta_1),\ R_2^{\,j-1}e^{-\alpha_2 z}\right\} \qquad (5.13)$$

Consider this Wronskian if L is large. When X = L, the first three columns of the determinant become very small. If X = 0, the last three become very small. It is shown in Ref. 22 that for an Nth order differential equation which does not contain the (N - 1st) derivative, the Wronskian is constant through the domain. To see that this is the fact for this Wronskian, it is helpful to factor the Wronskian into the three factors.

$$[W(X)] = [R]\ [T(X)]\ [E(X)] \qquad (5.14)$$

Where [R] is the diagonal determinant whose elements are $R_1^{\,j}$, [E(X)] is the diagonal determinant whose elements are the exponential terms

$$\left\{ e^{-\alpha_1 x},\ e^{-\alpha_1 x},\ e^{-\alpha_2 x},\ e^{-\alpha_1 z},\ e^{-\alpha_1 z},\ e^{-\alpha_2 z} \right\}$$

and T (x) is the remaining determinant whose jth row may be written

$$(-1)^j\cos(\beta_1 x - j\theta_1),\ (-1)^j\sin(\beta_1 x - j\theta_1),$$

$$\left(-\frac{R_2}{R_1}\right)^j,\ \cos(\beta_1 z + j\theta_1),\ \sin(\beta_1 z + j\theta_1),\ \left(\frac{R_2}{R_1}\right)^j \qquad (5.15)$$

then

$$[R] = R_1^{(1+2+3+4+5+6)} = R_1^{21}$$

$$[E(X)] = e^{-(\alpha_1+\alpha_1+\alpha_2)(x+z)} = e^{-(2\alpha_1+\alpha_2)L} \qquad (5.16)$$

With considerable manipulation of rows, the [T(X)] determinant can be seen to contain factors of the form

$$\cos^2(\beta x) + \sin^2(\beta x)$$

The Wronskian of basis (5. 2) is less manageable, as may be seen from the single term:

$$w_{1,7} = (\alpha^5 - 10\alpha^3\beta^2 + 5\alpha\beta^4)\sinh(\alpha x)\cos(\beta x)$$

$$- (5\alpha^4\beta - 10\alpha^2\beta^3 + \beta^5)\cosh(\alpha x)\sin(\beta x) \qquad (5.17)$$

Numerical difficulties arise because columns 1 and 2 and columns 5 and 6 of the complete basis may be obtained from each other by replacing $\sinh\alpha x$ with $\cosh\alpha x$ or vice versa. Thus, the linear independence depends upon the ability to distinguish the hyperbolic functions at large values of X with a computer number set. Through considerable algebra the Wronskian may be shown to contain the factors $(\cosh^2(\alpha x) - \sinh^2(\alpha x))$ and $(\cos^2(\beta x) + \sin^2(\beta x))$.

The difficulties with basis (5. 3) interestingly enough do not arise in the same manner as with (5. 2). Except for a minor phase difference in angles βx and βz the most significant difference between (5. 3) and (5. 4) lies in their respective [E(X)] determinants. For basis (5. 3), this becomes

$$e^{(-\alpha_1 - \alpha_1 - \alpha_2 + \alpha_1 + \alpha_1 + \alpha_2)x} = e^o = 1$$

so that on this consideration the basis is seen to be at least as good as basis (5. 4). A similar pattern is observed if we consider the Gramian determinant.

For a basis of functions (ϕ_i), the Gramian determinate is given by

$$G_{i,j} = \int_o^L \phi_i \phi_j \, dx \qquad (5.18)$$

41

we prefer to normalize the Gramian as to

$$G^*_{i,j} = \frac{\int_o^L \phi_i \phi_j \, dx}{\sqrt{\int_o^L \phi_i^2 \, dx} \sqrt{\int_o^L \phi_j^2 \, dx}} \tag{5.19}$$

In this form we may think of the functions ϕ_i and ϕ_j as being coordinate vectors in a function space. The terms $G_{i,j}$ are then cosines of angles between the coordinate vectors. One may also think of them as simple correlation coefficients between the base functions.

Now consider the functions $e^{-\alpha x}\cos(\beta x)$ and $e^{-\alpha z}\cos(\beta z)$.

Then

$$G^*_{1,3} = \frac{\int_o^L e^{-\alpha x}\cos(\beta x) e^{-\alpha z}\cos(\beta z) dx}{\sqrt{\int_o^L e^{-2\alpha x}\cos^2(\beta x) dx} \sqrt{\int_o^L e^{-2\alpha z}\cos^2(\beta z) dx}}$$

$$= \frac{e^{-\alpha x} \int_o^L \cos(\beta x) \cos(\beta z) \, dx}{\int_o^L e^{-2\alpha x}\cos^2(\beta x) \, dx} \tag{5.20}$$

If L is great the definite integral in the denominator will be approximated by the lower limit so that the order of magnitude of $G_{1,3}$ is $e^{-\alpha L}$. Thus, in this basis the functions separate into nearly orthogonal sets associated with the respective boundaries.

A similar situation holds for basis (5.3). The exponentials fall out of the numerator before integration and the denominator contains $e^{\alpha L}$. For basis (5.2), however:

$$G^*_{1,2} = \frac{\int_o^L \cosh(\alpha x)\cos(\beta x)\sinh(\alpha x)\cos(\beta x)dx}{\sqrt{\int_o^L \cosh^2(\alpha x)\cos^2(\beta x)dx}\ \sqrt{\int_o^L \sinh^2(\alpha x)\cos^2(\beta x)dx}}$$

$$= \frac{\int_o^L (e^{2\alpha x} - e^{-2\alpha x})\cos^2(\beta x)dx}{\sqrt{\int_o^L (e^{2\alpha x} + 2 + e^{-2\alpha x})\cos^2(\beta x)dx}\ \sqrt{\int_o^L (e^{2\alpha x} - 2 + e^{-2\alpha x})\cos^2(\beta x)dx}} \quad (5.21)$$

$$\longrightarrow 1$$

How may we distinguish between basis (5.3) and (5.4)? Let us recall that our solution is in the form

$$\tau(x) = \Sigma\, a_i\ \phi_i(x) \qquad (5.22)$$

This can be seen to be the vector $\tau(x)$ in a function space, written in terms of its components a_i along the coordinate axes $\phi_i(x)$. We have seen that basis (5.2) is unattractive because the axes ϕ_i become parallel. For the basis (5.3), the problem is the scaling of the axes. The length of the coordinate vectors is exactly the integral in the denominator of the Gramian. Thus, for a basis with distinct axes, such as (5.3), the choice of a second coordinate origin essentially normalizes the coordinate vectors.

Let us now consider the full matrix for the derivative boundary conditions. We may partition this matrix into four partitions as

$$\left|\begin{array}{c:c} S_{11} & S_{12} \\ \hdashline S_{21} & S_{22} \end{array}\right| \qquad (5.23)$$

Where above the horizontal line we write the conditions to be met at $X = 0$ and below the conditions to be met at $X = L$. To the left of the vertical line we write solution components associated with exponentials with negative real parts and to the right exponentials with positive real parts (or for basis 5.4 solution components in Z).

We have two matrix equations (at $X = 0$ and $X = L$). They may be represented as:

$$[B]\ [R]\ [T(X)]\ [E(X)] = [S] \tag{5.24}$$

The matrices [B] are rectangular. They and the matrices [R] are unchanged from basis (5.3) to (5.4). The changes in [T(X)] are only differences in phase angle of the trigonometric functions. The matrix [E(X)] , however, is very different. At $X = 0$, the E matrix for basis (5.3) is the identity matrix. The E matrix for basis (5.4) is the diagonal matrix

$$\left\{ 1, 1, 1, e^{-\alpha_1 L}, e^{-\alpha_1 L}, e^{-\alpha_2 L} \right\}.$$

This same matrix multiplies the E matrix of (5.3) at $X = L$.

To give the E matrix of basis (5.4)

$$\left\{ e^{-\alpha_1 L}, e^{-\alpha_1 L}, e^{-\alpha_2 L}, 1, 1, 1 \right\}$$

If we denote the complete matrix for the boundary conditions for basis (5.3) as S and the diagonal transformation matrix

$$\left\{ 1, 1, 1, e^{-\alpha_1 L}, e^{-\alpha_1 L}, e^{-\alpha_2 L} \right\}$$

as C, we may write the boundary conditions in basis (5.4) as

$$[S]\ [C]\ [A] = [F]$$

except for the differences in phase angle noted above.

If we chose to pre-multiply [A] , rather than post-multiply [C] , it is seen that C represents a scaling of the coefficients [A] . This problem is identified by Lanczos (Ref. 5) as artificial ill-conditioning. His recommendation is just the sort of rescaling accomplished by the choice of basis (5.4).

VI. REMARKS AND CONCLUSIONS

Using basis (5.4), we finally see the boundary condition equations in the partitioned form of the previous section in a physical light. The matrix S_{11} represents the semi-infinite problem at X = 0. The matrix S_{22} represents the semi-infinite problem at X = L. The matrices S_{12} and S_{21} represent cross-coupling between the semi-infinite solutions.

The final four figures show the solution and components for the seventh order problem solved with basis (5.4). Figure 4 is a plot of the shear stress for a typical joint. Figure 5 shows the components of the shear stress for this particular problem. The components associated with complex roots were too small to show on the same scale. Notice that the physical problem — two transients moving in from the boundaries, the solution, the solution components and the partitioned form of the matrix — all reflect the same pattern.

Figure 6 shows the peel stress solution. Figure 7 shows the components of the peel stress. R_1 is the absolute value of the complex root $\alpha_1 + i\,\beta_1$. R_2 is the value of α_2. Recall with basis (5.4) the derivatives contain successively higher powers of the moduli R_i. Now R_1 is 6.78, whereas, R_2 is 1.77. One is not too surprised, then, when σ which is given by

$$\sigma(X) = \frac{1}{\xi_2}\left(\frac{d^3\tau(X)}{dX^3} - \xi_2 \frac{d\tau(X)}{dx}\right)$$

contains larger components or the functions associated with the roots $\pm\alpha_1 \pm i\beta_1$.

Notice, also, that the magnitude of the real parts serve to determine how local the effects will be. Notice the components associated with $\pm\alpha_1 \pm_i \beta_1$ are much more rapidly damped than the components associated with α_2.

This same problem was solved using basis (5.2). An interesting comparison of the numerical difficulties is provided by comparison of the conditioning number of the matrices which had to be inverted. The conditioning number is the ratio of the largest Eigenvalue to the lowest.

The logarithm of the conditioning number provides an estimate of the number of digits which will be lost in the inversion. For the basis (5.2) solution, a 6 x 6 matrix was inverted. The conditioning number was $2.6.10^8$. For the basis (5.4), two 3 x 3 matrices are inverted (solution by partioning Ref. 10). Their conditioning numbers were $1.10.10^2$.

Our conclusions follow Hamming's beautiful statement (Ref. 1): "The Purpose of computing is insight, not numbers." Recognition of matrix products added tremendously to our insight and provided an unusual opportunity to see the nature of our numerical difficulties.

We confess to a strong interest in writing more solvable equations. The work which has been done on best approaches to the problem of solving poor equations, while very useful, has already run its course. Nothing but more digits will improve on the best methods available.

The problem of writing better equations is certainly not simple. Nor do we feel that we now know how. We do believe that the close imitation of the physical problem is a good clue. Further, for this problem we identified two mechanisms which could affect the equations. The basis (5.2) led to badly skew axes, the basis (5.3) to badly scaled axes. The double coordinate system improved the scaling. It is interesting to note that the normal equations of a least square approximation problem become highly skewed if the coordinate origin is very distant from the center of gravity of the function being approximated.

The possibility of writing a generalized program for the class of problems treated here looks good. If we were to write it, the first thing we should like to do is be very sure of our polynomial root routines. Another difficulty would be choosing the boundary conditions before the nature of the characteristic polynomial was established.

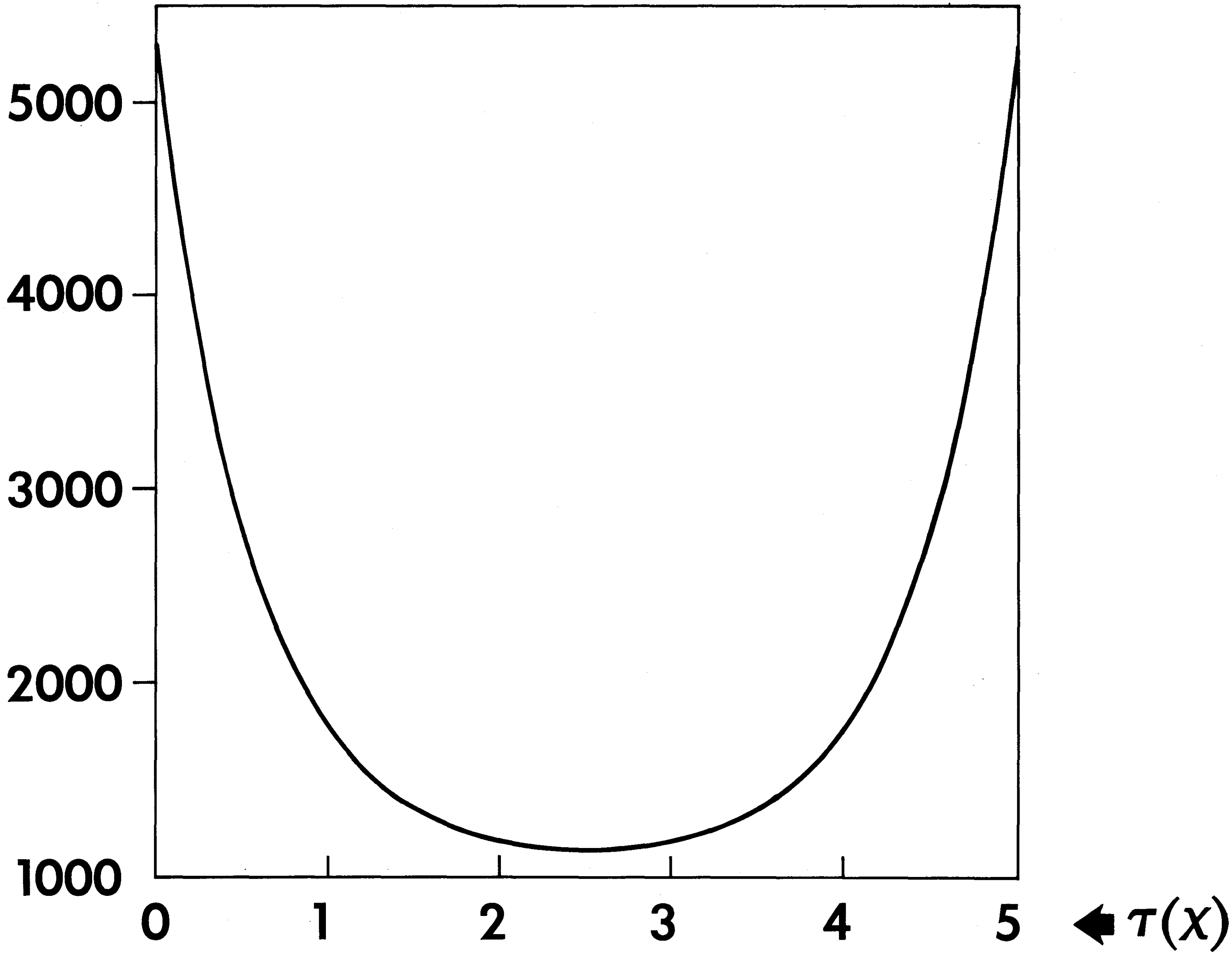


Figure 4. Shear Solution of the Seventh Order Problem

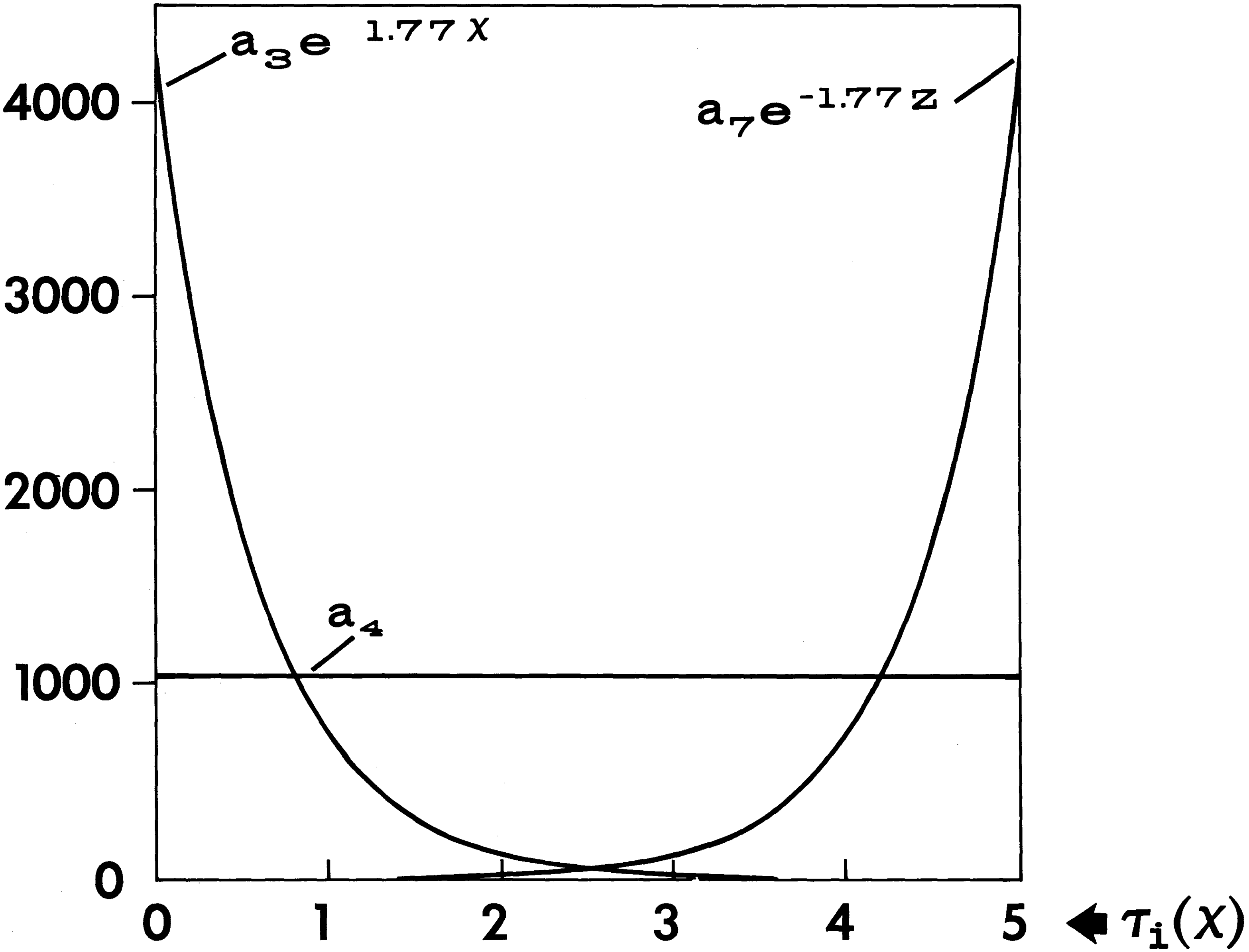


Figure 5. Components of Shear Solution (Note: Complex Components Were Too Small for Scale)

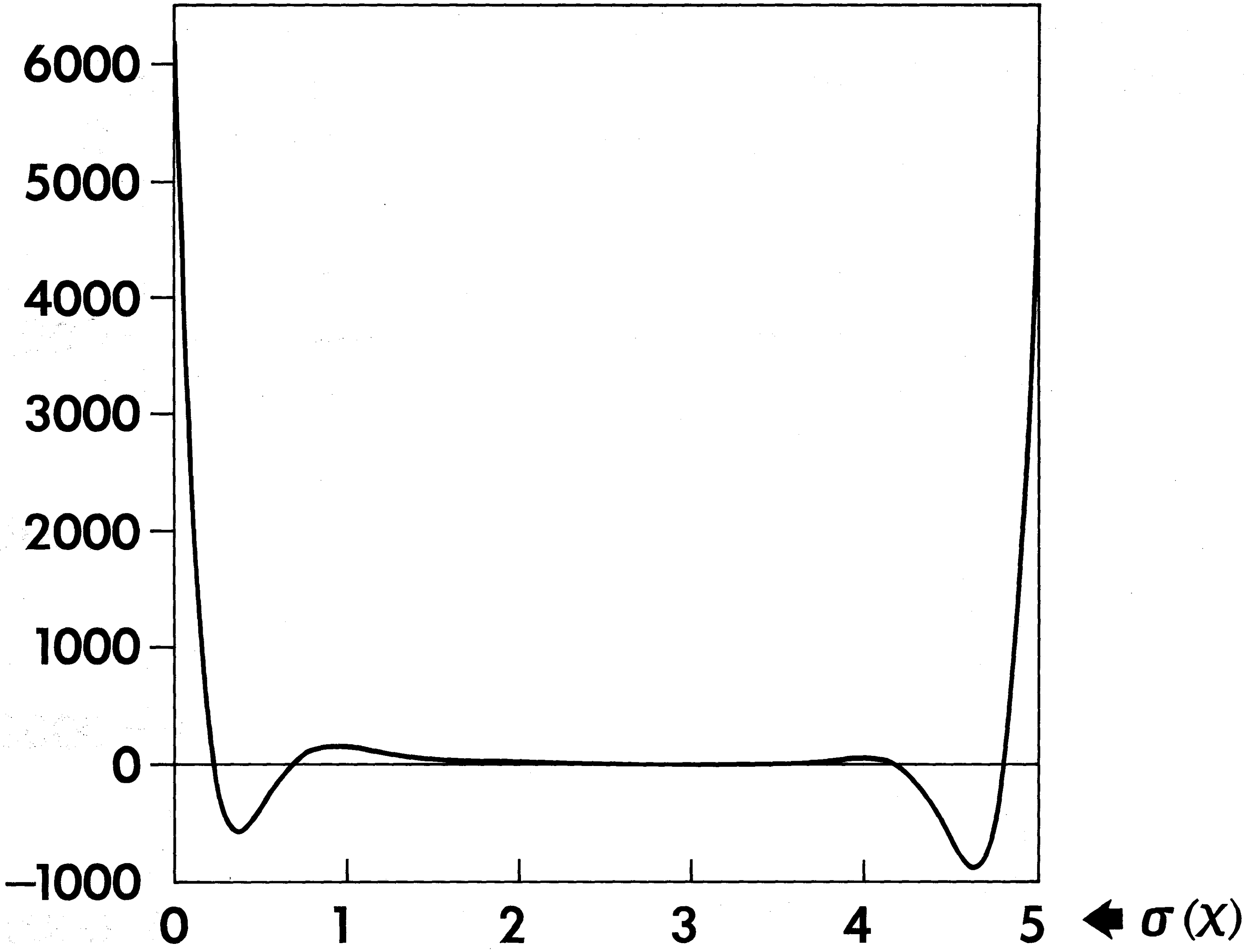


Figure 6. Peel Stress Solution Curve for Seventh Order Problem

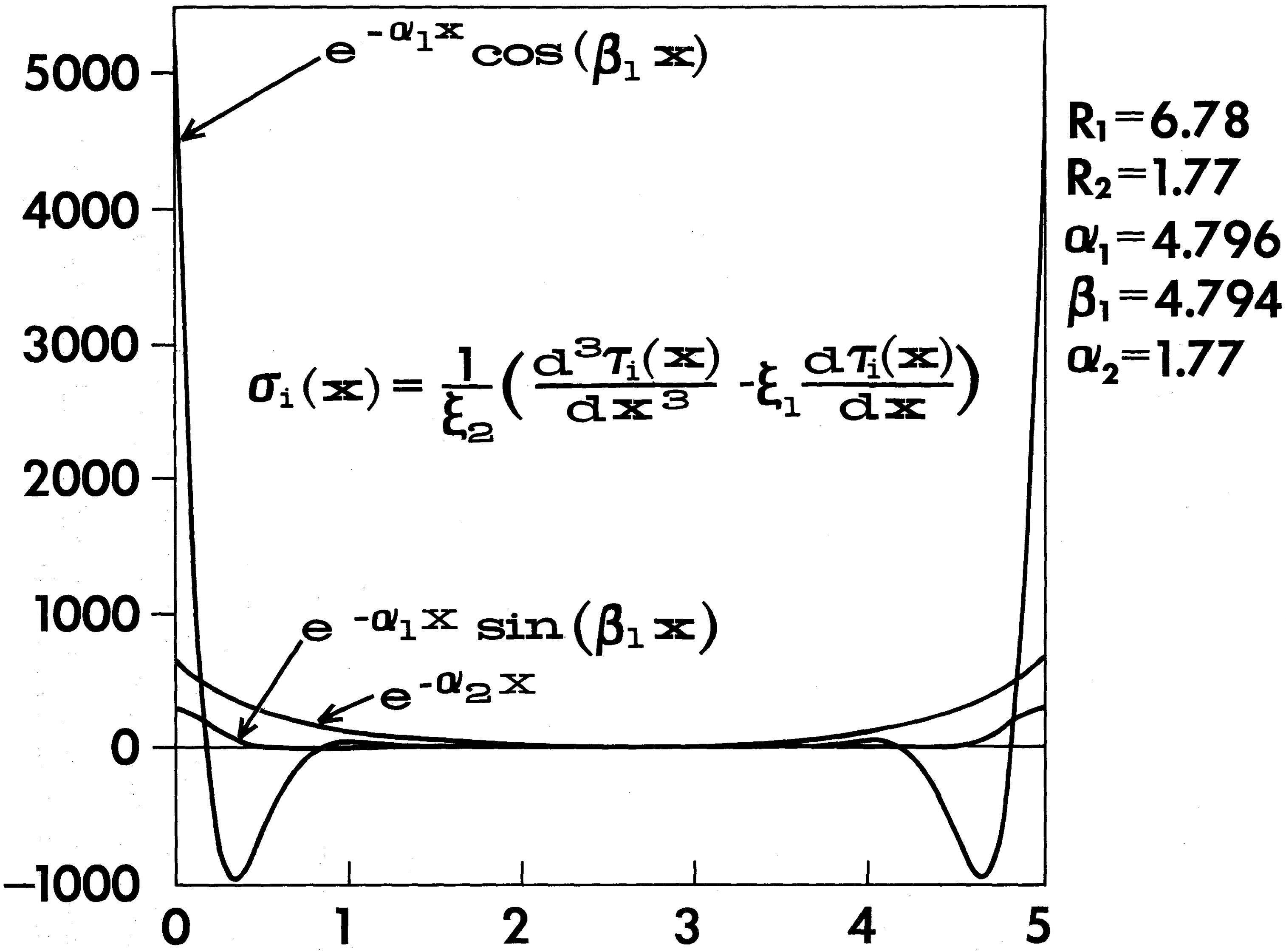


Figure 7. Components of Peel Stress Solution, Seventh Order Problem*

*Curves are labeled with corresponding functions from the basis of the shear solution. As can be seen from peel stress expression, the true functions of this diagram are linear combinations of high order derivatives of the respective shear basis functions.

REFERENCES

1. Hamming, R.W. Numerical Methods for Scientists and Engineers. New York: McGraw-Hill Book Co., Inc. (1962).

2. Goland, M., and Eric Reisner. "The Stresses in Cemented Joints," Journal of Applied Mechanics (March 1944), pp. A17-A27.

3. National Physical Laboratory. Modern Computing Methods. Second Edition, London: Her Majesty's Stationery Office (1961).

4. Burnside, W.S., and A.W. Panton. The Theory of Equations, Vol. I. New York: Dover Publications, Inc. (1928).

5. Lanczos, C. Applied Analysis. Englewood Cliffs, N.J.: Prentice Hall, Inc. (1956).

6. Wilkinson, J.H. "The Evaluation of the Zeros of Ill-Conditioned Polynomials," Numerische Mathematik, Vol. I, pp. 150-180.

7. Lance, G.N. Numerical Methods for High-Speed Computers. London: Iliffe & Sons, Ltd. (1960).

8. Golomb, M., and M. Shanks. Elements of Ordinary Differential Equations. New York: McGraw-Hill Book Co., Inc. (1950).

9. Kaplan, W. Advanced Calculus. Cambridge: Addison Wesley Publishing Co., Inc. (1953).

10. Fadeeva, V.N. Computational Methods of Linear Algebra. New York: Dover Publications, Inc. (1958).

11. Kaplan, W. Ordinary Differential Equations. Palo Alto: Addison Wesley Publishing Co., Inc. (1958).

12. Ralston, A., and Henry Wilf. Mathematical Methods for Digital Computers. New York: John Wiley & Sons, Inc. (1960).

13. Hildebrand, F.B. Introduction to Numerical Analysis. New York: McGraw-Hill Book Co., Inc. (1956).

14. Householder, A.S. Principles of Numerical Analysis. New York: McGraw-Hill Book Co., Inc. (1953).

15. Hetenyi, M. Beams on Elastic Foundation. Ann Arbor: The University of Michigan Press (1946).

16. Courant, R. Differential and Integral Calculus, Vol. I. New York: Interscience Publishers, Inc. (1934).

17. Paige, L.J., and J.D. Swift. Elements of Linear Algebra. New York: Ginn and Company (1961).

18. Churchill, R.V. Fourier Series and Boundary Value Problems. New York: McGraw-Hill Book Co., Inc. (1941).

19. Lanczos, C. Linear Differential Operators. New York: D. Van Nostrand Co., Ltd. (1961).

20. Edwards, J. An Elementary Treatise on the Differential Calculus. London: MacMillan and Co. (1892).

21. Morse, P.M., and H. Feshbach. Methods of Theoretical Physics. New York: McGraw-Hill Book Co., Inc. (1953).

22. Ince, E.L. Ordinary Differential Equations. New York: Dover Publications, Inc. (1926).

23. Ford, L.R. Differential Equations. New York: McGraw-Hill Book Co., Inc. (1955).

READER

A program to read and execute elementary
machine language laboratory exercises

R. C. Steinbach (5145)

Introduction

Grossmont College is one of California's many public two-year colleges. These colleges provide three educational programs: (1) General education courses for the community, (2) Technical-vocational courses, (3) Transfer courses for students going on to four year institutions. Within the technical-vocational area Grossmont College has a data processing program containing a one year (four units per semester) computer programming course which begins with machine language. Students are capable of writing miniature machine language programs after approximately two lecture hours. The program described here monitors the student programs, allowing the student to see his program executed and relieving the instructor of the job of reading machine language programs.

Student Program Format

During the first six weeks of the programming course the students are assigned specific problems to code. Examples of these problems can be found at the end of this paper. For each problem, each student hands in a deck of cards as follows: (See Figure 1) The first card or Header Card is used to identify each student's program. This card contains the student's name beginning in column one and ending with a record mark. It also contains a five digit identification number beginning in column 75 and a record mark in column 80. Reader uses this latter record mark to recognize the header card. The Program Cards follow the header card. The student machine language program is punched 72 digits (6 instructions) per card into as many cards as is necessary to a maximum of ten. A record mark in column 73 of a program card indicates that column one of the next card follows column 72 of the card just read. Thus the last card (it may be the first and hence the only card of the program) has no record mark in column 73.

All programs return control to READER with a branch to 00000. This allows a manual restart (INSERT, RELEASE, START) if the student program hangs up and has not destroyed the READER program.

OPTIONS

During the time that the student has no knowledge of input/output instructions READER outputs the work area so that the student (and the instructor) may check the program results. This output may be suppressed using console switch 3 after the student is familiar with output instructions. The output device, either card punch or typewriter, for READER may be selected using console switch 4. This latter option allows remarks from READER to be output on the same device required of the student in a given problem.

A TYPICAL RUN

For each problem, the programs written by the students form a single deck which follows the READER object deck and four special data cards. (See Figure 2)

The first special data card contains program identification, console switch settings and tabulator information for the operator. The next three cards contain data for the student work area, e.g. numbers to add or subtract, negative numbers to count.

It is advisable to add an instructor written solution to this deck of 4 special data cards so that the students can see the right answers and see one way of writing the program. As far as READER is concerned, this is the first student program. Note that the 4 specialdata cards and the instructor written program form a package which separates the reader object deck from the deck of student programs and which is easy to include for any given assignment.

READER types the program identification and operator message and halts. It then reads the three data cards, initializes the student work area and reads and executes the student programs as follows:

1. Search for Header Card. (Go to 3 when found; go to 2 on last card indicator.)
2. Type "All programs read" and halt. Press start to read next 4 special data cards and new batch of student programs.
3. Type student identification number.
4. Input student program, output student name and number of cards required for program.
5. Branch to student program. Return to 6 is automatic by student or manual by operator.
6. Output work area if switch 3 is on.
7. Initialize student work area.
8. Go to 1.

REMARKS

One should list the student program deck before doing anything else so that there is a permanent record of who turned in what. This is at least a partial defense against a charge of deck shuffling at execution time.

A clumsy student can wipe out core with a TF or TR. The only thing to do is reload the READER, but at least you have his identification number on the typewriter.

A loop, checkstop, or bad operation code can be noted by hand on the typewriter output and the READER restarted manually.

It is possible for a student to read the next student's program as data. As soon as this is obvious, a STOP, INSERT, R/S, will restart the READER. A comparison of the initial listing and the run listing will determine who was left out and his (their) program(s) can be placed at the end of the student program deck.

Conclusion

I would appreciate comments and criticism from any interested person. I do not plan to submit this to the Users Group Library until at least one more class has tried the system; they may think up new ways of giving the READER trouble.

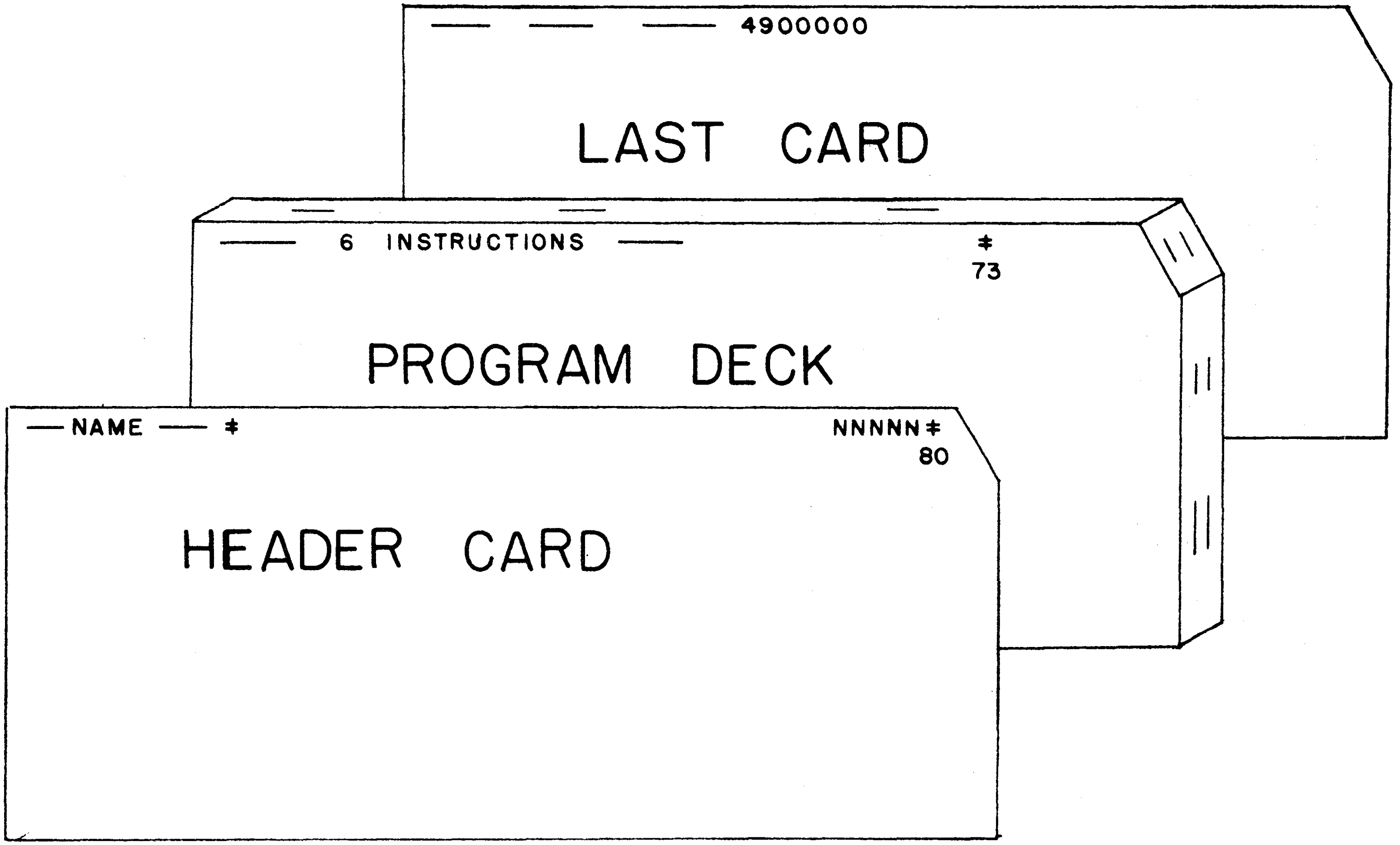

4900000
LAST CARD
6 INSTRUCTIONS
‡
73
PROGRAM DECK
NAME
‡
NNNNN ‡
80
HEADER CARD

FIGURE I

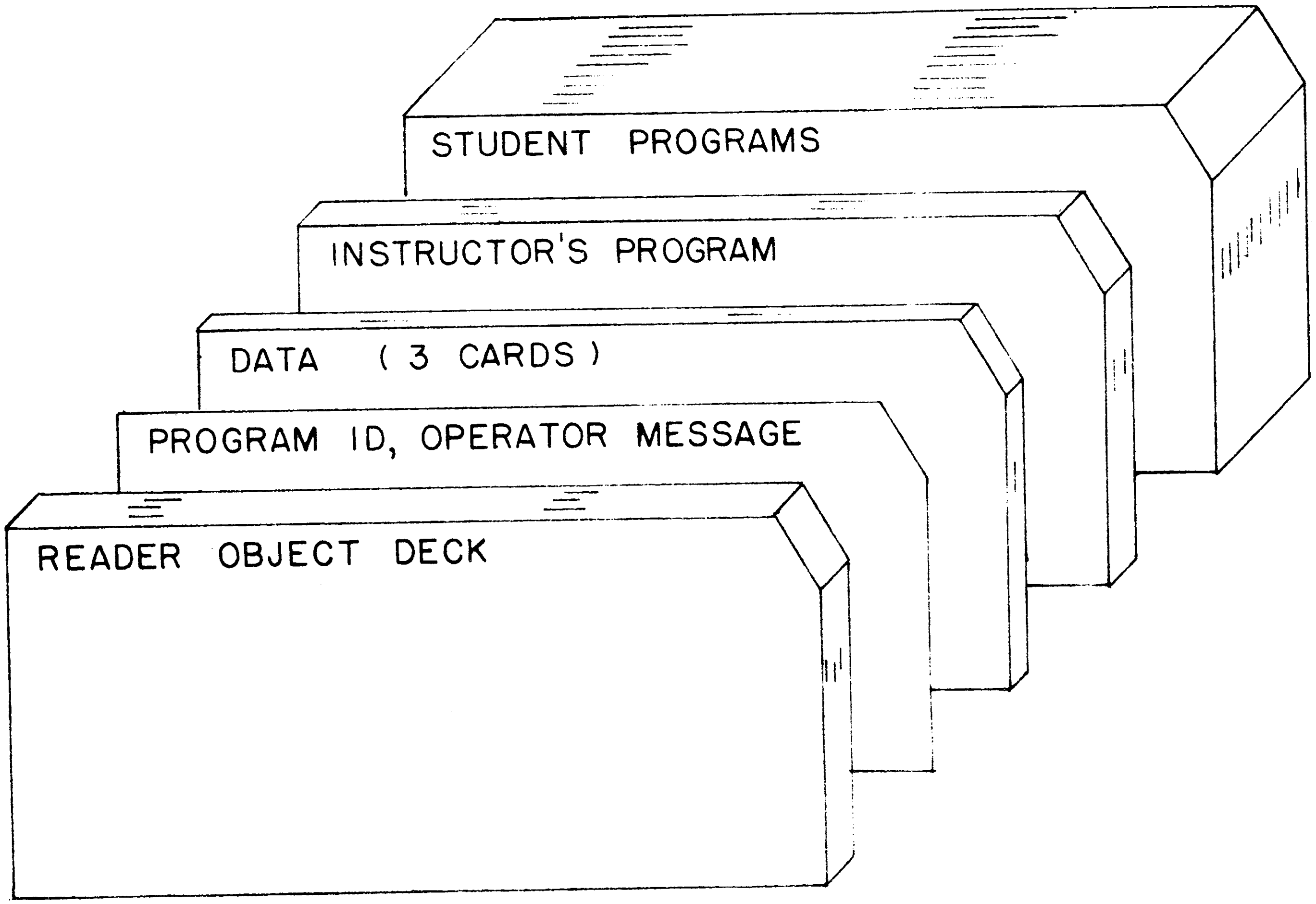

STUDENT PROGRAMS
INSTRUCTOR'S PROGRAM
DATA (3 CARDS)
PROGRAM ID, OPERATOR MESSAGE
READER OBJECT DECK

FIGURE 2

LAB EXERCISE I

Numbers, described below, are in storage with the most significant digit flagged.

Number	Number of Digits	Address of least significant digit
A	6	7006
B	2	7010
C	4	7016
D	3	7021

1.1 Assume no overflow, numbers are integers.

Replace A by A+B
Replace C by C-B
Replace D by D-658

1.2 Assume no overflow, numbers are integers.

Replace A by the integer A-2B+C-D

1.3 Assume no overflow. Assume decimal locations as follows:

A = xxx.xxx
B = .xx
C = x.xxx
D = xx.x

Replace A by A - D
Replace D by C + D
Replace C by C + 2.93

LAB EXERCISE 3

Note: Memory addresses above 11000 are available for your use. The first digit of your program is in 07300.

3.1 Return the carriage on the typewriter. Type out the numerical contents of 7001 - 7009, space the typewriter, type out the alphameric contents of 7030 - 7047. Return the carriage and type the numeric contents of 7030 - 7047. There are no record marks in place.

3.2 Return the carriage, type your name (25 character maximum), tabulate and type your code number.

3.3 As input to your program have one card with your name beginning in col. 1, and the words "1620 I/O PROGRAM" in col. 32-47, and a second card with 5 zeros, 5 ones, 5 twos, etc., and 5 nines in col. 1-50. Duplicate the two cards.

3.4 I will supply you with 3 cards which you will use as input to your program. Each card will have the following format:

A five digit number A in col. 6 - 10.
A nine digit number B in col. 17 - 25.

You are to punch out three cards with the following format:

A and B as above
A+B with low order digit in col. 40
A.B with low order digit in col. 60

There are no flags on the input cards, and there should be no flags on the output cards.

LAB EXERCISE 5

5.1 Type a message to turn on console switch 2 and then halt. If the switch is not set properly repeat the message and halt. Continue this process until the switch is on.

5.2 Two flagged 4 digit integers have their units position in 7005, and 7010 respectively. If the n-th integer is

$$\left.\begin{array}{l}\text{less than } 2222\\ \text{equal to } 2222\\ \text{greater than } 2222\end{array}\right\}\quad \text{put a}\left\{\begin{array}{c}1\\ 2\\ 3\end{array}\right\}\quad \text{in } 7011 + n$$

5.3 35 flagged 4 digit integers have their units position in 7004, 7008, ..., 7000 + 4n, ..., 7140. Tabulate the typewriter and type the number of negative numbers in the list.

5.4 Three 5 digit integers are located in 7005, 7010, and 7015 respectively; arrange them in ascending order in locations 7020, 7025, 7030.

A 519 Simulator
R. C. Steinbach (5145)

Introduction

Card reproduction on the 1620 is not new; the most straight forward approach is to insert 371111100500 391111100400 4900000 R/S. The problem becomes slightly more complex if information is to be deleted, the columns permuted, sequence numbers added, and/or information gang punched into the cards. This paper describes one method of handling these other possibilities.

Method

During the first phase, the simulator sets up a table of source addresses. The first entry in the table is the address of the two digit field to be placed in column one of the output deck; the second entry addresses the source field for column two; etc. During the second phase, a card is read into an input buffer, 80 two-digit fields are transmitted from the appropriate source (the source table is addressed indirectly) sequentially into an output buffer. A card is punched and the next card read, and so on.

Format Cards

The deck to be reproduced is preceded by three format cards called INPUT, OUTPUT, and EMIT. All three cards must be there, however, the INPUT and EMIT cards may be blank. The input format card identifies the source of characters from the deck to be reproduced; the output format card identifies the destination of all characters to be punched in the new deck; the emit format card contains characters to be gang punched into all cards of the new deck.

The simulator produces the source table by scanning the output format card. All columns that are blank in the output format card will be blank in the new, or output, deck. A field of 1's in the OUTPUT card indicates that the source is the same field on the old, or input, deck. A field of 2's (up to 5) indicates a sequence number field on the output deck. Note that this requires the OUTPUT card to be scanned from right to left. A field of 3's indicates that characters are to be emitted from the corresponding columns of the EMIT card. If a field of any other character, e.g. AAA or))))), is encountered on the OUTPUT card, then the INPUT card is searched for a corresponding field. The location of the field on the INPUT card determines the columns to be picked up in the old deck; the location of the field on the OUTPUT card determines the destination in the new deck. If the OUTPUT card contains a character other than the four special characters (blank, 1, 2, 3), that same character must appear on the input format card; furthermore, the field length defined must be the same. If either of these conditions fail, "Format card mismatch" is typed and the program will then accept new format cards. Figure 1 shows an example of the three format cards.

Anomalies

Although it is not immediately obvious, the method chosen to set up the source table allows one field of the input deck to be placed in more than one field of the output deck. To accomplish this, a field indication on the INPUT card appears in several (non-adjacent) fields of the OUTPUT card. Two non-adjacent fields on the input card designated by the same non-special character will not be correctly interpreted.

Sequence numbers (even of different length) may also be punched in several non-adjacent fields.

Modifications

Often, one wishes to change the emit characters whenever a master card is detected. The variety of ways in which a master card may be indicated, and the number of possible reactions to a master card suggests one of the following manual solutions to the problem rather than a fully automated system.

If there are just a few decks headed by master cards, the same INPUT and OUTPUT cards may be used with a different EMIT card. The Master card may be used for an EMIT card if the master card is not to be duplicated and the characters to be emitted are in the correct columns.

If there are many master cards in a particular run, they may be detected using a compare or compare immediate after each card is read. A special routine is then added to the source deck to transmit characters from the Master card to the EMIT card image. The bulk of the routine can be instructions of the form TF EMIT-2+2*ecn, IN-2+2*mcn where ecn stands for emit column number and mcn stands for master column number. With the detect routine and transmit routine added, the source deck is reassembled.

Conclusions

Any suggestions on ways to improve this program will be greatly appreciated. It will be submitted to the Users Group Library after these improvements are incorporated.

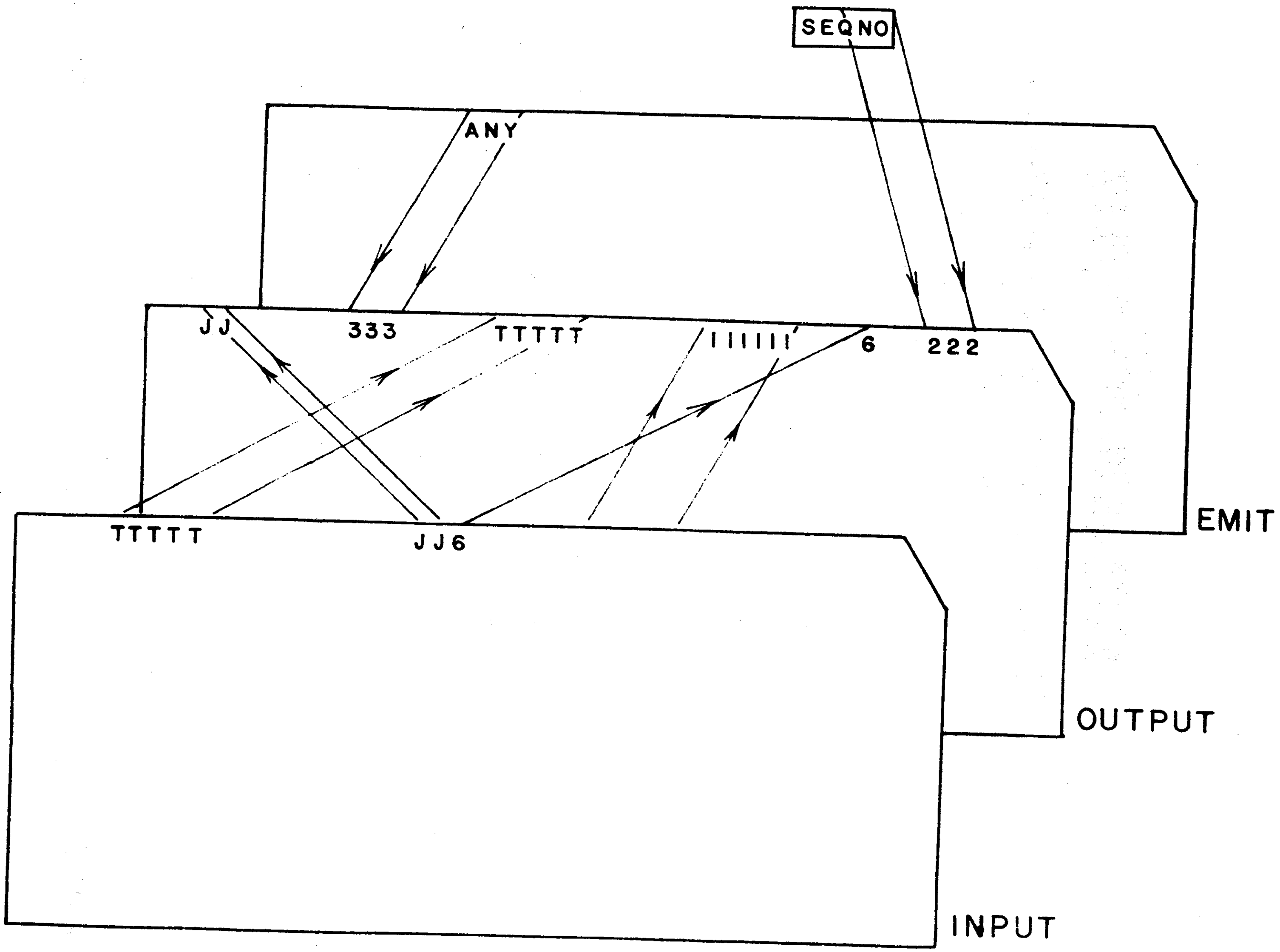

SEQNO
ANY
JJ
333
TTTTT
IIIIII
6
222
TTTTT
JJ6
EMIT
OUTPUT
INPUT

FIGURE 1

GOODYEAR

GOODYEAR AEROSPACE
CORPORATION
ARIZONA DIVISION
LITCHFIELD PARK, ARIZONA

SIMULTANEOUS LINEAR EQUATIONS
WITH COMPLEX COEFFICIENTS

By

N. Kuffel

AAP-18906 May 1, 1964

SIMULTANEOUS LINEAR EQUATIONS
WITH COMPLEX COEFFICIENTS

By

N. Kuffel

INTRODUCTION

This program solves simultaneous linear equations with complex coefficients resulting in complex roots. It was originally developed to solve large systems and has applications in mechanical and electrical engineering problems.

Of the numerous programs available for matrix inversion and simultaneous equations, very few take into account the under-and-overflow problems encountered on large matrix systems. There are no programs published at the present time for the 1620 for solutions of complex simultaneous equations, and very few available even for other machines. Several programs are available on the 1620 for real systems.

This program will solve up to 20 simultaneous linear equations with complex coefficients. Two forms of output results, A+jB and Ke$^{j\psi}$, are available for either a specified limited number of unknowns, or for all unknowns up to 20. The program is written in Fortran with Format and requires 40 K memory.

GENERAL

Given a system of N simultaneous linear equations, in N unknowns, with complex (or real) coefficients, the program solves for the desired number of unknowns in terms of complex numbers. In certain situations, only a few of numerous unknowns are needed. Those desired can be rearranged to appear first in the equations. By specifying the number desired, only that number will be solved for, saving considerable computer time in the case of large systems.

SIMULTANEOUS LINEAR EQUATIONS WITH COMPLEX COEFFICIENTS

Those equations to be solved are set in determinants of the form:

$$|Z| = |a| + j\ |b| \qquad \text{where a and b are the coefficients.}$$

The application of Cramers' Rule gives us:

$$\frac{|\omega|}{|Z|} = \frac{|c| + j\ |d|}{|a| + j\ |b|} \cdot \frac{|a| - j\ |b|}{|a| - j\ |b|}$$

$$= \frac{(|a|\ |c| + |b|\ |d|) + j\ (|a|\ |d| - |b|\ |c|)}{|a|^2 + |b|^2}$$

where Z is the determinant of the coefficients and ω is the same determinant with the coefficients of the desired unknown replaced by the constant terms.

All determinants are evaluated by the triangular method, in which all elements to one side of the leading diagonal are computed to be zero. The determinant is equal to the product of the elements in the leading diagonal of the triangular determinant. This method of evaluation is preferable to that of expansion in terms of minors or the pivotal method because of the storage and time problem involved in the large complex systems.

Previous programs have made it necessary to do a manual rearrangement of data when a zero element is encountered on the diagonal, resulting either from the original coefficients or from subsequent computations. This program will check elements in the same column of the remaining rows of the determinant for a non-zero element. If such a value is found, a row interchange is performed, changing also the sign of the determinant. If no non-zero element is found we have the case of a zero determinant. If this occurs for the coefficient determinant, a message is typed out and a different method of solution must be found for this case of a nonsingular solution. A zero numerator determinant evaluates an unknown equal to zero, which is the correct result.

Over and underflow problems are quite common in matrix problems when doing accumulative operations, such as computing the product of the diagonal elements of the determinant. A scaling procedure has eliminated such difficulties in this program. Before multiplying, each diagonal element is scaled to the range between .1 and 1.0, storing an accumulative characteristic (or power of ten) for the determinant, which is output with the product and then applied in the final division of determinants so that the end results have the correct magnitude.

Especially in the case of large systems, this program has been found to be as efficient even for real systems as most existing programs, particularly because of the row interchange and scaling procedures.

As many as 20 equations in 20 unknowns may be handled by this program on a 40K machine, which is minimum core for the program. The largest system run up to this time has been 18 equations, but no difficulties can be foreseen on any larger problems because of the scaling procedure.

The results are indicated in two forms. The actual outputs are the real and imaginary parts of the solution, as well as the magnitude and phase angle. These will give results in the forms:

$A + jB$ and $Ke^{j\psi}$ where

A = real part

B = imaginary part

K = magnitude

ψ = phase angle in degrees

$$K = \sqrt{A^2 + B^2} \qquad \psi = \tan^{-1} \frac{B}{A}$$

SUMMARY

This program has been used numerous times for several months now, on systems from 3 equations to 18, both partial and complete solutions. Execution times on the 1620 MOD II have run:

3rd and 4th order - 1 min.
15th order - 20 min.
17th order - 29 min.

It should be noted that these times are dependent on the original set up of the coefficients and how many row interchanges are necessary.

SIMULTANEOUS LINEAR EQUATIONS WITH COMPLEX COEFFICIENTS

The program is written in Fortran with Format and uses an ABSOLUTE VALUE subroutine. This can be easily changed in the source program if the subroutine is not readily available. Although the program presently begins at 6600, there is ample storage to recompile with a starting position of 8300 for other machine configurations. It would be a simple matter to change input and output modes to fit other needs and equipment. No sense switches are used.

Sample input and output data follow in Appendix A and a program listing is in Appendix B.

APPENDIX A

Sample input and output data listing follow. Input data follows the same sequence for all programs although Case 1 will be the only one described.

Case 1 - 3rd order complex system, complete solution

Input

1st Card - NSOL = 3 (number of solutions desired) - I3 format
Note statements 500 and 101 in program listing (Appendix B)

2nd Card - N = 3 (order of system) - I3 format

N X N (9) Cards - AR and AI (real and imaginary parts of the coefficients) - both values are on the same card in E14.8 format and are entered row-wise.
Note statement 100 in Appendix B.

N(3) Cards - FR and FI (real and imaginary parts of the constants) - both values on the same card as were the coefficients.

Output

Real and imaginary parts of the input coefficients

Real and imaginary parts of the input constants

Real and imaginary diagonal products, value of the coefficient determinant, scale factors for the products and the determinant, phase angle and magnitude.

Real and imaginary diagonal products, value of the determinant and scale factors for NSOL(3) solutions which include real and imaginary parts (A and B), phase angle (ψ) and magnitude (K).

Case 2 - 4th order real system, complete solution

Case 3 - 4th order real system, partial solution

Case 4 - 3rd order real system, zero determinant

```
  3
  3
+.20110300E+04+.13140000E+03
-.20550000E+04-.22700000E+01
+.00000000E-99+.00000000E-99
-.20550000E+04-.22700000E+01
+.16102980E+05+.21747000E+03
-.14170000E+05-.18500000E+03
+.00000000E-99+.00000000E-99
-.14170000E+05-.18500000E+03
+.22498000E+05+.18500000E+03
+.00000000E-99+.00000000E-99
+.00000000E-99+.00000000E-99
+.83300000E+04+.00000000E-99
```

```
              SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
                      WITH COMPLEX COEFFICIENTS
                            PROG. 223-63

ORDER   3

     REAL                IMAGINARY
  .20110300E+04        .13140000E+03
 -.20550000E+04       -.22700000E+01
  .00000000E-99        .00000000E-99
 -.20550000E+04       -.22700000E+01
  .16102980E+05        .21747000E+03
 -.14170000E+05       -.18500000E+03
  .00000000E-99        .00000000E-99
 -.14170000E+05       -.18500000E+03
  .22498000E+05        .18500000E+03

CONSTANTS
  .00000000E-99        .00000000E-99
  .00000000E-99        .00000000E-99
  .83300000E+04        .00000000E-99

   REAL PROD.          IMAG PROD.          DETERMINANT
  .22940795E-01       .25515200E-02       .53279032E-03      0
              MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      13
              MULTIPLY DETERMINANT BY 1.0E     26

PHASE ANGLE = .63464644E+01DEGREES
MAGNITUDE = .23082251E-01 * 1.0E     13


   REAL PROD.          IMAG PROD.          DETERMINANT
  .24256068E-01       .34347995E-03       .58847480E-03      1
              MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      13
              MULTIPLY DETERMINANT BY 1.0E     26

   REAL PROD.          IMAG PROD.          DETERMINANT
  .23717163E-01       .18609040E-02       .56596678E-03      2
              MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      13
              MULTIPLY DETERMINANT BY 1.0E     26

   REAL PROD.          IMAG PROD.          DETERMINANT
  .23433940E-01       .21191024E-02       .55364013E-03      3
              MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      13
              MULTIPLY DETERMINANT BY 1.0E     26


SOLUTIONS OF THE SIMULTANEOUS LINEAR EQUATIONS

ORDER    3

      REAL              IMAGINARY              PHASE ANGLE           MAGNITUDE
```

```
.10460585E+01    -.10137221E-00    -.55351769E+01    .10509588E+01
.10301213E+01    -.33454431E-01    -.18600954E+01    .10306643E+01
.10191628E+01    -.20980622E-01    -.11793317E+01    .10193786E+01
```

Sample Case 2
Input

```
  4
  4
+.30000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
-.10000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
-.10000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.40000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.30000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.50000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.30000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
+.30000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.00000000E-99+.00000000E-99
```

```
                    SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
                            WITH COMPLEX COEFFICIENTS
                                 PROG. 223-63

ORDER   4

      REAL             IMAGINARY
  .30000000E+01     .00000000E-99
  .20000000E+01     .00000000E-99
 -.10000000E+01     .00000000E-99
  .10000000E+01     .00000000E-99
  .10000000E+01     .00000000E-99
 -.10000000E+01     .00000000E-99
 -.20000000E+01     .00000000E-99
  .40000000E+01     .00000000E-99
  .20000000E+01     .00000000E-99
  .30000000E+01     .00000000E-99
  .10000000E+01     .00000000E-99
 -.20000000E+01     .00000000E-99
  .50000000E+01     .00000000E-99
 -.20000000E+01     .00000000E-99
  .30000000E+01     .00000000E-99
  .20000000E+01     .00000000E-99

CONSTANTS
  .10000000E+01     .00000000E-99
  .30000000E+01     .00000000E-99
 -.20000000E+01     .00000000E-99
  .00000000E-99     .00000000E-99

   REAL PROD.          IMAG PROD.         DETERMINANT
  .49999993E-02     .00000000E-99     .24999993E-04     0
             MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      4
             MULTIPLY DETERMINANT BY 1.0E      8

PHASE ANGLE = .00000000E-99DEGREES
MAGNITUDE = .49999992E-02 * 1.0E     4


   REAL PROD.          IMAG PROD.         DETERMINANT
  .19000000E-00     .00000000E-99     .36100000E-01     1
             MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      2
             MULTIPLY DETERMINANT BY 1.0E      4

   REAL PROD.          IMAG PROD.         DETERMINANT
 -.28999992E-02     .00000000E-99     .84099953E-05     2
             MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      4
             MULTIPLY DETERMINANT BY 1.0E      8

   REAL PROD.          IMAG PROD.         DETERMINANT
 -.50999986E-02     .00000000E-99     .26009985E-04     3
             MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E      4
             MULTIPLY DETERMINANT BY 1.0E      8
```

```
  REAL PROD.         IMAG PROD.         DETERMINANT
 .48999998E-02     .00000000E-99      .24009998E-04     4
             MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E    -3
             MULTIPLY DETERMINANT BY 1.0E    -6


SOLUTIONS OF THE SIMULTANEOUS LINEAR EQUATIONS

ORDER    4

      REAL               IMAGINARY            PHASE ANGLE           MAGNITUDE

 .38000004E-00     -.00000000E-99      .00000000E-99      .38000003E-00
-.57999987E-00     -.00000000E-99      .00000000E-99      .57999986E-00
-.10199998E+01     -.00000000E-99      .00000000E-99      .10199997E+01
 .98000006E-07     -.00000000E-99      .00000000E-99      .98000005E-07
```

```
  4
  4
+.30000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
-.10000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
-.10000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.40000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.30000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.50000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.30000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
+.30000000E+01+.00000000E-99
-.20000000E+01+.00000000E-99
+.00000000E-99+.00000000E-99
```

```
                 SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
                        WITH COMPLEX COEFFICIENTS
                              PROG. 223-63

ORDER   4

     REAL               IMAGINARY
 .30000000E-02      .00000000E-99
 .20000000E-02      .00000000E-99
-.10000000E-02      .00000000E-99
 .10000000E-02      .00000000E-99
 .10000000E-02      .00000000E-99
-.10000000E-02      .00000000E-99
-.20000000E-02      .00000000E-99
 .40000000E-02      .00000000E-99
 .20000000E-02      .00000000E-99
 .30000000E-02      .00000000E-99
 .10000000E-02      .00000000E-99
-.20000000E-02      .00000000E-99
 .50000000E-02      .00000000E-99
-.20000000E-02      .00000000E-99
 .30000000E-02      .00000000E-99
 .20000000E-02      .00000000E-99

CONSTANTS
 .10000000E-02      .00000000E-99
 .30000000E-02      .00000000E-99
-.20000000E-02      .00000000E-99
 .00000000E-99      .00000000E-99

  REAL PROD.         IMAG PROD.        DETERMINANT
 .49999993E-02      .00000000E-99     .24999993E-04     0
          MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E    -8
          MULTIPLY DETERMINANT BY 1.0E  -16

PHASE ANGLE = .00000000E-99DEGREES
MAGNITUDE = .49999992E-02 * 1.0E    -8


  REAL PROD.         IMAG PROD.        DETERMINANT
 .19000000E-02      .00000000E-99     .36100000E-05     1
          MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E    -8
          MULTIPLY DETERMINANT BY 1.0E  -16

  REAL PROD.         IMAG PROD.        DETERMINANT
-.28999992E-02      .00000000E-99     .84099953E-05     2
          MULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E    -8
          MULTIPLY DETERMINANT BY 1.0E  -16


SOLUTIONS OF THE SIMULTANEOUS LINEAR EQUATIONS

ORDER   4
```

```
      REAL              IMAGINARY          PHASE ANGLE         MAGNITUDE

  .38000004E-00     -.00000000E-99     .00000000E-99     .38000003E-00
 -.57999987E-00     -.00000000E-99     .00000000E-99     .57999986E-00
```

```
  3
  3
+.10000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.30000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
+.10000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.20000000E+01+.00000000E-99
+.10000000E+02+.00000000E-99
+.50000000E+01+.00000000E-99
+.15000000E+02+.00000000E-99
```

```
                    SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
                              WITH COMPLEX COEFFICIENTS
                                    PROG. 223-63

ORDER   3

       REAL             IMAGINARY
  .10000000E+01      .00000000E-99
  .20000000E+01      .00000000E-99
  .20000000E+01      .00000000E-99
  .30000000E+01      .00000000E-99
  .10000000E+01      .00000000E-99
  .10000000E+01      .00000000E-99
  .20000000E+01      .00000000E-99
  .20000000E+01      .00000000E-99
  .20000000E+01      .00000000E-99

CONSTANTS
  .10000000E+02      .00000000E-99
  .50000000E+01      .00000000E-99
  .15000000E+02      .00000000E-99

ZERO DETERMINANT - USE DIFFERENT METHOD OF SOLUTION
```

APPENDIX B

```
PAGE 01

SOURCE PROGRAM

07000 C      SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS WITH COMPLEX
07000 C        COEFFICIENTS USING CRAMERS RULE
07000 C      PROG NO 223-63
07000 C      DETERMINANTS EVALUATED BY THE TRIANGULAR METHOD
07000 C      ELEMENTS ENTERED ROW-WISE
07000 C
07000 C      AR-REAL PART OF DETERMINANT
07000 C      AI-IMAGINARY PART OF DETERMINANT
07000 C      REAL AND IMAGINARY PARTS OF A COEFFICIENT ENTERED ON SAME CARD
07000 C      FR-REAL PART OF CONSTANT TERM
07000 C      FI-IMAGINARY PART OF CONSTANT TERM
07000 C      N-ORDER OF THE SYSTEM
07000 C      NSOL - NUMBER OF SOLUTIONS DESIRED (EQUAL TO OR LESS THAN N)
07000        DIMENSION AR(20,20),AI(20,20),FR(20),FI(20),WR(20,20),WI(20,20)
07000        DIMENSION XR(20), XI(20)
07000        CONV = 180./3.14159265
07048    500 READ 101, NSOL
07072        READ 101, N
07096    101 FORMAT (I3)
07118        PUNCH 104
07142    104 FORMAT (//15X41HSOLUTION OF SIMULTANEOUS LINEAR EQUATIONS)
07296        PUNCH 105
07320    105 FORMAT (22X, 25HWITH COMPLEX COEFFICIENTS/29X12HPROG. 223-63/)
07552        PUNCH 119, N
07576    119 FORMAT(5HORDER,I3//5X,4HREAL,11X,9HIMAGINARY)
07712 C      INPUT AND PUNCH MATRIX
```

```
07712        DO 1 I = 1, N
07724        DO 1 J = 1, N
07736        READ 100, AR(I,J), AI(I,J)
07892    100 FORMAT (E14.8, E14.8)
07920        PUNCH 116, AR(I,J), AI(I,J)
08076 C      SET UP WORKING MATRIX
08076        WR(I,J) = AR(I,J)
08232        WI(I,J) = AI(I,J)
08388      1 CONTINUE
08460        PUNCH 103
08484    103 FORMAT (/9HCONSTANTS)
08532 C      INPUT AND PUNCH CONSTANTS
08532        DO 2 I = 1, N
08544        READ 100, FR(I), FI(I)
08628      2 PUNCH 116, FR(I), FI(I)
08748        MN = 0
08784        LIM = N-1
08832     50 SIGN = 1.0
08868 C      DIAGONALIZATION OF DETERMINANT
08868        DO 23 I = 1,LIM
08880        NUM = I
08916        L = I+1
08964     18 DEN = WR(NUM,I)*WR(NUM,I)+WI(NUM,I)*WI(NUM,I)
09300        IF(DEN) 14, 15, 14
09356     14 IF(NUM-I) 914, 914, 24
09424     15 NUM = NUM+1
```

```
09472         IF (NUM-N) 18, 18, 53
09540      53 IF(MN) 4, 4, 5
09596       4 PUNCH 110
09620         PRINT 110
09644     110 FORMAT(//42HZERO DETERMINANT - USE DIFFERENT METHOD OF9H SOLUTION)
09788         STOP
09836      24 DO 16 J = 1, N
09848         WRT = WR(I,J)
09944         WIT = WI(I,J)
10040         WR(I,J) = WR(NUM,J)
10196         WI(I,J) = WI(NUM,J)
10352         WR(NUM,J) = WRT
10448      16 WI(NUM,J) = WIT
10580 C       CHANGE SIGN OF DETERMINANT IF ROWS ARE INTERCHANGED
10580         SIGN = -SIGN
10628     914 DO 23 J = L, N
10640         WRT = WR(J,I)
10736         WIT = WI(J,I)
10832         DO 23 K = I, N
10844         X1 = WRT*WR(I,K)-WI(I,K)*WIT
11072         X2 = WR(I,K)*WIT+WI(I,K)*WRT
11288         WR(J,K) = WR(J,K)-(WR(I,I)*X1+WI(I,I)*X2)/DEN
11660         WI(J,K) = WI(J,K)-(WR(I,I)*X2-WI(I,I)*X1)/DEN
12020      23 CONTINUE
12128 C       ADJUST MAGNITUDE TO AVOID OVER OR UNDERFLOW
12128       5 IS = 0
```

```
12164          DO 200 I = 1, N
12176     220 ABWR = ABSF(WR(I,I))
12272          IF(ABWR) 200, 200, 213
12328     213 IF(ABWR-1.) 211, 200, 210
12396     211 IF(ABWR-.1) 212, 200, 200
12464     212 TTEN = 10.
12500          IS = IS-1
12548          GO TO 214
12556     210 TTEN = .1
12592          IS = IS+1
12640     214 WR(I,I) = WR(I,I)*TTEN
12808          WI(I,I) = WI(I,I)*TTEN
12976          GO TO 220
12984     200 CONTINUE
13020 C        EVALUTION OF DETERMINANT TAKING PRODUCT OF DIAGONAL ELEMENTS
13020          DO 7 I = 2, N
13032          J = I-1
13080          PRODR =(WR(J,J)*WR(I,I)-WI(J,J)*WI(I,I))
13428          PRODI =(WR(I,I)*WI(J,J)+WI(I,I)*WR(J,J))
13764          WR(I,I) = PRODR
13860        7 WI(I,I) = PRODI
13992          PRODR = PRODR*SIGN
14040          PRODI = PRODI*SIGN
14088          DET = PRODR*PRODR+PRODI*PRODI
14184          IF (DET) 111, 121, 111
14240     121 IF(MN) 4, 4, 111
```

```
14296    111 PUNCH 115
14320    115 FORMAT (/2X, 10HREAL PROD., 7X, 10HIMAG PROD. 7X, 11HDETERMINANT)
14478        PUNCH 116, PRODR, PRODI, DET, MN
14538    116 FORMAT (E14.8, 3X, E14.8, 3X, E14.8, I5)
14602        ISD = IS+IS
14650        PUNCH 117, IS
14674    117 FORMAT (10X,44HMULTIPLY REAL AND IMAGINARY PRODUCTS BY 1.0E, I5)
14818        PUNCH 118, ISD
14842    118 FORMAT (10X, 28HMULTIPLY DETERMINANT BY 1.0E, I5)
14954        IF(MN) 8, 9, 8
15010 C      DETERMINANT OF THE COEFFICIENTS IS SAVED FOR LATER COMPUTATIONS
15010      9 BOT = DET
15046        ISZ = IS
15082        PRDIZ = PRODI
15118        PRDRZ = PRODR
15154        PHID =ATANF(PRODI/PRODR)*CONV
15226        AMAG = SQRTF(DET)
15262        PUNCH 109, PHID
15286    109 FORMAT(/13HPHASE ANGLE =, E14.8, 7HDEGREES)
15368        PUNCH 125, AMAG, IS
15404    125 FORMAT(11HMAGNITUDE =, E14.8, 7H * 1.0E,I5/)
15486 C      SET UP DETERMINANTS WITH COEFFICIENTS OF UNKNOWNS REPLACED BY
15486 C        KNOWN TERMS
15486        DO 10 MN = 1, NSOL
15498        DO 11 I = 1, N
15510        DO 11 J = 1, N
```

```
15522          WR(I,J) = AR(I,J)
15678       11 WI(I,J) = AI(I,J)
15906          DO 12 J = 1, N
15918          WR(J,MN) = FR(J)
16038       12 WI(J,MN) = FI(J)
16194          GO TO 50
16202 C        SOLUTION OF THE UNKNOWNS
16202 C        POWER OF 10 READJUSTS TO CORRECT MAGNITUDE
16202        8 VAL = (10.**(IS-ISZ))/BOT
16298          XR(MN) = (PRODR*PRDRZ+PRODI*PRDIZ)*VAL
16430          XI(MN) = (PRDRZ*PRODI-PRODR*PRDIZ)*VAL
16574       10 CONTINUE
16610          PUNCH 106
16634      106 FORMAT (//46HSOLUTIONS OF THE SIMULTANEOUS LINEAR EQUATIONS/)
16766          PUNCH 3, N
16790 3FORMAT(5HORDERI4//5X4HREAL11X9HIMAGINARY7X11HPHASE ANGLE7X9HMAGNITUDE/)
17026          DO 13 I = 1, NSOL
17038          IF(XR(I)) 122, 123, 122
17118      123 PHID = 90.
17154          GO TO 124
17162      122 PHID = ATANF(XI(I)/XR(I))*CONV
17282      124 AMAG = SQRTF(XR(I)*XR(I)+XI(I)*XI(I))
17486          PUNCH 120, XR(I), XI(I), PHID, AMAG
17594      120 FORMAT(E14.8, 3X, E14.8, 3X, E14.8, 3X, E14.8)
17670       13 CONTINUE
17706          PRINT 900
17730      900 FORMAT(31HPAUSE, PUSH START FOR NEXT CASE)
```

```
PAGE 07

17816        PAUSE

17828        GO TO 500

17836        END

SYMBOL TABLE

39999 SIN
39989 SINF
39979 COS
39969 COSF
39959 ATAN
39949 ATANF
39939 EXP
39929 EXPF
39919 LOG
39909 LOGF
39899 SQRT
39889 SQRTF
39879 ABSF
39869 ABSFF
39859 AR     35869
35859 AI     31869
31859 FR     31669
31659 FI     31469
31459 WR     27469
27459 WI     23469
23459 XR     23269
23259 XI     23069
23059 CONV
23049  18000000+03
23039  31415926+01
23029 000
23019 *0500
23009 *0101
22999 *0101
22989 NSOL
22979 N
22969 *0104
22959 *0104
22949 *0105
22939 *0105
22929 *0119
22919 *0119
22909 *0001
22899 I
22889 J
22879 *0100
22869 *0100
22859 *0116
22849 *0116
```

```
PAGE 08

22839 *0103
22829 *0103
22819 *0002
22809 MN
22799  0000
22789 LIM
22779  0001
22769 *0050
22759 SIGN
22749  10000000+01
22739 *0023
22729 NUM
22719 L
22709 *0018
22699 DEN
22689 001
22679 *0014
22669 *0015
22659 *0914
22649 *0024
22639 *0053
22629 *0004
22619 *0005
22609 *0110
22599 *0110
22589 *0016
22579 WRT
22569 WIT
22559 K
22549 X1
22539 X2
22529 002
22519 IS
22509 *0200
22499 *0220
22489 ABWR
22479 *0213
22469 *0211
22459 *0210
22449  10000000+00
22439 *0212
22429 TTEN
22419  10000000+02
22409 *0214
22399 *0007
22389 PRODR
22379 PRODI
22369 DET
22359 *0111
22349 *0121
22339 *0115
22329 *0115
22319 ISD
```

```
22309 *0117
22299 *0117
22289 *0118
22279 *0118
22269 *0008
22259 *0009
22249 BOT
22239 ISZ
22229 PRDIZ
22219 PRDRZ
22209 PHID
22199 AMAG
22189 *0109
22179 *0109
22169 *0125
22159 *0125
22149 *0010
22139 *0011
22129 *0012
22119 VAL
22109 *0106
22099 *0106
22089 *0003
22079 *0003
22069 *0013
22059 *0122
22049 *0123
22039  90000000+02
22029 *0124
22019 *0120
22009 *0120
21999 *0900
21989 *0900
```

APPLICATIONS OF NUMERICAL FILTERS IN THE POWER SPECTRAL ANALYSIS OF STATIONARY TIME SERIES

BY

ALEXANDER A. J. HOFFMAN
TEXAS CHRISTIAN UNIVERSITY
FORT WORTH, TEXAS

Presented At
Western Region 1620 Users Group Meeting
Denver, Colorado

June 17, 1964

We will focus our attention on the spectral analysis of finite length recordings of a physical process which is assumed to be random in nature. For deterministic functions such as periodic and aperiodic functions a harmonic analysis is usually carried out by Fourier series analysis and by Fourier integral analysis, respectively. The discrete line spectrum for a periodic function and the continuous spectrum for the aperiodic function may be determined analytically because these deterministic functions are "known for all values of time". Random series are a class of functions which are not deterministic and do not lend themselves to the same harmonic analysis techniques used for deterministic functions; that is, statistical methods must be used.

The Tukey technique, which is used here, is applicable to random time series which very closely approximate a stationary random ergodic process. This computational procedure yields the variance spectrum of a time series. Other names for the resultant computation are power density spectrum, second-degree spectrum, or quadratic spectrum; all of which refer to the distribution of variance as a function of frequency.

One begins with a recording of a physical process which is assumed to represent a sample of a random process. The record must be free of "pure tone" or periodic components and transients. After sampling the record at equi-spaced intervals the linear trends and average should be removed.

Briefly, the Tukey method consists of computation of statistical estimates of the spectrum of a finite discrete time series by a numerical approximation of the Wiener-Khinchine equations. The procedure involves two steps. First, one computes a set of mean lagged products of the time series. Another name for the set of mean lagged products is the autocorrelation function. The raw power spectral estimates are computed by application of a discrete finite Fourier cosine transform to the autocorrelation function. This transformation gives the desired frequency domain representation of the time series. Systematic statistical errors resulting from use of a finite amount of data appear in the raw power spectral estimates. The Tukey technique to obtain improved spectral estimates involves a smoothing or refining operation performed on the raw estimates.

Slide 1 shows the Tukey equations.

Slide 2 shows an example of a time series to which one might apply the Tukey analysis.

Slide 3 shows the power density spectrum of the time series. Eighty percent confidence intervals are shown.

Your attention is directed to the fact that the power density graph has an upper bound at a point marked f_N and that no power estimates of higher frequency are plotted. This upper band set is known as the Nyguist frequency and is a function of the length of the sampling interval. A full discussion of sampling theory is beyond the scope of this presentation. However, a few brief remarks are in order.

When a continuous function is sampled at equi-spaced intervals, the question should be asked: "How well will the discrete set of sampled values represent the original function?" A continuous function of time is completely determined by its values at equally spaced intervals provided that the continuous function contains no frequencies higher than, say, W cycles per second, and the ordinates are given at points spaced 1/2 W seconds apart, the series extending for all time. This is a statement of the popularly referred to Shannon theorem. Under consideration here is an analysis which is to be based on sampled values obtained from continuous records which are not infinite in extent and are not band limited. Analysis based on finite amounts of data is common to statistical work.

Of immediate concern is the selection of the sampling interval and the problem of aliasing. Consider two sine waves of equal amplitude, but different frequencies.

(See Slide 4)

Attention here is directed to a particular set of sine waves, differing in frequency, but having a common set of equally spaced sample values. Thus, given only the sampled values, a sine wave of a given frequency may be confused with a sine wave of higher frequency.

Specifically, if a harmonic time function X(t) is sampled at equally spaced time intervals Δt, then a frequency

$$f_N = \frac{1}{2\Delta t}$$

called the Nyguist or folding frequency, exists such that the functions with frequencies

$$f \pm nf_N \text{ , for } n = 0,2,4,\ldots,$$

are not distinguishable.

Obviously, then, power contributed to a power spectrum at a given frequency f cannot be distinguished from powers contributed by frequencies $f \pm n\ f_N$. This translation of frequencies is known as aliasing. If the data actually contain power at frequencies greater than f_N, this power will be "folded back" into the principal band which extends from 0 to f_N. Power that is folded back results in a distortion of the true power spectrum in the principal band.

To make the effect of aliasing negligible it is necessary to select a sampling interval "small enough" to place the Nyguist frequency beyond all significant power contributions.

Associated with each spectral estimate there is a confidence interval which depends on the number of degrees of freedom in the computation. If one assumes the distribution of the data to be Gaussian and that the distribution of the variability in the spectral estimates follows the so called "chi-square" distribution, then the number of degrees of freedom may be computed by the convenient formula:

$$k = \frac{2}{m} (N - \frac{m}{3})$$

where k = number of degrees
N = number of sampled values
m = number of the maximum log

The confidence intervals are then computed using the number of degrees of freedom. As the number of degrees of freedom is increased the confidence intervals decrease in size and the computed estimates are more reliable. The number of degrees of freedom is, generally speaking, directly proportional to the number of data points and inversely proportional to the maximum number of lags. Acquisition of more data may be impossible or economically unfeasible and reducing the number of lags reduces the number of spectral points in the frequency range from zero to the Nyguist frequency.

This brings us to the point of this paper.

In many physical processes the power density decreases very rapidly with increasing frequency. Often at the higher frequencies the power density of the process under investigation is of the same order of magnitude as the noise background. One must sample the processes often enough to avoid aliasing which would cause the noise to "fold back" into the frequency range of interest. Then one must take many lags and compute many power density estimates in order to have a good look at the lower frequencies. The consequences of this are large confidence intervals and much computation.

In order to get around this problem one can operate on the original sampled data with a linear operator which is often called a numerical filter because of its mathematical resemblance to an electrical filter. Through use of filters one can change the frequency spectrum in a known and desireable way. In particular, a low-pass filter may be used to suppress the power near the Nyguist frequency and not significantly disturb the low frequency spectrum of a time series.

Slide 5 shows a power density spectrum computed before and after low-pass filtering.

Slide 6 shows a comparison between the mathematical model of an electrical filter which operates on a continuous electrical signal and a linear operator (a numerical filter) which operates on a set of equispaced sample values of a time series. Note that the time domain representation of the electrical filter is characterized by W, the impulse response or memory of the filter. The time domain representation of the linear operator is simply an array of numbers. In the frequency domain both the electrical and numerical filters have representations called the frequency response. It can be shown that the numerical filter is simply a numerical approximation to the mathematical model of the electrical filter.

Slide 7 shows a plot of the coefficients of a low-pass filter.

Slide 8 shows the frequency response of both a high-pass and a low-pass filter.

After the time series has been operated on by say, a low-pass filter, the new time series may be resampled using a larger sampling interval. That is, the set of sampled values may be decimated by taking every other value, every third value, etc. A new lower Nyguist frequency is associated with the power spectrum of the new time series since the new sampling interval is larger than the original one. The low-pass filter has suppressed the power at the higher frequencies and thus all but eliminated possible distortion caused by aliasing. Now the low frequency range may be investigated using fewer lags and thus keep the size of the confidence intervals small.

After the power spectrum has been computed the effect of the filter is removed using the frequency domain representation of the filter.

In various applications high-pass, band-pass as well as low-pass filters have been used. Such computations are used in geophysical applications such as analysis of temporal variations in the earth's magnetic field and in biomedical applications such as analysis of EEG recordings.

Slide 9 shows a macro-flow chart of a computer program, written in 1620 Fortran II, to accomplish the computations discussed in this presentation.

Listings of the program are available from the author. (User 5130).

TUKEY EQUATIONS

AUTOCORRELATION

$$C_r = \frac{1}{N-r} \sum_{q=0}^{N-r} X_q \cdot X_{q+r} \quad , \quad r = 0, 1, 2, \cdots, m < N$$

RAW POWER DENSITY

$$V_r = \left[C_0 + 2 \sum_{q=1}^{m-1} C_q \cos\left(\frac{qr\pi}{m}\right) + C_m \cdot \cos(r\pi) \right] \cdot \Delta t$$

REFINED POWER DENSITY

$$P_0 = \frac{1}{2}(V_0 + V_1)$$

$$P_r = \frac{1}{4} V_{r-1} + \frac{1}{2} V_r + \frac{1}{4} V_{r+1}$$

$$P_m = \frac{1}{2}(V_{m-1} + V_m)$$

$$f_r = \frac{r}{2m\Delta t}$$

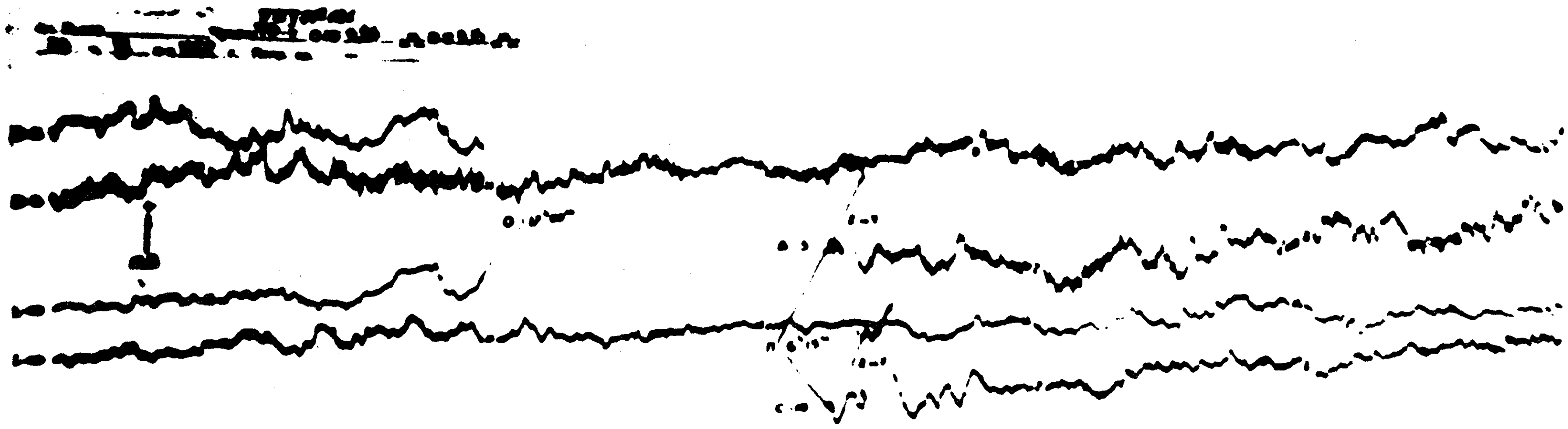
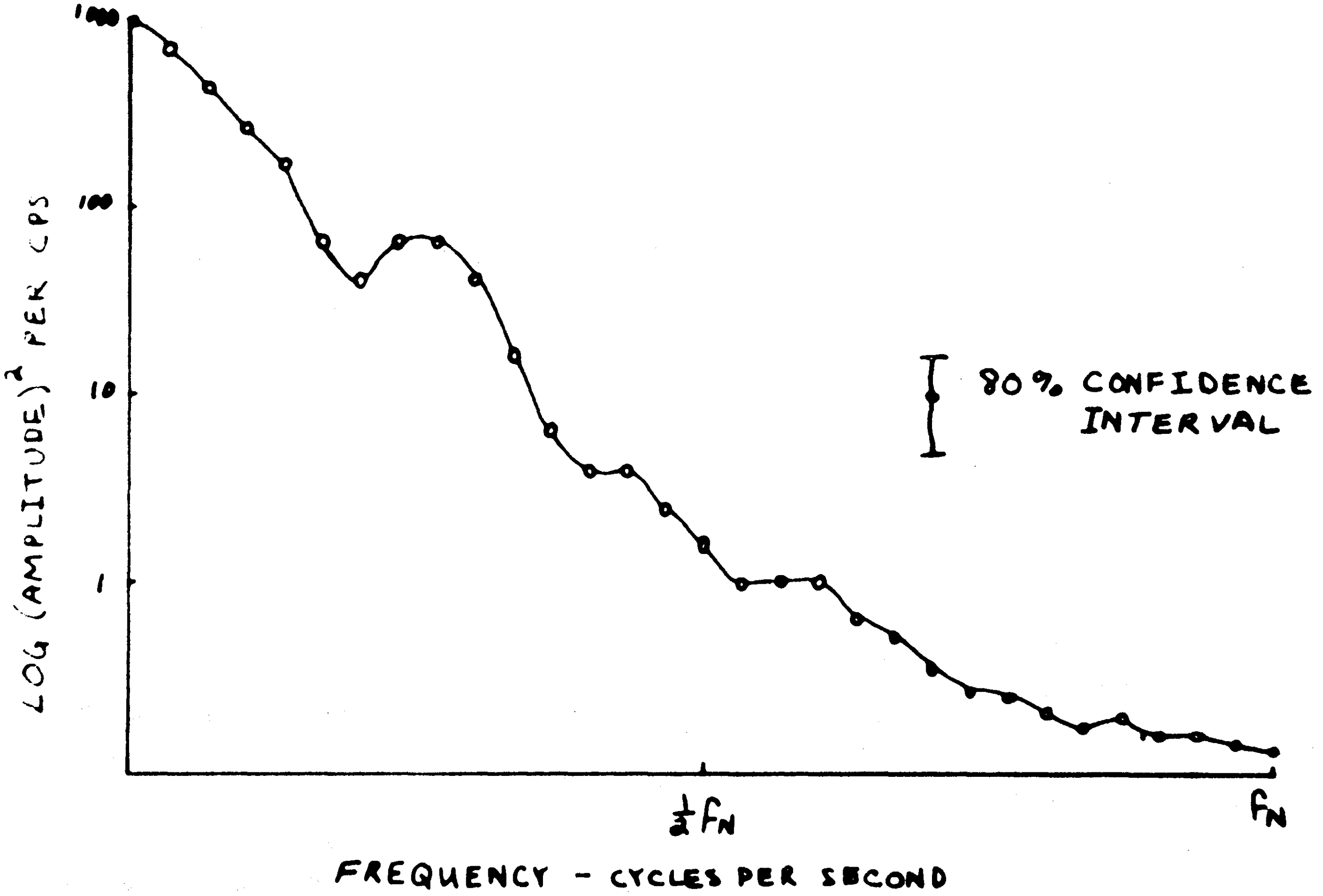


FIGURE 7

TELLUROGRAM FOR LVOV 20 SEPTEMBER 1957

SLIDE 2

POWER DENSITY SPECTRUM



SLIDE 3

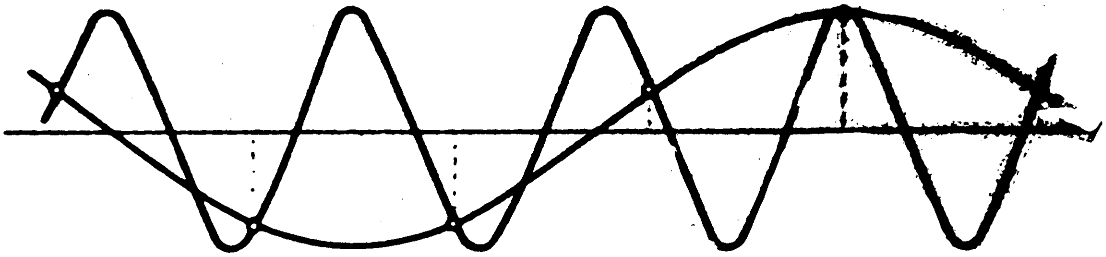
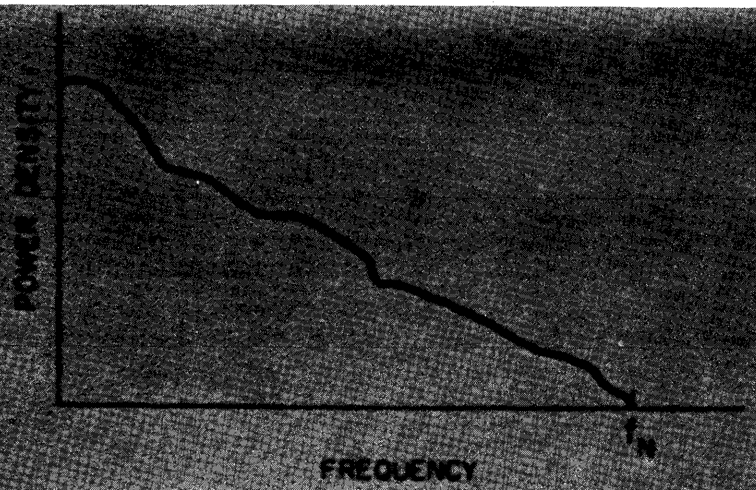
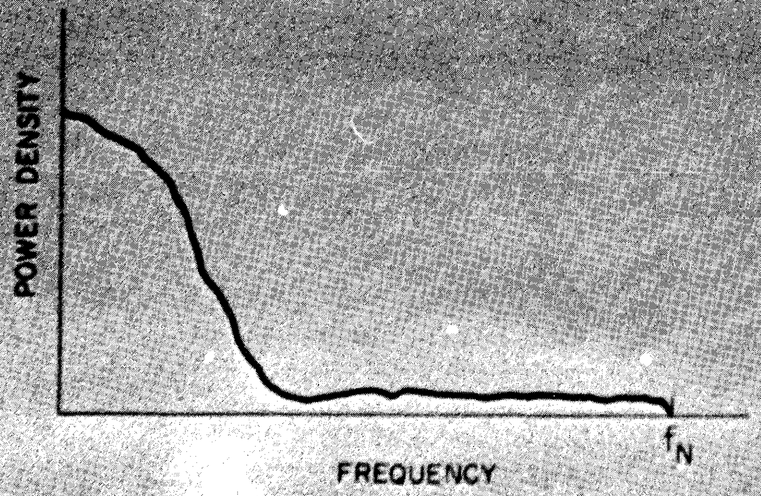


FIGURE 9
SINE WAVES OF DIFFERENT FREQUENCIES WITH THE SAME SET OF EQUALLY SPACED SAMPLE VALUES

SLIDE 4



POWER SPECTRUM BEFORE LOW-PASS FILTERING



POWER SPECTRUM AFTER LOW-PASS FILTERING

FIGURE 10
EXAMPLE OF A POWER SPECTRUM BEFORE AND AFTER LOW-PASS FILTERING

SLIDES

ELECTRICAL FILTER	NUMERICAL ANALOGUE
 	$a_k = W(k\Delta t)$
$H(t) = \int_{0^-}^{\infty} G(t-\tau)\, W(\tau)\, d\tau$	$v(t) = \sum_{k=0}^{K} a_k u(t-k\Delta t)$
$G(t,f) = e^{j2\pi ft}$	$u(t,f) = e^{j2\pi ft}$
$H(t) = \int_{0^-}^{\infty} e^{j2\pi f(t-\tau)}\, W(\tau)\, d\tau$	$v(t) = \sum_{k=0}^{K} a_k e^{j2\pi f(t-k\Delta t)}$
$H(t) = e^{j2\pi ft} \int_{0^-}^{\infty} e^{-j2\pi f\tau}\, d\tau$	$v(t) = e^{j2\pi ft} \sum_{k=0}^{K} a_k e^{-j2\pi fk\Delta t}$
$H(t,f) = Y(f)\, G(t,f)$	$v(t,f) = u(t,f) \cdot Y(f)$

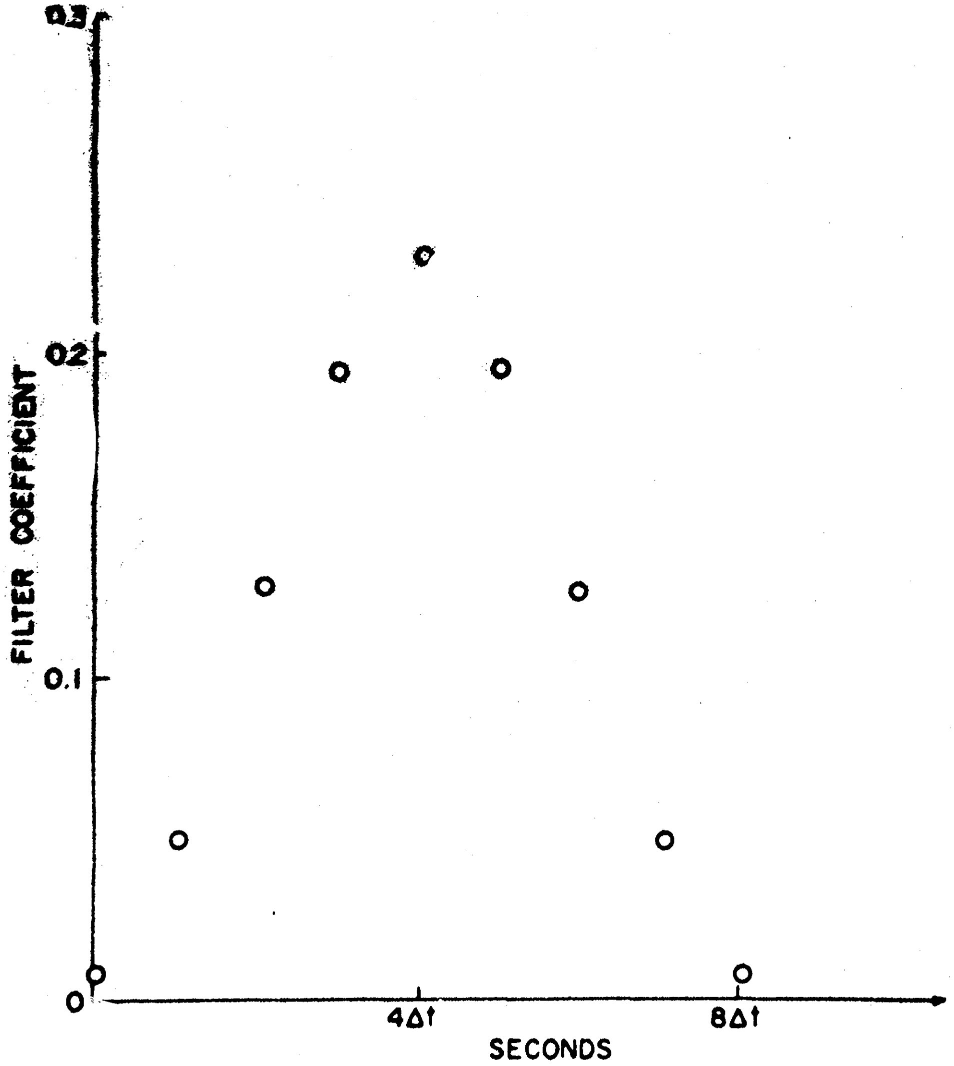


FIGURE 13
THE FILTER COEFFICIENTS OF THE LOW-PASS FILTER, CASE B OF TABLE I

SLIDE 7

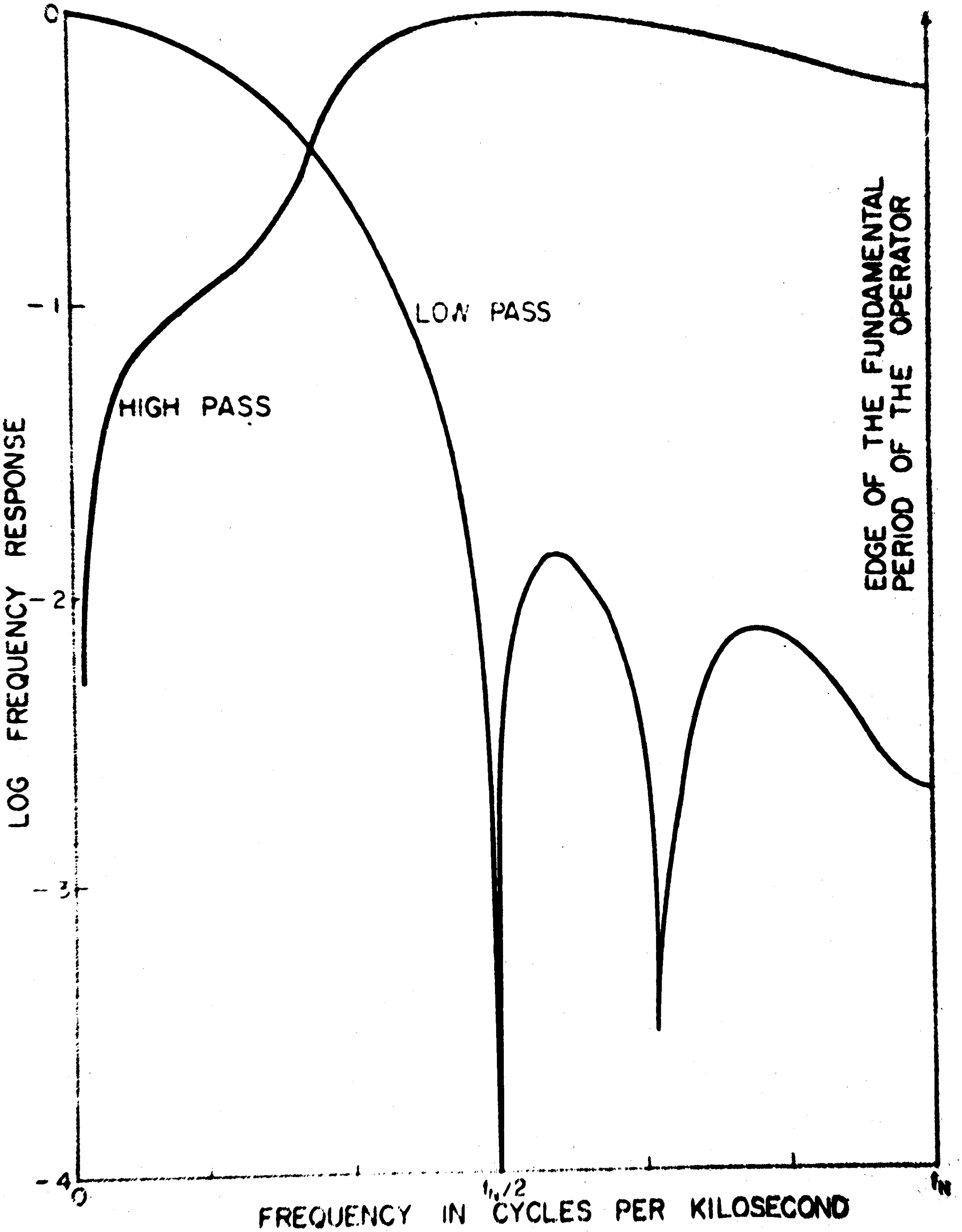


FIGURE 14
THE FREQUENCY RESPONSE OF LOW-PASS
AND HIGH-PASS FILTERS, TABLE I.

SLIDE 8

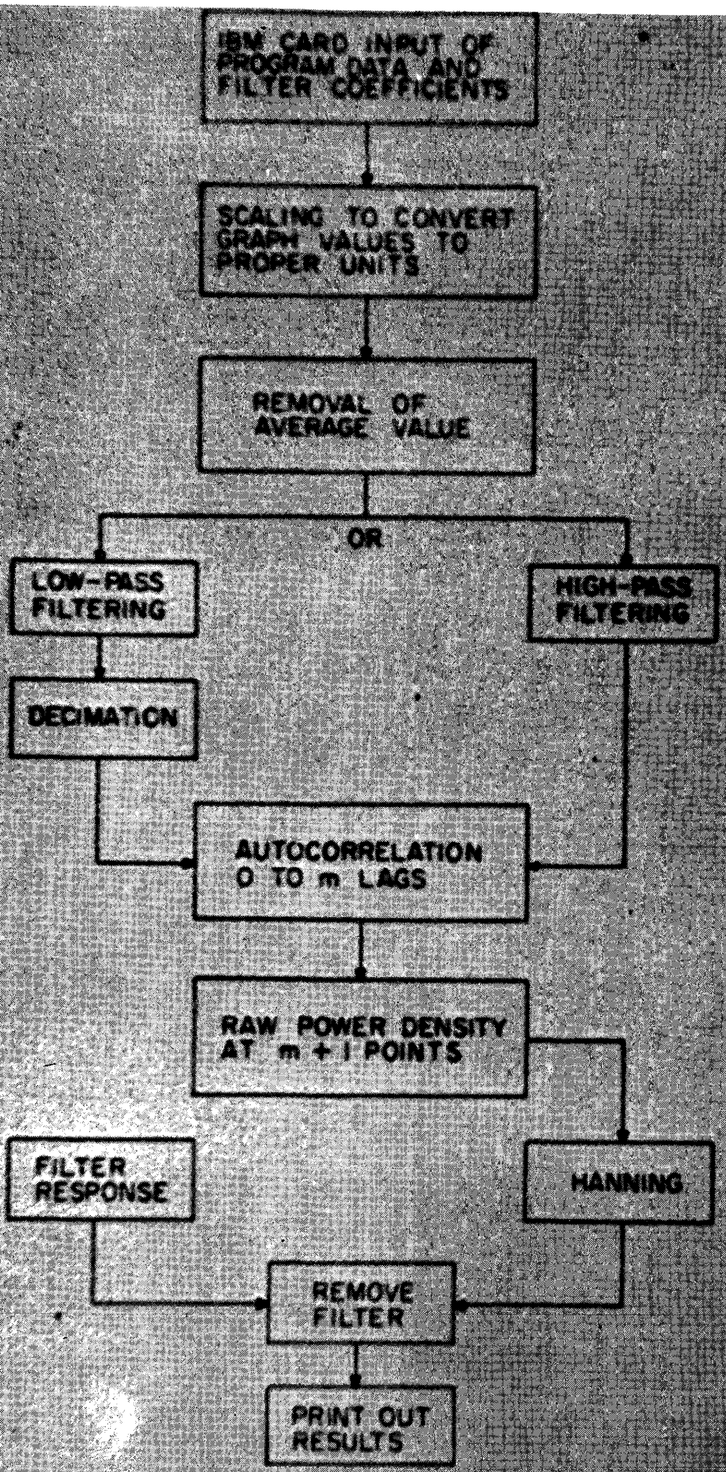


FIGURE 8
MACHINE COMPUTATION FLOW CHART FOR A SINGLE TIME SERIES

SLIDE 9

IBM 1620 ASSISTS STUDENT COUNSELORS AT JUNIOR COLLEGE

Paul S. Chan
IBM CORPORATION
3610 - 14th Street
Riverside, California

May 18, 1964

TABLE OF CONTENTS

ABSTRACT

IBM 1620 ASSISTS STUDENT COUNSELORS AT JUNIOR COLLEGE

The present study reports findings, based on the computed results from the IBM 1620, concerning the extent to which test scores on the college freshman testing program - such as the ACE, SCAT, Co-operative English Tests - are able to predict academic success or failure in specific junior college courses. Scattergrams have been created for those correlations of highest significance to assist counselors in estimating the incoming student's aptitude for college level study and in making a more accurate appraisal of the student's competence in a particular subject area.

Paul S. Chan
May 15, 1964

INTRODUCTION

Unlike private colleges, the state colleges, or the state university, California's public junior colleges are required by law to admit any resident of their districts who is a high school graduate or who is over 18 and able to profit from instruction.

Junior college administrators have interpreted this as meaning that they cannot deny admission to any applicant who has reached his 18th birthday, although virtually all now have retention policies which deny re-enrollment to students who fail to maintain a "satisfactory" grade point average. At one time, many administrators interpreted the legislative mandate to mean that they could not set any qualification for registration in any class. An apparent change in legislative sentiment has combined with the realities of post-war enrollment pressures to cause most junior colleges to search for some equitable means of screening from classes (particularly from transfer classes) those students who have little opportunity to succeed.

The freshman testing program has been an established practice at Riverside City College, a public junior college, for the past years. Although the counselors and admissions officers have been making extensive use of these tests to assist in laying out the academic path of many students, there have been no attempts until recently to make regular evaluations of the measuring instruments in use. Recently an IBM 1620 was installed at the college. One of the first projects to use the system was ab attempt to determine the relationship between the test scores and the final grades in specific courses. It is anticipated that the results will improve placement of students in appropriate sections or courses, and selection of students for particular areas of concentration or pre-professional training.

PURPOSE OF THE STUDY

The battery of tests - ACE, SCAT and others - were administered to the in-coming new students for the dual purpose of counseling and placement. Since this investigation was the initial application, the present study was to demonstrate the validity of the battery for these purposes.

Another aim of the study was to modify the battery to include only those tests best suited for the screening program. Excessive overlap of abilities measured by one test and those measured by another results in a waste of the student's time. Also, too great an array of scores for academic counselors might prove more confusing than helpful.

It was anticipated, too, that critical cut-off scores could be developed for each test, making it both practical and possible to advise the individual student, upon the basis of his score, just what his chances for success of failure in a specific course would be.

DATA

This study involved over 800 students who were enrolled in Psychology 49, a freshman orientation course, and who had completed one or more of 25 courses which the college wished to examine.

There were fifteen predictors. These included:

1. (1) three scores from the ACE (Quantitative, Linguistic, and Total)
2. (2) the R.C.C. Arithmetic Competency Test of 40 items
3. (3) three scores from the School and College Ability Tests (SCAT, Quantitative, Verbal, and Total)
4. (4) six scores from the Cooperative English Tests, Form 1A-1960 EDITION (Vocabulary, Level of Comprehension, Speed of Comprehension, Total Reading, English Expression, and Total English)
5. (5) overall high school grade point averages (to obtain this figure academic subjects and others such as typing, speech, journalism, and music courses were used. Physical education, military science and driver education were not used. Shop courses were used where it was the student's high school major.)
6. (6) academic grade point averages (to obtain this figure only solids such as English, foreign languages, math at the algebra and higher level, history and sciences, but not including general science, were used.)

ACE and Arithmetic scores were easily obtained because they are a part of the placement battery of tests required of all new students. The SCAT and Cooperative English test scores were obtained by testing in the Psychology 49 classes and the two high school grade point averages were rather tediously obtained by employing an individual to compute the figures by hand.

The courses included chiefly transfer courses with a few not-transfer type courses and represented a cross-section of the major divisions within the college.

DEPARTMENT	Course No.	Descriptive Title
Anthropology	2	Cultural Anthropology
Art	1A	History and Appreciation of Art
Biology	1	General Biology
Business	1A	Principals of Accounting
Business	18A (hour)	Business Law
Business	50A (51A)	Elementary Accounting
Business	81A (50A)	Business Mathematics
Chemistry	1A	Chemistry
Chemistry	2	Introductory General Chemistry

DEPARTMENT	Course No.	Descriptive Title
Electronics	51	Electrical Fundamentals of Electronics
English	1A	English Composition
Geography	1	Introductory Physical Geography
History	3	American History
History	4A	History of European Civilization
History	6A	Political and Social History of the US
Math	3A	Analytic Geometry and Calculus
Music	20	History and Appreciation of Music
Nursing	1A	Introduction to Nursing
Philosophy	6A	Introductory Philosophy
Physical Science	1	Introduction to Physical Science
Physics	2A	General Physics
Political Science	3	American Political Institutions
Psychology	1A	General Psychology
Sociology	1	Introduction to Sociology
Spanish	1	Elementary Spanish

This battery of tests was originally selected to provide a basis for predicting over-all scholastic success and success in specific subject-matter areas: The ACE for general scholarship, with its Q and L sub-scores for areas of primarily quantitative and verbal content respectively; the Co-operative English Test for English and other areas which require considerable reading; the Mathematics Tests for placement in mathematics and allied physical science courses.

Final grades of the students in each of the chosen freshman courses were compared with their scores on each of the tests, The courses were chosen from four areas: Language, Humanities, Social Sciences, and Natural Sciences.

METHOD OF ANALYSIS

Results of the test battery were separated into ten test variables:

(1) Three scores, the Q, L, and T, were derived from the ACE;

(2) Six scores from the Co-operative English Test;

(3) Three scores, the V, Q, and T, from the SCAT; and

(4) One score from Mathematics Placement Test.

All the test scores and course grades were recorded in punched cards. An analysis program was written in Fortran.

Coefficients of correlation were computed by the 1620 between scores on each of these tests and final grades in each course. To substantiate the validity of the results, besides the correlation coefficient, regression line coefficients, standard error of estimate, and standard error of regression coefficient b, the significance of r and of b were analyzed. A summary of equations for these calculations can be found in Appendix D.

CORRELATIONS BETWEEN TEST SCORES AND FINAL GRADES

		MATH	ACE			CO-OP ENGLISH						SCAT		
	cases		Q	L	T	1	2	3	4	5	6			
						(Vo)	(Le)	(Sp)	(Tr)	(Exp)	(Tot Eng)	V	Q	T
Language														
Spanish	33	064	044	295	297	268	326	196	233	179	232	281	161	362"
Humanities														
Anthropology	22	131	287	397	290	089	122	018	040	228	182	478*	049	159
Art 1A	18	488*	009	491"	362	656"	287	100	546"	794*	737*	382	388	519"
Music 20	22	463"	233	363	363	286	458*	404	389	404	456*	396	330	018
Philosophy 6A	24	415	117	196	084	596*	457"	051	027	154	091	065	045	501"
average		399	159	362	275	407	441	144	251	495	367	330	203	299
Social Science														
Geography	18	201	169	488"	438	525"	612"	688*	675"	336	548"	616*	093	242
Natural Science														
Biology	40	125	436*	326"	433*	433*	334"	449*	379"	547*	536"	391"	375"	375"
Chemistry 1A	55	442*	185	165	189	160	033	048	106	146	116	208	329"	266"
Chemistry 2	30	599*	217	406"	386"	416"	316"	421"	473*	551"	598*	411"	405"	124
Electronics	13	101	400	348	359	439	421	405	425	501	492	392	588"	484
Mathematics 3A	23	578*	121	245	144	360	357	324	368	305	324	413	225	126
Nursing 1A	25	127	209	376	144	596*	457"	310	503"	294	452"	579*	558*	384"
Physical Science	21	534"	342	306	450"	133	582"	449	377	528"	538"	065	491	328
average		278	218	241	218	326	290	260	295	320	342	275	385	232

*	Indicates .01 level of significance
"	Indicates .05 level of significance

RESULTS

The results of this study are reported in Appendix A, a table presenting the correlations between test results and course grades. Within each curricular area, the average correlation with each test is also given. All the correlations coefficients in Appendix A at the .01 level of significance are marked with an asterisk and at the .05 level with double primes.

Grades in some courses appear to correlate relatively well with scores on all the tests, while those in orther courses showed low correlations with most of the test scores, For example, biology has 12 out of 13 subscores with correlation at either .05 or .01 level of significance and chemistry has ten out of 13, whereas Spanish and electronics have only one out of 13 at .05 level of significance. Some explanations may be offered for this phenomenon. One is that the differential magnitude of the correlations depends partly on the magnitude of the reliabilities of the grades in those courses. Sectionings of a course will certainly be a factor to affect the magnitude. Another factor is that grades in some courses are based on objective-type examinations, while in others on a more subjective basis.

The relatively high predicitive power of the mathematics placement test in the Natural Science Division is more or less expected. However, an almost equivalent result was found in the Q part of the SCAT Test. This is an indication that it may be possible to obtain the same predictive information from either of the tests, so duplication of student's effort can be avoided. It is quite unexpected that Spanish correlates with only the total score of the SCAT Test in the entire battery. Also, electronics correlates only at .05 level of significance, with Q part of the SCAT Test. It is possible that this phenomenon is due to the fact that SCAT Tests involves not only the psychological functions commonly measured by tests of verbal ability, but also a particular type of reasoning ability important in academic success which is not assessed by any other tests employed in the present battery.

The two parts, speed of comprehension and total reading, of the Cooperative English Test show high correlation with geography. This can be explained because of the fact that the Social Studies courses normally require more speed in reaidng and in comprehension. The significant correlation at .01 level between philosophy and the vocabulary part of the Cooperative English Test certainly implies the requirements to succeed in the course.

In general, the six parts of the Co-operative English Test correlate relatively better than any of the three parts in the ACE Tests, with all the selected courses. This is illustrated by the r-values of .794 with Art, .458 with Music, etc.

The tendency was noted also for correlations to be relatively high or low with reference to separate courses rather than to the different tests. It was hypothesized that this phenomenon might be the result of difference among the courses in inter-section standardization reliability of grading, or use of objective examinations.

SUMMARY

In this study of the value of a battery of aptitude and achievement tests for the prediction of junior college freshman grades, test scores were correlated with final grades in a variety of freshman courses. The individual correlations appeared small, but the relative predictive power was demonstrated clearly.

The following major conclusions concerning the predictive significance of the present battery appear to be warranted:

(a) Overlapping of tests in the battery used is evidenced, suggesting that such an extensive array of examinations is somewhat superflous and repetitive. Both over-all and individual course predictions could be made with even greater accuracy with a more abbreviated battery.

(b) From the scattergram, it was found that it is feasible to determine the cut-off score in screening and to obtain more insight in the statistical probability of achievement of a student in a particular course.

(c) Because of the small number of cases in this particular study, a caution against placing too much weight on individual test scores in guidance, selection or placement is in order.

SCATTERGRAM OF SCAT T-SCORE vs. CHEMISTRY 1A

	F	D	C	B	A
99-95			3		1
94-90		2	3		
89-85	1	2	2	2	
84-80	2	3	5	1	
79-75	2	3	1		1
74-70	1	3	4	1	
69-65	4	1			
64-60	3				
59-55			1		
54-50	1				
49-44		1			1
TOTAL	14	14	19	4	3

SCATTERGRAM OF MATHEMATICS PLACEMENT
vs.
CHEMISTRY 1A

	F	D	C	B	A
44-40				1	
39-35	3	3	11	3	2
34-30	9	7	7	1	
29-25	4	4	1		1
24-20	1				
TOTAL	17	14	19	5	3

APPENDIX (D)

SUMMARY OF EQUATIONS

(1) Variances

$$S_x^2 = \frac{n\sum x^2 - (\sum x)^2}{n(n-1)}$$

$$S_y^2 = \frac{n\sum y^2 - (\sum y)^2}{n(n-1)}$$

(2) Regression Line

$$b = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$

$$a = \frac{1}{n}\left(\sum y - b\sum x\right)$$

(3) Correlation Coefficient

$$r = \sqrt{\frac{\left[(n\sum xy) - \sum x \sum y\right]^2}{\left[n\sum x^2 - (\sum x)^2\right]\left[n\sum y^2 - (\sum y)^2\right]}}$$

(4) Standard Error of Estimate

$$S_{y/x} = \sqrt{\frac{n-1}{n-2}\left(S_y^2 - b^2 S_x^2\right)}$$

(5) Standard Error of Regression Coefficient b

$$S_b = \frac{S_{y/x}}{S_x\sqrt{n-1}}$$

(6) Signficance of r
Compare $|r|$ with the critical value in statistical table for 2 variables and n-2 degrees of freedom.

(7) Signficance of b
Compare $t = \frac{|b|}{S_b}$ with the critical value in statistical table for n-2 degrees of freedom.

1620 COMPUTER UTILIZATION IN A WIND TUNNEL DATA ACQUISITION SYSTEM

by

Stanley E. Wisniewski
Programming Operations Section
NORTHROP DATA PROCESSING

Presented to the
1620 Users Group
Brown Palace Hotel
Denver, Colorado

18 June 1964

NORTHROP CORPORATION

TABLE OF CONTENTS

ABSTRACT

This paper describes how the IBM 1620 computer was teamed with a high-speed digital data acquisition system and two tape units to perform on-line processing of wind tunnel test data. The total installation is located in the Northrop Norair wind tunnel complex comprised of three tunnels: subsonic, transonic-supersonic, and hypersonic. The processing installation provides a central data acquisition and reduction function for all three tunnels, even simultaneously when necessary.

The high-speed data acquisition section scans, measures, and digitizes test data, introduces identification information, and records the data on magnetic tape for instantaneous reading by the 1620, in a read-after-write manner. The 1620 then reduces the data into tabulations meaningful to the aerodynamics research engineers, enabling them to make early evaluation of test run results and to proceed with model changes if called for.

During off-line operations, the computer is available for other applications, and has full control of the tape units.

INTRODUCTION

Today, more than ever, competition in the aerospace industry is very keen and time is one of the most important elements to be utilized. For this reason, a company that makes use of wind tunnels must also have a satisfactory test data acquisition system and a means of automatically reducing the collected data as soon as it becomes available.

In the following paragraphs you will learn how we at Northrop have improved our techniques in this area. Our wind tunnels will be described as well as our data acquisition system to which a 1620 computer is coupled. Also of interest will be the changes we designed into the 1620 computer to make it suitable for our applications and the programs we have written to fulfill our objectives.

To the general public, wind tunnels are environmental chambers used to test model planes, but to the aerodynamicist, wind tunnels are probably the most superior devices used in aeronautical and aerospace research and development. Because of modern wind tunnels, today's test pilots are no longer the nerveless stunt men of the past, but professional engineers. Wind tunnels offer both fast and accurate data as well as the ability to simulate the different types of atmospheric conditions of any time of day or year. However, they are by no means new tools. Years before the Wright Brothers famed flight at Kitty Hawk, wind tunnels, crude as they were, gave valuable aerodynamics data which proved the feasibility of powered flight. The original wind tunnel employed by Orville and Wilbur Wright is on exhibit at the Air Force Museum, Wright-Patterson Air Force Base in Dayton, Ohio.

WIND TUNNEL TESTING FACILITIES

Dominated by its 100,000-cubic-foot vacuum sphere is the supersonic-hypersonic wind tunnel facility at Hawthorne, California. This space age test facility provides test velocities from Mach 0.5 to Mach 14 with temperatures to 3000 degrees and simulated altitudes to 200,000 feet. To my knowledge, no privately-owned wind tunnel in the United States can produce the combined heat, pressure, velocity and run time that are obtainable with the one at Hawthorne. This relatively new, dual-circuit facility provides a greatly expanded capability for aerodynamics testing on advanced aircraft, missiles and space systems. It consists of two separate wind tunnel circuits: transonic-supersonic (Mach 0.5 through Mach 5) and hypersonic (Mach 6 through Mach 14). Design models can be tested for periods of at least 30 seconds in the supersonic circuit and up to one minute in the hypersonic circuit. The hypersonic tunnel can accommodate up to six 30-second runs each eight-hour shift. More test runs of proportionately shorter duration are possible.

Test sections, in which the models are mounted for aerodynamic study, measure two feet square in the supersonic circuit and 30 inches in diameter in the hypersonic circuit. A special "free jet" section in the hypersonic circuit allows removal of a model from the air flow while air flow is being established, thus protecting the model from excessive heat loads. The pressing of a button promptly injects the model into the flow stream. In a transonic or supersonic run, air passes from storage through a settling chamber (to smooth the airflow and remove any turbulence), is expanded through a nozzle (to establish Mach number), flows through the test section and then is forced through a "second throat" to reduce its velocity and to recompress it to atmospheric pressure before it exhausts through a muffler.

In a hypersonic run, air must be expanded so much (to achieve the higher velocities) that its temperature could actually be reduced to a point where the air would turn to liquid. To prevent liquefaction, an electrically fired heater containing a 16-ton bed of 3/8-inch alumina pebbles heats the air to temperatures as high as 3000 degrees Fahrenheit before it reaches the hypersonic nozzle. When the air is cooled by expansion, its temperature is therefore still high enough to keep it from liquefying.

From the test section, the hypersonic air passes through a "second throat" as in the supersonic circuit, to reduce velocity and then through a cooler to remove heat. It is then discharged into a large 100,000 cubic-foot vacuum sphere. The vacuum sphere is essential to hypersonic operations in order to achieve the high velocities desired in the test section. With storage pressure fixed at 3,200 pounds per square inch, the required pressure ratio obviously cannot be met by discharging the "used" air to atmospheric pressure (14.7 pounds per square inch). A low-pressure atmosphere is necessary and this is the function of the vacuum sphere.

About 100 feet from the supersonic-hypersonic facility and in another building is the 7' x 10' subsonic wind tunnel which went into operation in the year 1956 and was used in the very successful development of the Northrop T-38 Talon supersonic trainer, F-5 fighter, and Laminar Flow Control (LFC) airplane. During those tests, the output of test data was punched onto cards, carried to a remotely-located IBM 704 computer installation, processed and returned in a relatively long turn-around time (normally about three days; on emergency basis about four hours).

DATA ACQUISITION SYSTEM

Today in the same building that houses the subsonic tunnel, is the data acquisition system, which we are very proud to possess. It was designed and built to our specifications by the Astrodata Corp. It serves all three of our wind tunnels. The data from any two of the three remotely-located tunnels can be transmitted to this center simultaneously.

The analog data, supplied by transducers at each of the tunnels is digitized by an analog-to-digital converter (ADC) in the central data system. The digital data from the ADC is then sent to the formatting generator where it is joined by other digital data from the model-position encoders, the time-of-day clock and also the switch settings from both the transmitting site and the central data system. The switch settings provide fixed information such as the barometric pressure, the test number, the run number and the date. The formatting generator then assembles and prepares the data for recording on magnetic tape. The records produced by the formatting generator are of variable length and automatically padded to contain an integral multiple of six characters, so that the resulting magnetic tape recordings can be used with both the 7090 and 1620 IBM computers. The ability to read the system-generated tapes by the 7090 computer proved very valuable during system checkout, because the 1620 computer was not adapted to handle magnetic tapes until later.

There are two types of records produced by the data acquisition system. The first of these is the title run record which identifies the test run by a test number, a run number, four parameters, the barometric pressure, the day and the time of day, and the model position by roll, yaw and pitch. The activation of the title push button switch will initiate output of a title run record consisting mainly of the above information provided through manually-set, thumbwheel switches. The second of the two types of records produced by this system is the data record. A data record is generated when the data circuit is closed (manually or automatically). The data record consists of an identification header, the time of day, the model position, and data from all site input channels programmed for the specific test.

COMPUTER UTILIZATION

The 1620 computer employed is a Mod I with 40,000 core storage positions. It is equipped with most of the special, built-in features (indirect addressing, hardware divide, and floating point arithmetic). This computer is attached to the data acquisition system by an umbilical cord; it has been programmed to read and reduce the data as it is being recorded on any one of the two magnetic tape units. The reading is accomplished in a read-after-write manner, termed "eavesdropping." The information is introduced into the computer by the read gap, which is positioned a distance of .300-inch behind the write gap of a two-gap read-write head, almost immediately after the information is written onto the tape by the system. The normal function of the read gap, which is to provide parity checking during the recording process, was extended to make this possible. The two magnetic tape units used are Datamec D2020. These units are IBM compatible, using either 200 bpi or 556 bpi tape formats at 30 ips tape speed. The Central Data System (CDS) records at the 556 bpi density.

Eavesdropping allows the computer to sample the data as it is being recorded without interfering with the recording process itself. During the eavesdropping or on-line mode, as it is sometimes called, all the tape units are under the control of the CDS. Upon receipt of a signal from the 1620, the CDS causes the first character and associated parity bit to be transmitted to the 1620. Each character and associated parity bit continues to be transmitted until the longitudinal redundancy check character (LRCC) is encountered. The computer cannot initiate tape movement by attempting to read a tape while in this mode; therefore, a read tape instruction hangs up the computer until the CDS moves the tape to record new information. Besides the eavesdropping mode, the computer is also able to operate in an off-line mode. During the off-line mode, a selected tape unit (any of the two) may be read or written by the 1620 as if it were its own. These two modes of operation are manually selected.

Reduced punched card data is generally generated and plotted off-line during tests. An IBM 407 printer is also available in this center and is used to print much of the punched card output.

134

COMPUTER HARDWARE MODIFICATIONS

The 1620 computer performing the data reduction is unique. Three new instructions had to be designed and the computer modified to permit their use for this special application. The design and implementation of these instructions into the computer required several months. In addition, other instructions were adapted to permit the reading, writing and other handling of magnetic tapes.

The three new instructions pertain specifically to the use of magnetic tape.

BST, backspace magnetic tape (36XXXXX01300),
REW, rewind magnetic tape (36XXXXX02300) and
WEF, write end of file (36XXXXX01200)

Two instructions that refer to paper tape normally, RNPT, read numerically paper tape and WNPT, write numerically paper tape, were modified to read magnetic tape (RMT) and write magnetic tape (WMT), in the numerical mode.

RMT, read magnetic tape (36YYYYY00300) and
WMT, write magnetic tape (38YYYYY00200)

In order to allow for tape redundancy and end of file testing, the functions of the following sense switch testing codes were extended.

BC1, branch console switch 1 on (46YYYYY00100) and
BC2, branch console switch 2 on (46YYYYY00200)

When a BC1 instruction is executed, a branch takes place if either sense switch 1 is on or if a tape redundancy occurs. Likewise, the BC2 instruction also serves two purposes: a branch will occur if either sense switch 2 is on or an end of file mark is sensed. These two sense switches must be in their off position during magnetic tape operations. The redundancy and end of file indicators are not reset by any of these two instructions; they are reset only when the selected tape is put into motion again.

SOFTWARE

Although the magnetic tapes normally may be read and written with FORTRAN coded programs by utilization of the paper tape statements, the tapes produced by this system can only be read by SPS or machine type programs. This is due to the various field widths contained within the records written by the system. The problem of reading tapes was quickly resolved by the writing of an SPS subprogram that could be called by and loaded with FORTRAN coded programs.

The SPS subprogram was designed to operate in two modes. The first of these modes, as directed by the arguments of the FORTRAN program, causes a compacted system record to be read from the tape. Each of the fields of the record is then extracted and expanded to a six-character field width. Flags are placed over the leftmost positions of each of the fields and the fields are then transmitted to their prescribed COMMON locations as integers. The second of the two modes requires the subprogram to search through the tape (not used during the on-line operation) for a particular title run record that agrees with the run and test numbers indicated in the arguments of the calling program. When the appropriate record is found, the information from the record is processed in the same manner as it was in the first mode. If the record is not found an indicator is placed in a communication field, reserved in COMMON for this purpose.

The two tape reading modes of the subprogram, described above, are very useful and make the 1620 an even more important asset to the overall system. The first of these two modes provides the user with integer data that is FORTRAN-compatible. The second mode, in addition to performing the same task as the first mode, assists in retrieving previously recorded data.

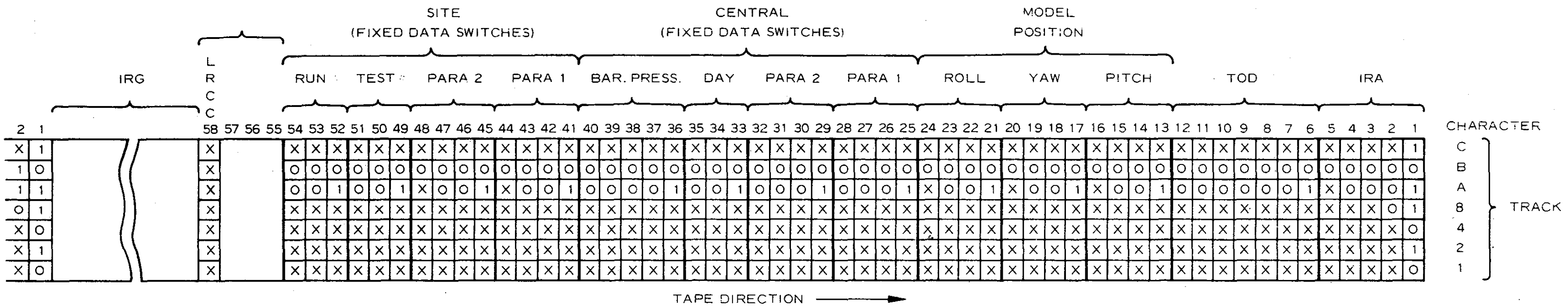
RECORD FORMAT

The formatting generator produces two types of records and these are the title run record and the data record. The purpose of the title run record is to provide identification for the data records that follow it. The title run record consists of 54 characters of the following information:

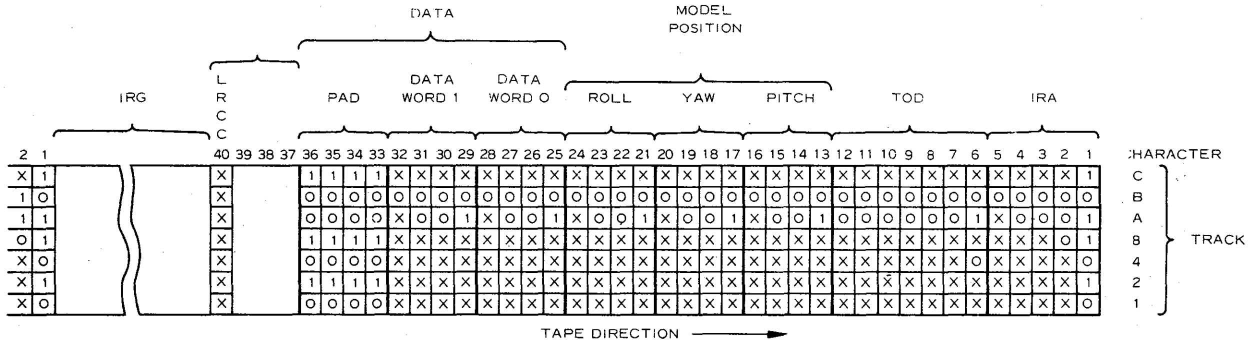
Characters 1 - 5	Information Retrieval Aid (IRA)
Characters 6 - 12	Time of Day (TOD)
Characters 13 - 24	Model Position (Pitch, Yaw and Roll)
Characters 25 - 32	Parameters from the CDS to be used for computations or further identification
Characters 33 - 35	Day (001 thru 366)
Characters 36 - 40	Barometric Pressure
Characters 41 - 48	Parameters from the transmitting site to be used for computations or identification
Characters 49 - 54	Test and Run numbers

The IRA indicates the record type and identifies the test site. In the data record, it also gives the number of channels (data words) that were recorded.

The data records produced by the formatting generator are of a variable length. The length varies with the number of data words that are recorded. Characters 1 - 24 of the data record are of the same format as those of the title run record. Characters 25 and above represent data words. Each data word consists of four characters. As many as 100 data words can be recorded in one record. The record is automatically padded to contain an integral multiple of six characters.



TITLE RUN RECORD



DATA RECORD

X, IN THE ABOVE ILLUSTRATIONS, MEANS THAT EITHER A 1 OR A 0 MAY BE GENERATED.

138

CONCLUDING REMARKS

This paper has described how an automatic data acquisition and processing system was developed to perform a vital function in the modern wind tunnel complex at Northrop Norair. The major benefits of this computerized system can be stated as follows:

1. It reduces wind tunnel data immediately when it is most needed.
2. It permits quicker and more effective adjustments to be made to the model within the test chamber.
3. It shortens the time spent in carrying out a series of tests.

REFERENCES

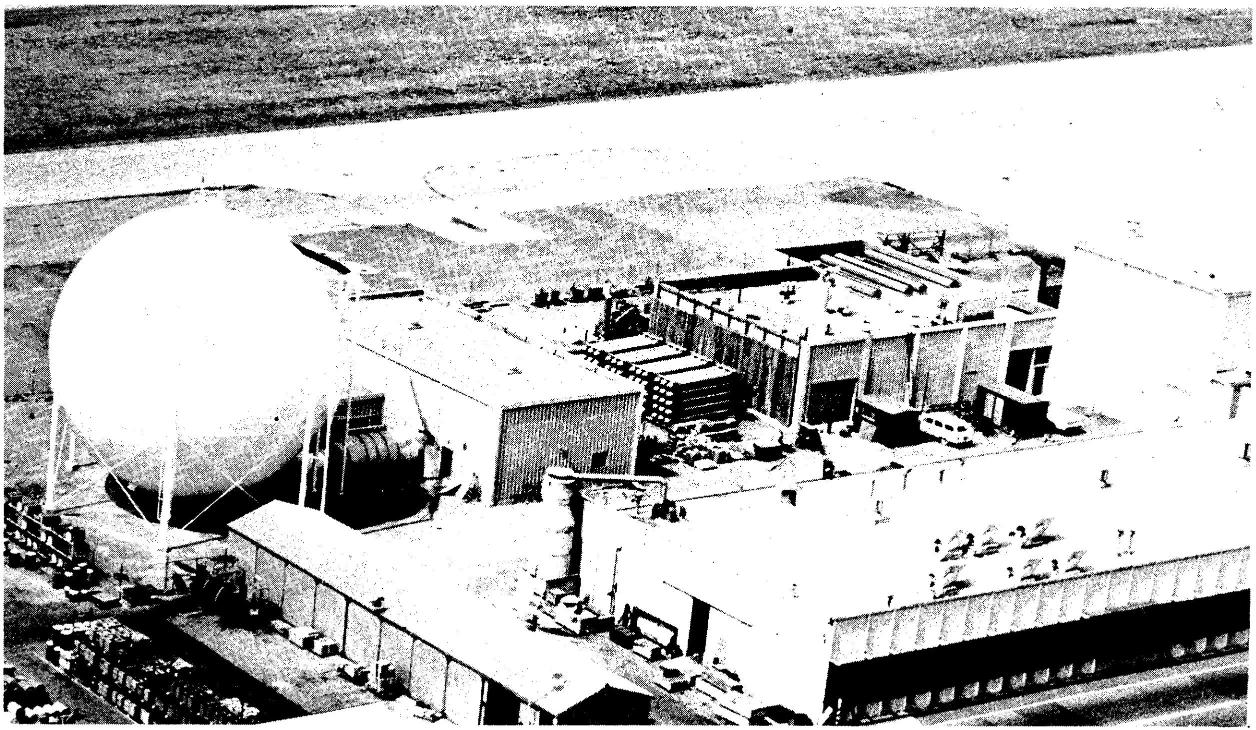
"The Programming Gap in Real Time Systems", R. V. Head, Datamation, February 1963.

"Pitfalls & Safeguards in Real Time Digital Systems", W. A. Hosier, Datamation, May 1962.

"Air in a Hurry", H. M. Karaszewski, Compressed Air, February 1963.

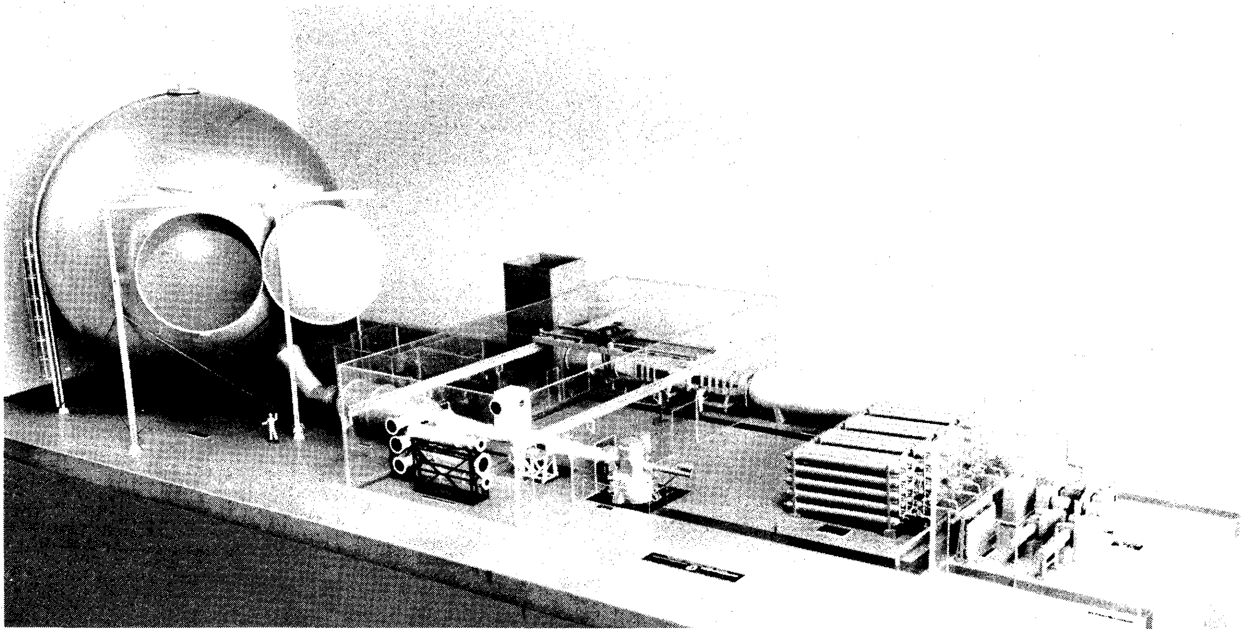
"Flight Simulation Without Forward Velocity", Henry M. Karaszewski, Compressed Air, December 1959.

"Northrop Inaugurates Mach 14 Wind Tunnel", Northrop Technical Digest, November 1962.



1043

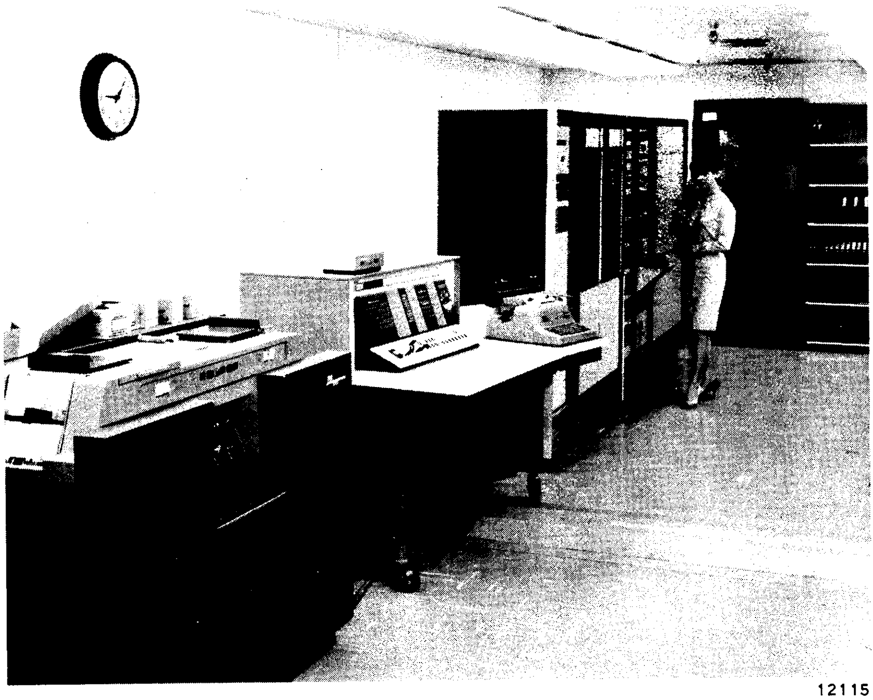
OVERALL VIEW OF WIND TUNNEL COMPLEX, SHOWING SUPERSONIC — HYPERSONIC FACILITY AT LEFT AND SUBSONIC AT EXTREME RIGHT



12113

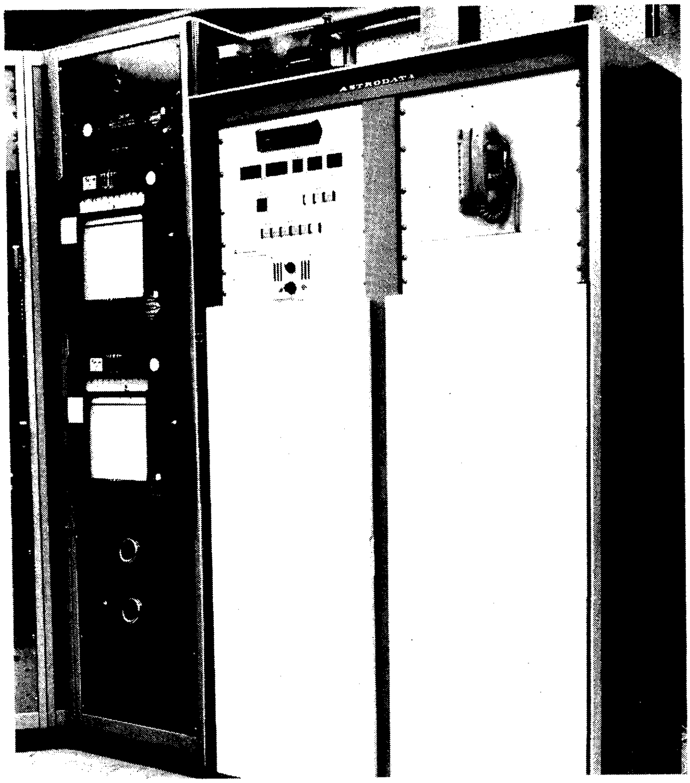
MODEL OF SUPERSONIC — HYPERSONIC CIRCUIT

140



12115

CENTRAL DATA ACQUISITION SYSTEM



SIGNAL CONDITIONING CABINET AT SUPERSONIC SITE

1620 IPL-V

A NON- NUMERIC PROBLEM SOLVING TOOL

by

Wendell Terry Beyer

An essay
submitted to the Department of Mathematics
of the University of Oregon
in partial fulfillment
of the requirements for the degree of
Master of Arts

April 1964

Acknowledgments

The development of the 1620 IPL-V system was done in part while the author was an IBM/WDPC Research Assistant at the University of Oregon. This assistantship was provided by the Western Data Processing Center at the University of California. Many long hours on an IBM 1620 computer were freely provided by the University of Oregon Statistical Laboratory and Computing Center.

Preface

This paper is composed of three section. Section I introduces the need for computer languages similar to IPL-V, section II outlines the IPL-V language, and section III describes the IPL-V implementation for the IBM 1620 computer. A detailed description of the IPL-V instructions and a sample problem are contained in the appendix. A list of selected references is given at the end.

I

Stored program digital computers were initially developed as devices for performing complex arithmetic calculations at high speeds. At first, the task of programming these machines was burdensome because all programming was done in machine language. However, programming languages were soon developed as an aid to the programmer, beginning with low level assembly languages for specific machines and eventually evolving into high level, machine independent languages such as FORTRAN, ALGOL, and COBOL. Due to the arithmetic origins of the computer, these languages were designed to assist the programmer in the coding of arithmetic or numeric problems.

For a long time, however, it had been known that the digital computer, with its ability to analyze data and take differential action, was not inherently limited in scope to numeric problems. Indeed the problem of translating source statements from a high level language like FORTRAN into machine code is itself a problem basically non-numeric in nature. Other problems for which computer solutions were sought include chess, bridge, analytic differentiation and integration, language translation, pattern recognition, study of learning and self-organizing systems, information retrival, theorem proving, and most recently theory developing.

As interest in these and similar problems grew, certain questions arose. Is the present form of digital computer, designed with numeric computations in mind, necessarily the best for non-numeric problems? If not, what better designs might there be? Is it in fact possible to believe that one design will be capable of handling the majority of non-numeric problems? Is it possible to develop a high level language which will do for non-numeric computation what FORTRAN does for numeric computation?

Today these questions remain largely unanswered. No one has succeeded in developing a high level language designed for non-numeric computing although work is being carried on in this area. Some computer designs have been developed which seem to yield a better method of attack on non-numeric problems than that afforded by numeric computers.

To more fully appreciate the problems confronting the designer of a non-numeric computer, it is necessary to examine some of the common characteristics of the non-numeric problems listed above. These problems cover a wide variety of topics and one might suspect that there is little in common among them; however, four characteristics do appear in most of the problems.

First, each problem is non-numeric in part or in whole. The great computational power of modern numeric computers is not needed.

Second, in most of the problems there is a need for a unit of data more complex than a simple number or array of numbers. For example in analytic differentiation, some method of representing algebraic formulae is needed. In language translation or theorem proving some method of representing syntax or theoretical relationships must be provided.

Third, in many of the problems the assignment of specific areas of memory to contain certain types of information is difficult or impossible since the form, structure, and amount of information is not known at the time a program is set in action. For example, in many cases it is not known what form a self-organizing system will take, or what concepts, and hence information, a theory developing program will yield.

Fourth, it is often desirable to have certain portions of a program call on themselves as subroutines. This is called recursion and is useful in differentiation or game playing where a routine may call on itself to look ahead a move.

A successful non-numeric computer, if it is to have general applicability, must be designed to meet these four needs. Similarly, any language aimed at non-numeric work must fill these needs.

IPL-V is an abbreviation for Information Processing Language V, a highly successful and widely used language designed for non-numeric computing. The IPL languages were developed at the RAND Corporation by Newell, Shaw and Simon, beginning in 1954 with IPL-I, a language for playing chess. Of the IPL languages, IPL-V is the only one which has seen widespread use. The language is well-documented and a manual for programmers is available.[1] In the next section IPL-V is outlined and the manner in which it meets the four problems posed above is discussed.

---

[1] See reference [3].

II

The IPL-V language may be regarded as an assembly language for a non-numeric computer, the IPL computer, or as a medium level language which is machine independent and is executed on numeric computers by an interpreter program. It is interesting to note that an IPL computer has never been built, and all work done in IPL-V is accomplished by means of interpreters. Nevertheless it is useful to describe the IPL-V language in terms of the IPL computer.

It is the function of the IPL computer to manipulate symbols, that is, to accept as data, members of a certain set of symbols, to store these symbols in memory, move the symbols from one location to another, compare the symbols, make decisions based on these comparisons. organize the symbols in memory in a meaningful manner and produce as output a sequence of symbols. For this reason IPL-V is often referred to as a symbol-manipulation language.

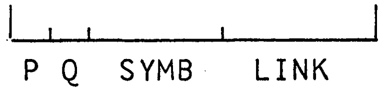
The memory of the IPL computer is divided into cells, and it is the addresses of these cells which form the symbol manipulated by the computer. That is, an IPL symbol is the address of a cell in the IPL memory. The meaning assigned to these symbols is arbitrary. Thus regardless of the contents of cell 14613,

149

the address 14613 may represent New York City in a military problem, Act II of Hamlet in a literature analysis. or the principle of mathematical induction in a theorem proving problem.

Since it is inconvenient for a programmer to deal directly with memory addresses, the IPL-V language allows a more convenient external representation of symbols. The thirty-six characters A B C ... Z $ = . + - * /) (and , are called regional characters. At the beginning of his program a programmer may assign to each regional character a continuous block of cells in memory. The block of cells assigned to say A is called the A region and the individual cells in this region are referred to by the symbols A0 (or simply A) for the first cell, A1 for the second cell, etc. Any symbol naming a cell in one of the thirty-six regions is called a regional symbol. The assembler translates regional symbols into the corresponding addresses. In addition the IPL computer has the ability to transform the address of any regional cell into the correct regional symbol during output operations. The address of any cell not assigned to a region is a non-regional symbol, and may be represented by the programmer in a variety of ways.

Each cell in the IPL memory contains two digits called the P and Q digits and two addresses called the SYMB for symbol and the LINK. A typical cell in memory is represented by the following diagram:



The individual portions of a cell are not addressable.

Cells may be used for one of three purposes: to contain an instruction for the IPL computer, to contain data, or to contain information necessary to the functioning of the IPL computer.

There are a fixed number of cells of the third type and three regions are automatically set aside to contain them. The H, W, and J regions always contain the same cells in memory. The cells of the H region function as registers and indicators in the IPL computer. The W region contains some cells usable by the programmer as temporary storage and other cells used in exercising a certain degree of control over the operation of the computer. Each cell in the J region represents and contains the first instruction of a built in subroutine, of which there are 188 in a complete system.

With the exception of the H, W, and J cells, any cell in memory may be used to contain data or an instruction, and during the course of a program, may contain both.

A cell containing data may be of two types. A data term is a cell containing special alphanumeric or numeric information, while the P and Q digits indicate the type of information.[2] A *standard data cell* is a cell used to store an IPL symbol. The symbol is stored in the SYMB and the P and Q digits indicate the type of symbol.[3] The LINK of a standard data cell also contains a symbol, the use of which will be described below. A data cell containing the symbol "+" might look as follows:

0	0	14613	00000
P	Q	SYMB	LINK

where 14613 is the address of the first cell in the "+" region. Unless attention is to be called to the P and Q digits, this will be represented by

+	0

.

---

[2] All data terms have a Q digit of 1 which serves to distinguish them from standard data cells which have a Q digit of 0, 2, or 4. The P digit of a data term indicates the type of information stored in the data term as follows:

P=0	Decimal integer
P=1	Floating point number
P=2	Alphanumeric
P=3	Octal number

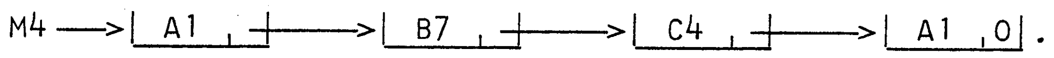
[3] Standard data cells usually have a P digit of 0 although they may be specially marked by a P digit of 1. The Q digit indicates the type of symbol contained in SYMB as follows:

Q=0	SYMB is regional
Q=2	SYMB is local
Q=4	SYMB is internal

The data terms play a rather minor role in the computer, usually serving as storage locations for numeric information; while the role of the standard data cell is central to the operation of the computer.

In dealing with symbols of arbitrary meaning, the IPL computer answers the first need of a non-numeric computer, that of dealing with non-numeric information. These symbols do double duty, serving sometimes as the addresses of cells in memory and at other times representing the concept assigned by the programmer. However, the IPL symbol, being an address, is basically no more complex than a number.

The need for a complex unit of data is fulfilled by the list, a basic unit of data in the IPL computer. A list is a sequence of data cells which are joined together by having the link of each cell contain the address or name of the following cell. A list of the symbols A1, B7, C4, and A1 in that order would be represented by the following diagram,



where the arrows indicate the cell referenced by the LINK of a cell. Note the use of the symbol 0 in the link of the last cell. This symbol is called the <u>termination symbol</u> and indicates that the list terminates at that point. The name of the first cell in the above list is

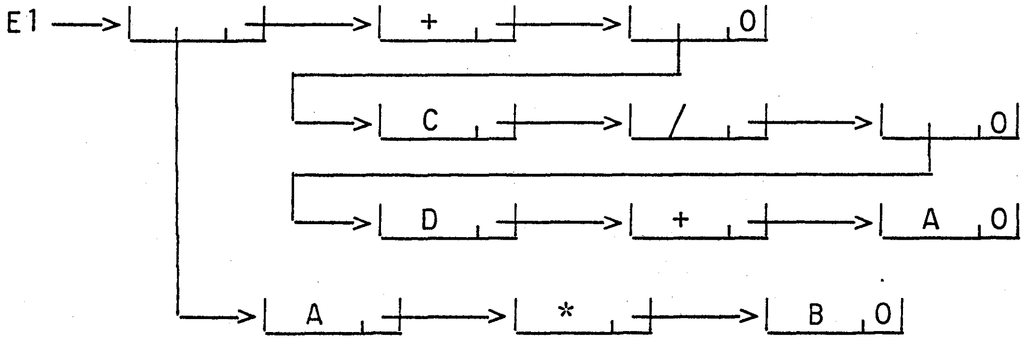
M4 and the list is also referred to by that symbol. Given cell M4, any symbol on the list may be reached by passing from link to link.

Far more complex structures may be created by using the SYMB of some cells on a list to contain the names of other lists. The Q digit of a cell on a list may be used to indicate whether the SYMB contains an abstract symbol or the name of a sublist which is to be considered part of the structure.[4] The number of structures possible is limited only by the programmer's imagination, but for simplicity only lists will be considered below.

Because of the list, IPL-V is called a list-processing language, as are other languages which use the same concept. The language contains subroutines for list manipulations such as copying, printing, searching, or erasing lists. An example is a subroutine which will

---

[4] For example, the algebraic expression A*B+C/(D+A) may be represented by the structure E1 below which expresses the structure of the expression in a manner not possible in a linear list representation.

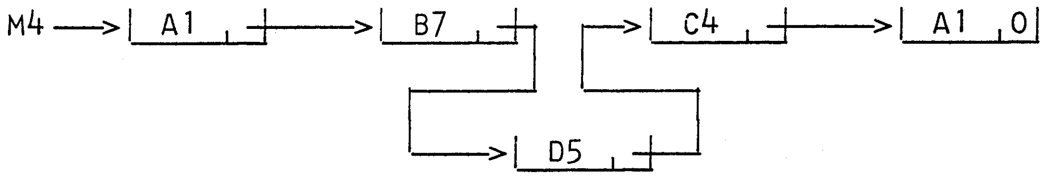


test whether a given symbol occurs on a list or not.

Consider again the list:

The location of the first cell of the list is important. Since the name of the list is M4, the first cell of the list must be cell M4, but the location of the remaining three cells is unimportant to the structure of the list. This fact has important consequences.

When inserting a new symbol on a list, it is not necessary to disturb the original cells of the list. For example, the symbol D5 may be inserted between B7 and C4 on list M4 above by finding any unused cell anywhere in memory, placing the symbol D5 in that cell, and rearranging the links as follows:



In this way a solution is achieved for the third problem of non-numeric computers, memory assignment. A block of memory need not be reserved for expansion of a data structure, since in expanding, a data structure may make use of any unused cells in memory, whether they lie in a continuous block or not. Even the names of new data structures may be kept on lists. Only the total number of cells in memory is of concern to the IPL-V programmer.

It might seem that locating an unused cell in memory would be difficult, but this problem is handled in an elegant and efficient manner. After assembly, all unused cells are linked together to form a list named H2 and called the <u>available space list</u>. During processing when a cell is needed, one is removed from H2 for use; and when a cell is no longer needed by the programmer, it is returned to H2.

The list organization also allows cells to be used as though they were capable of storing more than one symbol. Suppose for the moment we have a symbol stored in cell W0, say A7, and we need to temporarily store a second symbol, say B3, also in W0.

W0 ⟶ | A7 , 0 |

We execute an IPL instruction causing the computer to <u>push down</u> cell W0. That is, an unused cell is removed from H2, inserted behind cell W0, and a copy of the symbol in W0 is placed in the new cell, creating the following list:

W0 ⟶ | A7 , ┼ ⟶ | A7 , 0 |

Now that a copy of A7 has been made, B3 may be placed in WO.

WO ——> | B3 , ┼——> | A7 , 0 |

We may go even further and store C8 in WO before removing B3, by pushing down WO again, then storing C8.

WO ——> | C8 , ┼——> | B3 , ┼——> | A7 , 0 |

The list created in this manner is called a <u>push</u> <u>down</u> list but is no different from any other list.

When the symbol C4 is no longer needed in cell WO, a <u>pop</u> <u>up</u> instruction is executed. This operation copies the second symbol on the list into the first cell and removes the second cell from the list, returning it to H2.

WO ——> | B3 , ┼——> | A7 , 0 |

One more pop up, and WO is returned to its original state.

WO ——> | A7 , 0 |

The preceeding sequence of events may be summarized by writing the push down list vertically.

$$A7 \xrightarrow[\text{down}]{\text{push}} \begin{matrix}A7\\A7\end{matrix} \xrightarrow[\text{B3}]{\text{store}} \begin{matrix}B3\\A7\end{matrix} \xrightarrow[\text{down}]{\text{push}} \begin{matrix}B3\\B3\\A7\end{matrix} \xrightarrow[\text{C8}]{\text{store}} \begin{matrix}C8\\B3\\A7\end{matrix} \xrightarrow[\text{up}]{\text{pop}}$$

$$\begin{matrix}B3\\A7\end{matrix} \xrightarrow[\text{up}]{\text{pop}} A7$$

The push down and pop up instructions enable a subroutine and main routine to use the same storage cells. A set of working cells, W0 through W9, are provided for temporary storage. When a subroutine needs temporary storage, some of these cells are pushed down, then used as storage. Any information stored by the main routine in these cells is preserved by the push down operation. Before terminating, the subroutine pops up these cells, returning them to their original state.

The ability of the IPL computer to allow recursion, the fourth need of a non-numeric computer, is also based on the push down operation. The cell H1, called the current instruction address cell, contains at any given time the address of the instruction currently being executed by the IPL computer. When an instruction is completed, the address of the next instruction is obtained and placed in H1. Like any other cell in the memory, H1 may be pushed down. When one routine calls on another as a subroutine, H1 is pushed down by the computer, saving the address of the instruction in the main routine where processing is suspended. The address of the first instruction in the subroutine is placed in H1 and that instruction is executed. Processing now continues along the subroutine and the computer is said to have descended a level. Before terminating, the subroutine may call on itself or another subroutine.

Again H1 is pushed down, saving the point at which processing was suspended in the subroutine, and processing continues at a lower level. When a subroutine terminates, H1 is popped up and the routine one level up resumes action. A combination of the manner in which H1 is used and the ability of the working cells to keep the contents of routines on different levels from becoming mixed, allows a subroutine to call on itself.

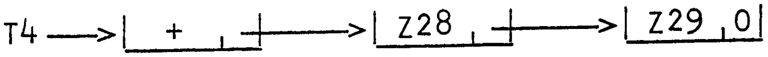
The instructions in the IPL computer are kept in lists. The P, Q, and SYMB of a cell make up the instruction and the LINK indicates the next instruction. The IPL computer follows instructions from cell to cell down a list rather than executing instructions sequentially in memory. This allows routines to be manipulated with the list processing subroutines. It is conceivable that a main routine could construct a subroutine using list processing subroutines, execute that subroutine, then erase it, that is, return all of its cells to the available space list.

In communicating information to a subroutine, a special cell H0, the <u>communication cell</u>, is used. The symbols required as inputs by the subroutine are placed in H0 using the push down operation. The subroutine accepts these inputs, removing them from H0, and before terminating, places all output symbols in H0 where they are recovered by the main routine.

In addition to producing output symbols, some subroutines produce a yes or no answer. For this purpose a cell called the test cell, H5, is provided. The test cell may be in one of two states, "+" or "-", and an instruction is provided to allow conditional branches or transfers within the program on the basis of the state of the test cell.

There are only eight basic instructions in IPL-V, most of the processing being done by the numerous subroutines. Two instructions are used for placing symbols in H0, one instruction for calling on subroutines, two instructions for removing symbols from H0, one instruction each for popping up cells or pushing down cells, and one instruction for conditional branching on the status of the test cell. The P digit determines the type of instruction and SYMB contains a symbol, the name of a cell, or the name of a subroutine, depending on the context. The Q digit is used in connection with SYMB for three levels of addressing. For more complete information concerning instructions and a sample routine, see the appendix.

The external form of IPL-V is quite simple. Lists, instructions or data, are written vertically on the coding sheet. Each line represents a cell and space is provided to indicate the name of the cell and the P, Q, SYMB, and LINK of the cell. If a link is left blank, the cell is assumed to link to the cell on the following line of the coding sheet and the name of the following cell may also be left blank if its memory location is unimportant. Thus to create the list



we write on the coding sheet

NAME	PQ	SYMB	LINK
T4		+	
		Z28	
		Z29	0

III

The University of Oregon IPL-V system for the IBM 1620 computer, developed and written by the author and John D. MacDonald, was designed with two objectives in mind. It was intended first as an educational device to acquaint students with list processing and symbol manipulation problems, and second as a system for checking out IPL-V programs before running them on larger computers. In view of the educational aim, operating speed was sometimes sacrificed for operating ease and additional safeguards. Because of the speed and size of the 1620, the system was never intended as a production tool.

The 1620 system is based on the specifications of IPL-V set forth in the manual[5] and is fully compatible with those specifications, though not all options are available on the 1620 system. Operating on any 1620 equipped with card I/O, indirect addressing, automatic divide, and special instructions, the system provides approximately 640 IPL cells at run time with a 20K memory. An additional 1,660 cells are available with each additional 20K of memory. The system operates at approximately 80 IPL instructions per second and is equipped

---

[5] See reference [3].

with all tracing and monitoring features specified by the manual. These features include operator or program controlled trace with output on any unit, automatic trapping on error conditions, and flexibility in trap recovery.

The system consists of three decks, the assembler, subroutines, and the interpreter, which are loaded in that order with the source deck placed between the assembler and subroutines. The assembler loads into the lower portion of memory and assembles the source deck directly into the upper portion, producing an assembly listing on option. Next the subroutine deck is read by the assembler and those subroutines called for are loaded into memory. After the last card of the subroutine deck has been read, the interpreter loads into the lower portion of memory, occupying the space previously occupied by the assembler; the computer halts; and execution begins when START is pressed.

The internal form of an IPL cell is a twelve digit field with an odd address. From low to high address the cell contains the P, Q, SYMB (five digits), and LINK (five digits).

Provisions are made for writing additional subroutines in SPS and including them in the source deck. It is also possible to reserve blocks of space in the 1620 memory for use by other systems. Methods of setting up linkage between systems are described in the documentation.

The documentation is in the form of an appendix to the manual[6] with cross references. A master copy of the documentation is maintained on cards for easy editing and reproduction.

During the summer and fall of 1963, a preliminary version of the system was written. This version was distributed to approximately twenty participating users for field testing and was used in a one term seminar in IPL-V programming at the University of Oregon. Students in this seminar used the system for problems such as analyzing poetical structure, construction of Farey sequences of numbers, calculation of all closed paths in a planar graph, and construction of a machine for playing Hex. The system has also been used for map coloring and analytic differentiation.

The preliminary version does not contain block handling, auxillary storage, read/write, floating point, save for restart, or post mortem dump routines. During the summer of 1964, a final version will be written, which will include all features except auxillary storage processes. The final version will be submitted to the 1620 Users Group's General Program Library for distribution.

---

[6] See reference [3].

Appendix

The IPL-V Instruction

The Q digit of an instruction operates on the SYMB to produce a transformed symbol S as follows:

Q=0	S is SYMB.
Q=1	S is the symbol contained in the cell whose name is SYMB.
Q=2	S is the symbol in the cell whose name is contained in the cell named SYMB.

For example, if we have the following cells in memory,

A1 ——> | T4 ,0 |
T4 ——> | J8 ,0 |

and the SYMB of the instruction contains A1, the Q digit produces the following transformations:

PQ	SYMB	S
0	A1	A1
1	A1	T4
2	A1	J8

The transformed symbol S is stored in a register; the SYMB portion of the original instruction is never altered in memory.

After the transformed symbol has been obtained the P digit determines the action as follows:

P=0	call on the subroutine whose first instruction is in cell S.
P=1	push down H0 and place a copy of the symbol S in H0.

P=2	copy the symbol in H0 into cell S, then pop up H0.
P=3	pop up cell S.
P=4	push down cell S.
P=5	same as P=1 except H0 is not pushed down first.
P=6	same as P=2 except H0 is not popped up afterward.
P=7	if H5 is -, transfer to cell S for the next instruction. if H5 is +, continue.

Sample Problem

As an example of how the instructions are used, we will write a short subroutine below. It will be necessary to understand the operation of two of the J routines.

J2 accepts two inputs in H0. Each input is a symbol. J2 compares the symbols and sets H5 "+" if they are equal and "-", if not. J2 leaves no symbols as output in H0 and the two input symbols are no longer in H0 after J2 terminates.

J60 accepts one input which is the name of a cell on a list. If that cell is the last cell on the list, J60 sets H5 "-" and leaves the input as output. If the cell is not the last cell on the list, J60 places the name of the following cell in H0 and sets H5 "+".

We now code the routine E4. E4 is a routine which evaluates a function of X at a given point. More clearly, E4 accepts a symbol representing a given point, say A, and a second symbol assumed to be the name of a list representing a function of X. For example, the list F1 below:

```
F1    X
      *
      L
      O
      G
      (
      B
      /
      X
      )     0          "$x\log (\frac{b}{x})$"
```

E4 then evaluates the function at A by replacing every occurrence of the symbol X on the list by the symbol A to yield the list:

```
F1    A
      *
      L
      O
      G
      (
      B
      /
      A
      )     0          "$a\log(\frac{b}{a})$"
```

E4 should leave no output in H0. In addition since E4 does not set H5 as part of its output, the status of H5 should be the same after execution of E4 as before. But E4 must call on J2, which does reset H5. For this reason, it will be necessary to push down H5 at the beginning of E4 to save its status, then to pop it up at the end to restore its status. Two storage cells, W0 and W1 will also be needed. It is assumed that the routine which called on E4 input the name of the function list first, then the symbol representing the point. A little study and liberal use of a black board as a simulator will make the operation of E4 clear. The symbols 9-1, 9-2, and 9-3 are called local symbols and are used for internal branching within the routine.

```
Name    PQ SYMB   LINK      Comments
E4      40 H5               Preserve H5
        40 WO               Preserve WO
        40 W1               Preserve W1
        20 WO               Move "point" to WO
9-1     12 HO               Input symbol in list cell
        10 X                Input X
        00 J2               Compare symbols
        70 9-2              Go to 9-2 if not equal
        60 W1               Copy list cell address in W1
        11 WO               Input point symbol
        21 W1               Move point symbol to list cell
9-2     00 J60              Find next list cell
        70 9-3    9-1       If no list cell, clean up
9-3     30 WO               Restore WO
        30 W1               Restore W1
        30 H5               Restore H5
        30 HO     0         Pop up HO, terminate
```

Selected References

IPL-V, Primary Sources

[1] Dupchak, Robert, *TIPL: Teach Information Processing Language*, The RAND Corporation, RM-3879-PR, October 1963.

[2] Newell, Allen, "Documentation of IPL-V", *Comm. ACM*, Vol. 6, No. 3, March 1963, pp. 86-89.

[3] Newell, Allen, et al., *Information Processing Language-V Manual*, (Second Edition), Prentice-Hall, Inc., Englewood Cliffs, New Jersey, January 1964.

IPL-V, Applications

[4] Newell, Allen, *A Guide to the General Problem-Solver Program GPS-2-2*, The RAND Corporation, RM-3337-PR, February 1963.

[5] Newell, Allen, and H. A. Simon, "GPS, A Program that Simulates Human Thought," *Lernende Automaten*, H. Billings (ed.), Oldenbourg, Munich, 1961.

[6] Newell, Allen, and H. A. Simon, "The Logic Theory Machine: A Complex Information Processing System", *IRE Trans. Info. Theory*, Vol. IT-2, No. 3, September 1956, pp. 61-79.

[7] Simon, H. A., "Experiments with a Heuristic Compiler," *J. ACM*, Vol. 10, No. 4, October 1963, pp. 493-506.

[8] Stefferud, Einar, *The Logic Theory Machine: A Model Heuristic Program*, The RAND Corporation, RM-3731-CC, June 1963.

General Information on Lists

[9] Baecker, H. D., "Mapped List Structures," *Comm. ACM*, Vol. 6, No. 8, August 1963, pp. 435-438.

[10] Banerji, R. B., "The Description List of Concepts," *Comm. ACM*, Vol. 5, No. 8, August 1962, pp. 426-432.

[11] Bowlden, H. J., "A List-Type Storage Technique for Alphanumeric Information," Comm. ACM, Vol. 6, No. 8, August 1963. pp. 433-434.

[12] Weizenbaum, J., "Knotted List Structures," Comm. ACM, Vol. 5, No. 3, March 1962, pp. 161-165.

Other List-Processing Languages

[13] Bobrow, D. G., and Bertram Raphael, A Comparison of List-Processing Computer Languages, Comm. ACM, Vol. 7, No. 4, April 1964, pp. 231-240.

[14] Cooper, D. C., and H. Whitfield, "ALP: An Autocode List-Processing Language," Comp. J., Vol. 5, No. 1, April 1962. pp 28-32.

[15] Gelernter, H., J. R. Hansen, and C. L. Gerberich, "A FORTRAN-Compiled List-Processing Language," J. ACM, Vol. 7, No. 2, April 1960, pp. 87-101.

[16] Green, B. F., Jr. "Computer Languages for Symbol Manipulation," IRE Trans. on Human Factors in Electronics, Vol. HFE-2, No. 1, March 1961, pp. 3-8.

[17] McCarthy, J., et al., LISP 1.5 Programmer's Manual, MIT Computation Center and Research Laboratory of Electronics, Cambridge, Massachusetts, 1962.

[18] Weizenbaum, J., "Symmetric List Processor," Comm. ACM, Vol. 6, No. 9, September 1963. pp. 524-544.

[19] Introduction to COMIT Programming, Research Laboratory of Electronics and MIT Computation Center, MIT Press, Cambridge, Massachusetts, 1961.

PETROLEUM EXPLORATION AND PRODUCTION APPLICATION
FOR THE IBM 1620 AND PLOTTER

By

Jack L. Morrison
Oil Information Center
University of Oklahoma Research Institute
Norman, Oklahoma

Delivered at:

IBM 1620 Users Group
Western Region Meeting
Denver, Colorado
June 17-19, 1964

It was refreshing to hear Dr. Edward N. Brandt, of the University of Oklahoma Medical School Biostatistical Laboratory, say in his keynote address that the problems dealing with computers in the field of medicine are such that they are basically related to and parallel the problems which are encountered in the oil industry. Dr. Brandt also related that the use of computers in medicine has required that the users better define their problems, which gives them a better understanding of the overall situation. The same can be said about the use of computers in the oil industry.

In the next 15-20 minutes, I plan to tell you a little about the Oil Information Center which is an integral part of the University of Oklahoma Research Institute. I will discuss the Oil Information Center:

1. Why and how it was established
2. The goals and objectives
3. How it is connected with the University computer usage generally and the IBM 1620 specifically
4. What we are presently doing, and
5. Where we are going

I. GENESIS OF THE OIL INFORMATION CENTER

Two independent oil men in Oklahoma, Mr. Ward Merrick, Ardmore, and Mr. Howard McCasland, Mack Oil Company, Duncan, were concerned about three apparently unrelated situations and problem areas in Oklahoma. These three problem areas were:

1. No attempt had ever been made to gather groups of oil field related information on a library basis.
2. The Oklahoma Corporation Commission needed an assist in some of their data processing problems and engineering calculations.
3. The computers at the University of Oklahoma were not being utilized as much as could be reasonably expected by local industries, particularly the oil industry.

The concern of these two independent oil operators led them to the concept of the Oil Information Center and as a direct result they furnished the impetus by supplying financial assistance through the medium of their personal foundations. A two-year budget was set up for the initial phase of this Center.

One obvious objective of the Oil Information Center was that sooner or later it must become self-supporting from earned income. It was felt by all concerned that these problem areas just mentioned would be the strong nucleus upon which the objective of self-support would be reached.

After a series of conferences, oil industry executives and University people agreed that the logical central location for libraries of oil information would be on the campus at the University of Oklahoma. The categories of information which seemed desirable to collect were electric logs, scout tickets, drillstem tests, sample logs, and Oklahoma Corporation Commission completion forms. The University of Oklahoma has been famous for years in the quality and quantity of graduates pointed toward the oil industry. The University has probably turned out as many petroleum geologists, petroleum geophysicists and petroleum engineers as any university in the United States.

The Oil and Gas Conservation Department of the Oklahoma Corporation Commission needed assistance with some data processing problems. They wished to work directly with a group who could help them in their work, on whose integrity they could rely and in whom they could have confidence. The Oil Information Center devised a plan to prepare computer programs to assist with some of these problems, and Commission representatives gladly accepted this plan.

II. OPERATIONS OF OIL INFORMATION CENTER

A. Introduction to University Relations

The actual operation of the Oil Information Center is concerned with various areas of effort. A major area is connected with university activities. These are:

1. The graduate program of the University of Oklahoma
2. The Oklahoma Geological Survey, The University of Oklahoma Schools of Geology and Petroleum Engineering

174

3. Conducting seminars on oil related topics

4. Attracting people in the oil industry to the campus

Geology Graduate Student

In checking the records I found no evidence indicating that any geology graduate student had used the computers or plotter to assist them in their master's thesis work. I sought out someone who might be interested in using the computer and found a Humble Oil geologist, on leave from his company to do master's work, and who was willing to work with me. Since the geologist was not a programmer, arrangements were made for his programs to be written for him and through the cooperation of the Computer Lab his key punching was accomplished. This graduate student's thesis was on the geology of an oil field in North Central Texas. His study of the electric logs on each well furnished him with formation tops, well elevations, etc. for his study of 25 different formations. With this information punched into cards he was ready to use the 1620 and plotter to prepare his isopach and subsea calculations and his many maps. The computer program as written was general enough that calculations could be made for isopach thicknesses, subsea formation tops, and sandshale-limestone ratios. This is an example of what can be done in working with graduate students and we hope to encourage others along these lines.

Oklahoma Geological Survey and University of Oklahoma Schools of Geology and Petroleum Engineering

The Oil Information Center has attempted to work closely with the Oklahoma Geological Survey and the University of Oklahoma Schools of Geology and Petroleum Engineering. The libraries of oil field information being gathered by the Oil Information Center are a valuable complement to the Core and Sample Libraries now existing at the University of Oklahoma. The Geological Survey uses the electric logs, sample logs, drillstem tests, etc. in their statewide geologic investigations. The Schools of Geology and Petroleum Engineering can use the same information as teaching aids.

Conducted Symposiums

An important activity in the university phase of our operation is the conduction of symposiums. The Oil Information Center, in conjunction with our libraries of information and computer services, has conducted two symposiums on the campus. One was related to our Drillstem Test Library to which we were able to get good industry speakers from all over the Southwest.

The second symposium was directly connected with the Mid-Continent Well Data System in Oklahoma City. In addition to the speakers at this meeting, the Oil Information Center in cooperation with the University of Oklahoma Computer Lab demonstrated an information retrieval program. I shall discuss this demonstration in more detail in a few minutes. These symposiums have been extremely helpful in our relationship with oil industry people, particularly on the operating level. The sharing of new ideas and approaches is always helpful.

Bring People to the Campus

Directly through the efforts of the Oil Information Center a large number of people have been directed to or through the University of Oklahoma campus. Our seminar on drillstem testing attracted 148 people for two days of meetings. The Mid-Continent Well Data System Symposium was for one day and was attended by 65 people.

Major oil company and consulting geologists from Tulsa, Ardmore, Norman, Ada, and Oklahoma City have been to the Oil Information Center libraries for various reasons. Major oil company representatives have also been to our computer installations using our computer and plotter services. Others have investigated the services which we have to offer in order to determine how this information could be beneficially used by their company.

Industry Effort

To the best of my knowledge, this is the first industry wide effort of information gathering undertaken by the University of Oklahoma. Acceptance of the oil libraries could well lead to the establishment of the gathering of information in other fields of endeavor.

Oil Industry in Oklahoma

With the advent of oil industry data retrieval pilot studies in West Texas, the Oil Information Center found it advisable to conduct their own pilot project on the digitizing of scout tickets and a retrieval program to recover this information. The Autwine field in Kay County, Oklahoma, was chosen for this study for several reasons. The field has more than one producing zone; it produces both oil and water; both major oil companies and independent oil operators have wells in the field. Scout tickets were received on 122 wells which included some surrounding dry holes, and the information was keypunched to our predetermined format.

A computer program was written for our 1410 to retrieve certain information from these cards. The program was written to gather certain usable groups of information:

1. List the wells which cored the Red Fork formation,
2. List the wells and the detailed results of all drillstem tests in the Red Fork formation,
3. List the casing programs in each well,
4. List the formation tops from some wells,
5. List each well that penetrated the Mississippi formation, and
6. List the details of the acid and fracture treatments on each producing Red Fork well.

These are some of the categories of information chosen to be retrieved for this demonstration. This information is typical of that which is used by the exploration geologist and the petroleum engineer in some of their everyday problems.

Oklahoma Corporation Commission

Preparation of Oklahoma Guymon-Hugoton Gas Allowable Schedule:

Due to the large amount of paper work which they process, and their general work load, Gas Conservation Department personnel often were two or three months late in the preparation and distribution of the Guymon-Hugoton Gas Allowable Schedule. By the time the operators of the well and the purchasers of the gas received the schedules they were practically of no value.

The Oil Information Center worked as liaison between Corporation Commission engineers and the Computer Lab programmer so that a computer program could be written to calculate the monthly gas allowable for each well in the field. When Corporation Commission personnel prepared this gas allowable schedule on a desk calculator, they required approximately 70-75 manhours per month. After an estimated five hours of keypunching and keyverifying per month, the IBM 1410 makes these calculations to prepare this gas allowable schedule in 0.4 hours per month.

Calculate one-point back pressure test:

An Oklahoma Corporation Commission statewide rule makes it mandatory for all allocated gas wells to annually report a one-point back pressure test. This information is used in assigning per well gas allowables for the following year.

An estimated 1,800 - 2,000 of these tests are filed with the Commission each year and the Gas Engineer is required to check each of the calculations. The Gas Engineer informed me that with no interruptions he could check five or six of these calculations per hour. This meant that two or two and one-half man-months per year was spent in checking these previously calculated tests. An O.U. Computer Lab programmer wrote a program for our 1410 to make these calculations. The 1410 processes these tests in 4.25 hours, which is a significant dollar saving estimated at 3-1/2 : 1. This Gas Engineer is now freed to do more productive and original work for the Commission, which represents the true saving.

B. Introduction to Commercial Applications

Our other major effort is the industrial commercial activities. We have worked directly with:

1. Major oil companies
2. Independent oil operators
3. Oil-field service companies
4. Petroleum consultants

In mid-1963 IBM released a group of programs from their 1620 library, which are called the Petroleum Package. These programs were written by experienced petroleum engineers, geophysicists, and geologists for a rather wide range of commonly encountered exploration and engineering problems. The engineering programs deal with primary oil recovery, secondary recovery, economic evaluations, casing design, gas production rates, flash calculations, etc. The exploration programs deal mainly with geophysics, but are also related to map contouring, electric log analysis, dipmeter calculations, map preparation, etc.

In the past ten years petroleum oriented companies have become more dollar conscious and overall economics have played an ever increasing part in top management decisions. Computers are being used more and more to funnel detailed geophysical, geological and petroleum engineering information to these top management people for their perusal in making their decisions.

In the recent past it was not feasible to make many groups of calculations in the fields of geophysics, geology and petroleum engineering. These calculations were known applications and approaches to their problems but were too detailed and too time consuming for the engineer or geologist to justify spending the time from his other daily duties. With the advent of computers, it became more realistic to consider making some of these calculations. Also, in the past, the necessary data to make these calculations were not gathered knowing that they would never be used. Such is not the case now, and it should be pointed out that the gathering of these data in many cases makes for a more efficient operation on all levels.

In several application areas the use of digital computers is becoming more valuable as magnetic tape recording devices are used in the field. Some of these instances are:

1. Electric logs (and their companion logs)
2. Dipmeter surveys
3. Geophysical field surveys

Many of the large oil field service companies are installing magnetic tape recording devices in their field trucks. This will lead to a more detailed study of data now being received but not efficiently used.

However, most of the commercial work which we have done in our 1620 Lab is related to geophysical problems. The reason is rather obvious when the users were questioned. In many instances geophysicists were not making certain known approaches to their problems because of the number of manhours required to prepare the data, make the calculations and plot certain information. The use of computers and digital plotters now makes it more practical to better utilize data gathered in the field by geophysical crews.

As some of you know, a reflection seismograph crew costs an oil company between $15,000 to $60,000 per month depending on the overall services rendered and the field equipment involved. As in most any other service operation, reflection seismograph field crews can and do have certain problems.

If the field data are being processed on computers as work progresses, the errors can quite easily be rectified. However, if there is a large time lag between the error and its discovery, it may not be so easy to make the necessary adjustments.

A geophysical group of a major oil company in Oklahoma City has been our largest user of commercial time on our 1620 and plotter. This District office is responsible for the geophysical work in all of Oklahoma, all of Kansas, the Texas Panhandle, North Central Texas and the northern 2/3 of Arkansas. In addition to the reflection seismograph field crews gathering new data, they are continually reviewing old seismic records previously shot by themselves or by other companies.

One geophysicist pointed out the following,relative to the information gathered from 300 shot-points. The time required to hand calculate and hand plot this data from 300 shot-points would be an estimated two man-months. To use computers, this same amount of work would require an experienced geophysicist one week, another week to key-punch, one to one and one-half hours on the 1620 for calculation, and five and one-half to six hours on the 1620 and on-line plotter. This represents a vast saving of time as well as money.

One geophysicist pointed out that the use of our 1620 computer on their reflection seismograph field data makes it possible for them to better utilize the information which can be gathered from seismic records. He said that they can now prepare ten to twelve useful sub-surface maps where previously they were fortunate if they were able to get five to six maps from a set of seismic records.

Dan Merriam of the Kansas Geological Survey and John Harbaugh of Stanford University through their joint effort developed a computer program to assist in the location of mineral deposits. (1) Based on certain known geological and/or geophysical information and certain mathematical computations trend surfaces are fitted so that the sum of the squared deviations is the least possible value. The trend surface analysis may be used to:

1. Predict projected depths to geological units within an area,
2. Delineate unconformities or changes in structural patterns, and
3. Extend better "geologic guesses" into adjacent unknown areas of no control.

Close agreement exists between local structural features and trend-surface residuals. The residual maps were found to stress or emphasize trend relationships not otherwise clearly observed from original data and to emphasize the local component of the structural pattern by essentially removing the regional component or regional dip. Inasmuch as in many regions the oil and gas producing areas are systematically associated with structural features, there is the possibility that a study of the residuals will indicate previously overlooked areas favorable for additional oil exploration.

184

The Oil Information Center plans to take advantage of the existence of this program but we plan to rewrite the program to use the IBM 1620 and plotter rather than using the printer to prepare the map.

John P. Dowds, a successful petroleum consultant in Oklahoma City, has worked on the laws of probabilities and the application of statistical methods to help analyze the problem of obtaining commercial oil or gas production. Dowds, in a recent paper, stressed that "exploration geologists and geophysicists need to become statistically minded and to think of locating oil and gas fields as a problem in applied possibilities." (2)

Dowds uses entropy for his mathematical model to learn of favorable trends and patterns in searching for logical locations for drilling new oil or gas exploration wells.

Dowds determined a long time ago that his calculations were too difficult and the number of these calculations required were too many to be done by hand. An Oil Information Center programmer recently wrote programs to Dowds' formulae for his entropy calculations. These are now being run on our 1620 and plotter. The final output to be studied for purposes of exploration is a series of contour maps. Dowds is representing a large independent oil operation in Oklahoma City in their search for sizable oil or gas reserves.

James M. Forgotson, Jr., research geologist with Pan American Petroleum Corporation in Tulsa, said in a recent Oil and Gas Journal article that the use of electronic computers to evaluate electric logs is very practical. He said, "The speed with which these computations can be performed makes the analysis of many zones or formations in thousands of wells practical." Forgotson went on to point out that "without the aid of the computer, approximately eight manhours are required to calculate shaliness, saturation ratio, and favorability criterion for four zones in one well." He also made an interesting comparison stating that "with the use of computers approximately one and one-half man-months would be required to process four zones in 1,000 wells while without the use of computers fifty-four man-months would be consumed." (3)

III. SUMMARY

The Oil Information Center is serving a useful purpose to the University of Oklahoma, to the Oklahoma Corporation Commission, and to the oil industry in general in Oklahoma.

With the 1410, 1620 and the plotter now in the University of Oklahoma Computer Lab, we are able to offer computer services to:

1. Major oil companies
2. Independent oil operators
3. Consultant geologists and petroleum engineers
4. Oil field service companies

Using the 1620 Petroleum Package of programs has proven successful up to a point even though the large majority of commercial time which we are able to sell has been to companies who have written their own programs.

REFERENCES

1. Merriam, D. and Harbaugh, J. "Balgol Program for Trend-Surface Mapping", distributed by the Kansas Geological Survey (Special Distribution Publication #3).

2. Dowds, John P. "Application of Information Theory in Establishing Oil Field Trends", presented in June 1963 at Stanford University during the 3rd Annual Conference on Computers in the Mineral Industries.

3. Forgotson, J.M., Jr. "How Computers Help Find Oil", Oil and Gas Journal, March 18, 1963.

A CONTROL SYSTEM APPROACH
TO
AUTOMATIC JET ENGINE TESTING

1620 User's Group
Western Region
June 17, 18, 19 - 1964

Aubrey D. Wood
IBM Systems Engineer
Oklahoma City, Oklahoma

TABLE OF CONTENTS

A. History of Jet Engine Testing:

After the first jet propelled airplane was captured from Germany by the United States in World War II, development of the jet type aircraft has proceeded in rapid fire fashion.

The first truly great use of the jet airplane came about as a result of the Korean War. In a few short years since the early 1950's, the development and production of the jet engine has proceeded at an amazing rate.

With the production of the first jet also came problems in the maintenance and overhaul of these complex, high thrust engines. At the beginning, especially during the Korean War, maintenance and repair was carried out in the remote airstrip locations and centralized repair facilities using the out-moded piston engine repair and test facilities. The piston engines had not required the highly substantial and instrumented test facilities that the newer high thrust jet engines were requiring; so, many of the first tests were performed in a crude makeshift manner.

In the initial stages, many of the repair personnel became engine test personnel. Because the jet engine development had proceeded in a hurried fashion, adequate testing procedures were lacking; so many of the first test cell personnel found themselves preparing their own through pooling, interchanging and accumulating their experiences. Many of the basic principles of these early testing technical procedures are still in use today. Also, the great majority of today's test cells are modified piston engine of low thrust jet engine test cells and their instrumentation leaves a lot to be desired. Much of the instrumentation was installed on a "guess and try" basis.

Since the early 1950's the production rate and number of jet engines in the air has risen considerably. With these increases also came increases in the number of engines to be overhauled and repaired. The test facilities in many instances have been updated with new instruments. The engine manufacturers have also been allowed time to adequately prepare better testing procedures. Even with all of these improvements there still remains two pressing problems. They are: (1) the large number of engines awaiting the testing facility and (2) the advent of the higher thrust (turbo fan, J75, etc.) jet engine has again outdated the test facilities.

B. Present Test Problems:

Because of the rapid expansion of the test facilities to accomodate the increased workload of jet engines and the complexities of the higher thrust engines, many problems arose in acquiring an adequate balance between the production and quality control functions.

These problems are presented in the following sections. They are grouped into areas in order to present a detailed view of each. It should, however, be noted that the problems actually overlap into other areas and even overlap each other. Many times a particular problem arises because of testing techniques, instrumentation, and the facilities being used.

1. Present Testing Methods

In order to fully understand the problems associated with the present testing techniques, the following is submitted as a general discussion of the overall testing procedure.

The typical jet engine test cell has two or three men assigned to it. During the initial installation phases two men perform all necessary physical connections. This will include steps (a) and (b) of the test procedure. During the running/testing of the engine, one of these men will control the throttle and instrumentation necessary to run the engine and make recordings while the other man makes the balance of the necessary recordings at the appropriate times and places. A third man acts as an inspector. His job is to observe the readings being made and perform a reasonableness check on certain limits to see if recording errors have been made. He also takes observed readings and corrects them to a standard day (sea level or other) condition for comparison with the technical order specifications.

On the final analysis he either accepts or rejects the engine based upon its performance within the limits and specifications of the manufacturer's technical order. As the engine is routed to test from final assembly it is complete as required by technical order to the final piece of safety wire.

a. Dressdown

Upon receiving the engine at the test area, numerous steps are necessary in order to prepare the engine for testing. The first step is checking for possible external damage which might have occurred during transportation. The engine normally is assigned to a particular test cell prior to dressing for test. Special plugs, fittings, and some harness have to be removed in order to install test equipment. Special test harness is installed in order to obtain individual thermocouple readings for temperature spread checks. Various pressure taps are installed throughout the engine in order to obtain internal air, oil and fuel pressures. Engines are so designed that internal pressures must meet certain limits. If engine internal ratios are below values outlined by engine manufacturer, it becomes necessary to change some specific internal clearance in order to obtain required ratios.

The next step is to install a workhorse tailcone or afterburner. Altogether, there are approximately ten test fittings and adapters that must be installed in addition to temperature harness. Engine mounting adapters and bellmouth adapter rings are installed. Finally the engine oil tank is filled to capacity. This about completes the initial dressing. If the engine is designed for an after-burner, then an AB is attached. There are additional functions of preparations to be performed after the engine enters the test cell.

b. Preliminary Check

After installation of the engine in the test stand, it is necessary to perform some inspections at particular times. This will include such inspections as freedom of compressor rotation and making sure no foreign objects are present in the compressor inlet. It is necessary to accomplish this type inspection prior to installing the bellmouth and inlet screen. If an inspection is performed after installation of the bellmouth, it is quite easy to overlook some small item which might result in compressor damage.

c. Preliminary Shakedown

After engine is properly secured in the test stand with all pressure and temperature connections, attached, a complete shakedown is accomplished by a quality inspector. This shakedown is necessary to pick up anything which may have been overlooked during the engine installation.

d. Functional Component Check

The next step is a functional component check out. This consists of selecting the main fuel control emergency system, afterburner system, and anti-icing valves for functional operation. These checks are necessary prior to starting the engine in order to replace such items that may be faulty.

e. Dry Run

Prior to starting the engine a dry run is performed in order to flush preservative oil from the fuel components, pressure fuel and oil system. Leaks are sometimes found during this check. Afterwards, the dry run oil system is replenished and the pressurizing valve sense line reconnected.

f. Running Prior to Acceptance

The engine is then ready for a start. After the engine has started and reached idle R.P.M. a complete shakedown is made to check for air, oil and fuel leaks. If no abnormal conditions are found, power is advanced toward top power and preliminary checks are made on oil pressure, E.G.T. and vibration.

g. Performance Runs

After the preliminary run has been completed, the engine is ready for a performance test run. This test run consists of numerous functions in order to test the basic engine and its attached components as a complete assembly.

Other checks that follow during the actual performance test are acceleration checks, simulated afterburner runs, emergency system runs, oil consumption check and performance calculations.

The test run begins with an initial power advance after start to approximately nine thousand RPM. This is necessary in order to obtain specific data for test run calculations and warm up the engine oil. The engine oil must be heated to actual operating temperature in order to obtain valid consumption during test run. Oil temperature must be noted at the time oil level is checked on a sight gauge and again at completion of the test run. Oil temperature at the time of the final check must be within ± 2 degrees F. of the initial temperature. Oil consumption is actually determined by visually observing a sight gauge. This sight gauge is calibrated to the engine oil tank and actually seeks the oil level within the engine tank. The oil level sight gauge is marked with ten increments to the inch and each increment represents a specific amount of oil.

The data collected during the initial warm up period is used to determine the exact power position required for various test runs. Four power runs ranging between seventy-five per cent and take off are required in order to help determine the quality of the engine.

Test run power positions are determined by charts representing given thrust positions. All data from such charts represent standard day conditions biased for temperature variation. Actual thrust requirements are subtracted from points corresponding to various power positions by using compressor inlet temperatures. Once having obtained required corrected thrust output, this data must be converted to actual time conditions. This correction is a function of present time condition variations from a standard day and test cell correction factors.

Each individual run has a time duration of five to twenty minutes depending upon the position of power. Recordings of internal pressures from compressor inlet to turbine discharge are made. Temperatures of air inlet, oil, fuel and turbine discharge air are logged. Other recordings such as fuel flow, thrust, turbine discharge pressure, RPM and vibration are necessary.

All data logged directly related with the functional operation of the engine must be corrected to a standard day condition. This data is also corrected for compressor inlet temperature, barometric pressure and test cell correction factors. There are approximately 175 calculations performed during the test run. Thirty-five points are plotted on special graphs in order to determine if any maximum limit has been exceeded. Also plot points are necessary in order to determine minimum RPM required to obtain guaranteed rated thrust. Other correction factors which are necessary pertain to the emergency fuel flow and cooling air ratio. Other checks of the emergency system consist of acceleration procedures and engine starts. Such steps are necessary in order to determine if the emergency system has the ability to operate properly and take over engine operation in the event the main system fail.

Cooling air ratio is a necessary factor in order to determine if a sufficient amount of air is being furnished to the hot section parts. If the air ratio is below values outlined by the engine manufacturer, damage could occur to some parts.

h. Simulated AB Runs

After completing the necessary performance checks, the engine afterburner system is simulated. The complete afterburner system is subjected to all functions of operation without actual firing. The method used is simply to rout afterburner regulator fuel back to the pump inlet. The ignitor valve will fire, nozzle control will function and afterburner regulator will meter fuel. This system actually is quite practical insofar as all fuel is returned to the inlet supply.

2. Testing Techniques

Jet Engine Testing has many problems associated with the techniques encountered using the present manual methods. Some of these problems can be directly associated with human capabilities and reactions during the test cycle. These represent man's inability to cope with the complex situations and the split-second decisions at a speed and with the accuracy required for maintaining a high quality test procedure.

Other problems can be attributed to inaccuracies in the existing mechanical and electrical means of transmitting test data to the test cell personnel from its primary source on the engine. These problems are created because a primary signal in the form of an electrical pulse, voltage or current, pressures, and temperatures must be converted to a mechanical means of display for use by the test operator.

a. Standard Tests

A standard test is defined as one in which the test procedure for each type, model, and series engine is conducted in the same manner each time it is conducted, e.g. all data are gathered the same, analyzed the same, and all decisions are made under the same rules without variance. This does not mean that the magnitude of each number in the recorded data will be the same each time, but the manner and intervals at which the recordings are made remain constant.

If an engine is tested and found to be acceptable under one set of ambient conditions, it should also be acceptable when tested under another set of changed ambient conditions. The procedure for testing after overhaul contains the necessary charts and calculations to correct all recordings to a standard day condition; thus, all data should be acceptable under the standard test limits each time it is taken, if it is acceptable at any one of the times.

Even though the testing instructions gives a description of the major procedures to be followed in testing a jet engine, it would become an insurmountable task to specify to the test cell personnel all the exact steps to be taken during the test.

Located at the test facility are many different operators and inspectors, (quality personnel). Because each man is capable of thinking and making individual decisions, he will conduct a jet engine test in a different manner. Because the technical order allows the variations in the manner in which a major test step may be conducted, each operator will not perform each step the same. This situation as well as inconsistencies in the test cell instrumentation will create many different techniques in testing and a possible multiple variations on the acceptance or rejection of an engine under varying conditions.

Not all of the problems associated with the Jet Engine Test can be completely removed by achieving the standard test alone. However, in the process of achieving this standard test many of the "ills" of the present method of testing would have to be eliminated.

The achievement of a standard test can only be realized after correction of the problems in the forthcoming sections.

b. Correlation of Test Cells

In most test facilities there are two or more test cells. In order to obtain a standard set of test data on a engine test in one or more of these cells, it is necessary to inter-correlate the cells.

Either a "gold plated" or standard engine that has been tested in the manufacturer's cells is tested in the production cells. This process is commonly known as calibrating a cell. It involves running the standard engine in the production cell, comparing the data gathered with the instrument recordings made in the manufacturer's cells. This will produce a correlation or correction factor to be used with each cell.

The correlation of one cell may require from five to eight hours to complete-longer if trouble is encountered. Trouble is common. Difficulties arise from changing cell ambient conditions (air temperature, humidity, etc.) inaccuracies of data from readout mechanisms, changing of test cell personnel, etc.

The accuracy of the data acquired in a final test phase will be directly dependent upon the degree of accuracy obtained in calculations of each cells correlation factor. Not only are the inaccuracies involved a problem, but there are extra man-hours, fuel costs, and engine wear characteristics encurred.

Rather than correlating every ninety days as is now required, $\bar{X}$ (average) and R (range or deviation) charts of all instrument reading deviations from those readings of the production correlator engine would produce cell correction factors. This would allow a constant updating of the cell correction or correlating factor as well as indicating trending abnormalities that may be developing. Using the present techniques of testing jet engines; it is impossible to gather sufficient data, calculate the $\bar{X}$ and R's of the recordings, and do the correlating.

The data recordings must be gathered and analyzed over a sufficient period of time to detect trending conditions. This usually involves such things as EPR's, EGT's, N_1 and N_2 speeds and their average and range deviations from the standard engine recordings.

Even if it were possible to gather the data, the magnitude of the calculation and analysis is enormous and would require many manhours.

c. Penalty Runs

It may become desirable after either a major test or a test segment completion, to conduct a penalty run. The penalty run would involve running a small segment of the test, several test segments, or the complete test.

After the completed test and the performance calculations have been made, the engine results could indicate an off specification; thus, requiring the need for a recheck of the calculations and test recordings.

Many times when a borderline situation exists, the inspector will call for the same recheck. Because of the inconsistency in testing methods, calculations, and decisions, the inspector may feel it necessary to repeat a portion of the test in order to gather additional data for analysis, or verification of calculations and recordings. Even when a penalty run is made, conditions may exist (the need for simultaneous readings) that cause the data accuracy to be insufficient, e.g., it is impossible to obtain simultaneous recordings under the manual methods. Because the operator and inspector know of the inconsistencies that exist, several extra minutes or hours along with many extra gallons of fuel may be consumed in conducting the penalty run in order to obtain suffieiently accurate data for a correct test.

3. Instrumentation

There are many and varied problems in the instrumentation areas. The sensing elements on most instruments are reliable and accurate. However, the actual readout mechanism is very difficult to keep within the calibration limits. Because most readout mechanisms present problems of nonlinearity in changing from one setting to another, time and manpower must be spent on a periodic (usually monthly or bimonthly) basis to insure accurate calibration. Many times an instrument can become erradic in its reading and the test cell personnel not become aware of it until a new calibration is made. In the mean time many "good" engines have been rejected and many possible "rejects" are flying or in storage.

4. Human Error

Throughout the test procedure recordings are being made on a second timing intervals. Many of these readings should be made simultaneously, but because of the human inability to observe and record on a "split" second basis many of the readings will change by large increments before they can all be recorded. This is especially true during acceleration and deacceleration of the engine.

Because all the instruments are not located at a 90° angle with the eyes of the man making the recording and because many of his recordings are made at a fast rate, it has been found that many recordings have been made with large errors (sometimes a completely gross transition error is made). A 5 lb. pressure error or 5% temperature error is enough in some readings to reject a "good" engine or accept a "bad" engine.

5. Rerun Statistics

If after the sequence of test events, calculations and plotting of data the engine does not perform according to the technical specifications it is not always rejected and sent to the rework area immediately. After a series of checks on the calculations made by himself, gross range errors on readings, or minor detectable instrument error, the inspector will apply his knowledge in conjunction with the trouble shooting points listed in the TO to diagnosing the area of trouble in the engine. These diagnostics will then be sent back with the engine to the rework area (overhaul line).

If he feels that some element of doubt is present in a reading or calculation, portions of the test or the complete test may be performed again. This re-running may consist of re-trimming the engine, re-running the performance runs, or giving an AB function check. Many times the ability to diagnose the problem area relys solely upon the experience and background of the inspector in charge of the test. The majority of the inspection personnel have not gained this type of experience. Because of this inexperience, many of the engines may be re-run or rejected needlessly. If the engine must be re-run several times in order to find the source of trouble, large quantities of time and fuel are consumed.

If we consider the price of a complete overhaul of an engine ranging from $12,000 to $15,000, the needless reject of a good engine or the improper diagnostic of an engine for overhaul becomes an expensive waste.

Because of the advance in design of the jet type engine year by year, it becomes a large task to keep the test personnel updated on the new techniques accompanying the advance design engines. During the period of time when the modernization of the cell and training of personnel are being done, many costly errors are made.

If we consider a facility that tests 2,000 engines per year, the annual fuel bill will be approximately $550,000 per year. It has been estimated that 40 per cent of this fuel bill can be attributed to running reworked engines or performing a portion of a test over again (because of improper readings or calcuation errors).

The preceding sections describe in general the testing procedures and some of its existing problems. Do not be "misled" by the seemingly simple test procedures described. There are many things not covered in as minute a detail as possible; also not mentioned are the many splitsecond decisions that must be made during the course of the test and at times when possible malfunctions occur.

6. Capacity

In cases of national emergency, or increased workload responsibility, the need for increased test capacity in the high thrust cells could develop into a major production "bottleneck."

Pressure could be relieved in these situations by creating extra shifts of men to handle testing and facilities maintenance; however, the increased utilization of the cells under the present test time and procedures would increase many fold the manhours of maintenance as well as cause the quality of the engines released to the field to be inferior because of this increased pressure.

In either case, the cost of an increased workload under these conditions can become enormous.

7. Safety

There are several events that could take place to endanger the lives of personnel working in the test cell while an engine is running. No "concrete" solution will be found to completely remove all these danger areas. The technical order regulations specify where and at what time personnel may be in the cells while the engine is running. Because of unusual circumstances, the rules are many times "bent" to fit the situation. In many of these cases, danger may be at its peak.

Examination of the possible dangers of these situations reveals that there is a possibility of fuel leaks and thus flash fire while trimming. The bleed valve may also dump excess air overboard while decelerating. The force of this air can be enough to knock a man off his feet. There are also dangers from any engine part or accessories not being securely fastened and thus breaking away.

II. SOLUTION TO THE PROBLEM

A. Introduction

Some of the problems existing in testing a turbo jet engine have been discussed in the first section of this paper. Not all of the intangible problems were brought out, but inference was made to them.

The forthcoming discussion is submitted as a possible approach to the solution of many of the problems that act as a plague to the efficient and correct testing of a jet engine.

There are many alternatives to the degree of automation that can be applied by the use of a computer in a jet test cell. The primary problem rests on two factors: (1) What degree of control should the system have and (2) Whether the system should be a primary "slave" to the operator or the operator a "slave" to the computer.

The one chosen discusses a completely closed loop operation (In this instance, the running of the test including start-up shut-down via an IBM 1710 Control System, related hardware and any special features). The advantages and disavantages of operating in a manual and open-loop mode as compared to the chosen approach are discussed.

A great majority of the following information has been derived by working with prospective customers in the jet engine test area; however, due to reasons which will not be discussed, customers' names will not be mentioned.*

B. Previous Work

1. Data Logging

One of the first attempts at applying an on-line device for the logging and reduction of engine test data was tried by the U.S. Naval Airforce. A special device for these purposes was built by Gilmore Industries (3) to perform such a function.

The primary design of this device was for gathering piston type engine data. Many of these were later modified to receive data from test cells geared for jet engines.

The data logger was usually located in a prototype cell where certain special test runs could be made.

The data logger was primarily an analog type sensing device. Its primary readouts were instrument faces, graphical x-y plotting and type writer data that had

---

* Contact author for further information.

199

been converted by an analog-to-digital converter to a scaled digital form. The number of channels or sensing and readout elements depended upon the elaborateness of the model ordered. The acceptance of this system was "poor" especially in commercial installations (where a few are found war surplus) where the price/performance ratio was much too great.

This piece of equipment contained the same hinderances as the analog computer does. No logical ability coupled with an "exponential" increase in price for flexibility, plus inadequate readout accuracy. Enough of this type of gear to log data in one cell often times cost as much as the digital computer components to control multi-cells.

2. Research and Development Jet Engine Testing

One of the first companies to apply a computer to the role of gathering and reducing test data was Pratt and Whitney. The computer is an IBM 1410 with a special interface (Analog-to-Digital Converter) to take data gathered during tests conducted for research purposes. The system acts as a data monitor. It logs and reduces data only during the time the engine is in the performance run phases. Special instrumentation has been added to detect malfunction of components at high temperatures and fast speeds. After one test has been conducted, the instrument leads are then automatically connected to an engine awaiting test in another cell.

Because the purpose of this system was to do only a data logging and data reduction job, no further effort has been made to perform a close-loop function.

Cases of research and development do not readily adapt themselves to a close-loop operation. There are many times when extraordinary or special tests need to be conducted which would not be compatible with the programs that had been written for test.

There are also under development in the NASA Space Program the adaptation of fast general purpose computer to missile checkout. This program like all other programs in jet engine control is in its infancy.

3. Industrial Testing Systems - Discrete Process

Industry has entered an era in which the processing of production and product performance information must be incorporated as a part of the manufacturing operation. As the profit squeeze continues along with the need for increased production, the cost of manufacturing the product must be reduced to maintain or improve the profit position. Much has been done and is being done to reduce the cost of making the product through advances in technology and by automation. However, the costly operation of quality assurance which continues to receive more demanding tasks is not keeping abreast of its production counterpart. To parallel the giant step made in manufacturing through automation, the quality assurance program in industry has made and continues to make drastic advances through in-process test and inspection systems. Some of the industrial testing applications using industrial process control systems are:

1. Space Vehicles - Analog to digital converter used in logging, reducing and analyzing data on space vehicles in environmental chambers.

2. Potentiometers - Final testing of potentiometers.

3. Automobiles - On-line quality control to determine defects in assembly as they happen.

4. Aerospace Nose Cones - FM Tape playback of data telemetered from missiles.

5. Nuclear Research - On-line recording of information from a spark chamber.

6. Atomic Powered Naval Ships - On-line measurements and computation of shielding experiments.

In general Industrial Testing with control systems controls plant test procedure, analyzes product test data and contributes to production test equipment the capabilities of:

1. Testing dynamically at production speeds.
2. Correlating the test data for each product.
3. Determining the classification of each produced unit based upon specification.
4. Sorting product unit after final test.
5. Storing test data for future analysis.
6. Initiating reports during production runs.
7. Checking and calibrating of test equipment during production runs.
8. Scheduling produced product.
9. Determining critical trends as they develop.[1]

The preceding paragraphs have shown the development of automatic control systems in the continuous process industries and manufacturing operations involving discrete processes. In each case one of the main objectives is increasing the quality of the end product. It can also be clearly seen that automatic testing is not an idea with unproven results but the missing link between production and quality.

---

[1] IBM Application Brief, No. K20-1725

C. Automatic Jet Engine Test Control System

1. System Design Requirements

In the preceding sections of this report the various phases of the actual jet engine test were discussed in moderate detail. These are functions performed by the operator, recorder, and inspector.

The following describes the functions the control system will perform in regard to the various test phases. The functions are necessary to deliver a high quality engine with minimum cost.

a. Control of the Independent Variables to Set Up and Sequence Tests

The jet engine control system will select various test phases for an individual type, model and serial number engine-use information gathered from the engine in the test cell, such as pressures, temperatures, flows, etc., and determine appropriate test sequences and procedures taken from the Technical Order to send control signals to the engine in the cell.

By designing the test phases as a series of logical steps, the system will use each test phase as a sub-program and execute the over-all series of sub-programs under control of a master monitor routine.

b. Data Acquisition and Control

Each instrument pick-up will be connected to a transducer which will be connected to a transducer which will be connected to a multiplexer and terminal unit which will be connected to an analog-to-digital converter. The analog-to-digital converter will provide a digital voltage to the control system main frame. The main frame will scan all instrument leads for each pressure, temperature, flow, etc and convert these by the use of equations into meaningful engineering values. These values will then be used to control the system.

The system will also convert a digital value to an analog voltage for control of the throttle, trimmer and other relay switches in order to control the speed, thrust, fuel flow, and other controllable variables.

c. Calculation of Performance Parameters

After gathering all data (instrument readings), one of the test phases will correct all data to a standard day (usually sea level) condition in order that all parameters may be compared against the T. O. limits for trimming and reject status.

d. Operator Guide for Engine Adjustment

Such things as warning messages, trim guides, test status, etc, will be logged for the operator. Any transducer reading will be available upon operator demand.

Any time the engine must be stopped or shut down by the control system, a message will be logged on the typewriter giving the reason, a complete diagnostic, and recommendations for repair or rework.

e. Automatic Instrument Calibration

This can be done by either or both of the following:

1. Comparison of a known standard signal with the transducer output from this signal.

2. Comparison of the transducer output to other related signals. This will, in essence, tell if the signal is abnormal (too high, too low, or fluctuating). From this an automatic calibration can be done. This will insure against catastrophic results from a faulty transducer.

f. Check Calibration of Installed Engine Transducers

Many pick-ups are installed on the engine during dress-down, thermocouples, tachometers, etc. It is possible for one of these to be faulty (disconnection or off specification in the thermocouple not detected during test). By using the calibrate feature, control system abnormalities may be detected before the actual test.

g. Conduct Penalty Runs

After the major test phases are completed and the acceptability of the engine is ascertained, it may be necessary to re-conduct portions of the major test or call upon special penalty run procedures to be executed. This need arises when certain T. O. limits have been exceeded or an engine has been accepted on a marginal condition. This will insure correctness of data and calculations as well as insuring that an out-of-limit condition was not a transient. The ability to automatically select and execute these routines under control of the automatic system will improve both speed and accuracy of the over-all test.

h. Engine Diagnostics

During the running of a sub-portion of the test or after completion of the major test, conditions may arise that will indicate off specifications in the engine or one of its components. By gathering data at high speeds, using past historical data on engine rejects, failure incidents and rework data, and building a series of logical steps or a mathematical model of certain sections of the engine, it will be possible to determine the exact cause of the abnormality and make recommendations for repair.

There will be a learning process by the system. As more and better data is gathered, the logical model will improve.

The ultimate aim of the system is to furnish complete re-work information to the engine penalty line. In many cases, this will save time and prevent unnecessary rework of an engine.

i. Logging and/or Punch-Out of Test Data and Engine Data

After a major test has been completed, all instrument readings, calculations, and diagnostics remarks will be stored on the disk storage unit. The operator in the cell control room will execute a request to the central control system room via the manual entry control. The control system then will print a completed log or run sheet giving the three items above for each test phase.

The log may be used by the operator to select penalty or re-runs if it appears that a component or recording is marginal to the limit.

Several carbon copies may be produced so that copies may be sent to all authorized personnel.

An engine data plate card will be printed to accompany the engine and a military run data card punched for Quality Analysis.

j. Store Test Programs and Parameters for All Type, Model and Serial Number Engines

The control system will use a mass random access unit for storage of the test parameters and limits for all engine models, types and serial number that will be tested. This type storage insures immediate access to all types of engine programs for complete asynchronous testing and control for the test facility.

Mass storage will allow the system to be open ended for expansion to future cells. By the use of this mass storage, a better and more complete engine diagnostic can be performed (as pointed out in the previous section). The system will be designed to allow the updating of all engine technical orders on a daily basis.

As a secondary function, statistical data will be stored for analysis. By storing summary data, critical trends can be detected early. All causes for rejects or defects can be stored by type, model, and serial number. Summary data will be quickly available upon management request.

204

k. Detect Emergency and Unsafe Conditions and Take Appropriate Action

The fast instrument scanning speed of the control system permits dangerous trends to be detected in many of the instrument readings and appropriate corrective action to be initiated to prevent occurence of out-of-limit conditions. In out-of-limit situations, the system will quickly bring the test and engine to a halt to prevent serious damage.

One of the most important things to consider when designing the actual control system is achieving a high degree of reliability. Two types of failures can occur. The failures and corrective actions are:

Type I - Transient Failures

These are internal system transmission errors and occur on a transient basis. In this case, the system will record the failure and try twice more to perform the operation. The recording will be used by maintenance engineers for regular preventive maintenance (once per week). A transient type error will usually be eliminated in three attempts.

Type II - Complete Component Failure

In this case, the system will try to by-pass the bad component switching to manual control or bring the engine to a safe stop. A by-pass procedure will be incorporated for emergency action.

l. Quality Analysis

1) The system will use store data to perform reliability calculations for engine and individual components.

2) The quality analysis will produce data assurance for a better test engine.

m. Production

1) Scheduling - Using advanced techniques such as linear programming, a master plan will be prepared for scheduling the cells.

2) Planning - Better methods of machine and manpower utilization can be prepared.

2. Control System and Interface Description

To approach the problem of determining the necessary hardware, one must keep three factors in mind. They are (1) design functions as determined in the meeting of section 1 (2) instrumentation-present and future, (3) and layout of the basic test cell.

If we notice the basic test cell layout as shown in illustration 1, it shows the location of the control room as being between two test cells. If there are more than two cells (there are usually several more) then it is logical there will be two or more control rooms. Because a typical control system will control more than two cells, it will be necessary to locate the computer in either a remote location or in the rear of one of the test cells. When this is done, there arises necessity for remote communication devices.

Attention should be drawn to the design function to operate in conjunction with this communications device. Whether the operator or the central computer system is the "slave". It will be necessary to place a device for the operator (inspector, etc.) to select the particular test function he wishes to perform. It will also be necessary for him to get return information from the instrument readings, pertinent calculations and emergency or troubleshooting messages.

Many times it will be necessary for the test cell foreman to have information concerning phases of test of engines in each cell in order to coordinate the overall movement of engines in test. He will also need access to stored statistical information pertaining to reject, re-run and other engine test functions. Many times upper level management will inquire of the cell foreman on these statistics. Things that could be available on an inquiry basis would be:

1. Number of rejects/month on a certain model number.
2. Major cause of rejects.
3. What was done for correction.
4. Ranges and standard deviations from set standard operating limits.
5. Etc.

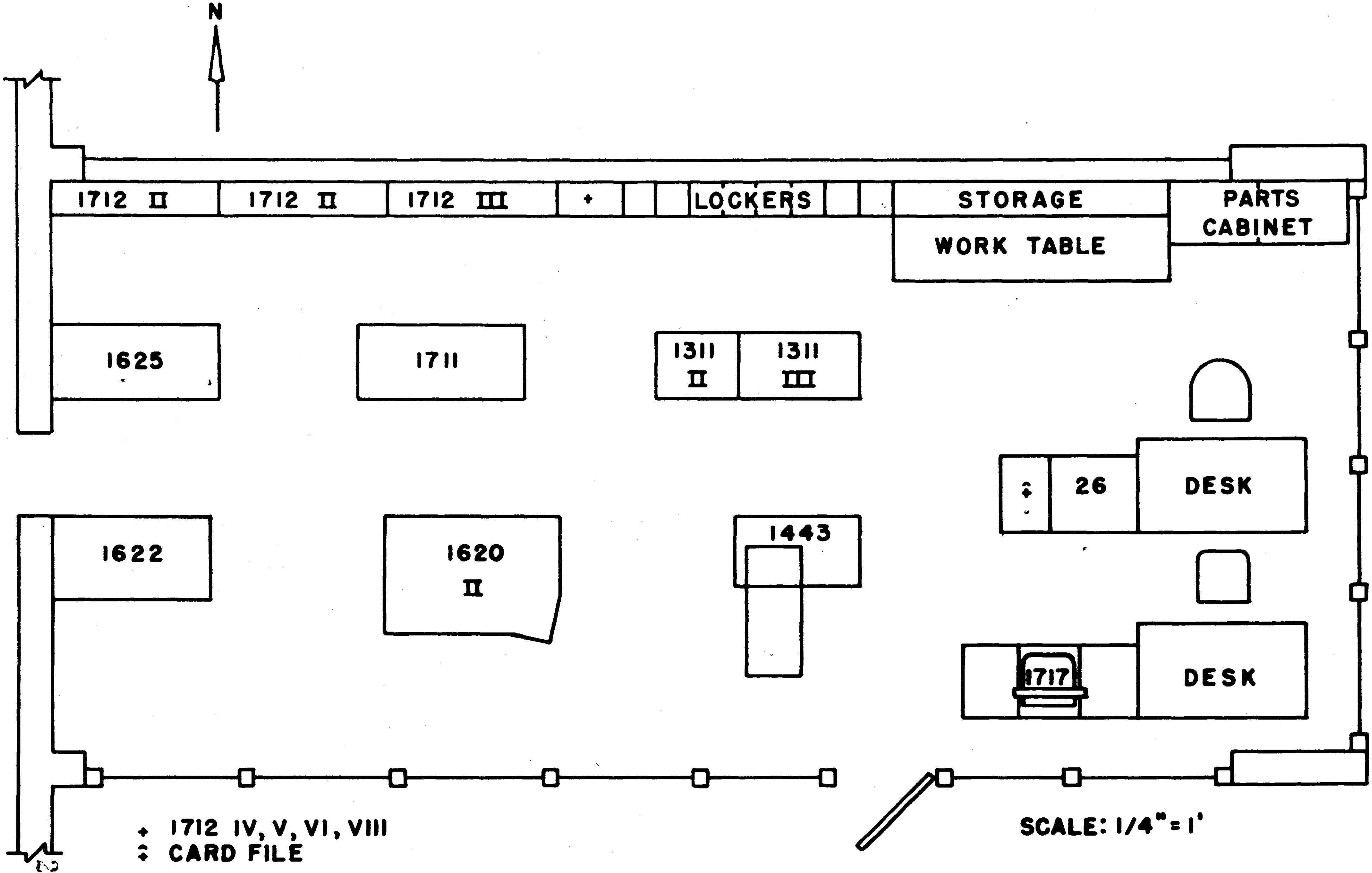
Using the system described in illustration 2 and treating the requirement as <u>8 test cells</u>, each component and its function will be discussed.

a. <u>Central Computer (1620 Model II)</u>

Because of the speed needed to accomplish the sampling of the necessary instrument leads, making all necessary calculations, actually sending output signals, for control and receiving feedback input signals for correction, the 1620 Model II with a 60,000 position memory was chosen. The 1620 as the heart of the 1710 Control System contains the necessary machine instructions and programming systems (Executive System-e.g., monitor) to operate in conjunction with an asynchronous test system design.

b. <u>Auxiliary Storage (2-1311 Disc Files)</u>

Even though the majority of the skeleton test functions are the same for all engines, there still remains different test parameters for each engine model. Each of these parameters must be stored for immediate access. Because no central computer memory would be large enough to contain all test program phases, these must



CONTROL ROOM

also be stored for immediate access as they are called by the skeleton control program. Also contained in auxiliary storage would be necessary diagnostic routines available upon request, as well as emergency limit and correction routines.

A second disc file would be used as in intermediate store area for input/output information, if all input/output devices are busy and they would be available to store quality and production control data gained as a by-product of each test. This data would be available upon inquiry from management.

c. Interface Equipment (1711 and 1712's)

In order to attach all necessary points for 8 cell (see Appendix B) and convert the analog (electrical) signal into a digital form in a sufficient time period, the analog to digital converter (1711) has the ability to convert 200 points/second. In order to handle all necessary analog input points, analog output points, contact indicating and operating relays for an eight (8) test cell facility, it is necessary to have three (3) multiplexing and terminal units (1712's) to the system.

d. Test Cell Input/Output Gear (1713, 1715, 1717)

Located adjacent to each instrument control panel will be an IBM 1713, 1715 and 1717.

The operator will have the option with the IBM 1713 manual entry device-through a set of coded instructions-to dial in either the command for a complete test or portions of a test. The command will be dialed through the use of twelve (12) rotary knobs with zero (0) to nine (9) selection ability.

An enter key will be hit, the information will go via the SIOC channel and interrupt the computer, the computer will read the rotary knobs and start the processing.

e. Interface

All instruments that furnish an electric signal of a standard form will be sent via shielded cabling to the 1712 multiplexing unit, all non standard (pulsed, etc) and pressure type signals will be transformed via transducers (in the test cell control room) to an electrical form and sent to the central system complex.

All pickup signals from the engine are easily adjusted to the standard 1710 signals; however, more specialized servos must be bought or designed to control the throttle and trimming mechanism. There are several types of stepping motors or feedback systems on the market today that can handle these tasks.

All existing cell instruments will remain intact as manual back-up for the system. Through a specially designed panel, the operator will be allowed the option of switching to either automatic or manual system at any time.

III. ECONOMIC JUSTIFICATIONS

The justifications for considering the "Control Systems Approach to Jet Engine Testing" can be broken into tangible, intangible and possible savings categories. The justifications can vary depending upon the application. Some of each are listed as follows:

A. Tangible

1. Increased Engine Throughput:

This can be accomplished by

a. Simplifying the testing procedure.

b. Decreasing delay in such things as trimming and shakedown.

c. Operator Guide Print-Out for prompt emergency and testing actions.

The best time estimate for engine throughput with no major hindrances is 5 hours 55 minutes. As previously mentioned, the average throughput is approximately eight (8) hours for an engine with time running up to twelve (12) hours if there are several re-runs or persistent trouble exists.

The control system would increase the capability of the cells to take on added workload without added facilities. This need would arise in wartime emergency for federal customers and with added contractual obligations for both commercial and federal.

2. Reduced Manpower Requirement/Engine

This would free inspection and operating personnel for a greater engine throughput. One operator would be substantial for testing procedures, where the present system utilizes an inspector and two (2) shops or production personnel.

3. Avoiding Re-Run of Engines

a. By eliminating bad instrument calibration--erroneous transducer signal.

b. Bad instrument reading--can be eliminated. The signal will originate completely at the transducer and eliminate the nonlinearity of the instrument read-out mechanism. Operator error in reading will also be eliminated, e.g., simultaneous reading of instruments.

4. Decreased Fuel Costs

This can be saved with automatic trim procedures and avoidances of excess penalty runs.

5. Decreased Calibration Costs

By automatically calibrating the transducers the computer will give a correction or tare factor for the back-up reading devices. The time between calibrations will decrease. The maintenance costs will correspondingly decrease.

B. Intangibles

1. Better Engine Quality

   a. Better checked out engines through more certain detection of off-specification units.

   b. Simultaneous recording of the instruments, thus insuring proper data for checking limit parameters.

   c. Consistent methods of testing, thus insuring proper acceptance or rejection of an engine.

2. Decreased Re-Work Costs Through Better Diagnostics

As mentioned previously, the system with its on-line mass storage can furnish pinpoint diagnostics to eliminate complete overhaul for minor abnormalities or defects.

3. Data Assurance

This assurance can be derived through getting simultaneous readings or reliability in instrument calibration and will result in better customer (the pilot or branch of the armed services) satisfaction.

4. Increased Safety for Personnel, Equipment and Property

5. Increased Readiness Program on First Line Aircraft

6. Reduced Paperwork Handling

Complete and accurate unit performance logging. Here we have better customer satisfaction through hard copy records. Diagnostics are automatically printed to be sent back for re-work.

IV. SUMMARY AND CONCLUSIONS

This paper has attempted to discuss one particular approach to the application of a digital computer to the closed loop control of a jet engine test cell.

As has been pointed out, there are many approaches to consider in designing a system for a particular application. In summary, the things that must be considered are repeated and listed as follows:

1. Degree of Control Desired-Open or Closed Loop
2. How Much the Operator is "Slave" to the Computer or Vise Versa
3. How Many Test Cells Must Be Controlled Simultaneously
4. The Functions that the Customer Wishes the System to Perform

The justifications for a control system can be varied, depending upon what the customer wishes to accomplish; however, the "state of the art" of automatic control in jet engine testing is in its infancy and there are many justifications in all cases.

It should be pointed out that the general approaches and ideas used are applicable to many other industries and are not limited to jet engine testing.

BIBLIOGRAPHY

1. Industrial Testing Systems at the IBM Components Division

   Poughkeepsie, New York.

   Application Brief, No. K20-1725

2. Sales Guide Industrial Testing Systems

   IBM Internal

3. Introduction to Control Systems

   General Information Manual F26-5577-0

GOODYEAR
GOODYEAR AEROSPACE
CORPORATION
ARIZONA DIVISION
LITCHFIELD PARK, ARIZONA

GENERALIZED FILTER NETWORK

A/C STEADY STATE ANALYSIS PROGRAM

by

D. H. O'Herren

AAP-18911 May 7, 1964

TABLE OF CONTENTS

2 1 4

GENERALIZED FILTER NETWORK

A/C STEADY STATE ANALYSIS PROGRAM

This program has been written to make possible comprehensive surveys of theoretical filter designs. It opens up a more sophisticated range of filters to theoretical consideration and evaluation. The program input is general enough that almost any filter network consisting of cascaded inverted-L or symmetrical lattice sections may be handled easily.

The minimum machine requirements are a 1620 with 40 K core storage, auto divide, and indirect addressing. The source language is Fortran II. There are 6 subprograms plus the mainline program.

Filter design has been speeded in recent years with the advent of tables of normalized low-pass filter element values*. Even if these tables are used, this program allows the designer to compute the effects of component tolerances, finite Q's, and mismatched terminations. These introductory remarks have centered around filter design, but it will be apparent that the program is useful for analyzing any RLC network, e.g., amplitude or phase equalizers.

The filter designer needs to know how a proposed design will perform over a particular range of frequencies before making recommendations to those who will implement the design. Manual calculations of the desired performance parameters over a range of frequencies can be quite tedious and are highly subject to human error. The problem is complicated because theoretical calculations do not always reflect the actual performance, particularly at higher frequencies due to stray reactances. The net result frequently is that a minimum number of proposed designs are evaluated at a few pertinent frequencies from which data plots of parameters versus frequency are made. Curve definition is rarely good, making it virtually impossible to compare proposed filter designs on important fine points.

---

*For example, see 1964 Microwave Engineers' Handbook, pp. 91-95, Horizon House-Microwave, Inc., Dedham, Mass.

This program will evaluate theoretical performance of filters over as broad a range of frequencies as desired giving precise values of the performance criteria. The net result is excellent definition of a greater number of performance curves in much less time. The designer is freed of computational drudgery and may consider more sophisticated designs without concern for the difficulties of manual evaluation.

The parameters computed by the program are insertion loss, impedance, input voltage phase angle (phase shift), and time delay (envelope or group delay). Insertion loss is defined as that loss resulting from inserting a given four terminal filter between the input and output terminals of a network where the output is non-reactive and the input can be a complex impedance driven by a voltage signal source. The phase angle of the input voltage is referenced to the voltage across the output terminals whose magnitude and phase are arbitrary. The impedance calculated is at the input of the filter looking toward the output and is in rectangular form.

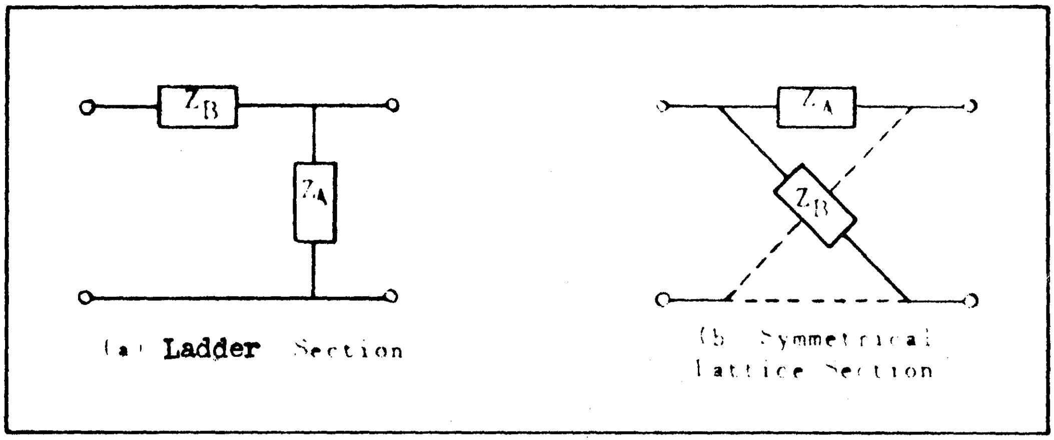
The program is based upon a report titled "Electronic Digital Computer Analysis of Cascaded Networks" by R. H. Tuznik and D. H. Wood of Westinghouse Electric Corporation. The report presents a method of defining in general terms a filter network and then, through recursion techniques, computing the impedance and voltage at the filter input for any desired frequency.

The program can handle networks having either basic ladder or symmetrical lattice sections. These sections consist of Z_A and Z_B basic section impedances as in Figure 1.

A maximum of four different sections in cascade can be handled. This is a limitation imposed by core storage. Note that the program will process a group of from one to four cascaded sections any desired number of times. Thus it is possible to analyze a filter consisting of four repeated groups of four cascaded sections by setting NCYCL, a program input variable, to four.

A section impedance Z_A or Z_B as in Figure 1 can be as simple as a single element or as complex as in Figure 2.

A basic section impedance may consist of from 1 to 3 orders each of which may be as complex as above or as simple as a single element. The maximum number of parallel branches in an order as well as the maximum number of orders again are limited by available core storage.



(a) **Ladder** Section

(b) Symmetrical Lattice Section

FIGURE 1

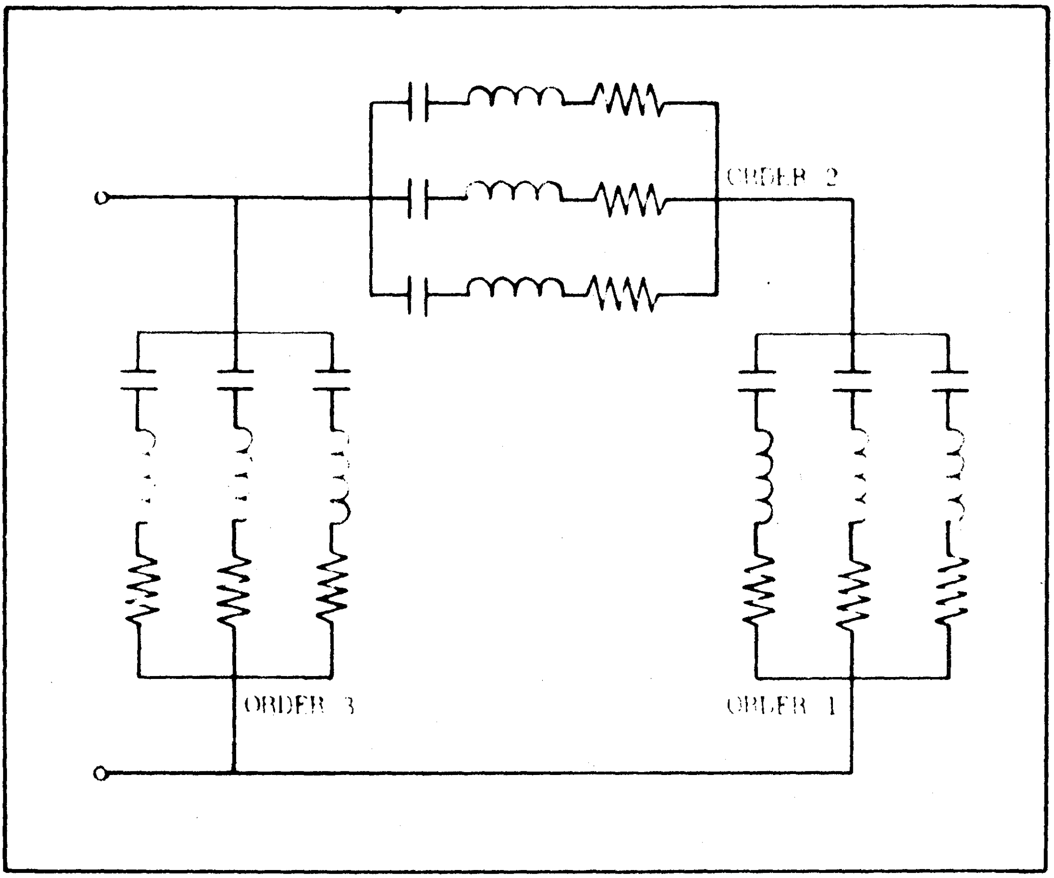


FIGURE 2 - BASIC SECTION IMPEDANCE

The program positions the filter to be analyzed between an input and output termination as in Figure 3.

The input termination may be as complex as shown or a single element. The output termination R_o must be a pure resistance. V_o is an input variable and usually is set equal to 1 volt at zero phase. The program computes V_i, the phase angle of which is used to determine time delay.

The designer may desire to analyze a proposed filter for a frequency sensitive network. The frequency sensitive network must be representable in the form shown in Figure 3 as the input termination. Another way is to include the frequency sensitive network as part of the filter with a single resistance as the input termination. The first way has the advantage of allowing the NCYCLE variable to be other than one.

The program begins by reading cards with the following information:

1. Frequency range including maximum, minimum, and increment.
2. Output termination resistance and voltage.
3. Number and type of cascaded sections in filter.
4. Number of times filter sections are to be cycled.
5. Filter section element values.
6. Input termination element values.

The program immediately punches out cards with these values. Then computation begins at the output termination and progresses section-by-section toward the input termination. Complex impedance and voltage necessary to produce the output termination conditions are calculated at the input of each filter section and so on recursively to the input termination.

The recursion equations programmed are for inverted-L or ladder sections and symmetrical lattice sections. These are in separate subprograms named LADDER and LATTICE respectively. These will be outlined in Appendix A. The program could be expanded to handle other types of basic sections such as the tee, bridged-tee, pi, etc., by writing additional recursion subprograms. Certain of these types of sections can be rearranged into ladder sections for evaluation with the existing program.

Examination of Figure 1 suggested a programming short cut. Both types of basic sections consist of two-terminal basic section impedances as in Figure 2. A single combination of two subprograms is used to compute these impedances.

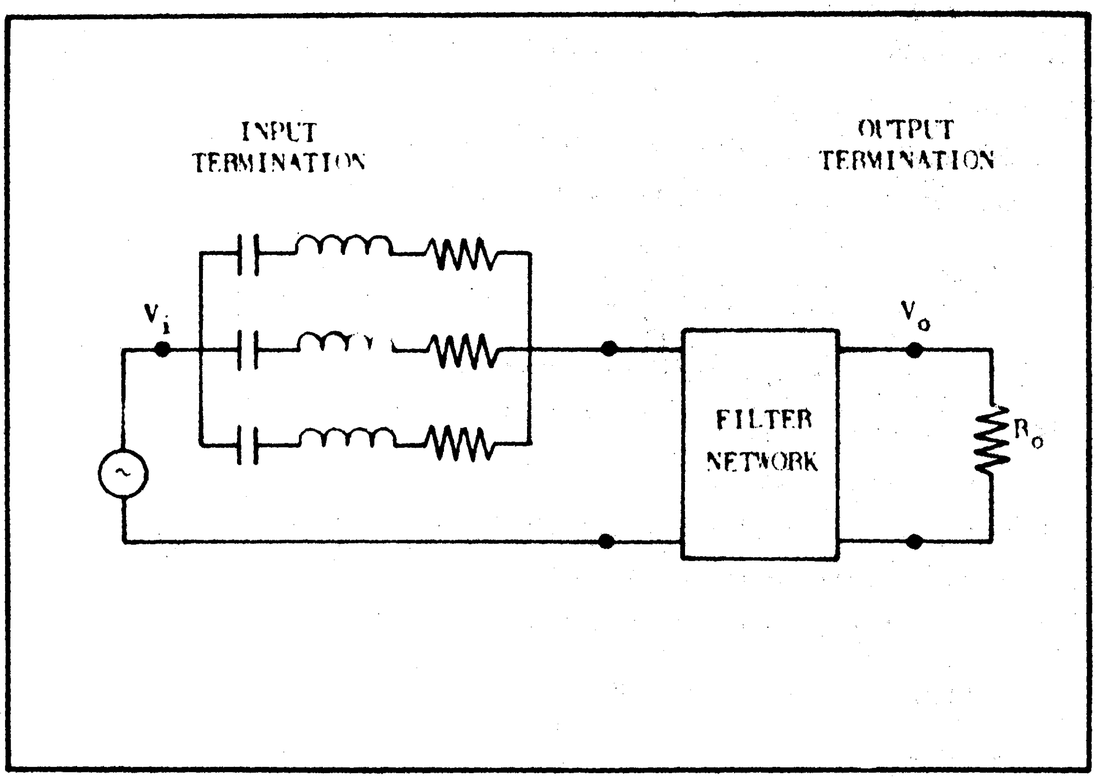


FIGURE 3 - FILTER NETWORK WITH TERMINATIONS

The first, DRPTZ (Driving Point Z) computes the immittances of an order. The second, ORDER, combines the immittances into the basic section impedance. When the first order of a particular basic section impedance Z_A has been processed, the program begins with the first order of the second basic section impedance Z_B. When all orders of a particular section have been processed the program shifts control to the appropriate recursion subprogram LADDER or LATTICE. All basic sections in a particular filter must be either inverted-L or symmetrical lattice type.

DETAILED DESCRIPTION OF PROGRAM INPUT DATA

Program input consists of four categories of cards which must be input in the order following for each network to be analyzed.

1. Title Card.
2. Frequency and Output Termination Card.
3. Basic Section Data Cards
4. Input Termination Cards

Note that jobs may be stacked.

Title Card

Any 80 alphanumeric characters for identification of output.

Frequency and Output Termination Card (F11.5 format)

Cols.	
1 - 11	FMIN - Minimum frequency in megacycles.
12 - 22	FDEL - Frequency increment in megacycles.
23 - 33	FMAX - Maximum frequency in megacycles.
34 - 44	ZOR - Output termination resistance in ohms.
45 - 55	VOR - Output termination voltage (real part) in volts.
56 - 66	VOI - Output termination voltage (imaginary part) in volts.

Basic Section Data

This data describes the basic filter sections. Data should be input section by section beginning at the section nearest the Output Termination.

Inverted-L (Ladder) Sections

Preliminary Card (I5 format)

Col.	
5	NSECT - Number of filter sections (Max. of four).
10	LETTR - Must be two.
15	NCYCL - Number of times basic sections are used.

Order Card (I5 format)

Col.	
5	IORD (K) - Maximum number of orders in Z_A or Z_B impedance of K^{th} section. Maximum of three.

Resonator Card (I5 format)

One card for each pair of Z_A and Z_B orders. This card must immediately precede the Z_A and Z_B parameter cards.

Col.	
5	MP - Number of resonators (parallel branches) in Z_A.
10	MQ - Number of resonators (parallel branches) in Z_B.

Z_A Parameter Cards (E14.8 format)

One card per MP resonator specified on preceding resonator card.

Cols.	
1 - 14	C - Capacitance in Farads.
15 - 28	AL - Inductance in Henries.
29 - 42	R - Resistance in Ohms.

Z_B Parameter Cards (E14.8 format)

One card per MQ resonator specified on preceding resonator card.

Cols.	
1 - 14	C - Capacitance in Farads.
15 - 28	AL- Inductance in Henries.
29 - 42	R - Resistance in Ohms.

Symmetrical Lattice Sections

This data is identical to Inverted-L data except that LETTR on the preliminary card in Col. 10 must be 1 and resistive pi pad parameters must be input. The card for the pi pad between the output termination and the first basic section must follow the preliminary card. Cards for pi pads following basic sections must follow the Order Card.

Pi Pad Card (E14.8 format)

Cols.

1 - 14 RA - Resistance of pad shunt arms in ohms.

15 - 28 RB - Resistance of pad series arm in ohms.

SAMPLE PROBLEMS

Three examples will be given, one of which will be quite simple and the other two slightly advanced.

Sample Problem 1

This problem will illustrate the input for a simple inverted-L (LADDER) filter network.

Given the two-section ladder filter in Figure 4 with element values as below, calculate filter parameters from 1.0 mc to 10 mc with a step of 1.0 mc. Use output voltage of 1.0 volts at zero angle as reference.

Output Termination

R_o = 50 ohms.

First Ladder Section

R_1 = 1000 ohms

C_1 = 1.0 μf

L_1 = 2.0 μh

Second Ladder Section

R_2 = 100,000 ohms

C_2 = .01 μf

L_2 = 1.0 μh

Input Termination Section

R_T = 50. ohms

Preparation of Input Data - Problem 1

Title Card

Sample Problem 1 - Network Analysis Program - Date

Frequency and Output Termination Card

Cols.	Value	
1 - 11	1.0	Minimum Frequency (mc) FMIN.
12 - 22	1.0	Frequency Increment (mc) FDEL.
23 - 33	10.0	Maximum Frequency (mc) FMAX.
34 - 44	50.0	Output Termination Resistance (ohms) ZOR
45 - 55	1.0	Output Termination Voltage Real Part (volts) VOR
56 - 66	0.0	Output Termination Voltage Imag. Part (volts) VOI

Basic Ladder Section Cards

Preliminary Card

Col.	Value	
5	2	Number of filter sections
10	2	Ladder section signal
15	1	Number of times basic sections are used

Order Card (First Section)

Col.	Value	
5	1	Maximum Number of Orders in First Section

Resonator Card (First Order)

Col.	Value	
5	1	The only order of the Z_A impedance of the first section has a single resonator.
10	1	The only order of the Z_B impedance of the first section has a single resonator.

Z_A Parameters Card

Cols.	Value	
1 - 14	+.10000000E-05	Capacitance in Farads - C_1.
15 - 28	+.00000000E-99	Inductance in Henries
29 - 42	+.10000000E+04	Resistance in Ohms - R_1.

NOTE: There is only one Z_A parameter card for the first section because there is only one resonator.

Z_B Parameter Card

Cols.	Value	
1 - 14	+.00000000E-99	Capacitance in Farads
15 - 28	+.20000000E-05	Inductance in Henries - L_1.

Order Card (Second Section)

Col	Value	
5	1	Maximum number of orders in second section.

Resonator Card (First Order)

Col .	Value	
5	2	There are 2 resonators in the only order of the Z_A impedance of the second section.
10	1	There is 1 resonator in the only order of the Z_B impedance of the second section.

Z_A Parameter Cards

Cols.	Value	
1 - 14	+.10000000E-07	Capacitance in Farads - C_2.

Cols.	Value	
1 - 14	+.00000000E-99	Capacitance in Farads
15 - 28	+.00000000E-99	Inductance in Henries
29 - 42	+.10000000E+06	Resistance in Ohms - R_2.

Z_B Parameter Card

Cols.	Value	
1 - 14	+.00000000E-99	Capacitance in Farads
15 - 28	+.10000000E-05	Inductance in Henries - L_2.

Order Card (Input Termination)

Col.	Value	
5	1	There is a single order in the input termination.

Resonator Card (First Order)

Col.	Value	
5	1	There is one resonator in the input termination and it is treated as a Z_A type impedance.
10	0	No Z_B resonators.

Z_A Parameter Card

Cols.	Value	
1 - 14	+.00000000E-99	Capacitance in Farads
15 - 28	+.00000000E-99	Inductance in Henries
29 - 42	+.50000000E+02	Resistance in Ohms - R_T

This completes the input data cards for sample problem 1. A listing of these cards is found on Page 22. A listing of the output is found on Page 23.

Sample Problem 2

This problem will illustrate the input for a two section symmetrical lattice filter network.

Given the network in Figure 5 with two symmetrical lattice sections separated by resistive pi pads:

Output Termination

R_o - 100 ohms

First Pi Pad

RA_o - 10000 ohms
RB_o - 5 ohms

First Section

C_1 - 30 μμf
C_2 - .02 μf
L_3 - .1 μh

Second Pi Pad

RA_1 - 10000 ohms
RB_1 - 5 ohms

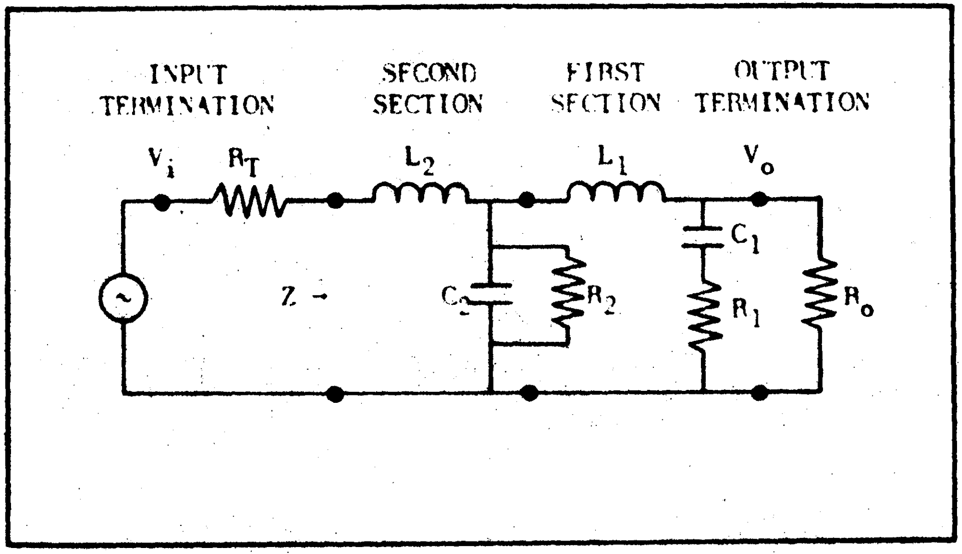


FIGURE 4 - SAMPLE PROBLEM 1

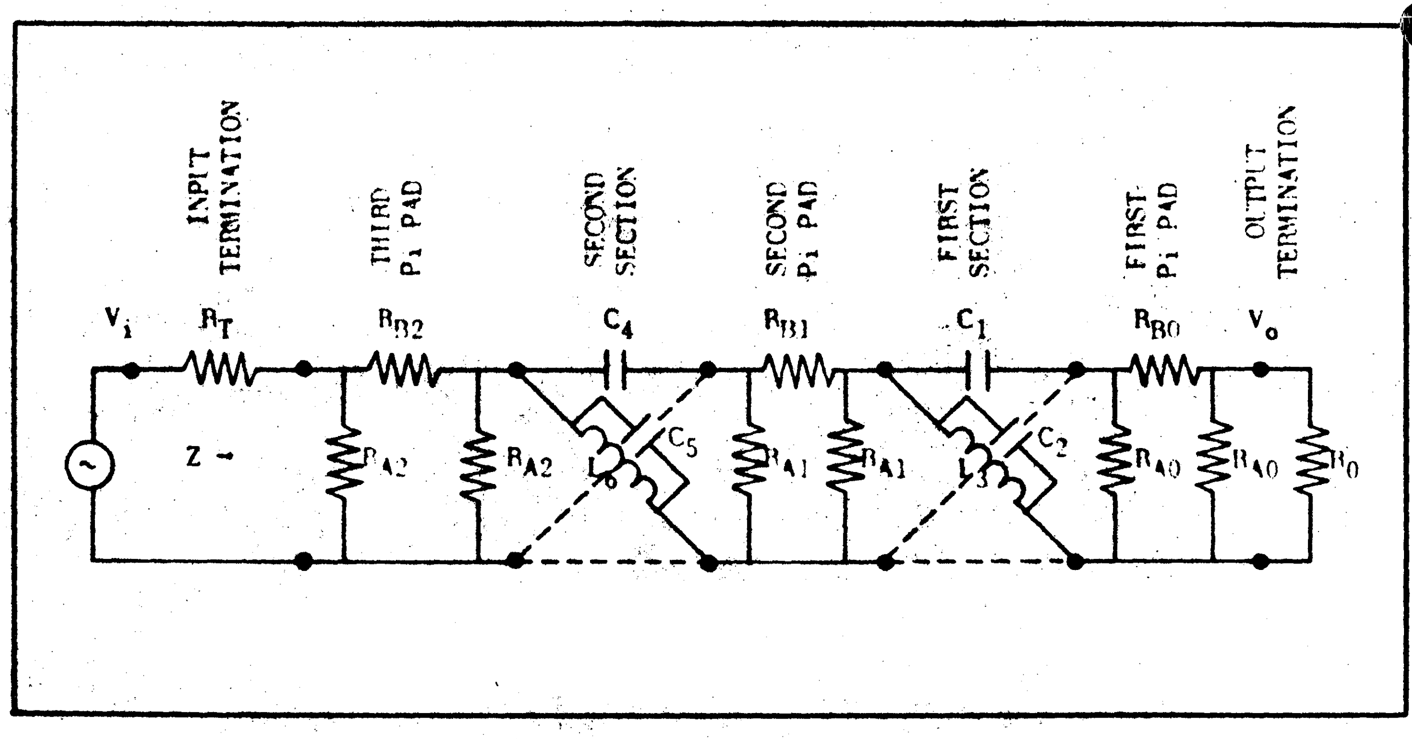


FIGURE 5 - SAMPLE PROBLEM 2

Second Section

C_4 - 300 μμf
C_5 - .4 μf
L_6 - .15 μh

Third Pi Pad

RA_2 - 10000 ohms
RB_2 - 5 ohms

Input Termination

R_T - 100 ohms

The system in Figure 5 is to be evaluated over a range of frequencies from 50 to 100 megacycles with 2 megacycle steps. The program always computes the same type of information, namely impedance, phase angle of V_1, insertion loss, and time delay.

Preparation of Input Data - Problem 2

Title Card

Sample Problem 2, Network Analysis Program - Date.

Frequency and Output Termination Card

Cols.	Value	
1 - 11	50.	Min. Freq. in Mc.
12 - 22	2.	Delta Freq. in Mc.
23 - 33	100.	Max. Freq. in Mc.
34 - 44	100.	R_o in Ohms.
45 - 55	1.0	V_o in Volts (real part).
56 - 66	0.0	V_o in Volts (imag. part).

Basic Lattice Section Cards

Preliminary Card

Cols.	Value	
5	2	Number of filter sections.
10	1	Lattice section signal.
15	1	Number of times basic sections are used.

First Pi Pad Card

Cols.	Value	
1 - 14	+.10000000 E+05	RA_o in ohms
15 - 28	+.50000000E +01	RB_o in ohms

Order Card (First Section)

Cols.	Value	
5	1	Maximum Number of Orders in Section

Second Pi Pad Card

Same as First Pi Pad Card

Resonator Card (First Order)

Col	Value	
5	1	A single resonator in Z_A.
10	2	Two resonators in Z_B.

Z_A Parameter Card

Cols.	Value	
1 - 14	+.30000000 E-10	Capacitance in Farads - C_1.

Z_B Parameter Cards

Cols.	Value	
1 - 14	+.20000000 E-07	Capacitance in Farads - C_2.

Cols.	Value	
1 - 14	+.00000000 E-99	Capacitance in Farads.
15 - 28	+.10000000 E-06	Inductance in Henries - L_3.

Order Card (Second Section)

Cols.	Value	
5	1	Max. Number of Orders in Section.

Third Pi Pad Card

Same as First Pi Pad Card.

Resonator Card (First Order)

Cols.	Value	
5	1	A single resonator in Z_A.
10	2	Two resonators in Z_B.

Z_A Parameter Card

Col.	Value	
1 - 14	+.30000000 E-09	Capacitance in Farads - C_4.

Z_B Parameter Cards

Cols.	Value	
1 - 14	+.40000000 E-06	Capacitance in Farads - C_5.

Cols.	Value	
1 - 14	+.00000000 E-99	Capacitance in Farads.
15 - 28	+.15000000 E-06	Inductance in Henries - L_6.

Order Card (Input Termination)

Col.	Value	
5	1	Maximum of One Order allowed in Input Termination.

Resonator Card (First Order)

Col.	Value	
5	1	There is one Z_A Resonator in the Input Termination.
10	0	The Input Termination Cannot Have A Z_B Resonator.

Z_A Parameter Card

Cols.	Value	
1 - 14	+.00000000 E-99	Capacitance in Farads.
15 - 28	+.00000000 E-99	Inductance in Henries.
29 - 42	+.10000000 E+03	Resistance in Ohms R_T.

This completes the input for sample problem 2. A listing of these cards is on page 24. The resulting output is on page 25.

Sample Problem 3

This problem will illustrate three features not displayed by the previous problems, namely the use of NCYCL for multiple usage of a group of cascaded sections, the use of a section having 3 orders, and a complex input termination.

This filter has a repeated group of two sections, that is, the First and Third Sections are identical, also the Second and Fourth Sections. The block labeled Z3 represents the two-terminal three-order impedance shown in Figure 7. The frequency range of interest is 5 to 100 Mc in steps of 5 Mc.

Element values are as follow:

Output Termination

R_o - 100 Ohms.

First Section

C_1 - 1 µf
R_2 - 1000 Ohms.

Second Section

C_3 - 1 µf
R_4 - 1 Megohm
R_5 - 1000 Ohms
L_6 - .2 µµh
C_7 - .3µf
C_8 - 1. µµf
R_g - 100 Ohms
l_{10}-1 µh
R_{11}-50 Ohms
C_{12}- .1 µµf
L_{13}- .2 µµh
R_{14}- 100 Ohms
L_{15}- .01 µh
R_{16}- 100 Ohms
C_{17}- .005 µf
L_{18}- .5 µh
R_{19}-100 Ohms
R_{20}- .1 Megohms
C_{21}- 1 µf

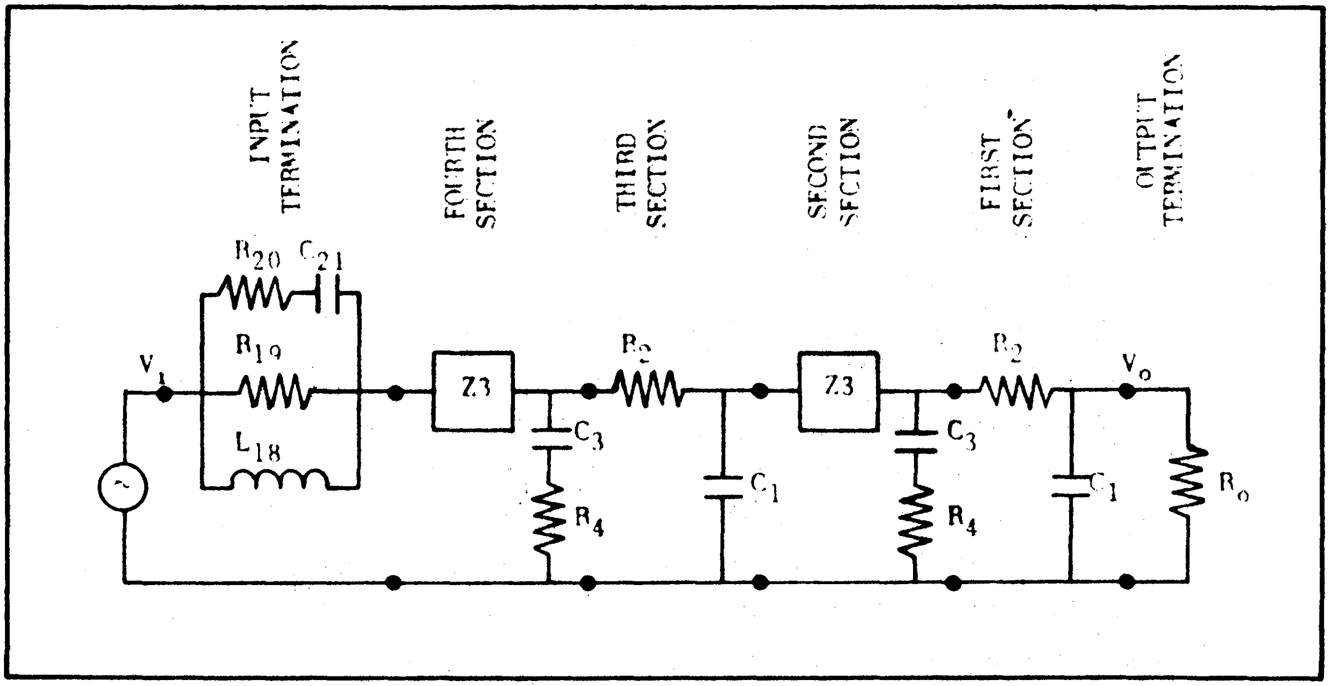


FIGURE 6 - SAMPLE PROBLEM 3

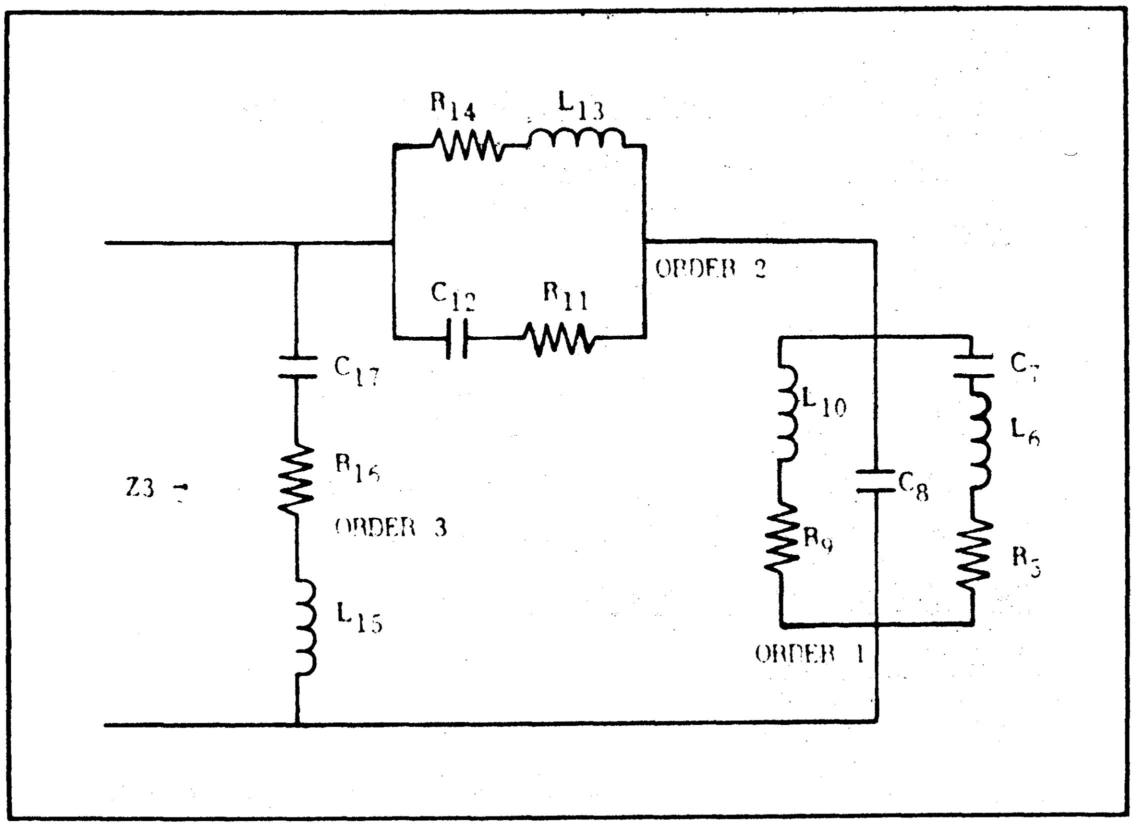


FIGURE 7 - Z3 IMPEDANCE IN SAMPLE PROBLEM

Sample Problem 3 (Continued)

For simplicity the output voltage will be assumed 1.0 volts at zero phase as before. The input to this problem is fairly complex but the fact that the first two sections are repeated in the next two sections means that only the first two sections must be input. Setting the NCYCL input variable equal to two (2) causes the program to process the two-section group twice before starting on the input termination. NCYCL can be set to any positive integer (less than 999). This feature can be used with either inverted -L or symmetrical lattice sections. As the program is written, the group of sections which is repeated can include a maximum of four (4) different sections. Only a single group can be in any one filter and the repeated group cannot be preceded or followed by any other sections excepting the input and output terminations. The three-order impedance Z3 in Figure 7 illustrates the complexity permitted by the program for all of the Z_A or Z_B type impedances in any or all basic sections. The complex input termination could have included three elements in each parallel branch or resonator but cannot exceed a single order.

Data Preparation for Sample Problem 3

Title Card

Sample Problem 3, Network Analysis Program - Date

Frequency and Output Termination Card

Cols.	Value	
1 - 11	5.0	Min. Freq. Mc
12 - 22	5.0	Delta Freq. Mc
23 - 33	100.0	Max. Freq. Mc
34 - 44	100.0	R_o in Ohms
45 - 55	1.0	V_o (Real Part) in Volt
56 - 66	0.0	V_o (Imag. Part) in Volt

Basic Ladder Section Data Cards

Preliminary Card

Col	Value	
5	2	There are two different sections to be repeated.
10	2	Ladder Section Signal
15	2	Cycle the two sections twice.

Order Card (First Section)

Col.	Value	
5	1	Max. Number of Orders.

Resonator Card (First Order)

Cols.	Value	
5	1	Number of Resonators in Z_A
10	1	Number of Resonators in Z_B

Z_A Parameter Card

Col.	Value	
1 - 14	+.10000000 E-05	C_1 Farads.

Z_B Parameter Card

Cols.	Value	
1 - 14	+.00000000 E-99	
15 - 28	+.00000000 E-99	
29 - 42	+.10000000 E+04	R_2 in Ohms.

Order Card (Second Section)

Col	Value	
5	3	Max. Number of Orders in a Section Impedance.

Resonator Card (First Order)

Col	Value	
5	1	Number of Z_A Resonators
10	3	Number of Z_B Resonators

Z_A Parameter Card

Cols.	Value	
1 - 14	+.10000000 E-11	C_3 in Farads
15 - 28	+.00000000 E-99	
29 - 42	+.10000000 E+07	R_4 in Ohms.

Z_B Parameter Cards

Cols.	Value	
1 - 14	+.30000000 E-06	C_7 in Farads.
15 - 28	+.20000000 E-12	L_6 in Henries.
29 - 42	+.10000000 E+04	R_5 in Ohms.

Cols.	Value	
1 - 14	+.10000000 E-11	C_8 in Farads.

Cols.	Value	
1 - 14	+.00000000 E-99	
15 - 28	+.10000000 E-05	L_{10} in Henries.
29 - 42	+.10000000 E+03	R_9 in Ohms.

Resonator Card (Second Order)

Col.	Value	
5	0	No Z_A Resonators
10	2	Two Resonators in Z_B Impedance.

Z_B Parameter Cards

Cols.	Value	
1 - 14	+.10000000E-12	C_{12} in Farads.
15 - 28	+.00000000E-99	
29 - 42	+.50000000E+02	R_{11} in Ohms.

Cols.	Value	
1 - 14	+.00000000E-99	
15 - 28	+.20000000E-12	L_{13} in Henries.
29 - 42	+.10000000E+03	R_{14} in Ohms.

Resonator Card (Third Order)

Cols.	Value	
5	0	No Z_A Resonator
10	1	1 Z_B Resonator

Z_B Parameter Card

Cols.	Value	
1 - 14	+.50000000E-08	C_{17} in Farads.
15 - 28	+.10000000E-07	L_{15} in Henries.
29 - 42	+.10000000E+03	R_{16} in Ohms.

Order Card (Input Termination)

Col.	Value	
5	1	One Order

Resonator Card (First Order)

Col.	Value	
5	3	Three Resonators in Z_A.
10	0	No Z_B Resonators.

Z_A Parameter Cards

Cols.	Value	
1 - 14	+.00000000E-99	
15 - 28	+.50000000E-06	L_{18} in Henries.

Cols.	Value	
1 - 14	+.00000000E-99	
15 - 28	+.00000000E-99	
29 - 42	+.10000000E+03	R_{19} in Ohms.

Cols.	Value	
1 - 14	+.10000000E-05	C_{21} in Farads.
15 - 28	+.00000000E-99	
29 - 42	+.10000000E+06	R_{20} in Ohms.

This completes the input to sample problem 3. A listing of these cards is found on page 27. A listing of the results is found on Page 28.

```
 SAMPLE PROBLEM 1 FOR GENERAL FILTER NETWORK ANALYSIS PROGRAM    29APR64
1.         1.         10.        50.        1.         0.0
    2    2    1
    1
    1    1
+.10000000E-05+.00000000E-99+.10000000E+04
+.00000000E-99+.20000000E-05
    1
    2    1
+.10000000E-07
+.00000000E-99+.00000000E-99+.10000000E+06
+.00000000E-99+.10000000E-05
    1
    1    0
+.00000000E-99+.00000000E-99+.50000000E+02
```

```
GENERAL NETWORK ANALYSIS PROGRAM

 SAMPLE PROBLEM 1 FOR GENERAL FILTER NETWORK ANALYSIS PROGRAM   29APR64

ZOR=   50.000   VOR=    1.000   VOI=    0.000

SECTIONS=   2   LETTR=   2   NCYCL=   1

ORDER   1IS   1

K=   1   I=   1   MP=  .1   MQ=   1
C( 1 , 1 , 1 ) =  .10000000E-05  L =  .00000000E-99  R =  .10000000E+04
C( 2 , 1 , 1 ) =  .00000000E-99  L =  .20000000E-05  R =  .00000000E-99

ORDER   2IS   1

K=   2   I=   1   MP=   2   MQ=   1
C( 1 , 2 , 1 ) =  .10000000E-07  L =  .00000000E-99  R =  .00000000E-99
C( 3 , 2 , 1 ) =  .00000000E-99  L =  .00000000E-99  R =  .10000000E+06
C( 2 , 2 , 1 ) =  .00000000E-99  L =  .10000000E-05  R =  .00000000E-99

ORDER   3IS   1

K=   3   I=   1   MP=   1   MQ=   0
C( 1 , 3 , 1 ) =  .00000000E-99  L =  .00000000E-99  R =  .50000000E+02

    F           DB          BETA         ZREAL        ZIMAG          T(F)

   1.0000      4.9387      76.462       5.2953      -9.2583
   1.0010      4.9450      76.513       5.2851      -9.2400     .14120277E-06
   2.0000     10.7053     114.502       1.1774       4.1843
   2.0010     10.7105     114.531       1.1760       4.1947     .81361108E-07
   3.0000     15.4775     138.972        .4043      13.2695
   3.0010     15.4819     138.993        .4039      13.2777     .57249998E-07
   4.0000     19.5824     156.885        .1711      20.9877
   4.0010     19.5862     156.901        .1709      20.9950     .43333332E-07
   5.0000     23.2046     170.717        .0829      28.1290
   5.0010     23.2081     170.729        .0828      28.1360     .34055554E-07
   6.0000     26.4500     181.741        .0443      34.9788
   6.0010     26.4531     181.751        .0443      34.9855     .27527777E-07
   7.0000     29.3887     190.725        .0256      41.6625
   7.0010     29.3915     190.733        .0256      41.6691     .22638888E-07
   8.0000     32.0716     198.173        .0157      48.2434
   8.0010     32.0741     198.180        .0157      48.2500     .18916666E-07
   9.0000     34.5371     204.434        .0101      54.7565
   9.0010     34.5395     204.440        .0101      54.7630     .15999999E-07
  10.0000     36.8156     209.760        .0068      61.2225
  10.0010     36.8178     209.765        .0068      61.2289     .13666666E-07
```

```
 SAMPLE PROBLEM 2 NETWORK ANALYSIS PROGRAM     1MAY64
50.        2.         100.       100.       1.         0.0
    2    1    1
+.10000000E+05+.50000000E+01
   .1
+.10000000E+05+.50000000E+01
    1    2
+.30000000E-10
+.20000000E-07
+.00000000E-99+.10000000E-06
    1
+.10000000E+05+.50000000E+01
    1    2
+.30000000E-09
+.40000000E-06
+.00000000E-99+.15000000E-06
    1,
    1    0
+.00000000E-99+.00000000E-99+.10000000E+03
```

```
GENERAL NETWORK ANALYSIS PROGRAM

 SAMPLE PROBLEM 2 NETWORK ANALYSIS PROGRAM     1MAY64

ZOR=  100.000   VOR=     1.000   VOI=     0.000
RAO= .10000000E+05   RBO= .50000000E+01

SECTIONS=    2   LETTR=   1   NCYCL=    1
R(1, 1 ) =   .10000000E+05  R(2, 1 ) =   .50000000E+01    1 ORDERS
K=    1    I=    1    MP=    1    MQ=    2
C( 1 , 1 , 1 ) =   .30000000E-10   L =   .00000000E-99   R =   .00000000E-99
C( 2 , 1 , 1 ) =   .20000000E-07   L =   .00000000E-99   R =   .00000000E-99
C( 4 , 1 , 1 ) =   .00000000E-99   L =   .10000000E-06   R =   .00000000E-99
R(1, 2 ) =   .10000000E+05  R(2, 2 ) =   .50000000E+01    1 ORDERS
K=    2    I=    1    MP=    1    MQ=    2
C( 1 , 2 , 1 ) =   .30000000E-09   L =   .00000000E-99   R =   .00000000E-99
C( 2 , 2 , 1 ) =   .40000000E-06   L =   .00000000E-99   R =   .00000000E-99
C( 4 , 2 , 1 ) =   .00000000E-99   L =   .15000000E-06   R =   .00000000E-99

ORDER    3IS    1

K=    3    I=    1    MP=    1    MQ=    0
C( 1 , 3 , 1 ) =   .00000000E-99   L =   .00000000E-99   R =   .10000000E+03

     F            DB           BETA          ZREAL         ZIMAG          T(F)

   50.0000      21.6168       89.216        5.2596       -4.8518
   50.0010      21.6170       89.216        5.2596       -4.8517      .55555554E-09
   52.0000      21.9569       89.613        5.2426       -4.6662
   52.0010      21.9571       89.613        5.2425       -4.6661      .55833331E-09
   54.0000      22.2846       89.994        5.2274       -4.4943
   54.0010      22.2848       89.994        5.2274       -4.4943      .52499998E-09
   56.0000      22.6008       90.361        5.2138       -4.3347
   56.0010      22.6009       90.361        5.2138       -4.3346      .52777776E-09
   58.0000      22.9061       90.716        5.2016       -4.1859
   58.0010      22.9063       90.716        5.2016       -4.1859      .47222220E-09
   60.0000      23.2015       91.060        5.1906       -4.0471
   60.0010      23.2016       91.060        5.1906       -4.0470      .47222220E-09
   62.0000      23.4874       91.393        5.1806       -3.9172
   62.0010      23.4875       91.394        5.1806       -3.9171      .47222220E-09
   64.0000      23.7646       91.718        5.1715       -3.7953
   64.0010      23.7647       91.718        5.1715       -3.7953      .41666665E-09
   66.0000      24.0334       92.033        5.1633       -3.6809
   66.0010      24.0336       92.034        5.1633       -3.6808      .41666665E-09
   68.0000      24.2946       92.341        5.1558       -3.5731
   68.0010      24.2947       92.342        5.1557       -3.5731      .44444443E-09
   70.0000      24.5484       92.642        5.1488       -3.4715
   70.0010      24.5485       92.643        5.1488       -3.4714      .41666665E-09
   72.0000      24.7953       92.937        5.1425       -3.3755
   72.0010      24.7954       92.937        5.1425       -3.3755      .41666665E-09
   74.0000      25.0356       93.225        5.1367       -3.2847
   74.0010      25.0358       93.226        5.1367       -3.2847      .38888887E-09
   76.0000      25.2698       93.508        5.1313       -3.1987
   76.0010      25.2699       93.509        5.1313       -3.1986      .38888887E-09
   78.0000      25.4982       93.786        5.1263       -3.1170
```

78.0010	25.4983	93.786	5.1263	-3.1170	.38888887E-09
80.0000	25.7209	94.059	5.1217	-3.0395	
80.0010	25.7210	94.059	5.1217	-3.0394	.36111110E-09
82.0000	25.9384	94.328	5.1174	-2.9657	
82.0010	25.9385	94.328	5.1174	-2.9657	.36111110E-09
84.0000	26.1509	94.593	5.1134	-2.8954	
84.0010	26.1510	94.593	5.1134	-2.8954	.36111110E-09
86.0000	26.3585	94.853	5.1097	-2.8284	
86.0010	26.3586	94.853	5.1097	-2.8284	.38888887E-09
88.0000	26.5616	95.110	5.1062	-2.7645	
88.0010	26.5617	95.110	5.1062	-2.7644	.36111110E-09
90.0000	26.7604	95.364	5.1030	-2.7033	
90.0010	26.7605	95.364	5.1030	-2.7033	.36111110E-09
92.0000	26.9549	95.614	5.1000	-2.6449	
92.0010	26.9550	95.615	5.1000	-2.6449	.36111110E-09
94.0000	27.1455	95.862	5.0971	-2.5889	
94.0010	27.1456	95.862	5.0971	-2.5889	.33333332E-09
96.0000	27.3323	96.107	5.0944	-2.5353	
96.0010	27.3324	96.107	5.0944	-2.5352	.36111110E-09
98.0000	27.5154	96.349	5.0919	-2.4838	
98.0010	27.5155	96.349	5.0919	-2.4838	.33333332E-09
100.0000	27.6950	96.588	5.0896	-2.4344	
100.0010	27.6951	96.588	5.0896	-2.4344	.33333332E-09

```
 SAMPLE PROBLEM 3 NETWORK ANALYSIS PROGRAM   4MAY64
5.          5.          100.        100.        1.          0.0
    2    2    2
    1
    1    1
+.10000000E-05
+.00000000E-99+.00000000E-99+.10000000E+04
    3
    1    3
+.10000000E-11+.00000000E-99+.10000000E+07
+.30000000E-06+.20000000E-12+.10000000E+04
+.10000000E-11
+.00000000E-99+.10000000E-05+.10000000E+03
    0    2
+.10000000E-12+.00000000E-99+.50000000E+02
+.00000000E-99+.20000000E-12+.10000000E+03
    0    1
+.50000000E-08+.10000000E-07+.10000000E+03
    1
    3    0
+.00000000E-99+.50000000E-06
+.00000000E-99+.00000000E-99+.10000000E+03
+.10000000E-05+.00000000E-99+.10000000E+06
```

```
GENERAL NETWORK ANALYSIS PROGRAM

 SAMPLE PROBLEM 3 NETWORK ANALYSIS PROGRAM    4MAY64

ZOR=  100.000    VOR=     1.000    VOI=      0.000

SECTIONS=    2    LETTR=    2    NCYCL=    2

ORDER    1IS    1

K=    1    I=    1    MP=    1    MQ=    1
C( 1 , 1 , 1 ) =  .10000000E-05   L =   .00000000E-99   R =   .00000000E-99
C( 2 , 1 , 1 ) =  .00000000E-99   L =   .00000000E-99   R =   .10000000E+04

ORDER    2IS    3

K=    2    I=    1    MP=    1    MQ=    3
C( 1 , 2 , 1 ) =  .10000000E-11   L =   .00000000E-99   R =   .10000000E+07
C( 2 , 2 , 1 ) =  .30000000E-06   L =   .20000000E-12   R =   .10000000E+04
C( 4 , 2 , 1 ) =  .10000000E-11   L =   .00000000E-99   R =   .00000000E-99
C( 6 , 2 , 1 ) =  .00000000E-99   L =   .10000000E-05   R =   .10000000E+03
K=    2    I=    2    MP=    0    MQ=    2
C( 2 , 2 , 2 ) =  .10000000E-12   L =   .00000000E-99   R =   .50000000E+02
C( 4 , 2 , 2 ) =  .00000000E-99   L =   .20000000E-12   R =   .10000000E+03
K=    2    I=    3    MP=    0    MQ=    1
C( 2 , 2 , 3 ) =  .50000000E-08   L =   .10000000E-07   R =   .10000000E+03

ORDER    3IS    1

K=    3    I=    1    MP=    3    MQ=    0
C( 1 , 3 , 1 ) =  .00000000E-99   L =   .50000000E-06   R =   .00000000E-99
C( 3 , 3 , 1 ) =  .00000000E-99   L =   .00000000E-99   R =   .10000000E+03
C( 5 , 3 , 1 ) =  .10000000E-05   L =   .00000000E-99   R =   .10000000E+06

     F           DB          BETA        ZREAL        ZIMAG          T(F)

    5.0000    180.7194     180.838    1065.2886        .2987
    5.0010    180.7228     180.838    1065.2887        .2999    .77777775E-09
   10.0000    192.0992     182.003    1066.2497       4.5705
   10.0010    192.1008     182.004    1066.2501       4.5712    .55555554E-09
   15.0000    198.3916     182.824    1067.7597       7.4989
   15.0010    198.3926     182.825    1067.7601       7.4994    .38888887E-09
   20.0000    202.7228     183.368    1069.6027       9.6923
   20.0010    202.7236     183.368    1069.6031       9.6927    .22222221E-09
   25.0000    206.0672     183.685    1071.5953      11.2810
   25.0010    206.0678     183.685    1071.5957      11.2812    .13888888E-09
   30.0000    208.8277     183.835    1073.5931      12.3601
   30.0010    208.8282     183.835    1073.5935      12.3602    .55555554E-10
   35.0000    211.1986     183.869    1075.4981      13.0270
   35.0010    211.1991     183.869    1075.4985      13.0271    .00000000E-99
   40.0000    213.2865     183.828    1077.2542      13.3759
   40.0010    213.2869     183.828    1077.2546      13.3760   -.27777777E-10
   45.0000    215.1560     183.741    1078.8370      13.4893
   45.0010    215.1563     183.741    1078.8373      13.4893   -.55555554E-10
   50.0000    216.8501     183.628    1080.2430      13.4347
```

```
 50.0010   216.8505    183.628   1080.2433    13.4347  -.55555554E-10
 55.0000   218.3995    183.501   1081.4807    13.2648
 55.0010   218.3998    183.501   1081.4809    13.2647  -.55555554E-10
 60.0000   219.8269    183.369   1082.5648    13.0190
 60.0010   219.8272    183.369   1082.5651    13.0189  -.55555554E-10
 65.0000   221.1500    183.237   1083.5125    12.7260
 65.0010   221.1502    183.237   1083.5127    12.7260  -.83333331E-10
 70.0000   222.3827    183.108   1084.3404    12.4067
 70.0010   222.3829    183.108   1084.3405    12.4066  -.55555554E-10
 75.0000   223.5363    182.985   1085.0647    12.0753
 75.0010   223.5365    182.985   1085.0649    12.0752  -.55555554E-10
 80.0000   224.6202    182.867   1085.6997    11.7419
 80.0010   224.6204    182.867   1085.6998    11.7418  -.55555554E-10
 85.0000   225.6422    182.757   1086.2579    11.4133
 85.0010   225.6424    182.757   1086.2580    11.4132  -.55555554E-10
 90.0000   226.6089    182.653   1086.7501    11.0938
 90.0010   226.6091    182.653   1086.7502    11.0937  -.55555554E-10
 95.0000   227.5257    182.555   1087.1857    10.7862
 95.0010   227.5259    182.555   1087.1857    10.7862  -.55555554E-10
100.0000   228.3976    182.464   1087.5724    10.4922
100.0010   228.3977    182.464   1087.5725    10.4922  -.55555554E-10
```

APPENDIX A

The following paragraphs will outline the equations programmed for the various subprograms. These assume steady state conditions on the imaginary axis ($S = j\omega$) with linear, lumped, bilateral, passive elements. The program processes basic sections starting with that section nearest the output termination. Regardless of section type, this section will have Z_A and Z_B impedances.

The DRPTZ subprogram calculates the impedances and admittances for a single order as in Figure 8 for both the Z_A and Z_B impedance.

The number of parallel branches, or resonators as they are called in the Westinghouse report, allowed is a function of available core storage. As written, three resonators per order are allowed although this could be increased by reallocating storage, i.e., reducing the number of orders per basic section impedance or the number of basic sections. The impedance for a single branch or resonator is:

$$Z_1 = R_1 + j\omega L_1 + \frac{1}{j\omega C_1} = R_1 + j(\omega L_1 - \frac{1}{\omega C_1})$$

$$Y_1 = 1/Z_1 = \frac{1}{R_1 + j(\omega L_1 - \frac{1}{\omega C_1})}$$

$$= \frac{R_1 + j(\frac{1}{\omega C_1} - \omega L_1)}{R_1^2 + (\omega L_1 - \frac{1}{\omega C_1})^2}$$

$$Y_1 = YRL_1 + jYIM_1$$

$$YRL_1 = \frac{R_1}{R_1^2 + (\omega L_1 - \frac{1}{\omega C_1})^2}$$

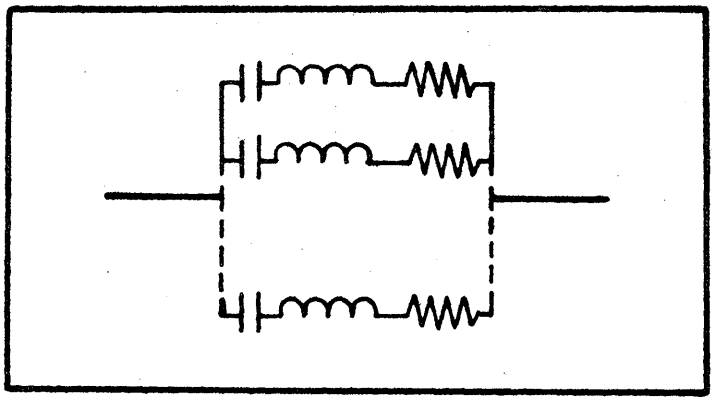


FIGURE 8 - A SINGLE ORDER

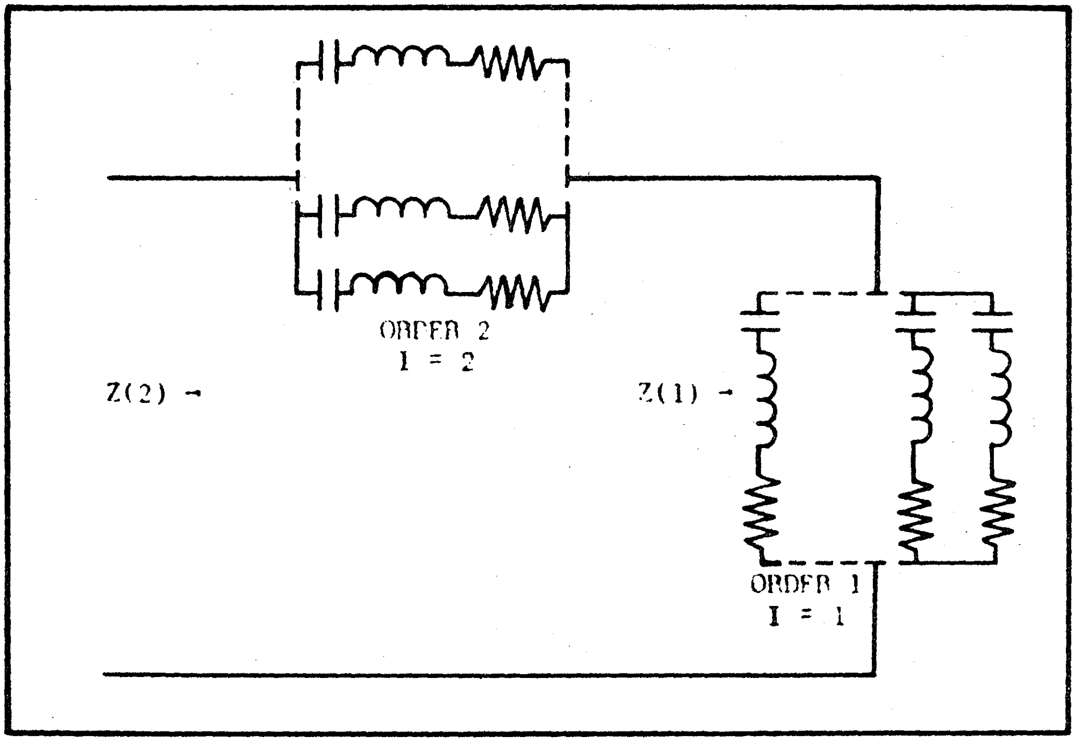


FIGURE 9 - TWO ORDERS OF A BASIC SECTION IMPEDANCE

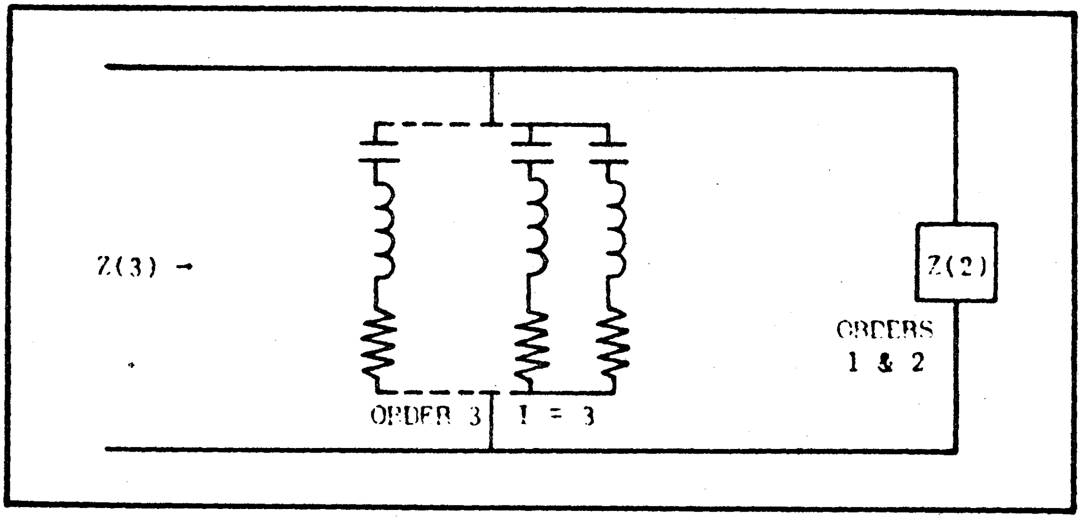


FIGURE 10 - THREE ORDERS OF A BASIC SECTION IMPEDANCE

$$YIM_1 = \frac{\frac{1}{\omega C_1} - \omega L_1}{R_1^2 + (\omega L_1 - \frac{1}{\omega C_1})^2}$$

Summing admittances for all k parallel branches in an order gives:

$$YRL_T = \sum_{i=1}^{k} YRL_i$$

$$YIM_T = \sum_{i=1}^{k} YIM_i$$

$$Y_T = YRL_T + jYIM_T$$

$$Z_T = 1/Y_T = ZRL_T + jZIM_T.$$

The DRPTZ subprogram stores values of ZRL_T, and ZIM_T impedances. Program control is then given to the ORDER subprogram.

The ORDER subprogram first stores the impedance values just computed by DRPTZ as subscripted impedances Z(I), with the order counter I set at 1. Control is returned to the main line which determines if either the Z_A or Z_B basic section impedance consists of more than one order. If not, control is given to the proper recursion subprogram. If there are more than one order in either Z_A or Z_B, the counter is incremented to 2 and control is returned to DRPTZ.

The DRPTZ subprogram then zeros the variables YRL_T and YIM_T and repeats the process described above using the element values for the second orders of the appropriate basic section impedances. Control is then given to the ORDER subprogram.

The ORDER subprogram determines if the order counter I equals 2 or 3. If I is 2, ORDER simply sums the impedances Z_T just computed by DRPTZ with those previously stored as subscripted impedances when I equalled 1. Figure 9 illustrates the situation.

$$Z(1) = ZTRL(1) + jZTIM(1)$$

$$Z(2) = Z(1) + Z_T = ZTRL(1) + jZTIM(1) + ZRL_T + jZIM_T = ZTRL(2) + jZTIM(2)$$

$$ZTRL(2) = ZTRL(1) + ZRL_T$$

$$ZTIM(2) = ZTIM(1) + ZIM_T$$

If the order counter I is 3, the ORDER subprogram has the situation in Figure 10. DRPTZ again has computed the impedance Z_T and admittance Y_T for the third order.

$$Y_T = YRL_T + jYIM_T \text{ (for Order 3)}$$

$$Y(3) = 1/Z(3) = \frac{1}{Z(2)} + Y_T = \frac{1}{ZTRL(2) + jZTIM(2)} + Y_T$$

$$= \frac{ZTRL(2) = jZTIM(2)}{A} + YRL_T + j\, YIM_T$$

where $A = \left[ZTRL(2)\right]^2 + \left[ZTIM(2)\right]^2$

$$Y(3) = \frac{1}{A}\left\langle\left[ZTRL(2) + A(YRL_T)\right] + j\left[A(YIM_T) - ZTIM(2)\right]\right\rangle$$

Let $B = ZTRL(2) + A(YRL_T)$

$C = A(YIM_T) - ZTIM(2)$

$$Z(3) = 1/Y(3) = \frac{A}{B + jC} = \frac{A(B - jC)}{B^2 + C^2}$$

Control is returned to the main line. If there are no further orders to be processed in the particular basic section impedances, control is given to the appropriate recursion subprogram LADDER or LATTICE.

The LADDER subprogram has the situation in Figure 11. As mentioned previously computation proceeds from right to left. The first step is to pick up the complex values of impedance just computed by the DRPTZ and ORDER subprograms, and the voltage and impedance existing at the output of the particular section being considered. If the first section to the left of the output termination is being processed, these latter two values are V_o and R_o. In the general case they are V_{K-1} and Z_{K-1}.

$$V_{K-1} = VRL_{K-1} + jVIM_{K-1}$$

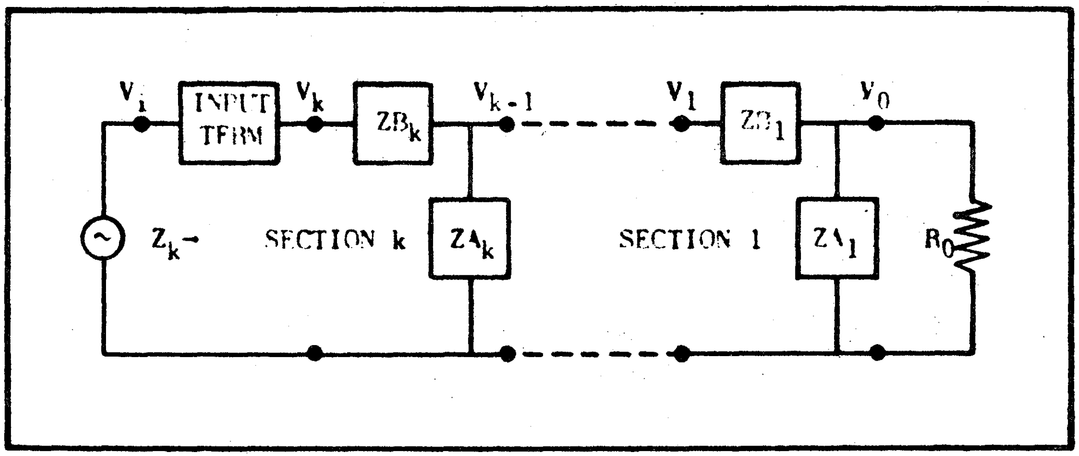


FIGURE 11 - CASCADED LADDER SECTIONS

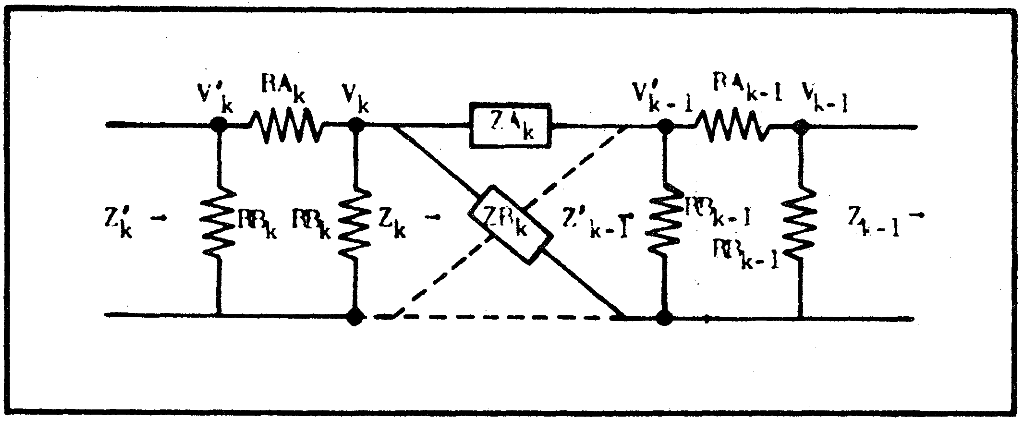


FIGURE 12 - SYMMETRICAL LATTICE SECTION

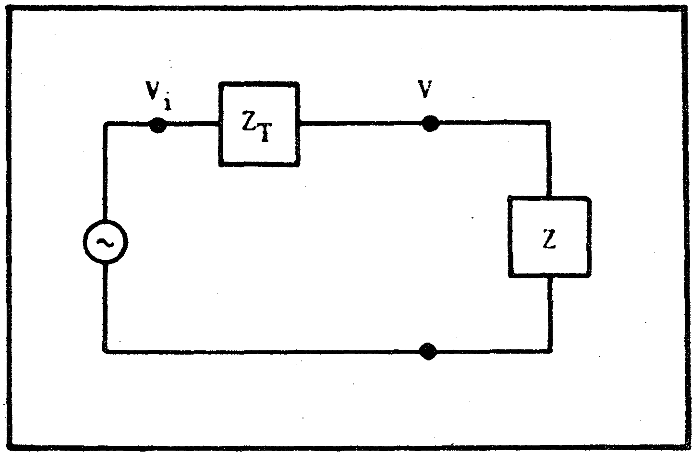


FIGURE 13 - FILTER NETWORK Z AND INPUT TERMINATION Z_T

$$Z_{K-1} = ZRL_{K-1} + jZIM_{K-1}$$

Note that ZA_K and ZB_K are equal to the appropriate Z(I) values determined in DRPTZ and ORDER where I is the maximum order of the kth section.

$$Z_K = ZB_K + \frac{ZA_K \, Z_{K-1}}{ZA_K + Z_{K-1}} = ZRL_K + jZIM_K$$

$$\frac{V_K - V_{K-1}}{ZB_K} = V_{K-1} \left(\frac{1}{ZA_K} + \frac{1}{Z_{K-1}} \right)$$

$$V_K = V_{K-1} \left\{ 1 + ZB_K \left(\frac{1}{ZA_K} + \frac{1}{Z_{K-1}} \right) \right\}$$

The LATTICE subprogram has the situation presented in Figure 12. It will be noted that symmetrical resistive pi sections are placed at each end of the basic symmetrical lattice section to permit a definite amount of attenuation or isolation between sections. This feature can effectively be eliminated by inserting zero values for the series elements R_B and very large values for the shunt R_A elements. Again ZA_K and ZB_K have been computed by the DRPTZ and ORDER subprograms and are equal to the appropriate Z(I) values where I is the maximum order of the Kth section. The recursion equations for the LATTICE subprogram follow. Computation proceeds from right to left as before. Note that the following equations hold for Fig. 12, in which the pi section shunt RA and series RB have been interchanged.

$$Z'_{K-1} = \frac{RB_{K-1} \left\{ RA_{K-1} + \frac{(RB_{K-1})\,(Z_{K-1})}{RB_{K-1} + Z_{K-1}} \right\}}{RB_{K-1} + RA_{K-1} + \frac{(RB_{K-1})\,(Z_{K-1})}{(RB_{K-1} + Z_{K-1})}}$$

$$V'_{K-1} = V_{K-1} \left\{ 1 + RA_{K-1} \left(\frac{1}{RB_{K-1}} + \frac{1}{Z_{K-1}} \right) \right\}$$

$$Z_K = \frac{Z'_{K-1} (ZA_K + ZB_K) + 2(ZA_K)(ZB_K)}{2 Z'_{K-1} + ZA_K + ZB_K}$$

$$V_K = \frac{V'_{K-1} \left\langle Z'_{K-1} (ZA_K + ZB_K) + 2(ZA_K)(ZB_K) \right\rangle}{Z'_{K-1} (ZB_K - ZA_K)}$$

$$Z'_K = \frac{(1/Z_K + 1/RB_K) RA_K + 1}{(1/RB_K) \left\langle (1/Z_K + 1/RB_K) RA_K + 1 \right\rangle + 1/Z_K + 1/RB_K}$$

$$V'_K = V_K \left\langle (1/Z_K + 1/RB_K) RA_K + 1 \right\rangle$$

The procedures described are repeated section-by-section until the last filter section before the input termination has been processed. The impedance of the entire filter plus the output termination is stored for subsequent output as ZREAL and ZIMAG. The Termin subprogram is called to monitor calculation of the input termination impedance and input voltage V_i required to produce the specified output voltage V_o. The subprogram uses the DRPTZ subprogram to determine the single-ordered input termination impedance, subsequently computing V_i. Figure 13 illustrates the situation with Z representing the input impedance of the filter plus the output resistance and Z_T the input termination. V is the voltage at the filter input.

$$\frac{V_i - V}{Z_T} = \frac{V}{Z}$$

$$\frac{V_i}{V} = \frac{Z_T}{Z} + 1$$

$$V_i = V \left(\frac{Z_T}{Z} + 1\right) = \frac{V}{Z} (Z_T + Z)$$

Z_T is stored as follows:

$$Z_T = RK + jXK$$

and control is returned to the main line which calls the BETADB subprogram.

BETADB calculates the phase angle β of the voltage V_i and the insertion loss A. These equations follow:

$$V_i = VPRL + jVPIM$$

$$\beta = \tan^{-1} (VPIM/VPRL)$$

$$A = 20 \log_{10} \left| \frac{R_o}{(RK + R_o + jXK)} \cdot \frac{V_i}{V_o} \right|$$

β is reported as a positive angle.

Following calculation of the voltage angle and insertion loss by the BETADB subprogram, control is returned to the main line which outputs the following information:

Frequency (in megacycles)
Insertion loss (in decibels)
Input voltage angle (in degrees)
Filter Impedance including output termination
(in ohms - rectangular form).

This completes one full pass thru the program.

Time delay is approximated by repeating the complete procedure with the frequency incremented by 1000 cycles, finding the change in input voltage angle, and dividing by a constant proportional to the increment in frequency.

$$T = \frac{\Delta\beta}{1000\ C}$$

As written, C = 360. This result approximates the slope of the phase versus frequency curve at the frequency point 500 cycles above the base frequency. The mainline then outputs the same information as above at the incremented frequency, as well as T. The frequency value is decremented by 1000 cycles, and tested against the maximum frequency value, FMAX. If less than that value, frequency is incremented by FDEL and the entire procedure repeated. Eventually frequency reaches FMAX and the program returns to its beginning point ready to read in another title card and complete set of frequency and element cards representing a new filter. If there are no cards in the input hopper the Reader-No-Feed light is turned on.

```
C     GENERAL FILTER NETWORK ANALYSIS PROGRAM FOR LADDER AND LATTICE
C          SECTIONS.
C     Z = COMPLEX DRIVING - POINT IMPEDANCE FUNCTION
C     Y = COMPLEX DRIVING - POINT ADMITTANCE FUNCTION
C     SUBSCRIPT (LKI) = (L) STANDARDIZED POSITION OF (K)
C       SECTION OF (I) ORDER
C     PROGRAM BASED UPON WESTINGHOUSE REPORT TITLED ELECTRONIC DIGITAL
C          COMPUTER ANALYSIS OF CASCADED NETWORKS BY RICHARD TUZNIK
C          AND DEAN H. WOOD.
C     PROGRAM CAN BE EXPANDED TO HANDLE PI, TEE, AND BRIDGED-TEE
C          SECTIONS ADDING APPROPRIATE SUBPROGRAMS AND MAKING
C          NECESSARY REVISIONS IN EXISTING ROUTINES.
C
      DIMENSION C(6,5,3), R(6,5,3), AL(6,5,3), MP(5,3), MQ(5,3),
     1RA(5), RB(5), IORDR(5), ZAKRL(3), ZBKRL(3), ZAKIM(3), ZBKIM(3)
      COMMON C, R, AL, MP, MQ, RA, RB, ZAKRL, ZBKRL, ZAKIM, ZBKIM, YARL,
     1YBRL, YAIM, YBIM, ZARL, ZBRL, ZAIM, ZBIM, F, F2, IREC, ZPRL, ZPIM,
     2VPRL, VPIM, ZRL, ZIM, VRL, VIM, K, I, IPUNC, RAO, RBO, DEN, L,
     3INDX, REAL, AIMG, RLDEN, ANMIN, RLNUM, DENIM, A, B, QUAN, IORDR,
     4LETTR, VAR1, IORD, ZARLK, ZBRLK, ZAIMK, ZBIMK,NSECT, NT
      COMMON XK, RK, ZOR, ZOI, VOR, VOI, BETA, DB, XIMAG
      PI = 3.14159265
      PI2 = 6.2831853
      CONST = 1.0/360000.
      CONV = 180.0/PI
  100 FORMAT (/32HGENERAL NETWORK ANALYSIS PROGRAM/)
  101 FORMAT (6F11.5)
  102 FORMAT (3I5)
```

```
  103 FORMAT (3E14.8)
C
C
C
   12 READ 116
C     READ TITLE CARD WITH UP TO 80 CHARACTERS.
      PUNCH 100
      PUNCH 116
  116 FORMAT(40H                                        ,40H
   ' 1
C
C
C
      READ 101, FMIN, FDEL, FMAX, ZOR, VOR, VOI
C     ZOR IS OUTPUT TERMINAL RESISTANCE IN OHMS, FMIN, FDEL, AND
C          FMAX ARE FREQUENCIES IN MC.
C     VOR AND VOI ARE OUTPUT TERMINAL VOLTAGES. FOR REFERENCE,
C          VOR = 1.0, VOI = 0.0
C
C
C
      READ 102, NSECT, LETTR, NCYCL
C     NSECT IS NUMBER OF INVERTED L-SECTIONS OR SYMMETRICAL LATTICE
C          SECTIONS.
C     LETTR SPECIFIES TYPE OF BASIC SECTION, = 1 FOR SYM LATTICE,
C          = 2 FOR LADDER.
C     NCYCL = NO OF TIMES GROUP OF SECTIONS IS USED BEFORE PROCEEDING
C          TO FINAL PORTION OF PROGRAM.
```

```
C
C
C
      GO TO (45,54), LETTR
   45 READ 103, RAO, RBO
C     READ INITIAL PI PAD VALUES, RAO IS SHUNT BRANCH, RBO IS SERIES
C          BRANCH. THESE VALUES ARE REQUIRED FOR LATTICE SECTIONS ONLY.
C
C
C     PUNCH OUT INPUT VALUES
C
   55 PUNCH 105, ZOR, VOR, VOI, RAO, RBO
  105 FORMAT (/4HZOR=,F9.3, 3X4HVOR=,F9.3, 3X4HVOI=,F9.3/ 4HRAO=,E14.8,
     13X4HRBO=,E14.8/)
      GO TO 56
   54 PUNCH 104, ZOR, VOR, VOI
  104 FORMAT (/4HZOR=,F9.3, 3X4HVOR=,F9.3, 3X4HVOI=,F9.3/)
   56 PUNCH 111, NSECT, LETTR, NCYCL
  111 FORMAT (9HSECTIONS=,I4, 3X6HLETTR=,I4, 3X6HNCYCL=,I4)
      NSECT = NSECT+1
      DO 2 K = 1, NSECT
C
C
      READ 102, IORDR(K)
C     IORDR IS NUMBER OF ORDERS PER SECTION IMPEDANCE ARM OF A SECTION.
C          IF SECTION IMPEDANCE ARMS HAVE DIFFERENT ORDERS, USE LARGER
C          VALUE.
C
```

```
C
C     DO NOT READ PI PAD VALUES FOLLOWING TERMINATION SECTION.
      GO TO (47,48),LETTR
   47 IF(NSECT-K) 48, 48, 64
C
C
   64 READ 103, RA(K), RB(K)
C     RA AND RB ARE PI PAD VALUES FOLLOWING SECTION K. RA - SHUNT
C          BRANCH.  RB - SERIES BRANCH.
C
C
      PUNCH 106, K, RA(K), K, RB(K), IORDR(K)
      GO TO 49
   48 PUNCH 110, K, IORDR(K)
  110 FORMAT (/5HORDER, I4, 2HIS, I4/)
  106 FORMAT(4HR(1,,I2,5H ) = ,E14.8,2X4HR(2,,I2,5H ) = ,E14.8,2X,I2,
     17H ORDERS)
   49 IORD = IORDR(K)
      DO 2 I = 1, IORD
C
C
      READ 102, MP(K,I), MQ(K,I)
C     MQ = NO OF RESONATORS IN (I)TH ORDER IMPEDANCE ZB IN (K)TH SECTION
C     MP = NO OF RESONATORS IN (I)TH ORDER IMPEDANCE ZA IN (K)TH SECTION
C     RESONATORS ARE NUMBER OF PARALLEL BRANCHES.
C
C
      PUNCH 112, K, I, MP(K,I), MQ(K,I)
```

```
  112 FORMAT (2HK=, I4, 3X2HI=, I4, 3X3HMP=, I4, 3X3HMQ=, I4)
      IREC = MP(K,I)
      L = 1
    9 IF (IREC-L) 7, 8, 8
    8 INDX = 2*L-1
C
C
      READ 103, C(INDX,K,I), AL(INDX,K,I), R(INDX,K,I)
C     READ C(FARADS),L(HENRIES),AND R(OHMS) VALUES OF ZA TYPE IMPEDANCE
C          WHICH IS PARALLEL ARM FOR INVERTED-L SECTIONS AND IS SERIES
C          ARM FOR LATTICE SECTIONS.
C
C
      PUNCH 107, INDX, K, I, C(INDX,K,I), AL(INDX,K,I), R(INDX,K,I)
  107 FORMAT (2HC(, I2, 2H ,, I2, 2H ,, I2, 5H ) = , E14.8, 2X4HL = ,
     1E14.8, 2X4HR = , E14.8)
      L = L+1
      GO TO 9
    7 IREC = MQ(K,I)
      L = 1
   10 IF (IREC-L) 2, 11, 11
   11 INDX = 2*L
C
C
      READ 103, C(INDX,K,I), AL(INDX,K,I), R(INDX,K,I)
C     READ C(FARADS),L(HENRIES),AND R(OHMS) OF ZB TYPE IMPEDANCE WHICH
C          IS SERIES ARM FOR INVERTED-L SECTIONS AND IS PARALLEL ARM FOR
C          LATTICE SECTIONS.
```

```
C
C
      PUNCH 107, INDX, K, I, C(INDX,K,I), AL(INDX,K,I), R(INDX,K,I)
      L = L+1
      GO TO 10
    2 CONTINUE
      FCOUN = FMIN
      KKLL = 1
      JFREQ = 1
      NSECT = NSECT-1
   20 ZRL = ZOR
      ZOI=0.
      ZIM = 0.0
      VRL = VOR
      VIM = VOI
      IREC = 0
      K = 1
      I = 1
      N = 1
      GO TO (42,43),LETTR
C
C     CALCULATE DROP FOR PI PAD PRECEDING FIRST SECTION.
C
   42 CALL LATTIS
   43 IREC = 2
      F = (FCOUN*1.0E+6)*PI2
      F2 = F*F
      N = 1
```

```
      K = 1
   25 IORD = IORDR(K)
      I = 1
   22 CALL DRPTZ
      CALL ORDER
      IF(IORD-I) 3, 3, 21
   21 I = I+1
      GO TO 22
    3 GO TO (36,37),LETTR
   36 CALL LATTIS
      GO TO 38
   37 CALL LADDER
   38 IF(NSECT-K) 27, 27, 24
   24 K = K+1
      GO TO 25
   27 IF(NCYCL-N) 30, 30, 29
   29 N = N+1
      K = 1
      GO TO 25
   30 CONTINUE
      GO TO (99,44), LETTR
   99 ZRLN = ZPRL
      ZIMN = ZPIM
      GO TO (97,98), KKLL
   44 ZRLN = ZRL
      ZIMN = ZIM
      GO TO (97,98), KKLL
   97 PUNCH 109
```

```
      KKLL = 2
C
C
C
   98 K = NSECT+1
C     SET  K AND I VALUES PERTAINING TO INPUT TERMINAL IMPEDANCE. K WILL
C          BE EQUAL TO NSECT+1, I = 1, WHERE THE IMPEDANCE IS CONSIDERED
C          ZA TYPE.
C
      I = 1
      CALL TERMIN
   32 GO TO (33,34), JFREQ
C
   33 JFREQ = 2
      CALL BETADB
C     CALCULATE DB AND BETA.
C
C
      BETA1 = BETA
C     ZRLN AND ZIMN ARE IMPEDANCE VALUES OF INSERTED NETWORK WITH OUTPUT
C          TERMINATION ZOR. THESE INCLUDE PI PADS FOR LATTICE SECTIONS.
      PUNCH 108, FCOUN, DB, BETA, ZRLN, ZIMN
C
C     INCREMENT F BY 1000 CYCLES AND REPEAT ENTIRE PROCEDURE IN ORDER
C          TO OBTAIN TIME DELAY.
C
      FCOUN = FCOUN+0.001
```

```
      GO TO 20
   34 CALL BETADB
      T = (BETA-BETA1)*CONST
      JFREQ = 1
      PUNCH 108, FCOUN, DB, BETA, ZRLN, ZIMN, T
C
C     DECREMENT FREQUENCY BY 1000 CYCLES.
C
      FCOUN = FCOUN-0.001
C
C     IS FREQUENCY AT MAXIMUM VALUE.
C
      IF(FMAX-FCOUN) 12, 12, 35
C
C     NO. INCREMENT BY AMOUNT AT INPUT AND REPEAT CALCULATIONS.
C
   35 FCOUN = FCOUN+FDEL
      GO TO 20
C
C     YES. GO TO START TO READ IN NEXT COMPLETE FILTER PROBLEM.
C
  108 FORMAT (F9.4, 2XF9.4, 2XF10.3, 2XF10.4, 2XF10.4, 2XE14.8)
  109 FORMAT (/4X1HF, 10X2HDB, 8X4HBETA, 8X5HZREAL, 7X5HZIMAG,9X4HT(F)/)
      END
```

```
      SUBROUTINE BETADB
C     CALCULATES BETA(RADIANS) AND DB FOR FREQUENCY
      DIMENSION C(6,5,3), R(6,5,3), AL(6,5,3), MP(5,3), MQ(5,3),
     1RA(5), RB(5), IORDR(5), ZAKRL(3), ZBKRL(3), ZAKIM(3), ZBKIM(3)
      COMMON C, R, AL, MP, MQ, RA, RB, ZAKRL, ZBKRL, ZAKIM, ZBKIM, YARL,
     1YBRL, YAIM, YBIM, ZARL, ZBRL, ZAIM, ZBIM, F, F2, IREC, ZPRL, ZPIM,
     2VPRL, VPIM, ZRL, ZIM, VRL, VIM, K, I, IPUNC, RAO, RBO, DEN, L,
     3INDX, REAL, AIMG, RLDEN, ANMIN, RLNUM, DENIM, A, B, QUAN, IORDR,
     4LETTR, VAR1, IORD, ZARLK, ZBRLK, ZAIMK, ZBIMK,NSECT, NT
      COMMON XK, RK, ZOR, ZOI, VOR, VOI, BETA, DB, XIMAG
      PI = 3.14159265
      PI2 = 6.2831853
      CONV = 180.0/PI
      BETA = (ATANF(VPIM/VPRL))*CONV
C
C     MAKE BETA POSITIVE ANGLE BETWEEN 0 AND 360 DEGREES.
C
      IF(VPRL) 1, 2, 2
    1 BETA = BETA+180.0
      GO TO 4
    2 IF(BETA) 3, 4, 4
    3 BETA = BETA+360.0
    4 RNUM = ZOR*VPRL-ZOI*VPIM
      XIMAG = VPRL*ZOI+ZOR*VPIM
      XNUM = SQRTF(RNUM*RNUM+XIMAG*XIMAG)
      RLDEN = RK*VOR+ZOR*VOR-XK*VOI-ZOI*VOI
      XIMAD = XK*VOR+ZOI*VOR+RK*VOI+ZOR*VOI
      DEN = SQRTF(RLDEN*RLDEN+XIMAD*XIMAD)
```

260

```
DB = 2.0*4.3429448*LOGF(XNUM /DEN)
RETURN
END
```

```
      SUBROUTINE LATTIS
C     RECURSION FORMULAE FOR SYMMETRICAL LATTICE SECTION
C     CALCULATES ZRL, ZIM, ZPRL, ZPIM, VRL, VIM, VPRL, VPIM
      DIMENSION C(6,5,3), R(6,5,3), AL(6,5,3), MP(5,3), MQ(5,3),
     1RA(5), RB(5), IORDR(5), ZAKRL(3), ZBKRL(3), ZAKIM(3), ZBKIM(3)
      COMMON C, R, AL, MP, MQ, RA, RB, ZAKRL, ZBKRL, ZAKIM, ZBKIM, YARL,
     1YBRL, YAIM, YBIM, ZARL, ZBRL, ZAIM, ZBIM, F, F2, IREC, ZPRL, ZPIM,
     2VPRL, VPIM, ZRL, ZIM, VRL, VIM, K, I, IPUNC, RAO, RBO, DEN, L,
     3INDX, REAL, AIMG, RLDEN, ANMIN, RLNUM, DENIM, A, B, QUAN, IORDR,
     4LETTR, VAR1, IORD, ZARLK, ZBRLK, ZAIMK, ZBIMK,NSECT, NT
      COMMON XK, RK, ZOR, ZOI, VOR, VOI, BETA, DB, XIMAG
      REAL = 1.0/RAO
      AIMG = RBO
C
C     IF FIRST SECTION, GO TO 4022 TO CALCULATE INTIIAL PI PAD DROP.
C          COME BACK TO CALCULATE FIRST SECTION AND FOLLOWING PI
C          PAD DROP.
C
      IF(IREC) 4021, 4022, 4021
 4021 ZARLK = ZAKRL(I)
      ZBRLK = ZBKRL(I)
      ZAIMK = ZAKIM(I)
      ZBIMK = ZBKIM(I)
      REAL = ZARLK+ZBRLK
      AIMG = ZAIMK+ZBIMK
      RLNUM =  ZPRL*REAL- ZPIM*AIMG
      RLNUM = RLNUM+2.0*(ZARLK*ZBRLK-ZAIMK*ZBIMK)
      RLDEN = 2.0* ZPRL+REAL
```

```
      DENIM = 2.0* ZPIM+AIMG
      ANMIN =  ZPIM*REAL+ ZPRL*AIMG
      ANMIN = ANMIN+2.0*(ZARLK*ZBIMK+ZAIMK*ZBRLK)
      DEN = 1.0/(RLDEN*RLDEN+DENIM*DENIM)
      ZRL = (RLNUM*RLDEN+DENIM*ANMIN)*DEN
      ZIM = (RLDEN*ANMIN-RLNUM*DENIM)*DEN
      REAL = ZBRLK-ZARLK
      AIMG = ZBIMK-ZAIMK
      RLDEN =  ZPRL*REAL- ZPIM*AIMG
      DENIM =  ZPIM*REAL+ ZPRL*AIMG
      DEN = 1.0/(RLDEN*RLDEN+DENIM*DENIM)
      A = VPRL*RLNUM-VPIM*ANMIN
      B = VPIM*RLNUM+VPRL*ANMIN
      VRL = (A*RLDEN+DENIM*B)*DEN
      VIM = (RLDEN*B-A*DENIM)*DEN
      REAL = 1.0/RA(K)
      AIMG = RB(K)
 4022 DEN = 1.0/(ZRL*ZRL+ZIM*ZIM)
      RLNUM = 1.0+REAL*AIMG+AIMG*ZRL*DEN
      ANMIN = AIMG*ZIM*DEN
      RLDEN = 2.0*REAL+AIMG*REAL*REAL+(ZRL*REAL*AIMG+ZRL)*DEN
      DENIM = (REAL*AIMG*ZIM+ZIM)*DEN
      DEN    = 1.0/(RLDEN*RLDEN+DENIM*DENIM)
      ZPRL = (RLNUM*RLDEN+DENIM*ANMIN)*DEN
      ZPIM = (-RLDEN*ANMIN+RLNUM*DENIM)*DEN
      DEN = 1.0/ (ZRL*ZRL + ZIM*ZIM)
      VPRL = VRL+VRL*REAL*AIMG+(VRL*AIMG*ZRL+VIM*ZIM*AIMG)*DEN
      VPIM = VIM+VIM*AIMG*REAL+(VIM*AIMG*ZRL-VRL*ZIM*AIMG)*DEN
```

```
4002 RETURN
     END
```

```
      SUBROUTINE ORDER
C     COMPUTES TOTAL COMPLEX IMPEDANCE FOR BASIC SECTION IMPEDANCES
C             ZA AND ZB FROM INDIVIDUAL ORDERS.
      DIMENSION C(6,5,3), R(6,5,3), AL(6,5,3), MP(5,3), MQ(5,3),
     1RA(5), RB(5), IORDR(5), ZAKRL(3), ZBKRL(3), ZAKIM(3), ZBKIM(3)
      COMMON C, R, AL, MP, MQ, RA, RB, ZAKRL, ZBKRL, ZAKIM, ZBKIM, YARL,
     1YBRL, YAIM, YBIM, ZARL, ZBRL, ZAIM, ZBIM, F, F2, IREC, ZPRL, ZPIM,
     2VPRL, VPIM, ZRL, ZIM, VRL, VIM, K, I, IPUNC, RAO, RBO, DEN, L,
     3INDX, REAL, AIMG, RLDEN, ANMIN, RLNUM, DENIM, A, B, QUAN, IORDR,
     4LETTR, VAR1, IORD, ZARLK, ZBRLK, ZAIMK, ZBIMK,NSECT, NT
      COMMON XK, RK, ZOR, ZOI, VOR, VOI, BETA, DB, XIMAG
      IVAL = I-1
      IF(IVAL) 5001, 5002, 5001
 5002 ZAKRL(I) = ZARL
      ZAKIM(I) = ZAIM
      ZBKRL(I) = ZBRL
      ZBKIM(I) = ZBIM
      GO TO 5006
 5001 IF(I-(I/2)*2) 5007, 5004, 5007
 5004 ZAKRL(I) = ZARL+ZAKRL(IVAL)
      ZAKIM(I) = ZAIM+ZAKIM(IVAL)
      ZBKRL(I) = ZBRL+ZBKRL(IVAL)
      ZBKIM(I) = ZBIM+ZBKIM(IVAL)
      GO TO 5006
 5007 RLNUM = ZAKRL(IVAL)
      DENIM = ZAKIM(IVAL)
      QUAN =RLNUM*RLNUM+DENIM*DENIM
      B = YARL*QUAN+RLNUM
```

```
      A = YAIM*QUAN-DENIM
      DEN = 1.0/(B*B+A*A)
      ZAKRL(I) = B*QUAN*DEN
      ZAKIM(I) = -A*QUAN*DEN
      RLNUM = ZBKRL(IVAL)
      DENIM = ZBKIM(IVAL)
      QUAN =RLNUM*RLNUM+DENIM*DENIM
      B = YBRL*QUAN+RLNUM
      A = YBIM*QUAN-DENIM
      DEN = 1.0/(B*B+A*A)
      ZBKRL(I) = B*QUAN*DEN
      ZBKIM(I) = -A*QUAN*DEN
 5006 RETURN
      END
```

```
      SUBROUTINE TERMIN
C     CALCULATES IMPEDANCE AND VOLTAGE AT INPUT TO INPUT TERMINATION.
      DIMENSION C(6,5,3), R(6,5,3), AL(6,5,3), MP(5,3), MQ(5,3),
     1RA(5), RB(5), IORDR(5), ZAKRL(3), ZBKRL(3), ZAKIM(3), ZBKIM(3)
      COMMON C, R, AL, MP, MQ, RA, RB, ZAKRL, ZBKRL, ZAKIM, ZBKIM, YARL,
     1YBRL, YAIM, YBIM, ZARL, ZBRL, ZAIM, ZBIM, F, F2, IREC, ZPRL, ZPIM,
     2VPRL, VPIM, ZRL, ZIM, VRL, VIM, K, I, IPUNC, RAO, RBO, DEN, L,
     3INDX, REAL, AIMG, RLDEN, ANMIN, RLNUM, DENIM, A, B, QUAN, IORDR,
     4LETTR, VAR1, IORD, ZARLK, ZBRLK, ZAIMK, ZBIMK,NSECT, NT
      COMMON XK, RK, ZOR, ZOI, VOR, VOI, BETA, DB, XIMAG
      GO TO (5,6), LETTR
    5 ZTRL = ZPRL
      ZTIM = ZPIM
      GO TO 7
    6 ZTRL = ZRL
      ZTIM = ZIM
C
C     USE DRPTZ TO CALCULATE INPUT TERMINATION IMPEDANCE.
C
    7 CALL DRPTZ
      ZBTRL = ZTRL+ZARL
      ZBTIM = ZTIM+ZAIM
      RK = ZARL
      XK = ZAIM
      REAL = VPRL*ZBTRL-VPIM*ZBTIM
      XIMAG= VPIM*ZBTRL+VPRL*ZBTIM
      DEN = ZTRL*ZTRL+ZTIM*ZTIM
      IF(DEN) 1, 2, 1
```

```
  , 2 PUNCH 101
  101 FORMAT (41HERROR,DENOMINATOR IN TERMIN EQUAL TO ZERO)
      PAUSE
C
C     CALCULATE INPUT VOLTAGE REQUIRED TO PRODUCE SPECIFIED OUTPUT
C         TERMINATION VOLTAGE.
C
    1 VKRL = (REAL*ZTRL+XIMAG*ZTIM)/DEN
      VKIM = (ZTRL*XIMAG-ZTIM*REAL)/DEN
      VPRL = VKRL
      VPIM = VKIM
      RETURN
      END
```

```
      SUBROUTINE LADDER
      DIMENSION C(6,5,3), R(6,5,3), AL(6,5,3), MP(5,3), MQ(5,3),
     1RA(5), RB(5), IORDR(5), ZAKRL(3), ZBKRL(3), ZAKIM(3), ZBKIM(3)
      COMMON C, R, AL, MP, MQ, RA, RB, ZAKRL, ZBKRL, ZAKIM, ZBKIM, YARL,
     1YBRL, YAIM, YBIM, ZARL, ZBRL, ZAIM, ZBIM, F, F2, IREC, ZPRL, ZPIM,
     2VPRL, VPIM, ZRL, ZIM, VRL, VIM, K, I, IPUNC, RAO, RBO, DEN, L,
     3INDX, REAL, AIMG, RLDEN, ANMIN, RLNUM, DENIM, A, B, QUAN, IORDR,
     4LETTR, VAR1, IORD, ZARLK, ZBRLK, ZAIMK, ZBIMK,NSECT, NT
      COMMON XK, RK, ZOR, ZOI, VOR, VOI, BETA, DB, XIMAG
C     COMPUTES RECURSION EQUATIONS FOR LADDER SECTIONS.
C     CALCULATES ZRL, ZIM, ZPRL, ZPIM, VRL, VIM, VPRL, VPIM.
      ZARLK = ZAKRL(I)
      ZAIMK = ZAKIM(I)
      ZBRLK = ZBKRL(I)
      ZBIMK = ZBKIM(I)
      VPRL = VRL
      VPIM = VIM
      RLDEN = ZARLK+ZRL
      DENIM = ZAIMK+ZIM
      DEN = RLDEN*RLDEN+DENIM*DENIM
      RLNUM = ZARLK*ZRL-ZAIMK*ZIM
      XIMAG = ZAIMK*ZRL+ZIM*ZARLK
      RNUM = RLNUM*RLDEN+XIMAG*DENIM
      XMAG = XIMAG*RLDEN-RLNUM*DENIM
      IF(DEN) 1, 2, 1
    2 PUNCH 101
  101 FORMAT (32HERROR,ZERO DENOMINATOR IN LADDER)
      PAUSE
```

```
1 ZPRL = RNUM/DEN
  ZPIM = XMAG/DEN
  ZRL = ZPRL+ZBRLK
  ZIM = ZPIM+ZBIMK
  DEN = ZPRL*ZPRL+ZPIM*ZPIM
  IF(DEN) 3, 2, 3
3 REAL = 1.0+(ZBRLK*ZPRL+ZBIMK*ZPIM)/DEN
  XIMAG = (ZBIMK*ZPRL-ZPIM*ZBRLK)/DEN
  VRL = VPRL*REAL-VPIM*XIMAG
  VIM = VPRL*XIMAG+VPIM*REAL
  VPRL = VRL
  VPIM = VIM
  RETURN
  END
```

```
      SUBROUTINE DRPTZ
C     CALCULATES IMPEDANCE AND ADMITTANCE FOR A TWO-TERMINAL IMPEDANCE
C          (ORDER).  IE. YARL, YAIM, YBRL, YBIM, ZARL, ZAIM, ZBRL, ZBIM
C       IE. YARL, YAIM, YBRL, YBIM, ZARL, ZAIM, ZBRL, ZBIM
      DIMENSION C(6,5,3), R(6,5,3), AL(6,5,3), MP(5,3), MQ(5,3),
     1RA(5), RB(5), IORDR(5), ZAKRL(3), ZBKRL(3), ZAKIM(3), ZBKIM(3)
      COMMON C, R, AL, MP, MQ, RA, RB, ZAKRL, ZBKRL, ZAKIM, ZBKIM, YARL,
     1YBRL, YAIM, YBIM, ZARL, ZBRL, ZAIM, ZBIM, F, F2, IREC, ZPRL, ZPIM,
     2VPRL, VPIM, ZRL, ZIM, VRL, VIM, K, I, IPUNC, RAO, RBO, DEN, L,
     3INDX, REAL. AIMG, RLDEN, ANMIN, RLNUM, DENIM, A, B, QUAN, IORDR,
     4LETTR, VAR1, IORD, ZARLK, ZBRLK, ZAIMK, ZBIMK,NSECT, NT
      COMMON XK, RK, ZOR, ZOI, VOR, VOI, BETA, DB, XIMAG
      IVAL = MP(K,I)
      YARL = 0.0
      YAIM = 0.0
      L = 1
   11 IF (IVAL-L) 9, 10, 10
   10 INDX = L*2-1
      VAR1 = R(INDX,K,I)
      QUAN = F2*C(INDX,K,I)
      IF (QUAN) 3, 4, 3
    4 QUAN = AL(INDX,K,I)*F
      GO TO 5
    3 QUAN =(AL(INDX,K,I)-1.0/QUAN)*F
    5 DEN = 1.0/(VAR1*VAR1+QUAN*QUAN)
      YARL = YARL+VAR1*DEN
      YAIM = YAIM-QUAN*DEN
      L = L+1
```

```
      GO TO 11
    9 DEN = YARL*YARL + YAIM*YAIM
      IF (DEN) 12, 13, 12
   12 DEN = 1.0/DEN
   13 ZARL = YARL*DEN
      ZAIM = -YAIM*DEN
      IVAL = MQ(K,I)
      YBRL = 0.0
      YBIM = 0.0
      L = 1
   16 IF (IVAL-L) 14, 15, 15
   15 INDX = 2*L
      VAR1 = R(INDX,K,I)
      QUAN = F2*C(INDX,K,I)
      IF(QUAN) 6, 7, 6
    7 QUAN = AL(INDX,K,I)*F
      GO TO 8
    6 QUAN =(AL(INDX,K,I)-1.0/QUAN)*F
    8 DEN = 1.0/(VAR1*VAR1+QUAN*QUAN)
      YBRL = YBRL+VAR1*DEN
      YBIM = YBIM-QUAN*DEN
      L = L+1
      GO TO 16
   14 DEN = YBRL*YBRL + YBIM*YBIM
      IF (DEN) 17, 18, 17
   17 DEN = 1.0/DEN
   18 ZBRL = YBRL*DEN
      ZBIM = -YBIM*DEN
```

```
RETURN

END
```

1620 USERS GROUP

WESTERN REGION MEETING

June 18, 1964

FORTRAN II - DEBUGGING TECHNIQUES AND AIDS

Leon P. Goldberg
Technical Staff
Princeton University

FORTRAN II

FLOATING HARDWARE

NON RELOCABLE SUBROUTINES

Address	Subroutine name	Function
01510	SWC	I/O
01768	COMPLT	I/O
03158	RATY	I/O
03182	RAPT	I/O
01418	RACD	I/O
01574	SLASH	I/O
00986	WATY	I/O
01022	WAPT	I/O
01058	WACD	I/O
01800	HTYPE	Hollerith conversion
02052	REDO	Multiple field specs.
02152	REP	Multiple parenthesized specs.
02380	ITYPE	I specification
03280	FTYPE	F specification
03300	ETYPE	E specification
06020	XTYPE	X specification
06052	ATYPE	A specification
06528	MATRIX	Reading arrays
07316	FXA	I+J
07348	FXSR	-(J*K)+I
07416	FXS	I-J
07440	FXM	I*J
07484	FXD	I/J
07570	FXDR	1/(I/J)
07604	RSGN	-I or -A
07698	FLOAT	A=I
07932	FIX	I=A
08152	FIXI	I**J
08586	FAXI	A**I
09044	FAXB	A**B
09356	FMFAC	FAC to A
09504	FSB	A-B
09528	FAD	A+B
09740	FSBR	-(A*B)+C
09808	FMP	A*B
09856	FD	A/B
09952	FDVR	1/(A/B)
10000	TOFAC	A to FAC
06710	TRACE	
02256	REPEAT (REP)	
00485	FAC	

```
C       INTERACTION OF COMMON AND EQUIVALENCE IN UNDIMENSIONED VARIABLES.
        COMMON X,Y,Z
        EQUIVALENCE (X,A),(Y,B,C),(Z,L)
        A=4.1527341
        B=2.*A
        R=X+2.
        S=B+R
        Z=X+Y+S
        STOP
        END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1043 4̅152730̅1
1̅1051 2̅000000̅1
5̅9999       X
5̅9999       A
5̅9991       Y
5̅9991       B
5̅9991       C
5̅9983       Z
5̅9983       L
1̅1059       R
1̅1067       S
END OF PASS I
```

```
171̅00001̅1043
170̅67105̅9999
171̅00001̅1051
170̅98085̅9999
170̅67105̅9991
171̅00005̅9999
170̅95281̅1051
170̅67101̅1059
171̅00005̅9991
170̅95281̅1059
170̅67101̅1067
171̅00005̅9999
170̅95285̅9991
170̅95281̅1067
170̅67105̅9983
340̅00000̅0102
390̅04170̅0100
480̅00000̅0000
```

```
C     COMMON STORAGE IN DIMENSIONED VARIABLES.
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      COMMON X,Y
      STOP
      END
T̅URN SW 1 ON F̅OR SYMBOL TABLE, PRESS START
5̲9959     X 5̲9999
5̲9859     Y 5̲9949
1̅1045     Z 1̅1635
END OF PASS I
```

```
C     EXAMPLES TO DEMONSTRATE THE COMPLEX INTERACTION OF EQUIVALENCE AND
C     COMMON STORAGE ASSIGNMENT IN DIMENSIONED VARIABLES.
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      COMMON X
      EQUIVALENCE (Y,Z)
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START
5̅9959      X 5̅9999
1̅1045      Y 1̅1135
1̅1045      Z 1̅1635
END OF PASS I
```

```
ENTER SOURCE PROGRAM, PRESS START
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      COMMON X
      EQUIVALENCE (X(5),Y(10),Z(60))
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

5̅9959      X 5̅9999
5̅9909      Y 5̅9999
5̅9409      Z 5̅9999
END OF PASS I
```

```
ENTER SOURCE PROGRAM, PRESS START
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      EQUIVALENCE  (Y(10),X(5),Z(60))
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045      Z 1̅1635
1̅1545      Y 1̅1635
1̅1595      X 1̅1635
END OF PASS I
```

```
C     EXAMPLES TO DEMONSTRATE THE COMPLEX INTERACTION OF EQUIVALENCE AND
C     COMMON STORAGE ASSIGNMENT IN DIMENSIONED VARIABLES.
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      EQUIVALENCE  (Z(60),X(5), Y(10))
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̄1045       Z 1̄1635
1̄1595       X 1̄1635
1̄1545       Y 1̄1635
END OF PASS I




ENTER SOURCE PROGRAM, PRESS START
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      COMMON X
      EQUIVALENCE (X,Y,Z)
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

5̄9959       X 5̄9999
ĒRROR 55
5̄9959       Y 6̄0049
ĒRROR 55
5̄9959       Z 6̄0549
END OF PASS I




ENTER SOURCE PROGRAM, PRESS START
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      COMMON Z
      EQUIVALENCE (X,Y,Z)
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

5̄9409       Z 5̄9999
5̄9409       X 5̄9449
5̄9409       Y 5̄9499
END OF PASS I

ENTER SOURCE PROGRAM, PRESS START
      DIMENSION X(5), Y(2,5), Z(3,4,5)
      COMMON Y
      EQUIVALENCE (X(3),Y(6),Z(15))
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START
5̄9909       Y 5̄9999
5̄9939       X 5̄9979
ĒRROR 55
5̄9819       Z 6̄0409
END OF PASS I
```

```
C     EQUIVALENCE STORAGE ASSIGNMENT.
      EQUIVALENCE (R,S),(T,U,V).(R,L)
      EQUIVALENCE (A,B,C,D)
      R=1.5
      A=R+3.
      L=B+1.
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̄1043 1̄50000 0̄1
1̄1051 3̄000000̄1
1̄1059 1̄000000̄1
1̄1067       R
1̄1067       S
1̄1067       L
1̄1075       T
1̄1075       U
1̄1075       V
1̄1083       A
1̄1083       B
1̄1083       C
1̄1083       D
END OF PASS I
```

```
17̄1̄00001̄1043
17̄0̄67101̄1067
17̄1̄00001̄1067
17̄0̄95281̄1051
17̄0̄67101̄1083
17̄1̄00001̄1083
17̄0̄95281̄1059
17̄0̄79320000̄0
17̄0̄67101̄1067
340̄00000̄0102
390̄04170̄0100
480̄00000̄0000
```

```
C     COMMON STORAGE ASSIGNMENT.
      COMMON W,X.Y Z
      W=1.5
      X=2.0
      Z=3.0
      Y=2.5
      A=W+X+Y+Z+0.5
      L=4. + 5.
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

11047 150000000001
11059 200000000001
11071 300000000001
11083 250000000001
11095 500000000000
11107 400000000001
11119 500000000001
59999       W
59987       X
59975       Y
59963       Z
11131       A
11141       L
END OF PASS I




171000011047
170671059999
171000011059
170671059987
171000011071
170671059963
171000011083
170671059975
171000059999
170952859987
170952859975
170952859963
170952811095
170671011131
171000011107
170952811119
170793200000
170671011141
340000000102
390041700100
480000000000
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
C     GENERAL I/O
      READ 1, X,Y,L
    1 FORMAT(F10.3,E15.8,I5,5X,3HXY=)
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045 0̅001
1̅1055       X
1̅1065       Y
1̅1070       L
END OF PASS I
```

```
27̅0̅14181̅1040
17̅0̅15101̅1055
17̅0̅15101̅1065
17̅0̅15101̅1070̅
17̅0̅17680000̅0
4̅91̅1045̅0̅0̅328̅
00̅100̅30̅33000̅
150̅80̅23800̅10
0̅60200̅100̅180
00̅066̅76̅83̅30̅2
0521̅1139340̅0
0000̅0102390̅0
4170̅0100480̅0
0000̅000004̅91̅
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
         READ 1, X,Y1,Y2,Y3
       1 FORMAT (A4,4(F10.3))
         STOP
         END
   TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

   11045 0001
   11055       X
   11065      Y1
   11075      Y2
   11085      Y3
   END OF PASS I
```

```
  270141811040
  170151011055
  170151011065
  170151011075
  170151011085
  170176800000
  491104500605
  200803280010
  030225611178
  040205211173
  340000000102
  390041700100
  480000000000
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
C       I/O OF MATRICES WITHOUT THE IMPLIED DO LOOP.
        DIMENSION A(15)
        READ 1, A
      1  FORMAT (F10.3)
        STOP
        END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045 ·0̅001
1̅1055       A 1̅1195
END OF PASS I
```

```
27̅0̅14181̅1040
160̅620300̅015
160̅626300̅000
17̅0̅65281̅1045
170̅1768000̅0
4̅91̅1045̅0̅0̅328
00̅100̅30̅20521̅
1263340̅00000̅
0102390̅04170̅
0100480̅00000̅
0000
```

```
ENTER SOURCE PROGRAM, PRESS START


C     I/O OF MATRICES WITH THE IMPLIED DO LOOP.
      DIMENSION X(20)
      READ 1, (X(I),I=1,20)
    1 FORMAT (8F10.3)
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1040 0̅0001
1̅1045 0̅0020
1̅1055 0̅001
1̅1065      X 1̅1255
1̅1260      I
END OF PASS I








27̅0̅14181̅1050
2̅6̅1̅112601̅1040
1̅31̅126000001̅0
11̅0̅00991̅1055
320̅00950̅00000
270̅15100̅00099
1̅11̅12600̅000̅01
2̅4̅1̅12601̅1045
4̅71̅128601100
170̅176800000̅0
4̅91̅1055̅0̅0̅328
00̅100̅30̅21521̅
13940̅80̅20521̅
1389340̅000000̅
0102390̅04170̅
0100480̅000000̅
0000
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
C     SAMPLE PROBLEM.
      X = A + B + C*D
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045      X
1̅1055      A
1̅1065      B
1̅1075      C
1̅1085      D
END OF PASS I

17̅1̅00001̅1055
17̅0̅95281̅1065
17̅0̅93561̅1139
4̅91̅11400̅0000
00000017̅1̅000
01̅107517̅0̅980
81̅108517̅0̅952
81̅113917̅0̅671
01̅1045340̅000
00̅0102390̅041
70̅0100480̅000
00̅00004̅

ENTER SOURCE PROGRAM, PRESS START
C     SAMPLE PROBLEM.
      X = C*D + A + B
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045      X
1̅1055      C
1̅1065      D
1̅1075      A
1̅1085      B
END OF PASS I



17̅1̅00001̅1055
17̅0̅98081̅1065
17̅0̅95281̅1075
17̅0̅95281̅1085
17̅0̅67101̅1045
340̅00000̅0102
390̅04170̅0100
480̅00000̅0000
```

```
C     SAMPLE PROBLEM.
      X = A*B*C**4
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1040 0̅0004
1̅1050      X
1̅1060      A
1̅1070      B
1̅1080      C
END OF PASS I
```

```
17̅1̅00001̅1060
17̅0̅98081̅1070
17̅0̅93561̅1135
4̅91̅11360̅0000
00000017̅1̅000
01̅108017̅0̅858
61̅104017̅0̅980
81̅113517̅0̅671
01̅1050340̅000
00̅0102390̅041
70̅0100480̅000
00̅00004̅91̅
```

```
C     SAMPLE PROBLEM.
      X = C**4 *A*B
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̄1040 0̄0004
1̄1050     X
1̄1060     C
1̄1070     A
1̄1080     B
END OF PASS I
```

```
17̄1̄00001̄1060
17̄0̄85861̄1040
17̄0̄98081̄1070
17̄0̄98081̄1080
17̄0̄67101̄1050
340̄00000̄0102
390̄04170̄0100
480̄00000̄0000
```

```
ENTER SOURCE PROGRAM, PRESS START


C     SAMPLE PROBLEM.
      X = C**4. *A*B
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045 4̅00000000̅1
1̅1055     X
1̅1065     C
1̅1075     A
1̅1085     B
END OF PASS I
```

```
17̅1̅00001̅1065
17̅0̅90441̅1045
17̅0̅98081̅1075
17̅0̅98081̅1085
17̅0̅67101̅1055
340̅00000̅0102
390̅04170̅0100
480̅00000̅0000
```

```
ENTER SOURCE PROGRAM, PRESS START


C     SAMPLE PROBLEM.
      X = A*B - C*D
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045      X
1̅1055      A
1̅1065      B
1̅1075      C
1̅1085      D
END OF PASS 1




 17̅1̅00001̅1055
 17̅0̅98081̅1065
 170̅93561̅1139
4̅91̅11400̅0000
 17̅0̅67117̅1̅000
 01̅1075117̅0̅980
 81̅108517̅0̅974
 01̅1139170̅671
 01̅1045340̅000
 00̅0102390̅041
 70̅0100480̅000
 00̅00004̅91̅
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
C       SAMPLE PROBLEM.
        TEMP = A * B
        X = C*D - TEMP
        STOP
        END
 TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

 1̅1045   TEMP
 1̅1055      A
 1̅1065      B
 1̅1075      X
 1̅1085      C
 1̅1095      D
 END OF PASS I
```

```
 17̅1̅00001̅1055
 170̅98081̅1065
 170̅67101̅1045
 17̅1̅00001̅1085
 170̅98081̅1095
 170̅95041̅1045
 170̅67101̅1075
340̅00000̅0102
390̅04170̅0100
480̅00000̅0000
```

```
C     SIMPLE SINGLE,DOUBLE AND TRIPLE SUBSCRIPTION EXAMPLE.
      DIMENSION A(5),B(2,4),C(2,2,2)
      EQUIVALENCE (A3, A(3)),(B2, B(2,3)),(C1,C(1,2,1))
 ERROR 05
      B2 = 23.
      C1 = 121.
      A2 = B2 + C1
      STOP
      END
END OF PASS I
ENTER SOURCE PROGRAM, PRESS START




C     SIMPLE SINGLE,DOUBLE AND TRIPLE SUBSCRIPTION EXAMPLE.
      DIMENSION A(5),B(2,4),C(2,2,2)
      EQUIVALENCE (A3, A(3)),(B2, B(7)),(C1,C(3))
      B2 = 23.
      C1 = 121.
      A2 = B2 + C1
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

11045 2300000002
11055 1210000003
11065      A 11105
11085     A3
11115      B 11185
11175     B2
11195      C 11265
11215     C1
11275     A2
END OF PASS I
CARDS NOT IN ORDER
SW1 ON TO PUNCH SUBROUTINES, PRESS START
END OF PASS II
SW1 ON TO PUNCH SUBROUTINES, PRESS START
END OF PASS II
```

```
131104500002
210000991040
260003900099
130003900010
110009911100
320009500000
260048500099
170935611395
491139600000
000000171000
011055270671
011395131106
000002210009
911040260003
900099130003
900002210009
911060260003
900099130003
900010110009
911140320009
500000260048
500099170935
611395171000
011070270671
011395131104
500010110009
911070320009
500000260048
500099170935
611395131104
500002210009
911040260003
900099130003
900010110009
911100320009
500000271000
000099131106
000002210009
911040260003
900099130003
900002210009
911060260003
900099130003
900010110009
911140320009
500000170935
611857491185
800000000000
271000000099
170952811857
270671011395
340000000102
390041700100
480000000000
```

```
C     SIMPLE SINGLE,DOUBLE AND TRIPLE SUBSCRIPTION EXAMPLE.
      DIMENSION A(5),B(2,4),C(2,2,2)
      B(2,3) = 23.
      C(1,2,1) = 121.
      A(3)=B(2,3) + C(1,2,1)
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1040 0̅0002
1̅1045 0̅0003
1̅1055 2̅30000000̅2
1̅1060 0̅0001
1̅1070 1̅2100000̅3
1̅1080       A 1̅1120
1̅1130       B 1̅1200
1̅1210       C 1̅1280
END OF PASS I
```

```
17̅1̅00001̅1045
17̅0̅67101̅1175
17̅1̅00001̅1055
17̅0̅67101̅1215
17̅1̅00001̅1175
17̅0̅95281̅1215
17̅0̅67101̅1275
340̅00000̅0102
390̅04170̅0100
480̅00000̅00000
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
C     SAMPLE PROBLEM SHOWING  POLYNOMIAL EXPRESSION  W/O NESTING.
      DIMENSION X(10),A(20)
      READ 1, NPTS,(X(I),I=1,NPTS)
      DO 30 I=1,20
      AI = I
   30 A(I) = AI
      DO 20 I=1,NPTS
      SUM = 0.0
      XX = X(I)
      DO 10 K=1,20
      KK= K-1
      AK= A(K)
   10 SUM = SUM + AK *XX**KK
   20 PRINT 2, SUM,I,XX
    2 FORMAT (2X,4HSUM=E15.8,7HWHEN X(I2,2H)=E15.8)
    1 FORMAT (I2/8E10.3)
      STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START
1̄1040 0̄0001
1̄1045 0̄0020
1̄1055 0̄000000009̄9̄
1̄1065 0̄001
1̄1070 0̄030
1̄1075 0̄020
1̄1080 0̄010
1̄1090 0̄002
1̄1100      X 1̄1190
1̄1200      A 1̄1390
1̄1395   NPTS
1̄1400      I
1̄1410     AI
1̄1420    SUM
1̄1430     XX.
1̄1435      K
1̄1440     KK
1̄1450     AK
END OF PASS I
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
C       SAMPLE PROBLEM SHOWING  POLYNOMIAL EXPRESSION   W. NESTING.
        DIMENSION X(10),A(20)
        READ 1, NPTS,(X(I),I=1,NPTS)
        DO 30 I=1,20
        AI = I
     30 A(I) = AI
        DO 20 I=1,NPTS
        XX = X(I)
        PROD = A(20)*XX + A(19)
        DO 10 K=3,20
        MK=21-K
     10 PROD = PROD*XX+ A(MK)
     20 PRINT 2,PROD,I,XX
      2 FORMAT (2X,4HSUM=E15.8,7HWHEN X(I2,2H)=E15.8)
      1 FORMAT (I2/8E10.3)
        STOP
        END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̄1040 0̄0001
1̄1045 0̄0020
1̄1050 0̄0019
1̄1055 0̄0003
1̄1060 0̄0021
1̄1070 0̄001
1̄1075 0̄030
1̄1080 0̄020
1̄1085 0̄010
1̄1095 0̄002
1̄1105        X 1̄1195
1̄1205        A 1̄1395
1̄1400     NPTS
1̄1405        I
1̄1415       AI
1̄1425       XX
1̄1435     PROD
1̄1440        K
1̄1445       MK
END OF PASS I
```

```
C     SAMPLE PROBLEM SHOWING COMPUTED AND UNCONDITIONAL GO TO @S.
      L=1
    3 Y=4
      X=2
      A=X+Y
      GO TO (1,2),L
    1 L=2
      GO TO 3
    2 STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1040 0̅0001
1̅1045 0̅0004
1̅1050 0̅0002
1̅1055 0̅003
1̅1060 0̅001
1̅1065 0̅002
1̅1070      L
1̅1080      Y
1̅1090      X
1̅1100      A
END OF PASS I


LOAD SUBROUTINES



17̅1̅00001̅1040
17̅0̅67101̅1070
17̅1̅00001̅1045
17̅0̅76980000̅0
17̅0̅67101̅1080
17̅1̅00001̅1050
17̅0̅76980000̅0
17̅0̅67101̅1090
17̅1̅00001̅1090
17̅0̅95281̅1080
17̅0̅67101̅1100
1̅31̅10700000̅5
12̅0̅00991̅1264
49̅0̅00991̅1060̅
1̅1065̅017̅1̅000
01̅105017̅0̅671
01̅10704̅91̅105
5̅0̅0000340̅000
00̅0102390̅041
70̅0100480̅000
00̅00004̅9
```

```
C      SAMPLE PROBLEM USING A DO LOOP
       DIMENSION X(10)
       DO 5 I=1,10
     5 X(I) = I
       STOP
       END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS
RT                   T               S


1̅1040 0̅0001
1̅1045 0̅0010
1̅1050 0̅005
1̅1060      X 1̅1150
1̅1155      I
END OF PASS I
```

```
2̅6̅1̅11551̅1040
1̅31̅11550001̅0
11̅0̅00991̅1050
320̅00950̅0000
260̅04850̅0099
17̅0̅93561̅1245
4̅91̅12460̅0000
000000̅5̅17̅1̅000
01̅115̲5170̅769
80000̅027̅0̅671
01̅12451̅11̅115
50̅00̅0012̅4̅1̅115
51̅10454̅71̅116
801100340̅000
00̅01023900̅41
70̅0100480̅000
00̅00004̅9
```

```
ENTER SOURCE PROGRAM  PRESS START
```

```
C     SAMPLE PROBLEM USING IF STATEMENT
      IF (A) 1,2,3
    1 B= -1.0
      GO TO 4
    2 B= 0.0
      GO TO 4
    3 B= 1.0
    4 STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START
1̅1045 1̅00000000̅1
1̅1055 0̅000000009̅9̅
1̅1060 0̅001
1̅1065 0̅002
1̅1070 0̅003
1̅1075 0̅004
1̅1085     A
1̅1095     B
END OF PASS I
```

```
17̅1̅00001̅1085
170̅67100̅0485
4̅31̅11440̅0476
4̅91̅1065̅00000
4̅41̅1070̅0̅0483
4̅91̅10600̅017̅1̅0
0001̅1045170̅7
604000̅0̅017̅0̅6
7101̅10954̅91̅1
075̅0̅000017̅1̅0
0001̅105517̅0̅6
7101̅10954̅91̅1
075̅0̅000017̅1̅0
0001̅104517̅0̅6
7101̅1095340̅0
0000̅0102390̅0
4170̅0100480̅0
0000̅00004̅91̅
```

```
ENTER SOURCE PROGRAM, PRESS START
```

```
C     SAMPLE PROBLEM USING IF STATEMENT
      B= 1.0
      IF (A) 1,2,4
    1 B= -1.0
      GO TO 4
    2 B= 0.0
    4 STOP
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START

1̅1045 1̅00000000̅1
1̅1055 0̅0000009̅9̅
1̅1060 0̅001
1̅1065 0̅002
1̅1070 0̅004
1̅1080       B
1̅1090       A
END OF PASS I



17̅1̅00001̅1045
17̅0̅67101̅1080
171̅00001̅1090
170̅67100̅0485
4̅31̅11640̅0476
4̅91̅10655̅00000
4̅41̅1070̅0̅0483
4̅91̅1060̅017̅1̅0
0001̅104517̅0̅7
6040̅000̅0170̅6
710̅1̅11080̅4̅91̅1
070̅0̅0000017̅1̅0
0001̅105517̅0̅6
7101̅1080340̅0
0000̅0102390̅0
4170̅0100480̅0
0000̅000004̅9
```

REFERENCES

Private Consultations with - - - - - - - - - - - L. Hoffman
P. Larrea

FORTRAN II AND THE 1443

Lanny L. Hoffman

PRINTER FN II SUB. LOCATIONS.

00485	FAC	FLOATING ACCUMULATOR
01314	TOFAC	A TO FAC
01358	FMFAC	FAC TO A
01402	TRACE	PRINT FAC, GO TO FMFAC
01482	FXA	I+J
01514	FXS	I-J
01538	FXSR	-(J*K)+I
01562	XRSGN	-I
01608	FXM	I*J
01652	FXD	I/J
01738	FXDR	1/(I/J)
01772	FIX	I=A
02112	FLOAT	A=I
02336	RSGN	-A
02336	FSBR	-(A*B)+C
02430	FAD	A+B
03002	FSB	A-B
03046	FMP	A*B
03218	FD	A/B
03434	FDVR	1/(A/B)
03538	FIXI	I**J
03960	FAXI	A**I
04550	FAXB	A**B
04920	WATY	I/O
04992	WACD	I/O
05028	PRA	I/O (PRINT ALPH)
05304	RATY	I/O
05364	RACD	I/O
05408	SLASH	I/O
05650	COMPLT	I/O
05690	REP	MULTIPLE PARENTHESIZED SPECS.
06066	SWC	I/O
06122	MATRIX	READING ARRAYS
06280	REDO	MULTIPLE FIELD SPECS.
05794	XTYPE	X SPECIFICATION
05826	HTYPE	H SPECIFICATION
06350	ATYPE	A SPECIFICATION
06842	ITYPE	I SPECIFICATION
07830	FTYPE	F SPECIFICATION
07850	ETYPE	E SPECIFICATION
01976	ERR MESS	PRINT ERROR MESSAGE

01924	PASS I	PASS I INIT.
00684	PASS II	PASS II INIT.
01938	SPS I	SPS PASS I INIT.

SUBPROGRAM

```
      SUBROUTINE X(A,B,C,D)
      C=A+B
      D=A-B
      RETURN
      END
```

MAIN LINE

```
      CALL X(ALFA,BETA,GAMMA,DELTA)
```

CALLING SEQUENCE

```
      BTM   XCELL,*+11,6
      DSA   ALFA,BETA,GAMMA,DELTA
```

SUBPROGRAM
GENERATED
MATERIALS

```
      DS    5
XPGM  AM    XPGM-1,5,010
      TF    A,XPGM-1,0111
      BNF   *+36,A,01
      CF    A,,0
      TF    A,A,0111
      AM    XPGM-1,5,010
      TF    B,XPGM-1,0111
      BNF   *+36,B,01
      CF    B,,0
      TF    B,B,0111
      AM    XPGM-1,5,010
      TF    C,XPGM-1,0111
      BNF   *+36,C,01
      CF    C,,0
      TF    C,C,0111
      AM    XPGM-1,5,010
      TF    D,XPGM-1,0111
      BNF   *+36,D,01
      CF    D,,0
      TF    D,D,0111
      AM    XPGM-1,1,010
      BT    TOFAC,A,1
      BT    FAD,B,1
      BT    FRMFAC,C,1
      BT    TOFAC,A,1
      BT    FSB,B,1
      BT    FRMFAC,D,1
      B     XPGM-1,,06
------------------------------------------------------------
```

FLAG CONVENTION...........

IF P IS RELOCATABLE, FLAG OPERAND IS 0
IF Q IS RELOCATABLE, FLAG OPERAND IS 1

NORMAL FLAG OPERANDS ARE STILL IN EFFECT FOR IMMEDIATE AND INDIRECT ADDRESSING.

FLAGS ARE USED OVER THE OPERATION CODE TO DENOTE RELOCATION TO THE LOADER. THESE FLAGS DO NOT ALTER THE OPERATION OF THE INSTRUCTION.

```
C      TEST PROGRAM FOR 1443 PRINTER PLOT, SIN (X) VS. X.
C
       DIMENSION X(500),Y(500)
        PRINT 10
    10 FORMAT(1H1)
       T=0.
        DO 1 I=1,200
       X(I)=T
       Y(I)=SINF(T)
       T=T+.01*3.14159
     1 CONTINUE
       PAUSE
       CALL PLOT(X,10.,0.,5,Y,1.,-1.,10,200)
       STOP
       END
```

```
C      FORTRAN SUBROUTINE FOR 1443 PRINTER PLOTTING,
C                 BY L. HOFFMAN, GUGGENHEIM LABS.
C
      SUBROUTINE PLOT(X,XMAX,XMIN,NX,Y,YMAX,YMIN,NY,N)
      DIMENSION DUMMY(2),OUTPUT(102),X(2),Y(2)
      XCHAR=.20
      YCHAR=.71
      CHAR=.14
      BLANK=0.
      XNO=100.
      NOX=XNO+1.
      YLABEL=YMAX
      DX=(XMAX-XMIN)/XNO
      DY=(YMAX-YMIN)/50.
C       MOVE MAX DOWN BY ONE-HALF BOX....
            YYMAX=YMAX+.5*DY
           XXMIN=XMIN-.5*DX
C
      KY=0
      NX1=NX+1
      DO 1 I=1,51
C     CALL INIT(OUTPUT,BLANK)
        DO 111 II2=1,NOX
  111   OUTPUT(II2)=BLANK
C     CALL GRID(OUTPUT,DX,DY,NX,NY,KY,I,XCHAR,YCHAR,IND)
      IND=0
        IF(I-1-50*KY/NY)211,222,211
  222     DO 332 JJ=1,NOX
  332    OUTPUT(JJ)=XCHAR
      IND=1
      KY=KY+1
  211   DO 444 JJ=1,NX1
         I2=((JJ-1)*(NOX-1))/NX
  444   OUTPUT(I2+1)=YCHAR
      ZI=I
      UP=YYMAX-(ZI-1.)*DY
      DOWN=UP-DY
C     CALL FINDY(X,Y,UP,DOWN,OUTPUT,N,DX,DY,XMAX,XMIN,CHAR)
         DO 1121 IF=1,N
         IF(Y(IF)-UP)2221,1121,1121
 2221    IF(Y(IF)-DOWN)1121,3331,3331
 3331    CONTINUE
           JJ=(X(IF)-XXMIN)/DX
         JJ=JJ+1
         OUTPUT(JJ)=CHAR
 1121    CONTINUE
    8 IF(IND)10,10,11
   10 PRINT 2,(OUTPUT(J),J=1,NOX)
    2 FORMAT(12X,50A1,51A1)
      GO TO 12
   11   PRINT 3,YLABEL,(OUTPUT(J),J=1,NOX)
    3 FORMAT(1X,E10.3,1X,50A1,51A1)
   12 YLABEL=YLABEL-DY
    1 CONTINUE
      RETURN
      END
```

FORTRAN II SPS SUBROUTINES, L. HOFFMAN, GUGGENHEIM LABS.

ASSEMBLY AND FINAL PHASE OF SPS SUBS. FOR FN II.

1) USE 1620/1710 SPS TO ASSEMBLE AND COMPRESS THE SPS PROGRAM.
2) REMOVE THE FIRST TWO (2) AND THE LAST SEVEN (7) CARDS FROM THE COMPRESSED DECK. (THIS DOES NOT INCLUDE THE TWO BLANK CARDS AT THE END OF THE DECK)
3) ADD HEADER CARD AS NO. 1.
4) ADD TRAILER CARD TO END OF DECK.
5) CORRECT DSA'S , IF ANY,USED IN THE SPS PROGRAM,OTHERWISE, GO TO 7.
6) PUNCH A FLAGGED ZERO IN COLUMN 62 OF ALL OBJECT DECK CARDS PRODUCED BY DSA'S IN SPS PROGRAM.
7) CHECK FOR RELOCATABLE CONSTANTS, IF NONE, THEN GO TO 9.
8) PUNCH A FLAGGED 1 IN COLUMN 62 OF ALL CONSTANTS NOT TO BE RELOCATED.
9) PUNCH NEW CARD NO. IN TRAILER CARD TO CONTINUE SEQUENCING.
10) THE DECK CAN NOW BE USED WITH A FORTRAN CALL STATEMENT.

THE HEADER CARD......................

COLS. 1-12	SUBROUTINE NAME IN TWO-DIGIT ALPHANUMERIC FORM WITH FLAG OVER HIGH ORDER DIGIT AND RIGHT JUSTIFIED.
COLS. 13-20	BLANK
COLS. 21-22	FF, LENGTH OF FLOATING MANTISSA, FLAG OVER HIGH ORDER DIGIT.
COLS. 23-24	KK, LENGTH OF FIXED MANTISSA, FLAG OVER HIGH ORDER DIGIT.
COLS. 25-62	BLANK
COLS. 63	RECORD MARK (0-2-8)
COLS. 64-80	BLANK, EXCEPT FLAG IN COL. 78

THE TRAILER CARD.....................

COLS. 1-62	BLANK
COLS. 63	FLAGGED 1
COLS. 64-80	BLANK, EXCEPT FOR CARD NO. IN COL. 78-80.

```
      *       PLOT FOR 1443 PRINTER, L. HOFFMAN GUGGENHEIM LABS.
      *
      *        AN EXAMPLE OF AN SPS SUBROUTINE FOR FN II .
      *
11036                             4/24           DORG 11036
11040        00005                AUTO    PLOT   DS   5
11045        00005                AUTO    X      DS   5
11050        00005                AUTO    XMAX   DS   5
11055        00005                AUTO    XMIN   DS   5
11060        00005                AUTO    NX     DS   5
11065        00005                AUTO    Y      DS   5
11070        00005                AUTO    YMAX   DS   5
11075        00005                AUTO    YMIN   DS   5
11080        00005                AUTO    NY     DS   5
11085        00005                AUTO    N      DS   5
11092        00007                AUTO           DS   7
                                          *       END OF ARGUMENT ADDRESSES/
11094   J1   11093   000-5        AUTO    START  AM   START-1,5,010
11106   KO   11045   1109L        AUTO           TF   X     ,START-1,0111
11118   MM   11154   11045        AUTO           BNF  *+36,X     ,01
11130   L3   11045   00000        AUTO           CF   X     ,,0
11142   KO   11045   1104N        AUTO           TF   X     ,X     ,0111
11154   J1   11093   000-5        AUTO           AM   START-1,5,010
11166   KO   11050   1109L        AUTO           TF   XMAX  ,START-1,0111
11178   MM   11214   11050        AUTO           BNF  *+36,XMAX  ,01
11190   L3   11050   00000        AUTO           CF   XMAX  ,,0
11202   KO   11050   1105-        AUTO           TF   XMAX  ,XMAX  ,0111
11214   J1   11093   000-5        AUTO           AM   START-1,5,010
11226   KO   11055   1109L        AUTO           TF   XMIN  ,START-1,0111
11238   MM   11274   11055        AUTO           BNF  *+36,XMIN  ,01
11250   L3   11055   00000        AUTO           CF   XMIN  ,,0
11262   KO   11055   1105N        AUTO           TF   XMIN  ,XMIN  ,0111
11274   J1   11093   000-5        AUTO           AM   START-1,5,010
11286   KO   11060   1109L        AUTO           TF   NX    ,START-1,0111
11298   MM   11334   11060        AUTO           BNF  *+36,NX    ,01
11310   L3   11060   00000        AUTO           CF   NX    ,,0
11322   KO   11060   1106-        AUTO           TF   NX    ,NX    ,0111
11334   J1   11093   000-5        AUTO           AM   START-1,5,010
11346   KO   11065   1109L        AUTO           TF   Y     ,START-1,0111
11358   MM   11394   11065        AUTO           BNF  *+36,Y     ,01
11370   L3   11065   00000        AUTO           CF   Y     ,,0
11382   KO   11065   1106N        AUTO           TF   Y     ,Y     ,0111
11394   J1   11093   000-5        AUTO           AM   START-1,5,010
11406   KO   11070   1109L        AUTO           TF   YMAX  ,START-1,0111
11418   MM   11454   11070        AUTO           BNF  *+36,YMAX  ,01
11430   L3   11070   00000        AUTO           CF   YMAX  ,,0
11442   KO   11070   1107-        AUTO           TF   YMAX  ,YMAX  ,0111
11454   J1   11093   000-5        AUTO           AM   START-1,5,010
11466   KO   11075   1109L        AUTO           TF   YMIN  ,START-1,0111
11478   MM   11514   11075        AUTO           BNF  *+36,YMIN  ,01
11490   L3   11075   00000        AUTO           CF   YMIN  ,,0
11502   KO   11075   1107N        AUTO           TF   YMIN  ,YMIN  ,0111
11514   J1   11093   000-5        AUTO           AM   START-1,5,010
11526   KO   11080   1109L        AUTO           TF   NY    ,START-1,0111
11538   MM   11574   11080        AUTO           BNF  *+36,NY    ,01
11550   L3   11080   00000        AUTO           CF   NY    ,,0
11562   KO   11080   1108-        AUTO           TF   NY    ,NY    ,0111
11574   J1   11093   000-5        AUTO           AM   START-1,5,010
11586   KO   11085   1109L        AUTO           TF   N     ,START-1,0111
11598   MM   11634   11085        AUTO           BNF  *+36,N     ,01
11610   L3   11085   00000        AUTO           CF   N     ,,0
```

```
11622  K0  11085  1108N      AUTO           TF   N     ,N     ,0111
11634  J1  11093  000-2      AUTO           AM   START-1,2,010
11646  M9  12074  00000      AUTO           B    AROUND,,0
11654                        AUTO           DORG *-3
                                     *            SYMBOL TABLE AND CONSTANTS HERE
00010      00000             3/17    FF     DS   ,10
00005      00000             3/17    KK     DS   ,5
00485      00000             3/17    FAC    DS   ,485
01402      00000             4/23    TRACE  DS   ,1402
01652      00000             4/23    FXD    DS   ,1652
02336      00000             4/23    RSGN   DS   ,2336
02112      00000             4/23    FLOAT  DS   ,2112
01772      00000             4/23    FIX    DS   ,1772
01358      00000             4/23    FMFAC  DS   ,1358
03002      00000             4/23    FSB    DS   ,3002
02430      00000             4/23    FAD    DS   ,2430
02336      00000             5012    FSBR   DS   ,2336
03046      00000             4/23    FMP    DS   ,3046
03218      00000             4/23    FDV    DS   ,3218
03434      00000             4/23    FDVR   DS   ,3434
01314      00000             4/23    TOFAC  DS   ,1314
05028      00000             4/23    WATY   DS   ,5028
06066      00000             4/23    SWC    DS   ,6066
05650      00000             4/23    COMPLT DS   ,5650
11658      00005             4/23    FMT    DC   5,5794
11661      00003             4/23           DC   3,002
11666      00005             4/23           DC   5,7850
11669      00003             4/23           DC   3,010
11671      00002             4/23           DC   2,03
11676      00005             4/23           DC   5,5794
11679      00003             4/23           DC   3,002
11684      00005             4/24    REDO   DC   5,6280
11687      00012                     DUMOUT DAC  12,
11711      00102             3/13    OUTPUT DAS  102
11915      00001                            DAC  1,'
11920      00005             3/22    I      DS   KK
11924      00004             3/22    JJ     DS   4
11927      00003             3/25    II2    DS   3
11932      00005             3/22    I2     DS   5
11937      00005             3/22    IF     DS   5
11947      00010             3/22    ZI     DS   FF
11957      00010             3/22    UP     DS   FF
11967      00010             3/22    DOWN   DS   FF
11972      00005             3/22    COUNT  DS   5
11980      00008             3/22    HALF1  DC   8,50000000
11982      00002             3/22    HALF   DC   2,00
11992      00010             3/22    YYMAX  DS   FF
12002      00010             3/22    DY     DS   FF
12012      00010             3/22    DX     DS   FF
12022      00010             3/22    YLABEL DS   FF
12023      00001             3/22    IND    DS   1
12028      00005             3/22    NDX    DS   KK
12036      00008             *****   XNO1   DC   8,10000000
12038      00002             *****   XNO    DC   2,03
12046      00008             3/22    ONE1   DC   8,10000000
12048      00002             3/22    ONE    DC   2,01
12056      00008             3/22    F1     DC   8,50000000
12058      00002             3/22    FIFTY  DC   2,02
12063      00005             3/22    KY     DS   KK
12073      00010             3/25    XXMIN  DS   FF
12074  1P  01314  J2038      3/22    AROUND BTM  TOFAC,XNO,17
```

```
12086   1P   02430   J2048        3/22           BTM   FAD,ONE,17
12098   17   01772   000-0        3/22           BTM   FIX,0,10
12110   K6   12028   00485        3/22           TF    NOX,FAC,0
12122   2P   01314   11070        3/22           BT    TOFAC,YMAX,1
12134   1P   01402   J2022        3/24           BTM   TRACE,YLABEL,17
12146   2P   03002   11075        3/22           BT    FSB,YMIN,1
12158   1P   03218   J2058        3/22           BTM   FDV,FIFTY,17
12170   1P   01402   J2002        3/24           BTM   TRACE,DY,17
                                  3/22   *
12182   2P   01314   11050        3/22           BT    TOFAC,XMAX,1
12194   2P   03002   11055        3/22           BT    FSB,XMIN,1
12206   1P   03218   J2038        3/22           BTM   FDV,XNO,17
12218   1P   01402   J2012        3/24           BTM   TRACE,DX,17
                                  3/22   *
12230   1P   03046   J1982        3/25           BTM   FMP,HALF,17
12242   2P   02336   11055        3/25           BT    FSBR,XMIN,1
12254   1P   01402   J2073        3/25           BTM   TRACE,XXMIN,17
12266   1P   01314   J2002        3/25           BTM   TOFAC,DY,17
12278   1P   03046   J1982        3/22           BTM   FMP,HALF,17
12290   2P   02430   11070        3/22           BT    FAD,YMAX,1
12302   1P   01402   J1992        3/24           BTM   TRACE,YYMAX,17
                                  3/22   *
12314   J6   12063   0-000        3/22           TFM   KY,0,08
12326   J6   11920   00-01        3/22           TFM   I,1,09
12338   J6   11927   00-01        3/25   RTN1    TFM   II2,1,09
12350   J0   11972   J1711        3/22   RTN111  TFM   COUNT,OUTPUT,017
12362   KJ   11972   11927        3/22           A     COUNT,II2,01
12374   KJ   11972   11927        3/22           A     COUNT,II2,01
12386   J6   1197K   000-0        3/22           TFM   COUNT,0,0610
12398   J1   11927   000-1        3/22           AM    II2,1,010
12410   KM   12028   11927        3/24           C     NOX,II2,01
12422   M6   12350   01300        3/24           BNN   RTN111,,0
12434   J5   12023   00000        3/22           TDM   IND,0,0
                                  3/22   *      GRID.........
12446   J3   12063   -005-        3/22           MM    KY,50,0711
12458   32   00095   00000        3/23           SF    99-KK+1
12470   26   00485   00099        3/22           TF    FAC,99
12482   2P   01652   11080        3/22           BT    FXD,NY,1
12494   12   00485   000-1        3/22           SM    FAC,1,10
12506   2J   00485   11920        3/22           A     FAC,I,1
12518   14   00485   0-000        3/22           CM    FAC,0,8
12530   M7   12662   01200        3/22           BNE   I211,,0
                                  3/22   *
12542   J6   11924   00-01        3/25   I222    TFM   JJ,1,09
12554   J0   11972   J1711        3/22   RTN332  TFM   COUNT,OUTPUT,017
12566   KJ   11972   11924        3/22           A     COUNT,JJ,01
12578   KJ   11972   11924        3/22           A     COUNT,JJ,01
12590   J6   1197K   000K0        3/22           TFM   COUNT,20,0610
12602   J1   11924   000-1        3/22           AM    JJ,1,010
12614   KM   12028   11924        3/24           C     NOX,JJ,01
12626   M6   12554   01300        3/24           BNN   RTN332,,0
                                  3/22   *
12638   J5   12023   00001        3/22           TDM   IND,1,0
12650   J1   12063   000-1        3/22           AM    KY,1,010
                                  3/22   *
12662   J6   11924   00-00        3/22   I211    TFM   JJ,0,09
12674   20   00485   12028        3/22   RTN444  TF    FAC,NOX,1
12686   12   00485   000-1        3/25           SM    FAC,1,10
```

```
12698  2L  00485  11924    3/22             M     FAC,JJ,1
12710  32  00095  00000    3/23             SF    99-KK+1
12722  26  00485  00099    3/25             TF    FAC,99
12734  2P  01652  11060    3/22             BT    FXD,NX,1
12746  J0  11932  J1711    3/22             TFM   I2,OUTPUT,017
12758  K1  11932  00485    3/22             A     I2,FAC,0
12770  K1  11932  00485    3/25             A     I2,FAC,0
12782  J1  11932  000-2    3/22             AM    I2,2,010
12794  J6  1193K  000P1    3/22             TFM   I2,71,0610
12806  J1  11924  000-1    3/22             AM    JJ,1,010
12818  KM  1106-  11924    3/24             C     NX,JJ,016
12830  M6  12674  01300    3/24             BNN   RTN444,,0
                           3/22    *
12842  20  00485  11920    3/22             TF    FAC,I,1
12854  33  00483  00000    3/24             CF    FAC-2
12866  32  00481  00000    3/24             SF    FAC-KK+1
12878  17  02112  000-0    3/22             BTM   FLOAT,0,10
12890  1P  01402  J1947    3/22             BTM   TRACE,ZI,17
12902  1P  03002  J2048    3/22             BTM   FSB,ONE,17
12914  1P  03046  J2002    3/22             BTM   FMP,DY,17
12926  1P  02336  J1992    3/22             BTM   FSBR,YYMAX,17
12938  1P  01402  J1957    3/24             BTM   TRACE,UP,17
12950  1P  03002  J2002    3/22             BTM   FSB,DY,17
12962  1P  01402  J1967    3/24             BTM   TRACE,DOWN,17
                           3/22    *
12974  J6  11937  00-01    3/22             TFM   IF,1,09
12986  J3  11937  000J0    3/22    RTN121   MM    IF,FF,010
12998  2J  00099  11065    3/22             A     99,Y,1
13010  27  01314  00099    3/22             BT    TOFAC,99
13022  1P  03002  J1957    3/22             BTM   FSB,UP,17
13034  M4  13246  00483    3/25             BNF   I1121,FAC-2,0
13046  J3  11937  000J0    3/22    I2221    MM    IF,FF,010
13058  2J  00099  11065    3/22             A     99,Y,1
13070  27  01314  00099    3/22             BT    TOFAC,99
13082  1P  03002  J1967    3/22             BTM   FSB,DOWN,17
13094  M4  13114  00483    3/22             BNF   I3331,FAC-2,0
13106  M9  13246  00000    3/22             B     I1121,,0
13114                      3/22             DORG  *-3
13114  J3  11937  000J0    3/22    I3331    MM    IF,FF,010
13126  2J  00099  11045    3/22             A     99,X,1
13138  27  01314  00099    3/22             BT    TOFAC,99
13150  1P  03002  J2073    3/25             BTM   FSB,XXMIN,17
13162  1P  03218  J2012    3/22             BTM   FDV,DX,17
13174  17  01772  000-0    3/22             BTM   FIX,0,10
13186  11  00485  000-1    3/22             AM    FAC,1,10
13198  13  00485  -0002    3/22             MM    FAC,2,7
13210  32  00095  00000    3/22             SF    95
13222  1J  00099  J1711    3/22             AM    99,OUTPUT,17
13234  16  0009R  000J4    3/22             TFM   99,14,610
13246  J1  11937  000-1    3/22    I1121    AM    IF,1,010
13258  KM  1108N  11937    3/24             C     N,IF,016
13270  M6  12986  01300    3/24             BNN   RTN121,,0
13282  ML  13318  12023    3/22             BD    I11,IND,01
13294  L9  11687  00900    4/2     I10      WA    OUTPUT-24,00900,0
13306  M9  13366  00000    3/13             B     I12,,0
13318  L9  11687  00901    4/2     I11      WA    OUTPUT-24,00901,0
                           3/13    *                ADD YLABEL OUTPUT .....
13330  1P  05028  J1653    4/24             BTM   WATY,FMT-5,17
13342  1P  06066  J2022    4/2              BTM   SWC,YLABEL,17
13354  17  05650  000-0    4/2              BTM   COMPLT,0,10
13366  1P  01314  J2022    3/13    I12      BTM   TOFAC,YLABEL,17
```

```
13378   1P   03002   J2002          3/13             BTM  FSB,DY,17
13390   1P   01402   J2022          3/13             BTM  TRACE,YLABEL,17
13402   J1   11920   000-1          3/13             AM   I,1,010
13414   J4   11920   000N1          3/13             CM   I,51,010
13426   M7   12338   01100          3/13             BNP  RTN1,,0
13438   M9   1109L   00000          3/13             B    START-1,,06
00000                               3/13             DEND
```

NUMERIC LISTING OF OBJECT DECK OF SPS PLOT SUBROUTINE FOR FN II.

L. HOFFMAN, GUGGENHEIM LABS.

```
    -               - -                                                   -
00005753566300000000805000000000000000000000000000000000000000R00000000000000001
-         - --         ---          -           --         -  - -    -     --
111109300005261104511093441115411045331104500000261104511045R0011109411154000002
-         - --         ---          -           --         -  - -    -     --
111109300005261105011093441121411050331105000000261105011050R0011115411214000003
-         - --         ---          -           --         -  - -    -     --
111109300005261105511093441127411055331105500000261105511055R0011121411274000004
-         - --         ---          -           --         -  - -    -     --
111109300005261106011093441133411060331106000000261106011060R0011127411334000005
-         - --         ---          -           --         -  - -    -     --
111109300005261106511093441139411065331106500000261106511065R0011133411394000006
-         - --         ---          -           --         -  - -    -     --
111109300005261107011093441145411070331107000000261107011070R0011139411454000007
-         - --         ---          -           --         -  - -    -     --
111109300005261107511093441151411075331107500000261107511075R0011145411514000008
-         - --         ---          -           --         -  - -    -     --
111109300005261108011093441157411080331108000000261108011080R0011151411574000009
-         - --         ---          -           --         -  - -    -     --
111109300005261108511093441163411085331108500000261108511085R0011157411634000010
-         - -                                                 - -    -     --
111109300002491207400000R0000000000000000000000000000000000000011163411658000011
-    -  -    -  - -    -  -                                   - -    -     --
0579400207850010030579400206280R000000000000000000000000000001011165411685000012
-                                                             - -    -     --
000000000000000000000000R0000000000000000000000000000000000001011168611710000013
-                                                             - -    -     --
0R000000000000000000000000000000000000000000000000000000000001011191411916000014
-       -                                                     - -    -     --
5000000000R000000000000000000000000000000000000000000000000001011197311983000015
-       - -       - -       -                                 - -    -     --
100000000310000000015000000002R000000000000000000000000000000001011202912059000016
 -     -     -     -              - -            -            - -    -     --
170131412038170243012048170177200000261202800485270131411070R0011207412134000017
 -     -     -           -     -     -     -     -            - -    -     --
170140212022270300211075170321812058170140212002270131411050R0011213412194000018
```

```
 -           -     -     -     -     -     -     -            - -    -     --
270300211055170321812038170140212012170304611982270233611055R0011219412254000019
 -     -     -     -     -     -     -           -     -      - -    -     --
170140212073170131412002170304611982270243011070170140211992R0011225412314000020
-       -   -        -  -        -  --     -    --            - -    -     --
161206300000161192000001161192700001161197211711211197211927R0011231412374000021
--          -     -   - -         - --          -             - -    -     --
211197211927161197200000111192700001241202811927461235001300R0011237412434000022
-           -      -   -                         -            - -    -     --
151202300000131206300050320009500000260048500099270165211080R0011243412494000023
          -  -                  -   -           -        -    - -    -     --
120048500001210048511920140048500000471266201200161192400001R0011249412554000024
--     -    --          --          -     -   - -         -   - -    -     --
161197211711211197211924211197211924161197200020111192400001R0011255412614000025
--          -           -           -         - -        -    - -    -     --
241202811924461255401300151202300001111206300001161192400000R0011261412674000026
 -                    -  -                                    - -    -     --
260048512028120048500001230048511924320009500000260048500099R0011267412734000027
 -          --     -    -           -           -         -   - -    -     --
270165211060161193211711211193200485211193200485111193200002R0011273412794000028
-     -   - -         - --    -     -            -            - -    -     --
161193200071111192400001241106011924461267401300260048511920R0011279412854000029
                                  -  -     -     -     -      - -    -     --
330048300000320048100000170211200000170140211947170300212048R0011285412914000030
 -     -     -     -     -     -     -     -     -     -      - -    -     --
170304612002170233611992170140211957170300212002170140211967R0011291412974000031
-        -  -         -  -                       -     -      - -    -     --
161193700001131193700010210009911065270131400099170300211957R0011297413034000032
-           -         -  -                       -     -      - -    -     --
441324600483131193700010210009911065270131400099170300211967R0011303413094000033
-           -                                                 - -    -     --
441311400483491324600000R000000000000000000000000000000000000011309413118000034
-         -  -                       -     -     -     -      - -    -     --
131193700010210009911045270131400099170300212073170321812012R0011311413174000035
          -           -        -                 -     -      - -    -     --
170177200000110048500001130048500002320009500000110009911711R0011317413234000036
      -   - -         - --    -     -           --            - -    -     --
160009900014111193700001241108511937461298601300431331812023R0011323413294000037
-           -           -            -     -     -     -      - -    -     --
391168700900491336600000391168700901170502811653170606612022R0011329413354000038
```

```
          -  -     -     -     -     -     -    -         -   - -    -     --
17056500000017013141202217030021200217014021202211119200000 1R0011335413414000039
-         - -           -     -                               - -    -     --
14119200005147123380110049110930000 0R00000000000000000000000000011341413450000040
                                                              -           -
00000000000000000000000000000000000000000000000000000000000000100000000000000041
```

```
C     AUTOMATIC LINKAGE GENERATOR FOR SPS SUBS. FOR FN II.
C
C     L. HOFFMAN, GUGGENHEIM LABS.
C
*1205
C     SPS - FN II LINKAGE AND CONSTANT AUTOMATIC GENERATOR......
      DIMENSION VAR(20)
   27 READ 1,N
    1 FORMAT(I5)
      READ 2,(VAR(I),I=1,N)
    2 FORMAT(A6)
    4 TYPE  3
    3 FORMAT(22H TYPE SUBROUTINE NAME. )
      ACCEPT 2,SNAME
      IF(SNAME)5,4,5
    5 PUNCH 6
      PRINT 6
    6 FORMAT(5HAUTO ,6X,9HDORG11036)
      PUNCH 7,SNAME
      PRINT 7,SNAME
    7 FORMAT(5HAUTO ,A6,5HDS   5)
      DO 8 I=1,N
      PUNCH 9,VAR(I)
      PRINT 9,VAR(I)
    9 FORMAT(5HAUTO ,A6,5HDS   5)
    8 CONTINUE
      PUNCH 10
      PRINT 10
   10 FORMAT(5HAUTO ,6X,5HDS   7)
      PUNCH 30
      PRINT 30
      PUNCH 31
      PRINT 31
      PUNCH 32
      PRINT 32
      PUNCH 45
      PRINT 45
      PUNCH 38
      PRINT 38
      PUNCH 33
      PRINT 33
      PUNCH 34
      PRINT 34
      PUNCH 37
      PRINT 37
      PUNCH 36
      PRINT 36
      PUNCH 35
      PRINT 35
      PUNCH 41
      PRINT 41
      PUNCH 40
      PRINT 40
      PUNCH 39
      PRINT 39
      PUNCH 42
      PRINT 42
      PUNCH 43
      PRINT 43
      PUNCH 44
      PRINT 44
```

```
   PUNCH 46
   PRINT 46
   PUNCH 49
   PRINT 49
   PUNCH 48
   PRINT 48
   PUNCH 47
   PRINT 47
30 FORMAT(5HAUTO ,6HFF     ,4HDS   ,6H,10         )
31 FORMAT(5HAUTO ,6HKK     ,4HDS   ,6H,5          )
32 FORMAT(5HAUTO ,6HFAC    ,4HDS   ,6H,485        )
45 FORMAT(5HAUTO ,6HTOFAC  ,4HDS   ,6H,1314       )
38 FORMAT(5HAUTO ,6HFMFAC  ,4HDS   ,6H,1358       )
33 FORMAT(5HAUTO ,6HTRACE  ,4HDS   ,6H,1402       )
34 FORMAT(5HAUTO ,6HFXD    ,4HDS   ,6H,1652       )
37 FORMAT(5HAUTO ,6HFIX    ,4HDS   ,6H,1772       )
36 FORMAT(5HAUTO ,6HFLOAT  ,4HDS   ,6H,2112       )
35 FORMAT(5HAUTO ,6HRSGN   ,4HDS   ,6H,2336       )
41 FORMAT(5HAUTO ,6HFSBR   ,4HDS   ,6H,2336       )
40 FORMAT(5HAUTO ,6HFAD    ,4HDS   ,6H,2430       )
39 FORMAT(5HAUTO ,6HFSB    ,4HDS   ,6H,3002       )
42 FORMAT(5HAUTO ,6HFMP    ,4HDS   ,6H,3046       )
43 FORMAT(5HAUTO ,6HFDV    ,4HDS   ,6H,3218       )
44 FORMAT(5HAUTO ,6HFDVR   ,4HDS   ,6H,3434       )
46 FORMAT(5HAUTO ,6HWATY   ,4HDS   ,6H,4920       )
49 FORMAT(5HAUTO ,6HPRA    ,4HDS   ,6H,5028       )
48 FORMAT(5HAUTO ,6HCOMPLT ,4HDS   6H,5650        )
47 FORMAT(5HAUTO ,6HSWC    ,4HDS   ,6H,6066       )
   PUNCH 11
   PRINT 11
11 FORMAT(5X,1H*,12X,25HEND OF ARGUMENT ADDRESSES/5X,1H*,14X,14HSTART
111 LINKAGE. )
   START=.6263415963
   BLANK=0.
   DO 12 I=1,N
   IF(I-1)13,14,13
14 PUNCH 15,START
   PRINT 15,START
15 FORMAT(5HAUTO ,A6,4HAM   ,13HSTART-1,5,010)
   GO TO 16
13 PUNCH 15,BLANK
   PRINT 15,BLANK
16 PUNCH 17,VAR(I)
   PRINT 17,VAR(I)
17 FORMAT(5HAUTO ,6X,4HTF   ,A6,13H,START-1,0111)
   PUNCH 18,VAR(I)
   PRINT 18,VAR(I)
18 FORMAT(5HAUTO ,6X,4HBNF  ,5H*+36,,A6,3H,01)
   PUNCH 19,VAR(I)
   PRINT 19,VAR(I)
19 FORMAT(5HAUTO ,6X,4HCF   ,A6,3H,,0)
   PUNCH 20,VAR(I),VAR(I)
   PRINT 20,VAR(I),VAR(I)
20 FORMAT(5HAUTO ,6X,4HTF   ,A6,1H,,A6,5H,0111)
12 CONTINUE
   M=N/2
   M=M*2
   IF(M-N)21,22,21
22 PUNCH 23
   PRINT 23
23 FORMAT(5HAUTO ,6X,4HAM   ,13HSTART-1,1,010)
```

```
      GO TO 24
   21 PUNCH 25
      PRINT 25
   25 FORMAT(5HAUTO ,6X,4HAM  ,13HSTART-1,2,010)
   24 PUNCH 26
      PRINT 26
   26 FORMAT(5HAUTO ,6X,4HB   ,9HAROUND,,0/5HAUTO ,6X,4HDORG,3H*-3/
   261 5X,1H*,12X,36HSYMBOL TABLE AND CONSTANTS HERE ...   )
      GO TO 27
      END
```

```
 DATA FOR AUTO-LINK.

    9
X
XMAX
XMIN
NX
Y
YMAX
YMIN
NY
N
```

A SURVEY OF THE BEGINNING PROGRAMMING COURSE

Clarence B. Germain
College of St. Thomas
February 20, 1964

Last Fall, a questionaire was sent to the 280 schools which are members of the USERS Group. 175 schools responded. The results are tabulated on the following pages.

1. No allowance has been made for non-respondents. This does bias the results.
2. Since the survey covers only schools having 1620's, the figures for the end of 1964 do not reflect the influence of schools which will acquire their first 1620 during the year.
3. A suprising number of respondents gave incosistent answers; e.g., they indicated floating-point hardware, but not divie hardware, or they indicated that 35% of their students run their own SPS programs, while they taught SPS only to 20% of their students.
4. Figures for index registers, binary capabilities, and the 1627 plotter may not be indicative since the questionaire was circulated too soon after announcement of these features.
5. Average enrollment in the beginning programming courses in 170 students per school per year.
6. Many of the Model II 1620's will supplement existing Model I's, not replace them.
7. Relatively few schools indicated any plans to obtain the 1443 printer.
8. The disk units will more than double in popularity during 1964 with 1/3 of all schools having at least one disk unit by the end of the year.
9. While 3% of the schools offered no course involving Fortran, 35% of the students were taught more than one version of Fortran.
10. At the end of 1963, 51% of the schools had the hardware necessary to run Fortran II; by the end of 1964, this figure will rise to 59%.
11. 85% of the students get "hands on" experience in running their own programs on the computer. This percentage is about the same regardless of what programming systems (SPS, GOTRAN, etc.) are taught.
12. Jim Moore's Multi-Trace, 1.4.003, was the most commonly mentioned trace program taught to students. However, 85% of the schools indicated that they used no trace program in their courses.
13. The figures for textbooks are for use in at least one course. Many schools use more than one text in a course. 31% of the schools use only IBM publications as texts. While a wide variety of texts, many unrelated to either Fortran or the 1620, are in use, only four commercial texts and a half-dozen IBM publications are used with any frequency. Of the non-programming type texts, numerical analysis books, particularly Stanton's, were most often mentioned.
14. The textbook percentages in no way indicate sales of books; these figures are quite different from the percentages shown here and were not a part of this study.

RESPONSES OF 175 SCHOOLS TO A SEPTEMBER 1963 QUESTIONAIRE

Results are given as a percentage of the number of schools replying to the questionaire. Probable errors do not exceed ±3% except for items marked with an asterisk (*) where the probable error is less than ±8%. Results are given for the end of 1963 and for the end of 1964. Changes for 1964 are only for equipment now on order. Slight discrepancies in the percentages are due to rounding.

1620 Model:	1963	1964
I	98%	89%
II	2	11

Special Features, Model I		
AFP, Div, IDA, Edit	31	31
AFP, Div, IDA	3	3
AFP, Div, Edit	0	0
AFP, Div	1	1
Div, IDA, Edit	31	31
Div, IDA	14	14
Div, Edit	1	1
Div	3	3
IDA, Edit	1	1
IDA	3	3
Edit	1	1
No special features	13	13

Summary:		
Automatic Floating-Point	34	35
Automatic Divide	82	82
Indirect Addressing	82	82
Additional (Edit) Instructions	64	64

Storage:		
20K core, no disk	48	38
40K core, no disk	21	18
60K core, no disk	17	13
20K core, disk	5	12
40K core, disk	4	9
60K core, disk	5	9

Input-Output:		
Paper Tape only	4	4
Paper Tape and Cards	10	10
Cards only	86	86
Magnetic Tape	4	4
Paper Tape	13	14
Cards, 1622-1	83	81
Cards, 1622-2	13	16
Cards, RPQ to read 800 cpm	3	3
1443 Printer		8
Disk, one or more	14	31
1627 Plotter	4	4
1710	2	3

Number of 1620's in the school:	
One	95%
Two	5

Special Features, Model II (1964)	
Automatic Floating-Point	65*
Index Registers	0*
Binary Capabilities	5*

Installations with Printer (1964)	
No disk	23*
1 disk	15*
2 disks	54*
3 disks	0*
4 disks	8*

Type of Courses Offered:	
Both credit and non-credit	5
Non-credit courses only	36
Credit courses only	13
No answer or no courses	47

Departments which offer courses:	
Engineering	40
Education	1
Mathematics	45
Business	31
Other	40

Subjects Taught:	
Machine Language	32
Operation of the Computer	66
SPS	29
GOTRAN	17
FORTRAN with FORMAT	47
FORTRAN II or II-D	33
FORGO, etc.	35
Use of some library trace	13
Block Diagramming	63
Monitor I	9

Disks:

No disk	86	68
1 disk	8	20
2 disks	5	11
3 disks	0	0
4 disks	1	1

Hardware necessary to run:

Fortran II only	37	29
Fortran II and II-D	9	19
Fortran II-D only	5	11

Students are expected to write and run their own programs using:

SPS II	25
GOTRAN	15
FORTRAN with FORMAT	43
FORTRAN Pre-Compiler	28
FORTRAN II	27

Required or recommended texts:

IBM Publications

1620 Reference Manual	74
1710 Reference Manual	4
SPS Reference Manual	49
GOTRAN Reference Manual	22
1620 FORTRAN Reference Manual	61
1620 FORTRAN II Bulletin	38
FORTRAN General Information Manual	23
1620 Program Writing and Testing Bulletin	12
Introduction to IBM Data Processing Systems	15
Programming and Block Diagramming Techniques	12

Commercial Publications

Germain—Programming the IBM 1620	27
Leeson-Dimitry—Basic Programming Concepts and the IBM 1620 Computer	39
Gruenberger-McCracken—Introduction to Electronic Computers	6
McCracken—A Guide to FORTRAN Programming	38
Organick—A FORTRAN Primer	38
Colman-Smallwood—Computer Language	6
Smith-Johnson—FORTRAN Autotester	3

Utah State University
Logan, Utah

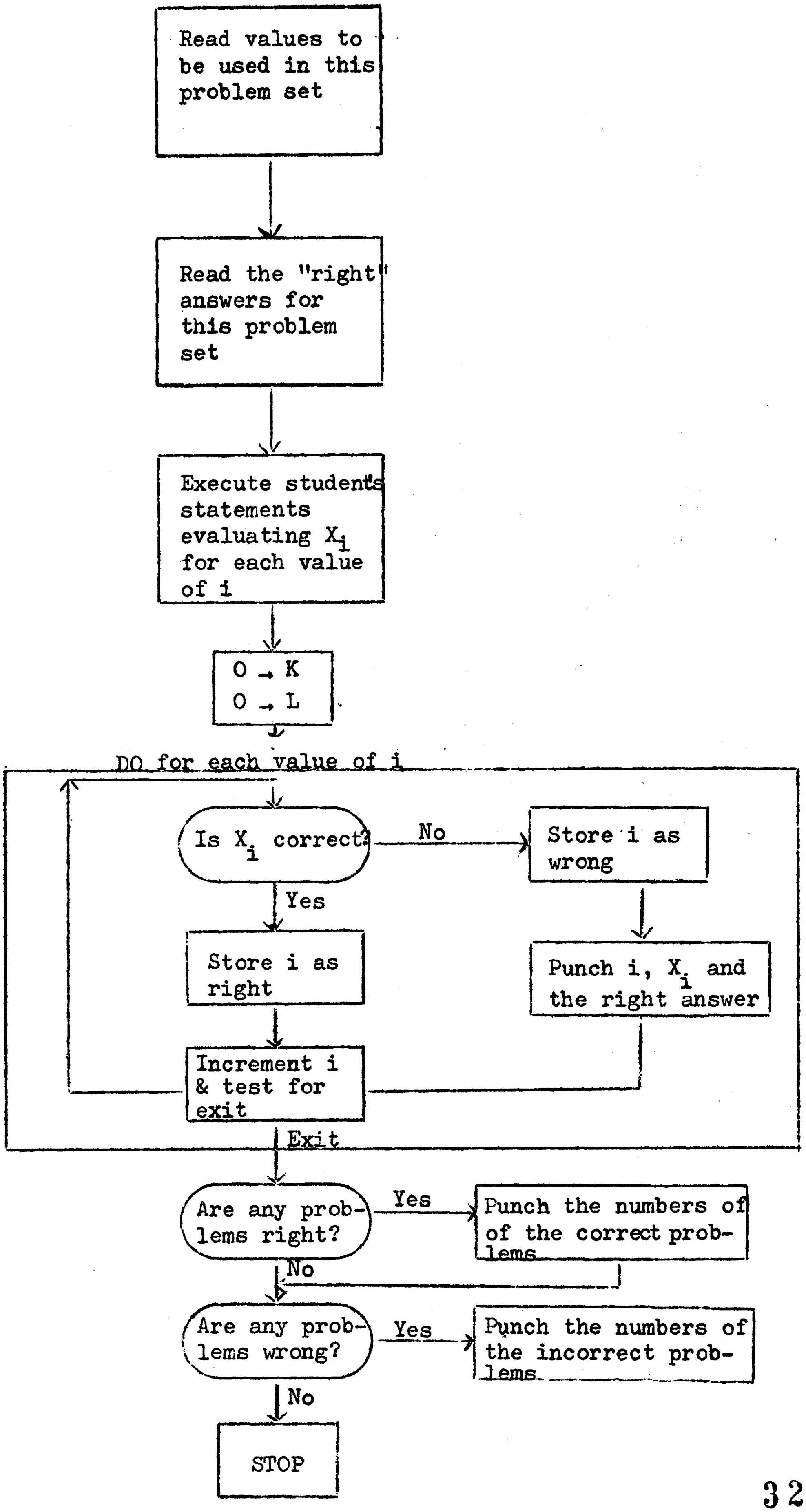
FORTRAN "TEACH" PROBLEMS

by

Wendell L. Pope

These problems are designed to be of assistance in introducing the neophyte to FORTRAN. Problem sets and programs to check them are provided for arithmetic statements, subscripted variables, fixed and floating point variables, functions and control statements, loops and input-output. The problems do not require that a student be able to write a complete program. They provide a means of acquainting him with the characteristics of FORTRAN in easy stages and help to bridge the gap between the introduction to computing and the writing of a complete program. The student's statements are checked for correctness by imbedding them in the appropriate checking program. They are checked for compilation errors by the FORGO processor, and for accuracy by the checking program itself. This is done by comparing the values computed by the student's statements to a predetermined set of "correct" values. For wrong answers, the number of the problem and the value computed are output, for right answers only the number of the problem is output.

FLOW CHART - TEACH Problem Checking Program



```
C TEACH PROBLEM NO 1   PLACE STUDENT HEADER CARD IN FRONT OF THIS CARD
      DIMENSION NRITE(10),NWRNG(10),RIGHT(10),X(10)
      READ,A,B,C,D,E,F,G,H,(RIGHT(I),I=1,10)
C     INSERT STUDENT STATEMENTS BEHIND THIS CARD
      K=0
      L=0
      DO 9950 I=1,10
      IF(X(I)-RIGHT(I)) 9912,9914,9912
 9912 L=L+1
      NWRNG(L)=I
      PUNCH 9920,I,X(I),RIGHT(I)
      GO TO 9950
 9914 K=K+1
      NRITE(K)=I
 9950 CONTINUE
      IF(K) 9922,9924,9922
 9922 PUNCH 9923,(NRITE(I),I=1,K)
 9923 FORMAT(6H RIGHT,10I5)
 9924 IF(L) 9925,9926,9925
 9925 PUNCH 9927,(NWRNG(I),I=1,L)
 9926 STOP
 9920 FORMAT(9H PROB NO. I3,4X,9HYOUR ANS=E16.8,4X,10HRIGHT ANS= E16.8)
 9927 FORMAT(6H WRONG, 10I5)
      END
10.341296 10.345599 8.6867569 -40.683394 15.683097 .0034784067
.00034329602 1.1234567
  .10664629E+02  0.41901625      0.84447317E+02  0.12039480E+02
  .65303219E+09  0.10259388E+01  0.14580430E+01  0.87982136E+01
  .46184027E+02  0.11729093E+03
```

```
C TEACH PROBLEM NO 2   PLACE STUDENT HEADER CARD IN FRONT OF THIS CARD
      DIMENSION NRITE( 5),NWRNG( 5),RIGHT( 5),X( 5),B(4),A(4,4)
      READ,K,L,((A(I,J),J=1,4),I=1,4),(B(I),I=1,4), (RIGHT(I),I=1,5)
C     INSERT STUDENT STATEMENTS BEHIND THIS CARD
      K=0
      L=0
      DO 9919 I=1,5
      IF(X(I)-RIGHT(I)) 9912,9914,9912
 9912 L=L+1
      NWRNG(L)=I
      PUNCH 9920,I,X(I),RIGHT(I)
      GO TO 9919
 9914 K=K+1
      NRITE(K)=I
 9919 CONTINUE
      IF(K) 9922,9924,9922
 9922 PUNCH 9923,(NRITE(I),I=1,K)
 9923 FORMAT(5HRIGHT,10I5)
 9924 IF(L) 9925,9926,9925
 9925 PUNCH 9927,(NWRNG(I),I=1,L)
 9926 STOP
 9927 FORMAT(5HWRONG,10I5)
 9920 FORMAT(8HPROB NO. I3,4X,9HYOUR ANS= E16.8,4X,10HRIGHT ANS= E16.8)
      END
  2   3
2.3964587  3.6241346  4.1357653  5.3422587  3.3524569  4.3687946  5.312
6.0247685  4.1024678  5.3751468  6.0347312  7.3107386  5.3420769  6.046
7.3214680  8.3469201  9.3704368  10.437695  -1.3579430  -2.5347962
5.4673001  41.125807  4913189.2  7701.4270  .86225934
```

```
C TEACH PROBLEM NO 3   PLACE STUDENT HEADER CARD IN FRONT OF THIS CARD
      DIMENSION NRITE(10),NWRNG(10),RIGHT(10),X(10),B(4)
      READ,K,L,(B(I),I=1,4),(RIGHT(I),I=1,5)
C     INSERT STUDENT STATEMENTS BEHIND THIS CARD
      K=0
      L=0
      DO 9919 I=1,5
      IF(X(I)-RIGHT(I)) 9912,9914,9912
 9912 L=L+1
      NWRNG(L)=I
      PUNCH 9920,I,X(I),RIGHT(I)
      GO TO 9919
 9914 K=K+1
      NRITE(K)=I
 9919 CONTINUE
      IF(K) 9922,9924,9922
 9922 PUNCH 9923,(NRITE(I),I=1,K)
 9923 FORMAT(5HRIGHT,10I5)
 9924 IF(L) 9925,9926,9925
 9925 PUNCH 9927,(NWRNG(I),I=1,L)
 9926 STOP
 9927 FORMAT(5HWRONG,10I5)
 9920 FORMAT(8HPROB NO. I3,4X,9HYOUR ANS= E16.8,4X,10HRIGHT ANS= E16.8)
      END
  2  3
                      9.3704368  10.437695  -1.3579430  -2.5347962
157.09495  -1.3579430  15.915392  205.01975  11.0
```

```
C TEACH PROBLEM NO 4   PLACE STUDENT HEADER CARD IN FRONT OF THIS CARD
C TEACH PROBLEM NO 4       PLACE STUDENT HEADER CARD IN FRONT OF THIS C
      DIMENSION NRITE(5),NWRNG(5),RIGHT (5),X(5)
      READ,A,B,C,D,E,F,G,H,K,(RIGHT(I),I=1,5)
C     INSERT STUDENT STATEMENTS BEHIND THIS CARD
      K=0
      L=0
      DO 9919 I=1,5
      IF (X(I)-RIGHT(I))9912,9914,9912
 9912 L=L+1
      NWRNG(L)=I
      PUNCH 9920,I,X(I),RIGHT(I)
      GO TO 9919
 9914 K=K+1
      NRITE(K)=I
 9919 CONTINUE
      IF(K)9922,9924,9922
 9922 PUNCH 9923,(NRITE(I),I=1,K)
 9923 FORMAT(5HRIGHT,10I5)
 9924 IF(L)9925,9926,9925
 9925 PUNCH 9927,(NWRNG(I),I=1,L)
 9926 STOP
 9927 FORMAT(5HWRONG,10I5)
 9920 FORMAT(8HPROB NO. I3,4X,9HYOUR ANS= E16.8,4X,10HRIGHT ANS= E16.8
      END
1.2457369  2.3580123  3.8609756  4.7602541  5.3025768  6.2047536
7.0367524  8.3205689  2
 1.1161260  1.3001178  .78878076  2.4429843  5.6480088
```

```
C TEACH PROBLEM NO 5   PLACE STUDENT HEADER CARD IN FRONT OF THIS CARD
      DIMENSION NRITE(5),NWRNG(5),RIGHT(5),X(5)
      READ,A,B,C,D,E,(RIGHT(I),I=1,5)
C     INSERT STUDENT STATEMENTS BEHIND THIS CARD
      K=0
      L=0
      DO 9919 I=1,5
      IF (X(I)-RIGHT(I))9912,9914,9912
 9912 L=L+1
      NWRNG(L)=I
      PUNCH 9920,I,X(I),RIGHT(I)
      GO TO 9919
 9914 K=K+1
      NRITE(K)=I
 9919 CONTINUE
      IF(K)9922,9924,9922
 9922 PUNCH 9923,(NRITE(I),I=1,K)
 9923 FORMAT(5HRIGHT,10I5)
 9924 IF(L)9925,9926,9925
 9925 PUNCH 9927,(NWRNG(I),I=1,L)
 9926 STOP
 9927 FORMAT(5HWRONG,10I5)
 9920 FORMAT(8HPROB NO. I3,4X,9HYOUR ANS= F16.8,4X,10HRIGHT ANS= E16.8
      END
2.3964587  3.6241346  4.1357653  5.3422587  3.3524569 112761.0 23922.7
11018.845 7478.2325 6160.7034
```

```
C TEACH PROBLEM NO 6   PLACE STUDENT HEADER CARD IN FRONT OF THIS CARD
      DIMENSION NRITE(5),NWRNG(5),RIGHT(5),X(5)
      READ,(RIGHT(I),I=1,5)
C     INSERT STUDENT STATEMENTS BEHIND THIS CARD
      K=0
      L=0
      DO 9919 I=1,5
      IF (X(I)-RIGHT(I))9912,9914,9912
 9912 L=L+1
      NWRNG(L)=I
      PUNCH 9920,I,X(I),RIGHT(I)
      GO TO 9919
 9914 K=K+1
      NRITE(K)=I
 9919 CONTINUE
      IF(K)9922,9924,9922
 9922 PUNCH 9923,(NRITE(I),I=1,K)
 9923 FORMAT(5HRIGHT,10I5)
 9924 IF(L)9925,9926,9925
 9925 PUNCH 9927,(NWRNG(I),I=1,L)
 9926 STOP
 9927 FORMAT(5HWRONG,10I5)
 9920 FORMAT(8HPROB NO. I3,4X,9HYOUR ANS= E16.8,4X,10HRIGHT ANS= E16.8
      END
   .97569023E+04  0.11699576E+04  0.15283722E+05
   .34042183E+03  0.23871681E+04
 14.369025  15.6753869  1.0367521  6.9851203
8.4357205
3.9857423
9.8530247
8.5432586
```

TEACH PROBLEM SET 1

Arithmetic Expressions

Assume that values of A, B, C, D, E, F, G and H are in storage. Write and keypunch correct FORTRAN statements to evaluate each of the following expressions.

1. $X(1) = A + \frac{-B}{C + D}$

2. $X(2) = \frac{A + B}{C - D}$

3. $X(3) = AB + C \frac{D}{E-F} - G$

4. $X(4) = A + \cfrac{B}{C + \cfrac{D}{E + \cfrac{F}{G-H}}}$

5. $X(5) = A + B^{C} - \frac{D}{E}$

6. $X(6) = \left(A + B^{\frac{C-D}{E}}\right)^{F}$

7. $X(7) = A^{(B^2+C)^D} + \frac{\pi}{\left(E - \frac{F}{G}\right)^H}$ $(\pi = 3.141592653...)$

8. $X(8) = \frac{A/B}{C} + \frac{A}{B/C}$

9. If $E > B > A > H > C > F > G > 0$, compute

   $X(9) = A + B + C + E + F + G + H$

   to obtain the most accuracy.

10. $X(10) = A + B^2 + \frac{3}{4+D}$

TEACH PROBLEM SET 2

Subscripted Variables

Assume the arrays $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$ and $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$

are in storage. Write and keypunch correct FORTRAN statements to evaluate each of the following expressions.

1. $X(1) = \frac{a_{12}}{a_{11}} + \frac{a_{13}}{a_{11}} + \frac{a_{14}}{a_{11}}$

2. $X(2) = \sum_{j=1}^{4} a_{1j}\, b_j$

3. $X(3) = \sum_{j=1}^{4} a_{2j}^{\,b_j}$

4. $X(4) = a_{31}X_1 + a_{32}X_1^2 + a_{33}X_1^3 + a_{34}X_1^4$

5. $X(5) = \frac{\{b_1\}}{b_2}$ where $\{b_1\}$ denotes the integer portion of b_1.

TEACH PROBLEM SET 3

Fixed point, Floating point, and subscripted variables

Assume that values of k and L and $B = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{pmatrix}$ are in storage.

Write and keypunch correct FORTRAN statements to evaluate each of the following expressions.

1. $X_1 = \sum_{i=1}^{4} b_i^{\,i}$

2. $X_2 = b_{k^3-5}$

3. $X_3 = \sum_{i=1}^{4} b_{ik-i}$

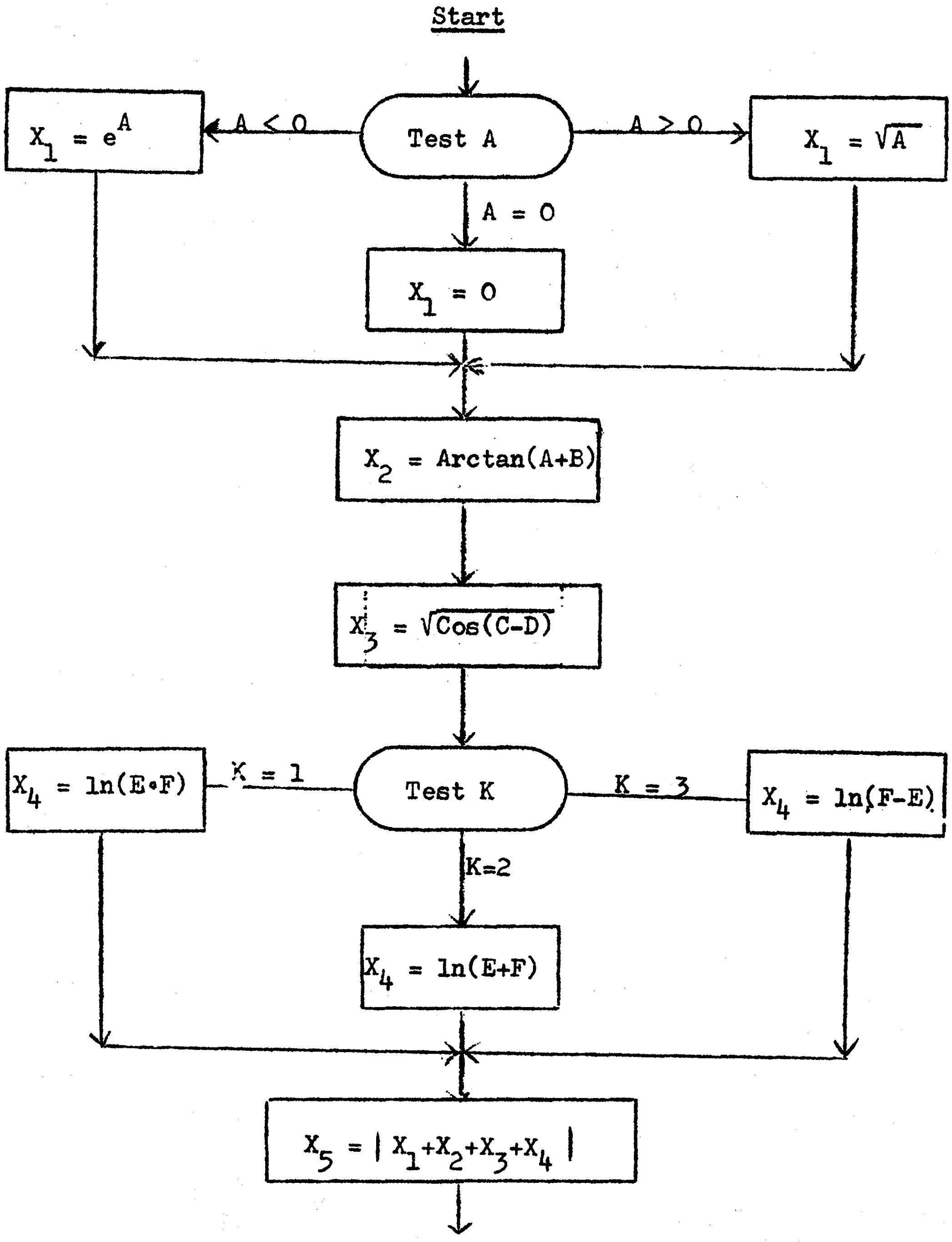
4. $X_4 = \sum_{i=1}^{4} b_i^{\,k}$

5. $X_5 = \frac{L}{k} + k^L + k^3 - 2$

TEACH PROBLEM SET 4

Functions and Control Statements

Write (and keypunch) statements to evaluate X_1, X_2, X_4 and X_5 according to the instructions in the flow chart below. Assume A, B, C, D, E, F and K to be defined.



TEACH PROBLEM SET 5

Loops

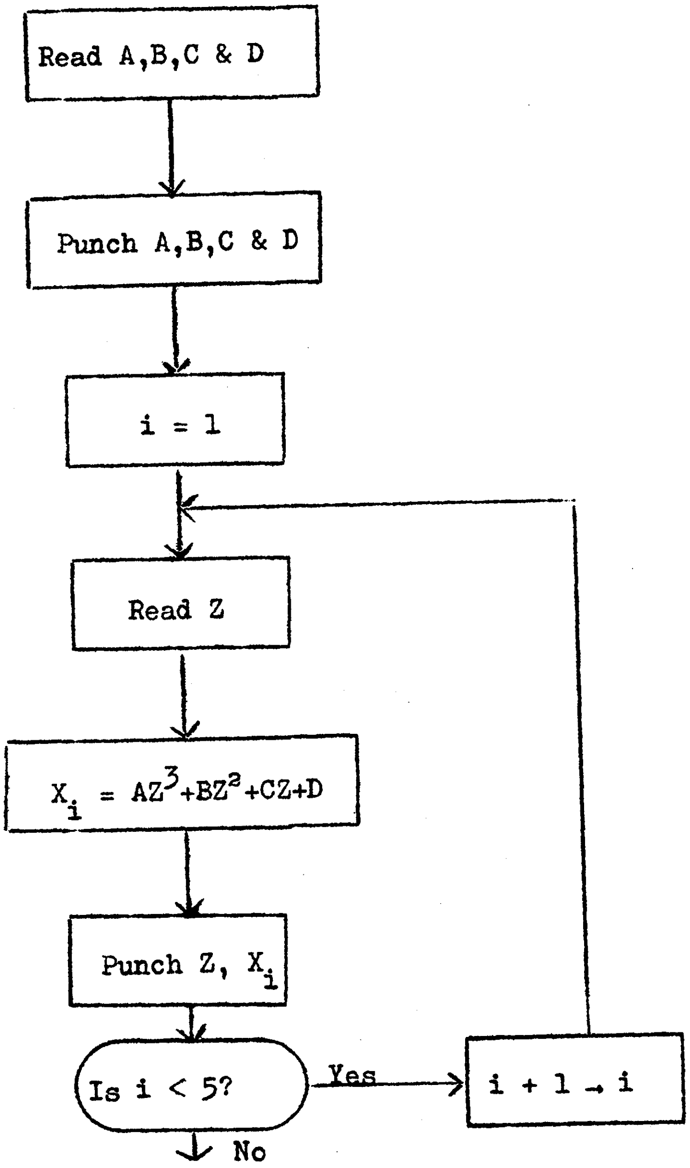
Write (and keypunch) statements to evaluate X_1, X_2, X_3, X_4 and X_5 in the exercises below.

1. $X_1 = 3^2 + 6^2 + 9^2 + 12^2 + \ldots + 99^2$.

2. $X_i = \frac{A + B^C}{D}\sqrt{E \cdot X_{i-1}}$, $i = 2, 3, 4, 5$.

TEACH PROBLEM SET 6

Input - Output (without formats)



LOAD-AND-GO SPS WITH MONITOR CONTROL

Kenneth M. Lochner and Glenn R. Ingram

MSC ASSEMBLY SYSTEM:

A LOAD-AND-GO SPS WITH MONITOR CONTROL

- Kenneth M. Lochner and Glenn R. Ingram

I.) INTRODUCTION

This paper will discuss a monitored, load-and-go type assembler developed at Montana State College and the conditions that prompted its development. It is a report on work done by Ken Lochner, formerly of the MSC Computing Center, and soon to assume duties as chief programmer at the Dartmouth Computation Center, in the actual writing of the processor.

To suggest some of the background reasons for this processor, it is well to admit that I am a relative newcomer to 1620 ranks in completing my second academic year with a 1620, after leaving a 709 installation. Anyone who follows this path finds himself wondering why in the world he did, and then develops a feeling somewhat akin to the fellow who had a job with a circus. This particular job consisted of following behind the animals during a parade, and cleaning the street with his little shovel. After an especially trying day, he complained so bitterly that his wife asked, "If it's so bad, why don't you quit and get another job?" The man replied, "What! And get out of show business?"

If the analogy isn't exact, it may be suggestive that some things could be cleaner in the 1620 tent.

II.) BACKGROUND AND MOTIVATION

To indicate some of the motivation for the work discussed here, Montana State College has, since 1958, offerred courses in symbolic programming. This programming is not viewed as an end in itself, but as a basis for the presentation of somewhat more sophisticated topics. This was a necessity while the computer was a 650, and the practice has continued with the Model I 1620, and more recently, the Model II. The clear advantages of FORTRAN have caused a change in the philosophy of the introductory course and in the development of library routines.

The ease of teaching FORTRAN, plus its relative machine independence and the ability of the casual user to obtain answers to real problems rather quickly, make it the logical choice for an introductory language. The newcomer is somewhat surprised to discover that FORTRAN will also produce a faster object program than SPS--despite the standard folklore--but is inclined to believe that a symbolic programming system still has a place in the academic world.

We feel that a symbolic system should be taught to students who have mastered the compiler language, and are interested in more depth in the computer. However, even the most innocent SPS devotee must admit that the pre-disk system was, at best, miserable from a teaching standpoint. Without entering into a discussion of whether SPS is a model of the mold in which symbolic systems should be cast, suffice it to say that there are rather obvious drawbacks to its use by a class of students.

Having mentioned the pre-disk system, let it be stressed that the system to be discussed, despite the "monitor" in the title, does not require disks, and has no relation to the disk monitor. Our 1620 has 60K storage and the standard features of the Model II, but the system will operate on a 40K Model I with automatic divide, indirect addressing and card input-output.

III.) CRITERIA FOR A GOOD ASSEMBLY SYSTEM FROM A TEACHING STANDPOINT

Returning to the motivation for this system, let it be noted that anyone who has taught a symbolic system to beginning programmers is aware that syntax and logical errors abound in the programs they produce. One can visualize the standard scene in a 1620 installation: a group of students loading the assembler, loading and unloading the punch hopper, entering the object deck, watching the typewriter anxiously, and then staring in increasing bewilderment at a machine which has halted, cleared or is in an infinite loop.

This scene, repeated many times, has its effect on the patience and morale of both students and teacher.

The elimination of this scene provides a good starting point in considering what is desirable in a symbolic system. We would submit that the essential points are:

1.) The ability to process a number of programs in a short time, which requires
    a.) Load and Go operation
    b.) Batch assembly

and 2.) Good diagnostics
    a.) During assembly
    b.) During program execution.

These considerations suggest that the system must always be in core, and provide the third condition:

3.) A "student-proof" processor; i.e., one that virtually cannot be erased.

IV.) FEATURES OF THE MSC ASSEMBLY SYSTEM

To show how these considerations dictated the type of processor developed, and to reinforce the assertion that it was superior to the regular SPS for teaching (and debugging), we list some features of this system beside those of the standard one.

Feature	MSC	Regular SPS
Language	Modified SPS-09	SPS-09 or -20
Operation	Load-and-go, batching	Two passes of source plus load object program
Assembly Diagnostics	Cards	Typewriter
Basic Floating Point Operations	Built-in (rewritten)	Additional deck to load
Execution Error Diagnostics	22 detected	None
Processor	Always in core	In and out

This table gives the essential comparison of pertinent points. To continue, under the MSC system, the program can be traced during execution (at the operator's option, by a console switch), execution can be halted by turning on a console switch, and a maximum number of instructions that a program can execute may be preset: if the program exceeds this number, the monitor will so indicate, and proceed to the next job. Hence, the only output will be the error messages, results of a trace, or answers computed by the program. Thus, the time consuming card handling of processor and object deck is eliminated, as well as the annoyance of punched output after errors have been detected. Also, the MSC processor has been made as nearly "student-proof" as possible. Once in two quarters the system was wiped out of core in a rare situation associated with a divide command, and we feel that our students are as inventive as anyone's.

V.) OPERATION OF THE SYSTEM

The operation of this system can be divided into three phases: a rather trivial "Job Card" search, the assembly of the source program, and the execution of the program under monitor control.

Each of the stacked programs must be preceded by a "Job Card," typically containing the student's name and program description. This card is identified by an asterisk in column 80, and the processor reads cards until such a card has been found. The job card is reproduced and assembly of the source program begins. The purpose of this is, of course, to allow this title card to be differentiated from data cards which may be stacked behind the previous program.

The assembly of the source program proceeds as in the SPS-09 assembler, except that

1.) During the equivalent of pass 1, the source statements are stored in core,
2.) During the equivalent of pass 2, the source statements are assembled,
3.) Diagnostic errors are punched,

and 4.) If syntax errors were discovered during assembly, an "End of Job" card is punched, and the processor returns to the job card search phase.

If no syntax errors were detected during assembly, monitor control of program execution is initiated.

The primary purpose of the monitor part of the processor is to provide an indication of execution errors while protecting the processor. After a program has been assembled, the monitor successively examines each instruction for possible execution errors, finds equivalent direct addresses, and if there are no errors, executes the instruction. The examination precludes executing an instruction that would, for example, carry out arithmetic on a field containing a record mark, operate on a field or record that hadn't been flagged or marked, etc.

This process continues until

1.) Program execution is complete,
2.) An error is detected,
3.) The number of instructions executed exceeds the number allowed,

or 4.) The operator terminates program execution by a console switch.

Any one of these four conditions causes an "End of Job" card to be punched, and the processor returns to the job card search phase. Any of the last three conditions causes an error message to be punched before the "End of Job" card.

VI.) SUMMARY OF CLASS USE

In evaluating the effectiveness of such a system, one is tempted to compare the results of student use with those of students using another system on a different computer; e.g., SOAP on the 650 or FAP, say, on the 709 or 90. However, such a comparison is probably more a reflection of the particular computer than of the programming system, because of the differences in size and complexity of the command structure, and the storage available for a processor.

Perhaps it is more meaningful to indicate that a class of 25 students was able to make effective use of this system to learn symbolic programming for the 1620. Each student was able to have several trials with his program in a two hour period. The programs written varied from a beginning one that simply read cards, rearranged fields, and punched cards, to rather complex ones. Other simple programs involved the various standard arithmetic operations and the floating point routines, and finding roots of polynomials. Other programs included random number generators, subsequent tests for randomness, Monte Carlo evaluation of an integral, square root subroutines, simulation of index registers, and some basic "building block" routines which will be used in the simulation of a symbolic system for a fixed word length computer.

VII.) CONCLUSION

Summing up, it may be gathered that we have been pleased with this processor. It has been useful in debugging programs other than those written by students. Its effectiveness has been purchased at the cost of the time used in the detailed scrutiny of the monitor. Balancing that cost against the time that would be used in the extensive card handling of the regular systems, the gain in convenience and the ability to handle many programs in one period,has made this processor a real contribution to our classes.

With the oncoming surge of disk storage, our interest in this system may sound comparable to the enthusiasm of the folks who invested in buggy whips just before the car was invented. If so, someone should have developed such a system three years earlier.

We feel this may have a place, even with disks, so it has been submitted to the program library, in four forms. Certainly it has been well-nigh foolproof and indestructible in core, thus providing the rapid on-off form desired. It may be obsolete, but we are too old to be innocent, and we have learned to be skeptical of Greeks bearing gifts and manufacturers bearing software.

EXAMPLES OF 1620 USE IN COLLEGE ADMINISTRATION

Noel T. Smith

INDIANA STATE COLLEGE

TEST SCORE ANALYSIS

Wayne E. Hoover
Computer Center
Indiana State College
Terre Haute, Indiana
May, 1964

DECK KEY

Deck 1	Source Program
Deck 2	Object Program
Deck 3	Sample Data
Deck 4	Sample Output (sorted alphabetically)

Indiana State College

Test Score Analysis

TABLE OF CONTENTS

ABSTRACT

Title: Indiana State College Test Score Analysis

Author: Wayne E. Hoover
Computer Center
Indiana State College
Terre Haute, Indiana

Purpose/Description: Suppose a student tells you that he received a score of 36 on a particular examination. This alone does not indicate his achievement in relation to the rest of the class. Before one can interpret such a number, more information is needed. It becomes evident that the degree to which the results of measurement are useful is proportional to the accuracy and thoroughness with which there results are analyzed.

In order to effectively evaluate the attainment of his students accurately and fairly, a professor must have adequate measurement techniques at his desposal. He must know how to use them properly, and how to interpret results obtained by their use. However, an unfortunate characteristic of evaluative techniques is that they are very time consuming.

It is the purpose of this project to write a program for the IBM 1620 Computer which will relieve the professors and their assistants of the burden of manually calculating from a given set of a maximum of 500 test scores, the arithmetic mean, standard deviation, alpha-three, alpha-four, reliability of the test, the rank, percentile rank, z-score, and t-score for each student.

EQUATIONS USED

Alpha-three:

$$\text{Alpha-3} = \frac{1}{n\sigma^3} \sum_{i=1}^{n} (X_i - X)^3$$

Alpha-four:

$$\text{Alpha-4} = \frac{1}{n\sigma^4} \sum_{i=1}^{n} (X_i - X)^4$$

Arithmetic Mean:

$$X = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Percentile Rank:

$$PR = \frac{100\,(n - \text{rank})}{n}$$

Reliability of Test:

$$R_t = \frac{I \quad ^2 - X\,(I - X)}{\sigma^2 \quad (I - 1)}$$

Standard Deviation:

$$= \sqrt{\frac{1}{N} \sum_{i=1}^{n} (X_i - X)^2}$$

T-Score (Sigma Score):

$$\text{T-Score} = 50 + 10\,(\text{Z score})$$

Z-Score (Standard Score):

$$\text{Z-Score} = \underline{X_i - X}$$

Legend:

I = Total Possible Points
n = Number taking the test
$N = n$ if $n \geq 30$
$n - 1$ if $n < 30$
R_t = Reliability of the test
= Standard Deviation
X = Arithmetic Mean
X_i = Individual Score

REFERENCES

Alpha-three:	p. 102[1]
Alpha-four:	p 104[1]
Arithmetic Mean:	p. 45[1]
Percentile Rank:	p. 37[2]
Reliability of Test:	p. 152[2]
Standard Deviation	p. 85[1]
T-Score	p 44[2]
Z-Score	p. 44[2]

---

[1] John E. Fruend, Modern Elementary Statistics (Englewood Cliffs, 1963), p. (indicated above).

[2] Victor H. Noll, Introduction to Educational Measurement (Boston, 1957), p. (indicated above).

COMMENTS

Alpha-three: A widely used measure of skewness and symmetry is alpha-three. When alpha-three is greater than zero, the tail of the distribution is skewed to the right, and the distribution is said to be positively skewed; it is negatively skewed if the tail is at the left and alpha-three is less than zero. The distribution is perfectly symmetrical when alpha-three is exactly equal to zero.

Alpha-four: Peakedness or kurtosis is commonly described by alpha-four. When alpha-four is exactly equal to three, the distribution is said to have the familiar bell shape of the normal distribution. When alpha-four exceeds three, the distribution is very peaked and has a relatively wide tail; the distribution is flat in the middle and has a relatively thin tail when the value of alpha-four is less than three.

Arithmetic Mean: The arithmetic mean is commonly known as the average. It is the sum of all the scores divided by the number of persons taking the test.

Percentile Rank: The percentile rank is a derived score stated in terms of the percentage of examinees in a specified group who fall below a given score point. In other words, it describes a person's relative standing within a particular group. Thus, a percentile rank of 80 means that a person's score was equal to or higher than the scores made by 80 percent of the people in a specified group.

Rank: The student with the highest score receives a rank of 1; the next highest score receives a rank of 2, and so on. The student with the lowest score thus receives the highest rank. In care of ties in rank, we assign to each of the tied observations the mean of the ranks which they jointly occupy. Hence, if two scores are tied for ranks 4 and 5, we assign each rank 4 1/2; if three observations are tied for ranks 10, 11, and 12, we assign each rank 11.

Reliability of the Test: This is a measure of the consistency with which a test measures what it is intended to measure; it enables one to determine how much reliance he can place on the scores yielded by the test. This particular formula is accurate enough for classroom tests and other ordinary situations; however, it usually gives an underestimate of the true reliability.

Standard Deviation: In most distributions, approximately 34 per cent of the scores lie between the mean or average and a point or score that is one standard deviation away from the mean (in either direction). In other words, approximately two-thirds of the cases or scores will fall within one standard deviation of the mean.

T-Score (Sigma Score): This is a standard score which has a mean of 50.0 and a standard deviation of 10.0. It is obtained by adding 50 to 10 times the z-score. For example, the z-score -1.4 and 1.4 correspond to the t-scores 36 and 64, respectively. It has been said that t-scores are the best method that has been devised for direct comparison of individual test results.

Z-Score (Standard Score): The basic standard score is the z-score. It tells in simple terms the difference or distance between a stated group's mean and any specified raw-score value. The mean is 0.0 and the standard deviation is 1.0.

For example, suppose a student had a score of 49 and he is to be compared with his class. Given that the mean and standard deviation are 40.0 and 6.0, respectively, we find his z-score is (49 - 40)/6 = 1.5, which corresponds to a t-score of 50 + 10 (1.5) = 65. In other words, this student's score is 1.5 standard deviations above the mean. This may also be interpreted to mean that he scored better than about 93 per cent of the group.

PREPARATION OF DATA

Data Cards: Each IBM data input card (except the header card) must be prepared in the following manner:

Columns	Field
1-5	Student Number
7-27	Name (see Exceptions, below)
28-33	Score (floating-point with decimal in column 32; may have one place to right of decimal in column 33)
40	0
46-47	0.
55-56	0.
64-65	0.
72-73	0.

Header Card: The first card of the data must contain in columns 28-33, the total possible points. Columns 40-73 are punched as stated above.

Sorting: The data cards are then sorted in descending order, according to columns 28-33. After sorting is completed, the header card is then placed on top of the deck; it is the first data card.

Limations: The program will handle a maximum of 500 individual test scores.

Exceptions: The program is so designed that columns 1-27 of the data input cards may be left blank without affecting the rest of the program.

OUTPUT

Printed Output: The arithmetic mean, standard deviation, alpha-three, alpha-four, and the reliability of the test are printed out first (in that order.) This information is printed regardless of the switch settings.

The remaining information may or may not be printed out, depending on the switch settings.

The heading, student number and name, score, number, rank, percentile rank, z-score, and t-score is printed out next.

Finally, each line of typed output contains the following information in the indicated order: student number, name, score, a "counter" number, rank, percentile rank, z-score, and t-score. Students receiving the same score are grouped together; the typewriter double spaces between groups.

When the last card has been processed, the word STOP is printed.

Punched Output: The first seven cards contain the arithmetic mean, standard deviation, alpha-three, alpha-four, the reliability of the test, and two blank cards. This information is punched in columns 1-35.

The Heading: Name, Scre, No, Rnk, Ptrk, Z-Scre, and T-Scre is punched on the eighth card.

The remaining cards contain the following information

Columns	Field
1-5	Student Number
7-27	Name
28-33	Score
38-40	A "counter" number
44-49	Rank
54-59	Percentile Rank
63-68	Z-Score
71-76	T-Score

Sorting Punched Output: The punched output cards may be sorted alphabetically by sorting in ascending order according to columns 1-5. (This assumes that the student numbers are assigned in sequence when the last names are in alphabetical order). The first eight cards of the output will be rejected by the sorter because columns 1-5 are blank; however, save these cards to process through the Alphabetic Interpreter.

Alphabetic Interpreting of Punched Output: The information contained in the punched output cards may be printed at the top of each card by means of the IBM 548 Alphabetic Interpreter.

The alphameric interpreter board should be wired as follows:

Wire Read Brushes	to	The Print Entries
7-35		1-29
38-40		30-32
44-49		34-32
54-59		41-46
63-68		48-53
71-76		55-60

MACHINE REQUIREMENTS

Equipment Specifications: IBM 1620, 1622

Storage Requirements: 20K

Source Language: Afit FORTRAN

Special Features: None

PROGRAM EXECUTION

Step-by-Step Procedure: Assume that the object deck is compiled and at hand. Then ---

1. Clear 1622 Card Read Punch.

2. Load Afit Fortran loader, object deck, and the Afit Fortran Subroutines in the 1622.

3. Check Switches.

PARITY Check Switch	STOP
I/O Check Switch	STOP
OVERFLOW Check Switch	PROGRAM

4. Press RESET and Reader LOAD.

5. Set Program Switches for desired option.

   Switch 2 ON for PUNCHED output.

   Switch 2 and 3 OFF for PRINTED output.

   Switch 2 and 3 ON for PRINTED and PUNCHED output.

   Switch 1 and 4 OFF.

6. Press START

7. Place data cards in the 1622 and press READER START.

8. Press PUNCH START if necessary.

9. Typewriter prints and/or 1622 punches.

10. Repeat Step 7. (i.e. two passes are required).

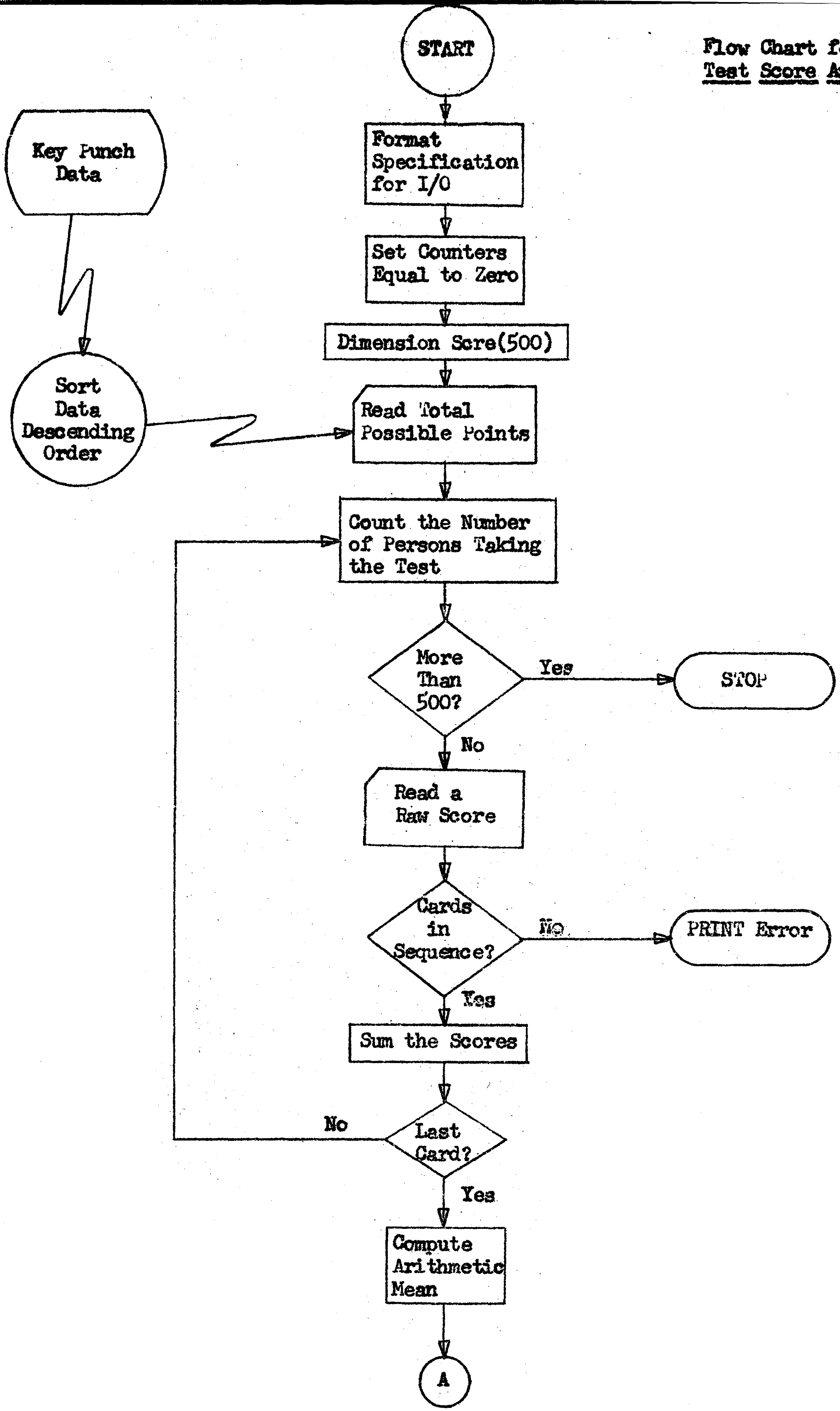
11. Typewriter prints and/or 1622 punches. When the last card has been processed, the word STOP is typed.

Expected Stops: If the data is not sorted in descending order, a "card out of sequence" message will be printed out and the program will branch to the STOP command. The restart procedure is then necessary.

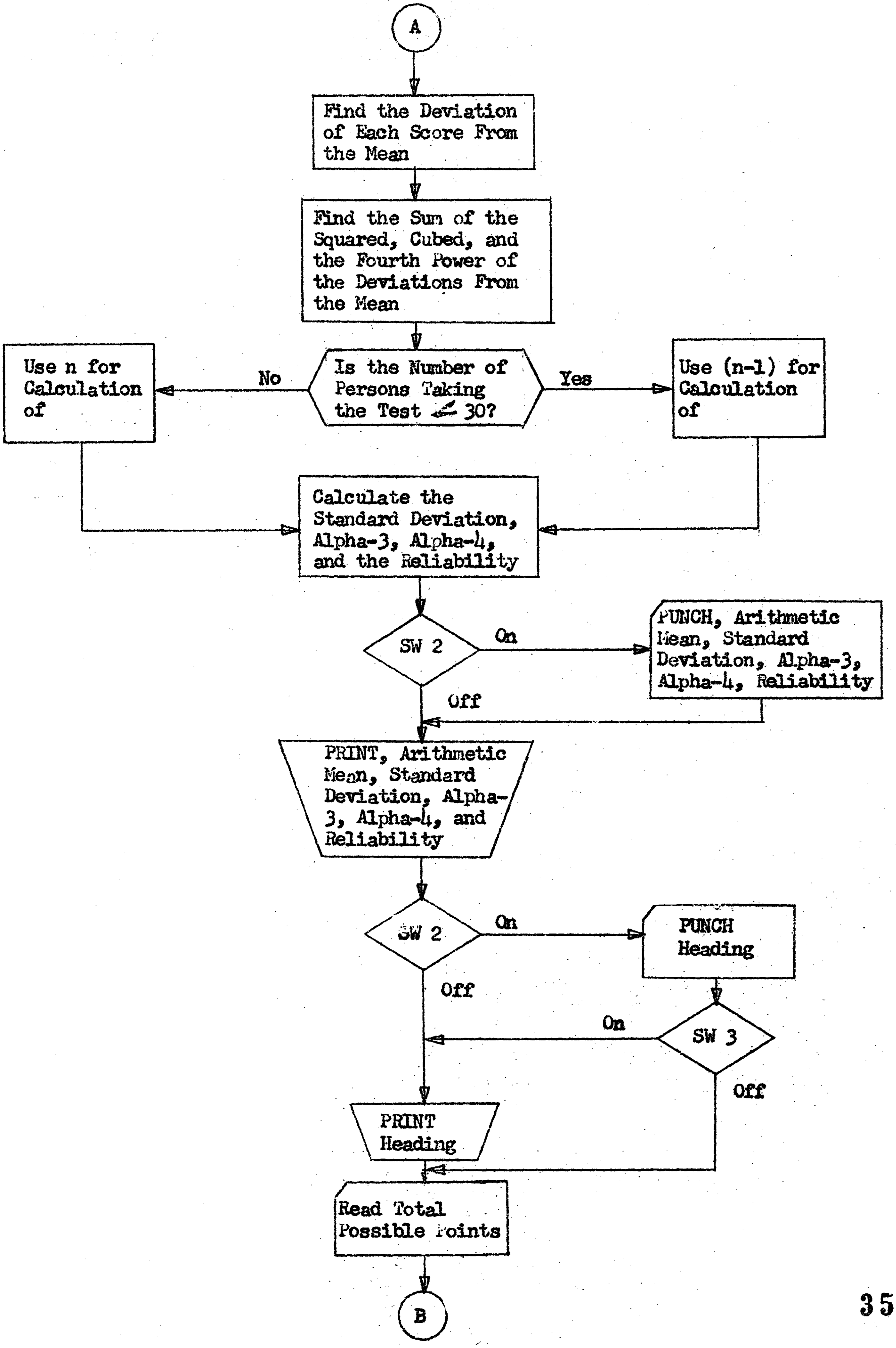
The computer will automatically stop if more than 500 scores are read in.

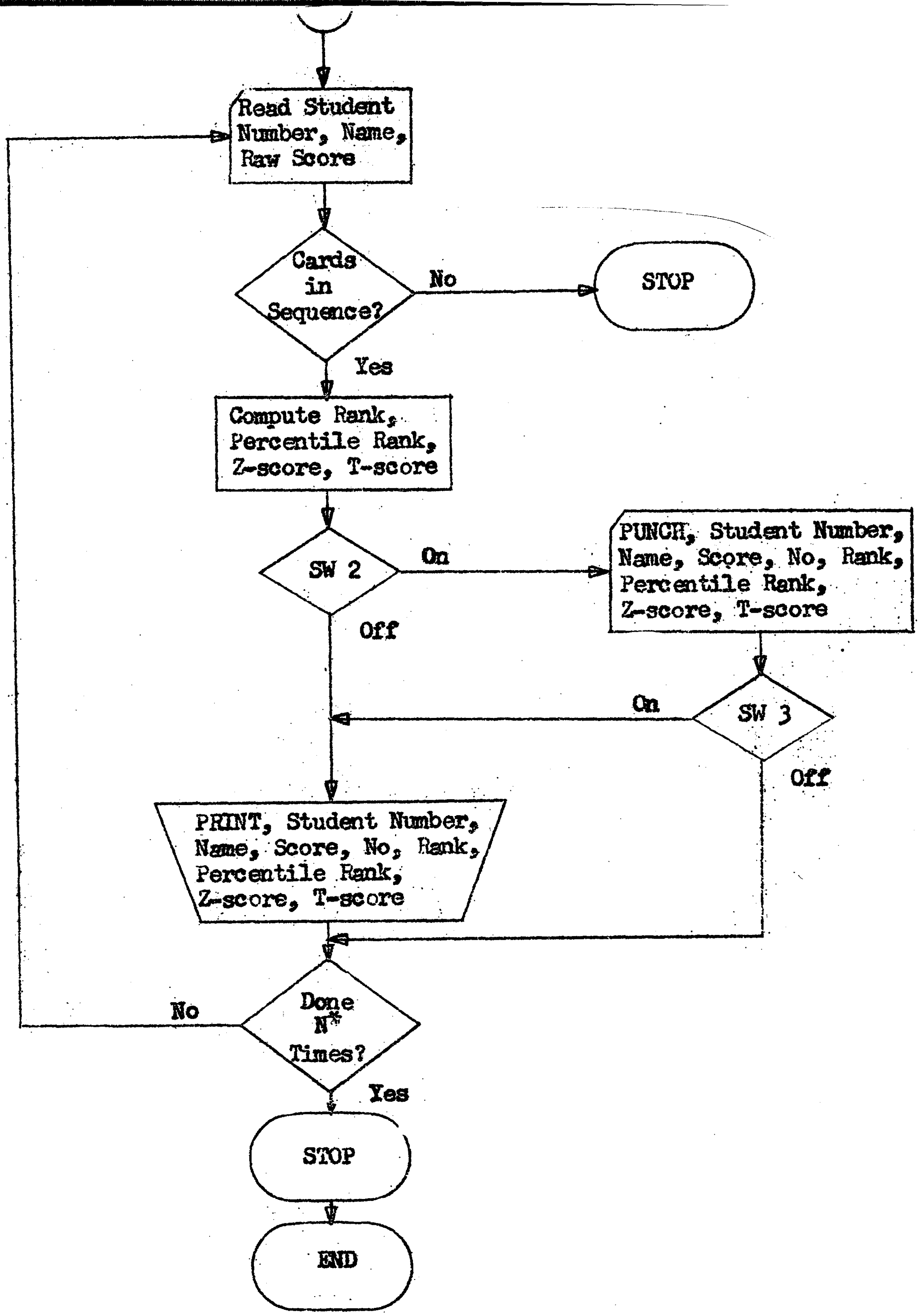
Restart Procedure: Press INSTANT STOP, RESET, INSERT, RELEASE, START. Repeat the step-by-step procedure for program execution.

Flow Chart for
Test Score Analysis



354


A
Find the Deviation of Each Score From the Mean
Find the Sum of the Squared, Cubed, and the Fourth Power of the Deviations From the Mean
Is the Number of Persons Taking the Test ≤ 30?
No
Yes
Use n for Calculation of
Use (n-1) for Calculation of
Calculate the Standard Deviation, Alpha-3, Alpha-4, and the Reliability
SW 2
On
Off
PUNCH, Arithmetic Mean, Standard Deviation, Alpha-3, Alpha-4, Reliability
PRINT, Arithmetic Mean, Standard Deviation, Alpha-3, Alpha-4, and Reliability
SW 2
On
Off
PUNCH Heading
SW 3
On
Off
PRINT Heading
Read Total Possible Points
B



*Where N equals the number of persons taking the test.

```
0̄8000 C   TEST SCORE ANALYSIS                              WAYNE E HOOVER
0̄8000 C   COMPUTER CENTER                                  INDIANA STATE COLLEGE
0̄8000 C   TERRE HAUTE, INDIANA                             MAY, 1964
0̄8000 100 FORMAT(27X,F6.1,I7,F9.2,F10.3,F9.3,F8.3)
0̄8108 390 FORMAT(6X4HNAME,19X4HSCRE,5X2HNO,5X3HRNK,5X4HPTRK,6X12HZSCRE  TSCRE)
0̄8360 442 FORMAT(6X19HSTUDENT NO AND NAME,3X,5HSCORE,5X,2HNO,5X,4HRANK)
0̄8532 443 FORMAT(11H  PCNT RANK,8H  Z-SCRE,8H  T-SCRE)
0̄8624 776 FORMAT(/)
0̄8646 778 FORMAT(//)
0̄8674 889 FORMAT(6X18HARITHMETIC MEAN IS F11.3)
0̄8758 900 FORMAT(6X22HSTANDARD DEVIATION IS F7.3,//6X11HALPHA 3 IS F18.3)
0̄8914 901 FORMAT(6X11HALPHA 4 IS F18.3,//6X23HRELIABILITY OF TEST IS F6.3)
0̄9072 999 FORMAT(//20HCARD OUT OF SEQUENCE)
0̄9146     SSCRE=0.0
0̄9170     OLD=1000.0
0̄9194     SAFA3=0.0
0̄9218     SAFA4=0.0
0̄9242     SDEV=0.0
0̄9266     N=0
0̄9290     DIMENSION SCRE(500)
0̄9290     READ 100,TPTS,Z1,Z2,Z3,Z4,Z5
0̄9374  13 N=N+1
0̄9410      IF (N-500)14,14,99
0̄9478  14 READ 100,SCRE(N),Z1,Z2,Z3,Z4,Z5
0̄9586      IF (OLD-SCRE(N))16,18,18
0̄9674  16 PRINT 999
0̄9698     GO TO 99
0̄9706  18 SSCRE=SSCRE+SCRE(N)
0̄9766     OLD=SCRE(N)
0̄9814      IF (SENSE SWITCH 9)21,13
0̄9834  21 TNMBR=N
0̄9858     AMEAN=SSCRE/TNMBR
0̄9894     NMBR=N
0̄9918     DO 30 N=1,NMBR
0̄9930     VAR=SCRE(N)-AMEAN
0̄9990     SDEV=SDEV+VAR**2
1̄0038     SAFA3=SAFA3+VAR**3
1̄0086  30 SAFA4=SAFA4+VAR**4
1̄0170      IF (N-30)37,37,35
1̄0238  35 TOTAL=N
1̄0262     GO TO 38
1̄0270  37 TOTAL=N-1
1̄0306  38 STDV2=SDEV/TOTAL
1̄0342     STDV=SQRT(STDV2)
1̄0366     ALFA3=SAFA3/(TNMBR*STDV**3)
1̄0426     ALFA4=SAFA4/(TNMBR*STDV**4)
1̄0486     REL=(TPTS*STDV2-AMEAN*(TPTS-AMEAN))/(STDV2*(TPTS-1.0))
1̄0654      IF (SENSE SWITCH 2)85,89
1̄0674  85 PUNCH 889,AMEAN
1̄0698     PUNCH 900,STDV,ALFA3
1̄0734     PUNCH 901,ALFA4,REL
1̄0770  89 PRINT 776
```

```
T0794       PRINT 889,AMEAN
T0818       PRINT 778
T0842       PRINT 900,STDV,ALFA3
T0878       PRINT 778
T0902       PRINT 901,ALFA4,REL
T0938       PRINT 778
T0962       IF (SENSE SWITCH 2)39,42
T0982    39 PUNCH 390
T1006    41 IF (SENSE SWITCH 3)42,44
T1026    42 PRINT 778
T1050       PRINT 442
T1074       PRINT 443
T1098       PRINT 776
T1122    44 OLD=1000.0
T1146       READ 100,TPTS,Z1,Z2,Z3,Z4,Z5
T1230       DO 83 N=1,NMBR
T1242       READ 100,SCRE(N),Z1,Z2,Z3,Z4,Z5
T1350       J=N
T1374       IF (N-NMBR)51,63,99
T1442    51 IF (SCRE(N)-SCRE(N+1))16,52,62
T1554    52 IF (J-1)99,54,53
T1622    53 IF (SCRE(N-1)-SCRE(N))16,64,54
T1734    54 SAVEN=N
T1758    55 N=N+1
T1794       IF (N-NMBR)57,58,99
T1862    57 IF (SCRE(N)-SCRE(N+1))16,55,58
T1974    58 THISN=N
T1998       ADD=(THISN-SAVEN)/2.0
T2046       SRANK=SAVEN+ADD
T2082       GO TO 64
T2090    62 IF (J-1)99,66,63
T2158    63 IF (SCRE(N-1)-SCRE(N))16,64,66
T2270    64 RANK=SRANK
T2294       GO TO 67
T2302    66 RANK=N
T2326    67 N=J
T2350       PRANK= ((TNMBR-RANK)*100.0)/TNMBR
T2410       ZSCRE=(SCRE(N)-AMEAN)/STDV
T2482       TSCRE=ZSCRE*10.0+50.0
T2530       IF (SENSE SWITCH 2)73,75
T2550    73 PUNCH 100,SCRE(N),N,RANK,PRANK,ZSCRE,TSCRE
T2658       IF (SENSE SWITCH 3)75,82
T2678    75 IF(OLD-SCRE(N))16,76,78
T2766    76 PRINT 776
T2790       GO TO 79
T2798    78 PRINT 778
T2822    79 PRINT 100,SCRE(N),N,RANK,PRANK,ZSCRE,TSCRE
T2930    82 OLD=SCRE(N)
T2978    83 CONTINUE
T3014    99 STOP
T3022       END
END OF COMPILATION T3541T3810
```

Legend: Source Program

ALFA 3 = alpha-three

ALFA 4 = alpha-four

AMEAN = arithmetic mean

NMBR = total number of persons taking the test

PRANK = percentile rank

REL = reliability of the test

SAFA 3 = $\sum_{i=1}^{n} (Xi - \bar{X})^3$

SAFA 4 = $\sum_{i=1}^{n} (Xi - \bar{X})^4$

SCRE = individual test score (raw score)

SDEV = $\sum_{i=1}^{n} (Xi - \bar{X})$

SSCRE = $\sum_{i=1}^{n} Xi$

STDV = standard deviation = σ

STDV 2 = σ^2

STDV 3 = σ^3

STDV 4 = σ^4

TNMBR = total number of persons taking the test.

TOTAL = TNMBR - 1.0 if TNMBR $\leqq$ 30

= TNMBR if TNMBR $>$ 30

TPTS = total possible points or total number of test items, which ever is greater.

TSCRE = T-SCRE

VAR = $(Xi - \bar{X})$

ZSCRE = Z-SCORE

Z1 - Z5 = Dummy Variables

```
      ARITHMETIC MEAN IS      82.256

      STANDARD DEVIATION IS  11.094

      ALPHA 3 IS              -.828

      ALPHA 4 IS              3.695

      RELIABILITY OF TEST IS   .890


      STUDENT NO AND NAME   SCORE      NO      RANK   PCNT RANK   Z-SCRE   T-SCRE

12550 BUSH DONALD JOSEPH     98.0       1      1.50     96.153    1.419   64.190
57795 MICKSCHL JAMES HAROLD  98.0       2      1.50     96.153    1.419   64.190

28336 FOSLER LARRY RICHARD   96.0       3      4.50     88.461    1.238   62.387
65600 PENNINGTON GAIL LEA    96.0       4      4.50     88.461    1.238   62.387
84890 TATEM JOHN DAVID       96.0       5      4.50     88.461    1.238   62.387
68090 PROPST DARRELL         96.0       6      4.50     88.461    1.238   62.387

74390 SCHAEFER JAMES MARTIN  95.0       7      7.00     82.051    1.148   61.486

42645 JENSEN JAY WAYNE       94.0       8      8.00     79.487    1.058   60.585

74565 SCHERB JOHN FREDERICK  92.0       9      9.00     76.923     .878   58.782

25890 EVANS MELVIN ERNEST    88.0      10     13.00     66.666     .517   55.177
33150 GRIFFITH JOHN H        88.0      11     13.00     66.666     .517   55.177
43336 JOHNSON STEPHEN R      88.0      12     13.00     66.666     .517   55.177
44700 KEEGAN GARY LEE        88.0      13     13.00     66.666     .517   55.177
47600 KRENKE GLEN LEE        88.0      14     13.00     66.666     .517   55.177
48006 KUYKENDALL JOHN A      88.0      15     13.00     66.666     .517   55.177
53125 MAPES REX MARSHALL     88.0      16     13.00     66.666     .517   55.177

03050 BAKER DOYLE SIMON      86.0      17     18.00     53.846     .337   53.374
70260 RICE JAMES HAROLD      86.0      18     18.00     53.846     .337   53.374
75778 SEIM KENNETH BRUCE     86.0      19     18.00     53.846     .337   53.374

54750 MC ATEE DON SCOTT      84.0      20     20.50     47.435     .157   51.571
55975 MC DOWELL MARSHA G     84.0      21     20.50     47.435     .157   51.571

39370 HOFFMAN DAVID CHARLES  82.0      22     22.50     42.307    -.023   49.768
76548 SHEEHAN JAMES EVERETT  82.0      23     22.50     42.307    -.023   49.768

60600 MOYER MICHAEL A        80.0      24     24.50     37.179    -.203   47.966
77387 SHOULDERS MICHAEL C    80.0      25     24.50     37.179    -.203   47.966

17800 CORBIN JAMES B         78.0      26     26.50     32.051    -.383   46.163
24965 ELLER LARRY ALLEN      78.0      27     26.50     32.051    -.383   46.163
```

```
24310 ECHARD WM RANDALL        76.0    28    29.50    24.358     -.563   44.360
64265 PAIGE WANDA ELAINE       76.0    29    29.50    24.358     -.563   44.360
64890 PATRICK GERRY WAYNE      76.0    30    29.50    24.358     -.563   44.360
 2725 WILDER RICHARD LYNN      76.0    31    29.50    24.358     -.563   44.360

55475 MC COY CAROL RUTH        74.0    32    32.00    17.948     -.744   42.558

22977 DOUGHERTY CHARLES E      70.0    33    33.50    14.102    -1.104   38.952
25383 EMMERT ROBERT CARL       70.0    34    33.50    14.102    -1.104   38.952

02025 ARMSTEAD ROBERT LEE      69.0    35    35.00    10.256    -1.194   38.051

69130 RAUSCH MICHAEL           64.0    36    36.50     6.410    -1.645   33.544
73936 SANDERS JAMES W          64.0    37    36.50     6.410    -1.645   33.544

62097 NEWHARD NANCY FAYE       62.0    38    38.00     2.564    -1.825   31.741

28065 FOLTA JAMES VINCENT      48.0    39    39.00      .000    -3.087   19.122
STOP
```

A

PROPOSAL FOR AN ADDITION IN GRADE REPORTING PROCEDURES TO ALLOW FOR AUTOMATIC PROCESSING OF PROBATION STUDENTS

Indiana State College
Computer Center
May 5, 1964

363

FLOW

OF

PROCEDURE

Explanation of Flow Chart

A All card files are merged together in alphabetical sequence by student number.

B Computer writes grade reports and updates student index file. Probation cards for failing students. Store used cards.

C Grade reports to students and school officials.

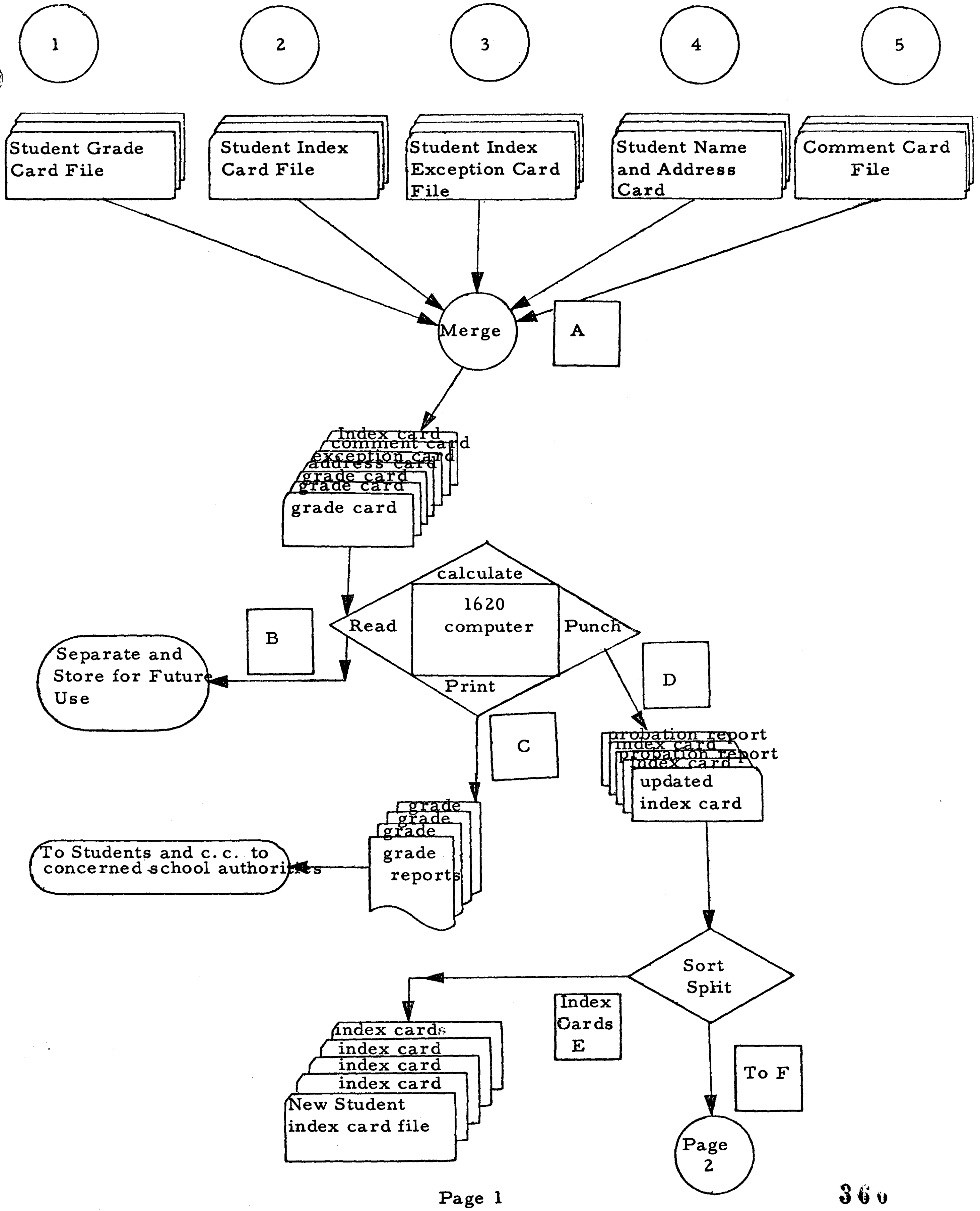
D Output: New student index cards and probation report writing cards.

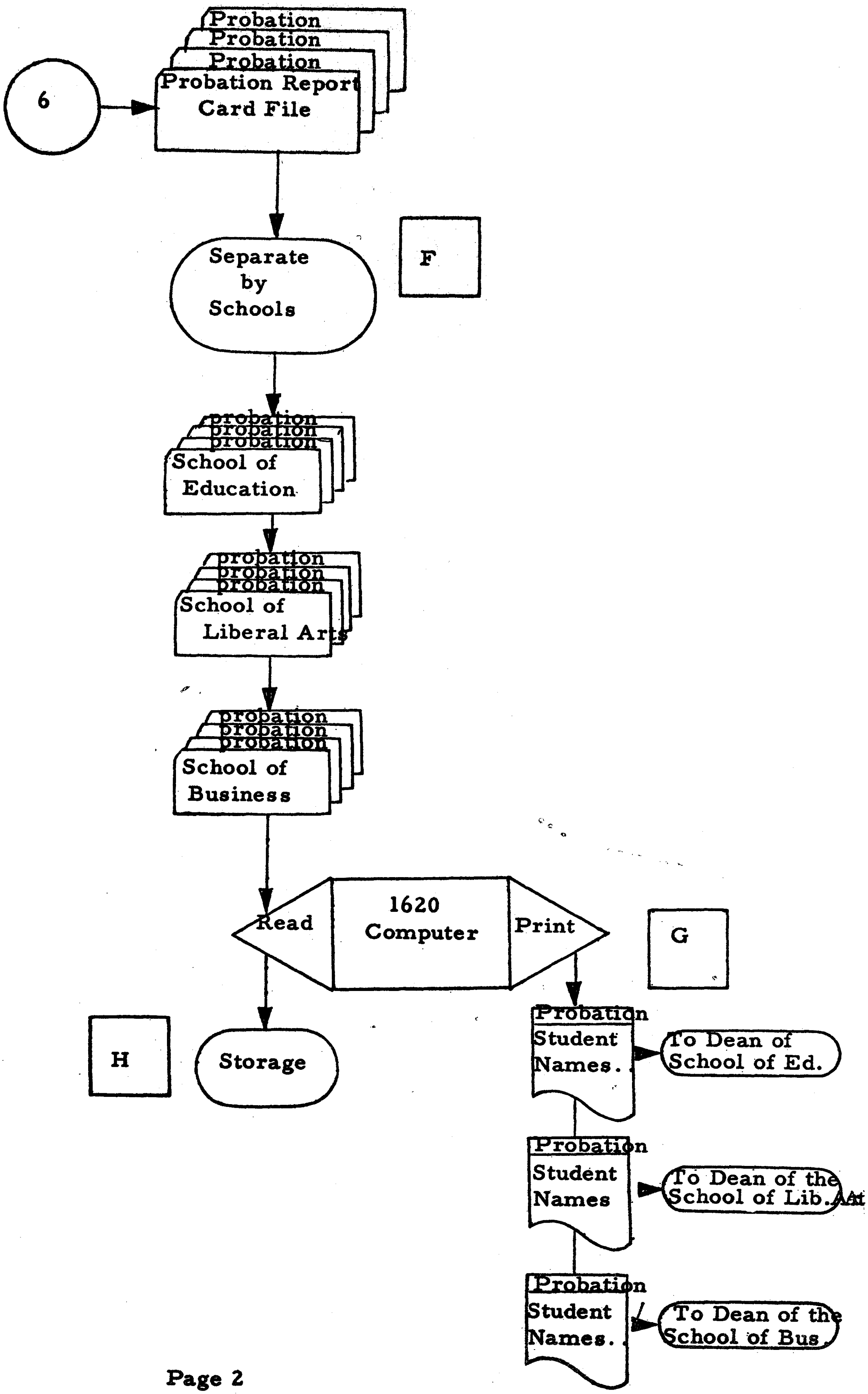
E New index cards go back into file for report use.

F Probation report cards are sorted by school.

G Probation reports are printed and sent to the deans of the schools.

H Probation cards are stored.


1
2
3
4
5
Student Grade Card File
Student Index Card File
Student Index Exception Card File
Student Name and Address Card
Comment Card File
Merge
A
Index card
comment card
exception card
address card
grade card
grade card
grade card
calculate
Read
1620 computer
Punch
Print
B
Separate and Store for Future Use
C
D
probation report
index card
probation report
index card
updated index card
grade
grade
grade
grade reports
To Students and c.c. to concerned school authorities
Sort Split
Index Cards E
index cards
index card
index card
index card
New Student index card file
To F
Page 2


6
Probation
Probation
Probation
Probation Report Card File
Separate by Schools
F
probation
probation
probation
School of Education
probation
probation
probation
School of Liberal Arts
probation
probation
probation
School of Business
Read
1620 Computer
Print
G
H
Storage
Probation Student Names.
To Dean of School of Ed.
Probation Student Names
To Dean of the School of Lib. Art
Probation Student Names.
To Dean of the School of Bus.

CARD

JUSTIFICATION

Explanation of Card File

After completion of Step A (shown by the flow chart), the input to the computer consists of multiple card groups, one per student. It is the purpose of this report to show the need for including each type of card.

The speed and efficiency of any data processing procedure are largely dependant upon the volume of data to be processed. Because card volume is of such importance, it is to the users advantage to keep it at a minimum.

The five types of input data cards necessary for the student academic progress reports follow. They are:

Card 1 - Student Class Grade Card

These cards enter the flow through the registration line. They are the yellow striped cards the student submits for each class he attempts. After registration, the cards are held until the instructors turn in their grades. Each grade, with the respective grade points, is entered on the correct grade card. The result of this activity leaves a workable file of all work completed on punched cards.

Card 2 - Student Index Card

A continuous file of student index cards is maintained by the Computer Center. This file records the complete scholastic history and present status of each student. It is this card that records the amount of college work completed, with the grade average earned for this work - on a cumulative basis, and also on a single semester basis.

Such information as where the student lives, what social organization does he belong to, the number of credit hours transferred in from other colleges, his first two major areas of study, his minor area of study, and the sex, is all recorded in a numerical code on this card.

It is the student index card that facilitates all reports on academic progress; from a report showing the current and cumulative index of each girl living on the third floor of Reeve Hall, to a report of the numbers of hours all Education majors carried any given semester.

Because this card is so vital to our work, and because the grade report contains both cumulative and current credit hours, grade points earned, and grade point ratio, it is necessary for this card to be re-computed at each semester's end.

Card 3 -- Student Required Index Exception Card

The purpose of this card file is to automate the detailed processing of students having scholastic problems. Because the student's academic progress is of upmost concern to the college, careful monitoring and guidance techniques are essential. The inclusion of this card greatly facilitates much of the detailed analysis work necessary.

The student index card allows the proper school authorities to carefully supervise the progress of a student. By submitting a probation form to the Computer Center, a school official can stipulate exactly what scholastic level of achievement must be met. This is done by simply stating what grade point ratio the student must earn, either on a cumulative, or semester basis. This information is then entered into the student's card group and allows the computer to analyze the student's work accordingly.

If the student fails to meet this requirement, the computer will generate a card from which a complete scholastic report can be written and sent to the appropriate official.

The card is labeled "exception" because, in the absence of such a card, the computer will use the standard required index schedule to analyze the student. See Probation Scaling.

Card 4 -- Student Name and Address Card

The name and address card allows for automatic addressing of the grade report.

Card 5 -- Comment Card

The comment card allows a school official a maximum of two lines (68 characters) of comment on the student grade report. Through the use of a comment code, the comment may be printed only if the student fails to meet specified grade conditions. It is also possible to print the comment under any conditions. See Comment Printing.

Output Data Cards

Two types of cards are generated by the computer. The first, an updated student index card, replaces the input student index card (see card 2 - student index card).

The second, a probation report card (card 6), is punched for every student falling below certain minimum grade average requirements.

The purpose of this card is to allow a file of cards to be maintained on all probation students. A more complete description of this will be found under Probation Report.

Comment Printing

The grade processing procedure utilizes two kinds of comments. The first type, those entered on the comment card, may be worded as the school official desires. However, because the probation report sent to the school authorities should explain what kind of comment was made, and who authorized the comment, the inclusion of an authority code, and of a comment classification code is necessary.

The classification of comments is as follows (without respect to wording):

Code	Description
1	Place on Probation
2	Contact the Dean of your school
3	Contact the Registrar
4	Withdrawn from school for scholastic reasons
5	Withhold Permit to Register
6	
7	
8	
9	Machine generated probation comment.

The authority codes are as follows:

Code	Description
1	Registrar
2	Dean of Students
3	Business Office
4	Dean of the School of Ed.
5	Dean of the School of Lib. Arts

6	
7	
8	
9	Computer Center

It should be noted that the presence of a comment does not necessarily mean a student is on probation.

A special punch over the comment code will suppress printing if the required index is met or exceeded. Thus, a code 1 and this special punch would read J, a 2 would read K, 3 - L, etc.

The second form of comment, machine-generated, are those printed on the grade report to notify the student he is being placed on probation, or that he is being removed from probation status. These comments will be indentified by a code 9.

Probation Scaling

As each student index card enters the computer during grade processing, the probation code found in card column 47 will be examined. A zero in this column indicates the student is not on probation this semester, any other digit means that the student was on probation last semester, i.e., a 3 would indicate a probation student for the 3 preceding semesters. If the student again fails to earn a satisfactory grade index, the probation code will be incremented by 1. If the student earns a satisfactory average, the probation code will be made zero.

As each student's grades are processed and the new cumulative hours and grade points are brought up to date, the computer will, in the absence of a required index exception card, scale the cumulative hours and find the required grade-point average. The scale is as follows:

Cumulative Hours	Required Grade-Point Average
0 - 16	1.00
17 - 32	1.25
33 - 45	1.50
46 - 60	1.80
61 +	2.00

Any student who does not meet or exceed this scale will be automatically placed on probation. The probation code for such students will be incremented and a "Probation Report Card" (card 6) will be generated.

The computer will notify such students of this condition by - in the absence of a comment card - printing on the grade report. See comment printing.

Special Provisions

If a "Required Index Exception Card" (card 3) is present, their grade-point average will be scaled as specified by this card.

CARD

FORMAT

Card
1

Student Class Grade Card

Card Column	Description
1-5	Student Number
6	School
7-27	Name
28	Classification
29	Curriculum
30-32	Number in Class
33-36	Department
37-40	Course Number
41-42	Section Number
43-57	Course Description
58-61	Course Number
62-65	Time Class Meets
66-70	Days Class Meets
71-72	Semester Code
73-74	Grade
75-77	Grade Points
79	Hours of Credit
80	Code 1

Card
2

Student Index Card

Card Column	Description
1-5	Student Number
6	School
7-27	Name
28-31	Cumulative Hours
32-35	Cumulative Points
36-38	Cumulative Grade Point Ratio
39-40	Semester Hours
41-43	Semester Points
44-46	Semester Grade Point Ratio
47	Probation Code
48	Grade Point Ratio
49	Hours
50-51	Housing
58-61	Total Hours Toward Graduation
62-63	Hall
64-65	House
66	Social Organization
67-69	Semester Code
70	New or Transfer Students
71	Teaching or Non-Teaching
72-73	Minor
74-75	Second Major
76-77	First Major
78	Sex
80	Code 2

Card
3

Student Required Index Exception Card

Card Column	Description
1-5	Student Number
6	School
7-27	Name
28	Authority
29-31	Required Cumulative Index to be Earned
32-34	Required Semester Index to be Earned
35-37	Semester Code
80	Code 4

Card Format

Card
4

Student Name and Address Card

Card Column	Description
1-5	Student Number
6	School
7-27	Name
28-51	Street Address
52-72	City and State
80	Code 1

Card
5

Comment Card

Card Column	Description
1-5	Student Number
6	School
7	Authority
8	Comment Code
9-11	Semester Code
12-45	First Line of Comment
46-79	Second Line of Comment
80	Code 5

Card Coding For Schools

Card Column 6 Of All Cards

Code	School
1	Education
2	Liberal Arts
3	Business
4	Nursing
5	Other

Card Coding For Classification

Card Column 49 Student Index Card

Card Column 75 of Probation Report Card

Code	Classification	Cumulative Number of Hours
1	Freshman	0 - 27
2	Sophomore	28 - 56
3	Junior	57 - 85
4	Senior	86 - 124
5	Other	

FORMS

Probation and Comment Request Form

To: Computer Center

Date: / 1964

Please ☐ Place on Probation ☐ Withhold the Permit to Register ☐ Make the Following Comment only,

Mr. Mrs. Miss ______________ ______________ If a ☐ cumulative ☐ current grade-point ratio ___.___ ___
Student Name Student Number Do Not Fill This Out For A Comment Only

is not earned.

Please make the following comment ☐ on the Grade Report ☐ if the student is placed on probation,
Comment Code

☐ regardless of the scholastic achievement.

First Line of Comment ----------------------------------
34 Characters only - Include Blank Spaces

Second Line of Comment ----------------------------------

Classification of Comment ☐ AUTHORITY CODE

Signature of Authority

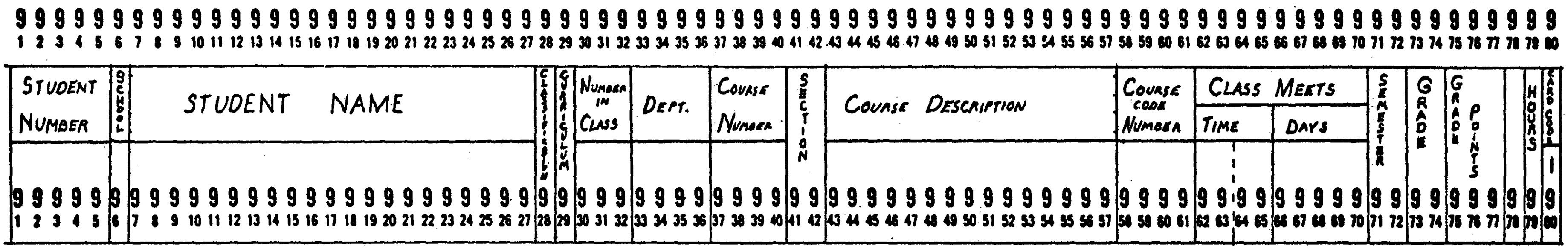
____

Probation Report

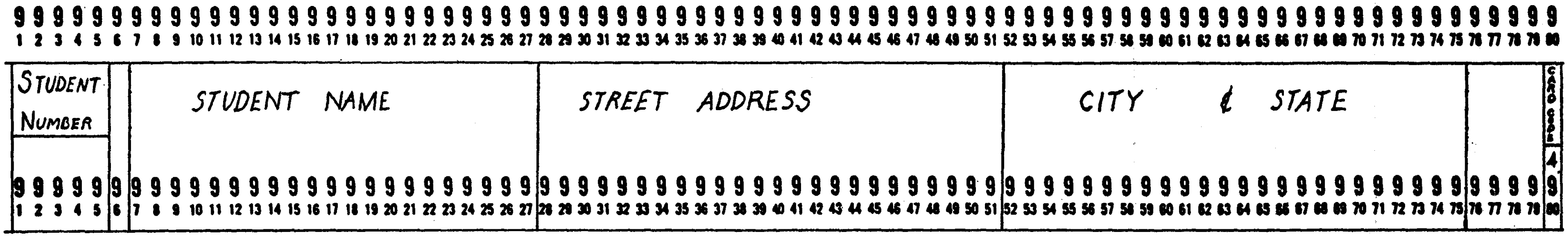
After grade processing the probation report cards will be used to generate a report of all students currently on probation. The format of this report is as follows:

Student Number	Student Name	Authority	Hours	Points	GPR	Hours	Points	GPR	Total Hrs Toward Grad.	Required Index Cum/Cur.	Comment Code	Probation Code	Social Organizations

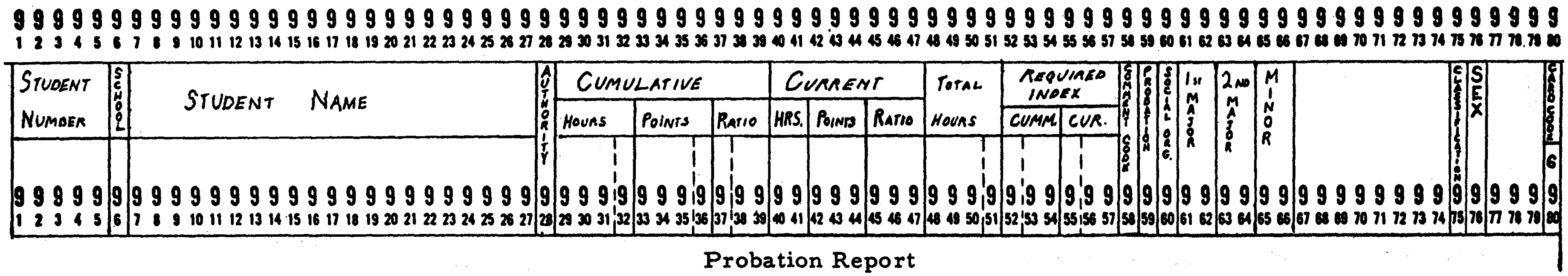
INDIANA STATE COLLEGE
COMPUTER CENTER



Grade

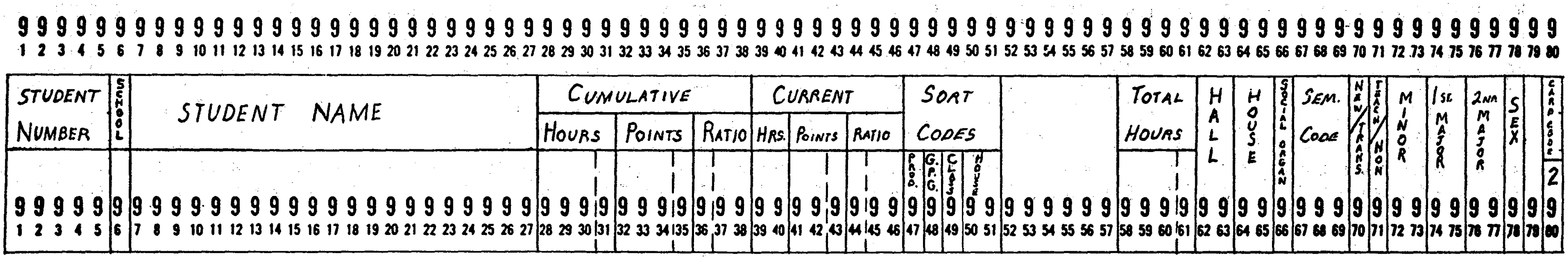


Address

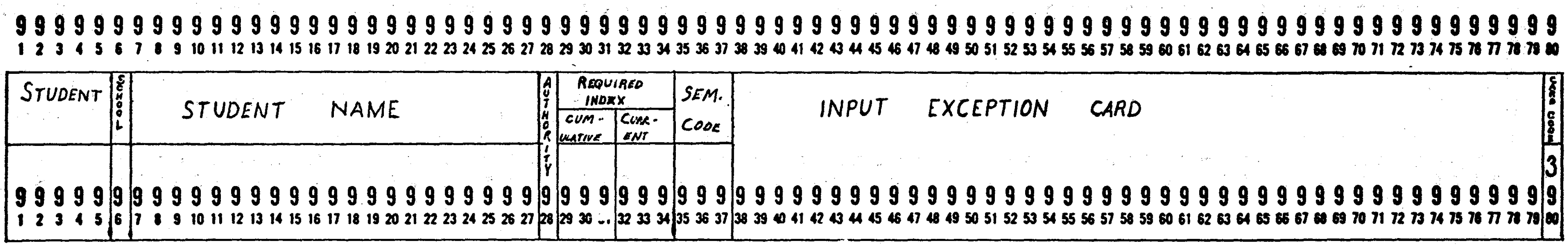


Probation Report

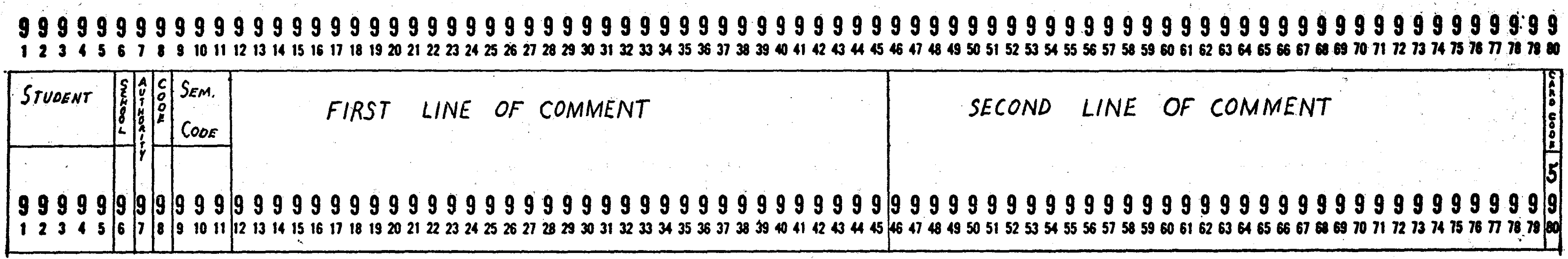
INDIANA STATE COLLEGE
COMPUTER CENTER



Index



Exception



Comment

COST STUDY
PROGRAM ABSTRACTS

Programs 1, 2, and 3

PROGRAM 1

Description:

This program extends the enrollment and class hour figures in the 2 cards. The input, the 2 cards sorted by Department Number, is read from the 1622 card reader. The computer multiplies the enrollment figure times the class hour figure. The entire 2 card is reproduced for output - containing the extended amount in card columns 26-31.

When a change in Department Number occurs, the machine will generate a 3 card. Therefore, there should be one 3 card for every department entered. The 3 card will contain: The Year, Dept. No., Total Student Class Figure, and the Total FTE Staff Figure. Both totals are from the 2 cards.

Error Conditions:

As the file goes through the machine, a sequence check for equal groups is performed. If a 2 card is out of order an error message will be typed on the typewriter ("error in Dept. sequence."). Because the computer recognizes an error condition on an "Not Equal and Not High" compare break, the department to which the error card belongs may have already passed completely through. In such a case it will be necessary to:

1. Adjust the 3 card for that department.
2. Rerun the department with the card included.

Irrespective of which method is used, the run must be started from the first department for which there is no 3 card.

Switch Settings:

I/O - Stop
Parity - Stop
Overflow - Stop

Console Switches:

Sw. 1	Not Used
Sw. 2	Not Used
Sw. 3	See Operating Suggestions
Sw. 4	Not Used

PROGRAM 2

Description:

This program is used to pro-rate the number two cards. After Program 1 is complete, the Department Total cards (3 cards) have two total expense amounts keypunched in them. These cards are then placed back in the file of 2 cards in front of their respective department files. This program (Program 2) then accepts that file as data. The program accepts the data in sequential order (by Dept. No. - 3 card first, followed by the two cards) and pro-rates the two amounts from each 3 card into each two card.

The two amounts to be allocated are found in card columns 19-24 and 25-30 of each 3 card. The first amount is pro-rated by weighting the student class hours figure on each 2 card against the total student class hour amount on its respective 3 card. This is then applied to the first total (stored in the machine) for each 2 card's share. The second amount is pro-rated on a similar basis, i.e., FTE figures are weighted. The remainders are carried into the following dividend, allowing the last card to zero-balance the amount to be allocated and the amount allocated.

Error Conditions:

The program tests for six error conditions during processing. If an error condition is detected, the computer will type "Error N" and halt. The six error conditions follow:

Error	
1	Sequence - 3 card
2	Sequence - 2 card
3	Amount allocated on student hours doesn't zero balance
4	Amount allocated on FTE doesn't zero balance
5	Student hours from 2 cards do not equal total from 3 card
6	.FTE percentages from 2 cards do not equal total from 3 card

On any of the above conditions, except 1 and 2, it will be necessary to start from the last department not completely processed.

PROGRAM 2

<u>Switch Settings:</u>

I/O - Stop
Parity - Stop
Overflow - Stop

<u>Console Switches:</u>

Sw. 1	Not Used
Sw. 2	Not Used
Sw. 3	Must be same setting as in Program 1
Sw .4	Not Used

PROGRAM 3

Description:

This program allocates the pro-rated totals on the two cards to a 7 card for each course. The output from Program 2 is sorted into course within department order (card columns 5-10) with the 7 cards then being merged in. If the data is in correct order, there will be one seven card in front of every group of 2 cards.

The program accepts its data through the card reader. The computer will read all the data cards for an entire department, then punch a new 7 card with the sum of the three expense totals from the two cards (FTE Expense + Salary Expense + Student Class Hours Expense) allocated to the six different grade levels (Freshmen, Sophomore, Junior, Senior, Special, and Graduate). The total enrollment of each class is weighted against the remainder from each division is carried to the following dividend to zero-balance the last card of each group.

The 7 card generated, then, has the complete cost of the course allocated to each class of students.

Error Conditions:

The program is extremely limited as to the number of error conditions it can check. The 7 cards are presumed to have been crossfooted, and the 2 cards have been through the machine checks of Program 1 and 2. Three error conditions might arise. They are:

1. An unmatched two card
2. A sequence error
3. The 7 card allocation did not zero balance - This would happen if a 2 card was missing.

On any error condition, the typewrite will describe the error and halt. It is then necessary to start over, from the last course 7 card not punched.

Switch Settings:

I/O - Stop
Parity - Stop
Overflow - Stop

Console Switch Settings:

Not Used

OPERATING SUGGESTIONS

1. Before running any program, clear the machine. This is done by pressing "Reset", then "Insert" on the computer console. Type in 260000800009, then press the R-S key. Wait about 1/8 of a second and depress "Instant Stop". Then press "Reset".

2. After clearing the machine, place one of the object decks in the read hopper and press "Load" (This is the yellow button). After the machine stops reading, press the green "Reader Start" to read in the last card. The machine will be in the manual mode at this point. Load the data in the read hopper, and blank cards in the punch hopper and press Start to begin processing.

3. If programs one and two are used with switch 3 off, there will be a duplication of output. That is, Program 1 extends the two cards with switch 3 off. This may be avoided by using the programs with switch 3 on (Programs 1 and 2). This method will allow just Program 2 to generate the 2 cards. Program 1 will then punch only the 3 cards and Program 2 will extend the 2 cards as it pro-rates them. Irrespective of the switch setting used, it must be the same for both programs. See the following flow chart for a more complete description.

Equipment Specifications

20K 1620 Computer
1622 Card/Punch Unit

Additional Features

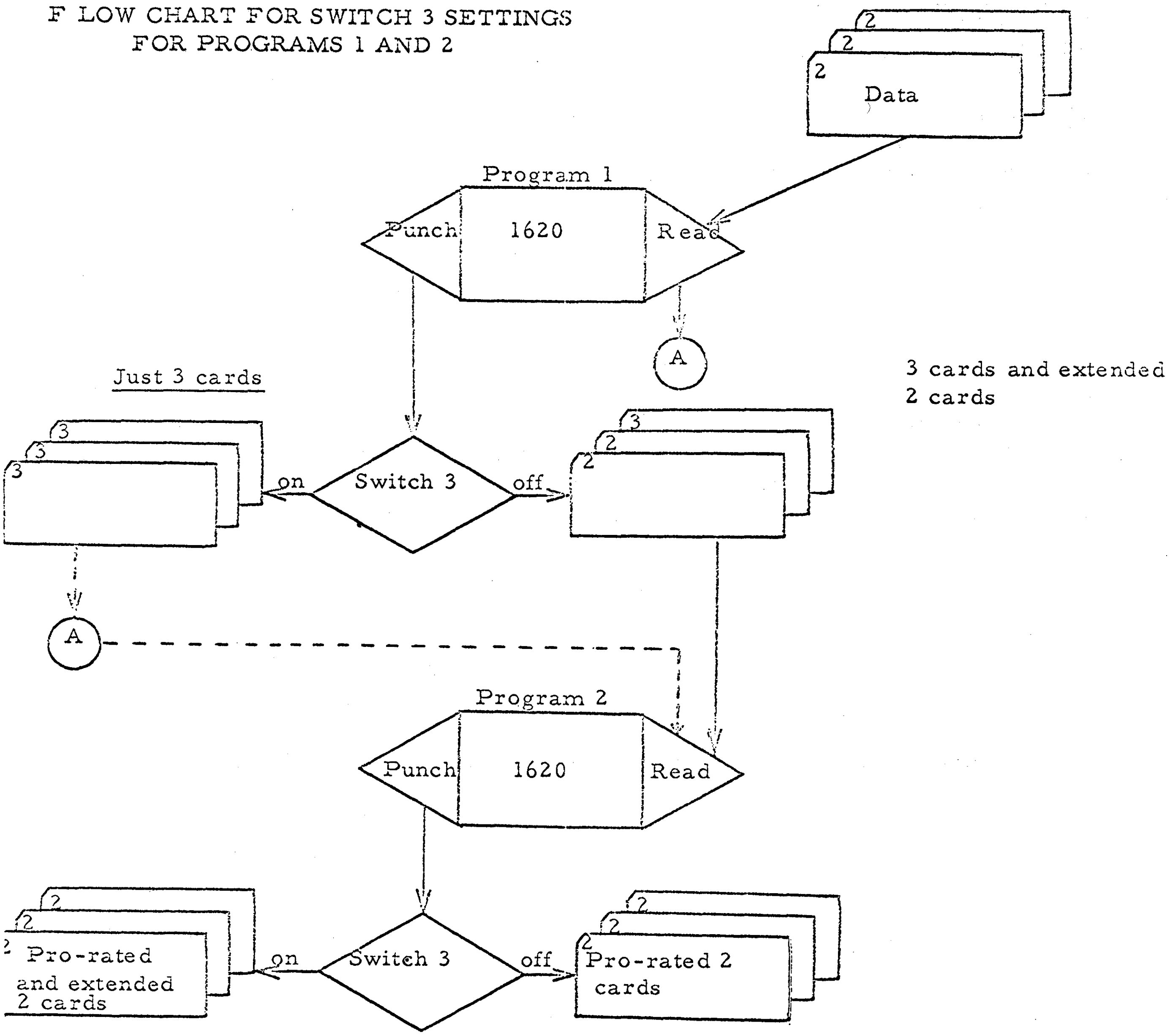
TNS and TNF Instructions*
Indirect Addressing Not Used

*These instructions greatly facilitate numeric conversion, but if not available, they may be simulated with several transmit digit instructions. As an example of this substitution, Program 1, line 01090 (TNS in+10, Now, Dept. No. 3-6) might be changed to:

```
TD         Now, in+10
TD         Now - 1, in+8
TD         Now - 2, in+6
TD         Now - 3, in+4
SF         Now - 3
```

This is because a number such as 0036 will be read as 70707376.

F LOW CHART FOR SWITCH 3 SETTINGS FOR PROGRAMS 1 AND 2



Ⓐ The output from Program 1 with Switch 3 ON is merged in the input file. This then becomes the input for Program 2.

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 7

2	YEAR	DEPT No.		COURSE No.	TYPE		DIV.	ENROLMENT	CLASS HOURS	STUDENT CLASS HOURS	EXPENSE / FTE STAFF		FTE STAFF	SALARY EXPENSE	EXPENSE / ST. CL. HRS	STAFF IDENTIFICATION	
																No.	SURNAME
9	9	9 9	9 9	9 9 9 9	9	9 9 9 9	9 9 9	9 9 9 9	9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9	9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9	9 9 9 9 9 9 9 9 9 9 9 9 9 9
1	2	3 4	5 6	7 8 9 10	11	12 13 14 15	16 17 18	19 20 21 22	23 24 25	26 27 28 29 30 31	32 33 34 35 36 37	38 39 40 41 42	43 44 45 46	47 48 49 50 51 52	53 54 55 56 57 58	59 60 61 62 63	64 65 66 67 68 69 70 71 72 73 74 75 76 77

3	YEAR	DEPT No.		TOTAL STATISTICS		TOTAL EXPENSES		
				STUDENT CLASS	FTE STAFF	TO BE ALL. ON STDT CLASS HRS	TO BE ALL. FTE STAFF	
9	9	9 9	9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1	2	3 4	5 6	7 8 9 10 11 12	13 14 15 16 17 18	19 20 21 22 23 24	25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77

7	YEAR	DEPT No.		COURSE No.		CURRICULUM	STUDENT ENROLLMENTS							TOTAL EXPENSES						
							TOTAL	FR.	SO.	JR.	SR	SP.	GR.	FR.	SO.	JR.	SR.	SP.	GR.
9	9	9 9	9 9	9 9 9 9	9 9 9 9	9 9 9	9 9 9 9	9 9 9 9	9 9 9 9	9 9 9	9 9 9	9 9 9	9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9 9
1	2	3 4	5 6	7 8 9 10	11 12 13 14	15 16 17	18 19 20 21	22 23 24 25	26 27 28 29	30 31 32	33 34 35	36 37 38	39 40 41	42 43 44 45 46 47	48 49 50 51 52 53	54 55 56 57 58 59	60 61 62 63 64 65	66 67 68 69 70 71	72 73 74 75 76 77

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77

9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77

1962-63 COST STUDY
I.SC – H^{3}

A
REPORT
OF
SCHOLASTIC ACHIEVEMENT
FOR
SPRING SEMESTER
1963-64

June 4, 1964

This report contains a summary of information by department, residence halls, sororities, fraternities, non-sorority, non-fraternity and by total student body.

This report deals exclusively with the undergraduate student body.

All Students

	Number of Students	Hours Attempted	Grade Points Earned	Grade-Point Ratio
Men	3,150	40,290	89,102.0	2.21
Women	2,321	29,697	74,730.0	2.52
Total	5,471	69,987	163,832.0	2.34

Fraternities*

	Number of Students	Credit Hours Attempted	Grade Points Earned	Grade-Point Average
Alpha Tau Omega	46	643	1,500.5	2.33
Lambda Chi Alpha	102	1,332	3,163.5	2.38
Pi Lambda Phi	31	430	988.0	2.30
Sigma Phi Epsilon	105	1,407	3,239.5	2.30
Tau Kappa Epsilon	87	1,186	2,705.0	2.28
Theta Chi	42	608	1,433.0	2.36
Total	413	5,606	13,029.5	2.32
Non-Fraternity Men	2,737	34,684	76,072.5	2.19

*Actives Only

Sororities*

	Number of Students	Credit Hours Attempted	Grade Points Earned	Grade-Point Average
Alpha Omicron Phi	55	774	2,072.5	2.68
Alpha Phi	39	594	1,642.5	2.77
Alpha Sigma Alpha	35	518	1,400.5	2.70
Chi Omega	50	699	1,916.0	2.74
Delta Gamma	54	796	2,231.0	2.80
Gamma Phi Beta	60	860	2,334.5	2.71
Sigma Kappa	55	778	2,109.0	2.71
Zeta Tau Alpha	53	779	2,080.0	2.67
Total	401	5,798	15,786.0	2.72
Non-Sorority Women	1,920	23,899	58,944.0	2.47

*Actives Only

By Area Major

Major	Number of Students	Hours Attempted	Grade Points Earned	Grade-Point Average
Art	113	1,286	2,974.5	2.31
Business	794	10,073	22,351.0	2.22
English	302	3,895	10,128.0	2.60
Foreign Language	137	1,300	3,696.0	2.84
Home Economics	165	2,131	5,162.0	2.42
Industrial Education	418	5,351	11,920.0	2.23
Journalism	5	37	69.5	1.88
Liberal Arts	102	972	1,801.0	1.85
Mathematics	267	3,552	8,550.0	2.41
Music	148	2,090	4,977.5	2.38
Nursing	85	790	1,950.5	2.47
Phy. Ed. Men	421	5,511	11,252.5	2.04
Phy. Ed. Women	108	1,384	3,220.0	2.33
Science	517	6,803	16,693.5	2.45
Social Studies	587	7,740	17,654.0	2.28
Speech and Radio	115	1,538	3,757.5	2.44
Special Education	148	1,939	4,819.5	2.49
Elem. - Fresh.	342	4,729	10,175.5	2.15
Elem. - Soph.	245	3,330	7,835.0	2.35
Elem. - Jr.	280	3,964	10,269.5	2.59
Elem. - Sr.	172	1,572	4,575.0	2.91
Total	5,471	69,987	163,832.0	2.34

By Residence Halls

Female

	Number of Students	Credit Hours Attempted	Grade Points Earned	Grade-Point Ratio
Burford Hall	286	4,194	10,117.0	2.41
Erickson Hall	277	4,046	10,004.0	2.47
Pickerl Hall	290	4,334	10,669.5	2.46
Reeve Hall	326	4,724	11,392.5	2.41
Total	1,179	17,298	42,183.0	2.44

Male

	Number of Students	Credit Hours Attempted	Grade Points Earned	Grade-Point Ratio
Gillum Hall	304	4,338	9,411.5	2.17
Hulman Center	309	4,163	9,049.0	2.17
Parsons Hall	298	4,326	8,452.0	1.95
Sandison Hall	298	4,360	9,748.5	2.24
Total	1,209	17,187	36,661.0	2.13

Other Housing

	Number of Students	Credit Hours Attempted	Grade Points Earned	Grade-Point Ratio
Men	1,941	23,103	52,441.0	2.27
Women	1,142	12,399	32,547.0	2.62
Total	3,083	35,502	84,988.0	2.39

By Residence Hall

Burford Hall

House	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
2	52	743	1846.0	2.48
3	58	902	2302.0	2.55
4	60	854	1975.0	2.31
5	59	867	1937.0	2.23
6	57	828	2057.0	2.48
Total	286	4194	10117.0	2.41

By Residence Hall

Erickson Hall

House	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
2	57	861	2048.5	2.38
3	56	832	2113.5	2.54
4	54	766	1879.5	2.45
5	52	757	1827.0	2.41
6	58	830	2135.5	2.57
Total	277	4046	10004.0	2.47

By Residence Hall

Gillum Hall

Hours	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
2	39	561	1149.5	2.05
3	39	553	1248.0	2.26
4	38	549	1157.5	2.11
5	35	492	968.0	1.96
6	38	533	1128.5	2.12
7	40	550	1131.5	2.06
8	36	521	1184.0	2.27
9	39	579	1444.5	2.49
Total	304	4338	9411.5	2.17

By Residence Hall

Hulman Center

House	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
3	33	442	956.0	2.16
4	53	711	1514.0	2.13
5	59	778	1737.5	2.23
6	57	730	1555.0	2.13
7	56	805	1837.5	2.28
8	51	697	1449.0	2.08
Total	309	4163	9049.0	2.17

By Residence Hall

Parsons Hall

House	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
1	57	843	1786.0	2.12
2	44	643	1312.0	2.04
3	56	820	1544.0	1.88
4	36	530	1061.0	2.00
5	57	832	1471.5	1.77
6	48	658	1277.5	1.94
Total	298	4326	8452.0	1.95

By Residence Hall

Pickerl Hall

House	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
2	56	827	2208.0	2.67
3	57	906	2111.0	2.33
4	58	846	2070.5	2.45
5	59	865	2071.0	2.39
6	60	890	2209.0	2.48
Total	290	4334	10669.5	2.46

By Residence Hall

Reeve Hall

House	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
1	19	270	610.0	2.26
2	87	1235	2853.5	2.31
3	114	1707	4172.5	2.44
4	106	1512	3756.5	2.48
Total	326	4724	11392.5	2.41

By Residence Hall

Sandison Hall

House	Number of Students	Credit Hours Taken	Grade Points Earned	Grade-Point Ratio
1	13	182	301.0	1.65
2	36	545	1324.5	2.43
3	35	521	1147.5	2.20
4	37	527	1230.0	2.33
5	34	473	1073.5	2.27
6	37	546	1362.5	2.50
7	36	542	1207.5	2.23
8	34	492	1002.5	2.03
9	36	532	1099.5	2.07
Total	298	4360	9748.5	2.24

AUTOMATIC PROCESSING OF AUTOSPOT AND AUTOMAP PROGRAMS WITH THE 1620-1311 DISC SYSTEM

A Paper Prepared For The:
Spring Meeting
1620 Users Group
Brown Palace Hotel
Denver, Colorado
June 19, 1964

BY:
Jack T. Dunn
AVCO Corporation
Huntsville, Alabama

And

Ernie G. Moore
IBM Corporation
Huntsville, Alabama

The authors of this paper gratefully acknowledge the work of Mr. Gary D. Douglas, programmer for AVCO Corporation, Huntsville, Alabama, who programmed a significant portion of this system and whose ideas are utilized throughout.

TABLE OF CONTENTS

I

TEXT

INTRODUCTION

Our purpose in presenting this paper is to illustrate a technique, developed at AVCO Corporation's Huntsville Engineering Office in conjunction with IBM Corporation's Huntsville Branch Office, Systems Engineering Staff, for processing AUTOSPOT and AUTOMAP programs on the 1620 equipped with an on line 1311 Disc Drive.

This is an automatic system in that it provides a means for producing a numerical control tape deck or paper tape from a deck of source statements without going through the multi-pass card system.

This technique is not reputed to be a panacea, nor will it be advantageous for all users.

It is, however, felt to be a significant step in improving the existing means of processing numerical control programs on the 1620.

During the next few minutes we will present to you information regarding:

(1) Rationale which led us to choose this particular method of attack.

(2) The methods utilized by and the operation of the system.

(3) Advantages and limitations of the system.

(4) Savings in time and money which might reasonably be expected, and

(5) Application to Fortran and SPS programs.

Following the discussion of these topics, Mr. Ernie Moore of IBM, Huntsville, will discuss some of the specific modifications which must be made to the machine language deck in order that this system may be utilized.

I wish to invite those of you who are not specifically interested in numerical control to visualize the decks which we will be discussing as nothing more or less than machine language object decks compiled under FORTRAN with FORMAT or SPS for 1620 Mod I.

In this way it may be possible for you to anticipate uses for this type of technique within your own organization.

RATIONALE

For the benefit of those who are unfamiliar with AUTOSPOT and AUTOMAP it would probably be worthwhile to take a brief glance at the nature of these two popular numerical control languages.

AUTOSPOT is a language for positioning machine tools -- basically it is for point to point operations but has limited machining capability. An AUTOSPOT source program is ordinarily written in fixed format English language statements, which are then punched into cards. These source statements are then processed through a three phase General Processor called the AUTOSPOT processor.

Following this General Processor will be a multi-phase post processor.

The post-processing required for the machine tools which we utilize at AVCO, consist of three to six additional phases. One of these phases is a two pass phase; thus a total of ten passes through the computer is required. All but one pass produces an intermediate output deck which must be subsequently loaded with the appropriate processor -- seemingly "ad infinitum".

AUTOMAP is a contouring language which requires an English language source deck and two phases of general processing. In addition, three phases of post-processing are required for our particular machine tools. Again each phase produces an intermediate output deck which must be subsequently reloaded with the appropriate processor.

The final phase of each of these produces a punch paper tape or a card deck which contains the input to the special purpose computer attached to the machine tool.

This special purpose computer actually controls the machine tool.

Upon exposure to this system, one immediately sees the need for streamlining the operation.

We chose to accomplish this streamlining by adapting AUTOSPOT and AUTOMAP to the 1620 with an on line 1311 disc.

This approach was chosen essentially because of the versatility offered by this system in other applications such as scientific/engineering computation under MONITOR/FORTRAN II-D as well as the numerical control application.

The choice of the particular technique we eventually utilized was the result of considerable thought as to what constituted the best approach. It was also the result of experience gained traveling a number of blind alleys.

The initial thinking indicated that the best approach would be to recompile each phase of AUTOSPOT and AUTOMAP under MONITOR with the read-write statements changed where necessary. This would, presumably, have involved updating the SPS source decks for all phases of the general and post processors applicable to our work; plus compiling them as if they were utility programs. It seemed at the time that this would be a simple straightforward approach which would yield results with a minimum of re-programming.

However, upon closer examination of this idea, a number of problems became evident.

The most serious of these involved the floating point subroutines which were required. AUTOSPOT and AUTOMAP originally were written and compiled to utilize the "excess fifty floating point" system. On the other hand, we came to learn that our SPS II-D (the SPS for MONITOR I) utilizes exponential numbers.

This problem in itself seemed to be insurmountable from a practical standpoint.

One other problem which we came to consider practically insurmountable is the fact that almost all phases of our general and post-processors fill 20,000 positions of core practically to the hilt. Thus no space would exist for the "supervisor" program which MONITOR requires.

These two problems alone seemed sufficient to eliminate the MONITOR approach as being simple.

One might reason that a 40 or 60K machine would eliminate the latter problem; and rightly so.

Unfortunately, we have a 20K machine -- moreover we desired, if possible, to keep AUTOSPOT and AUTOMAP in such a form that only the basic 1620 plus 1311 would be required.

There were other considerations in addition to these which led us to abandon the MONITOR approach. For example, our experience with MONITOR I led us to believe that, even if we were able to "Monitorize" AUTOSPOT and AUTOMAP, the execution speed would be agonizingly slow due to the disc relocatable subroutines, etc.

Thus after consultation with our systems engineer, we decided to investigate what we then called, a "quick and dirty" approach. This approach, we visualized, would incorporate machine language modifications to the existing compressed object decks of the various phases which we desired to implement. It was planned that these changes would link the various phases together in such a way that

415

each would call the next from its respective place in disc storage. In addition it was planned that the intermediate output would be loaded in an area reserved on the disc rather than being punched in cards.

It appeared, at the time, that this type of approach would result in serious problems in coding since all changes would necessarily be in machine language and since spare core storage locations were at a premium in most cases.

Still the benefits which were expected seemed to warrant proceeding along this particular avenue.

Some of the anticipated benefits were:

(1) Relatively high speed operation

(2) Simple operation

(3) The AUTOSPOT and AUTOMAP decks would remain in essentially the same form as "non disc" decks so that Users Group library patches could be utilized.

After a number of troublesome but not really serious problems, our expectations began to be realized -- first with AUTOMAP -- then with the post-processor used for the Bendix controlled Milwaukee Matic Model II -- finally with AUTOSPOT and the post-processor for the G. E. controlled Milwaukee Matic Model II.

True to its name the system was quick but it was not at all "dirty" -- rather it was quite smooth and clean.

THE SYSTEM

In essence the system as it applies to AUTOSPOT operates in the following manner.

(Slide #5A)

Phase I of the AUTOSPOT general processor resides in a specified position on the disc, its READ statements remaining unchanged and its PUNCH commands changed to load each output card image in an area on the disc reserved for intermediate output. This is accomplished, in general, by changing all the PUNCH commands to BRANCH to a routine which:

(1) Writes the card image on the disc

(2) Increments the sector address

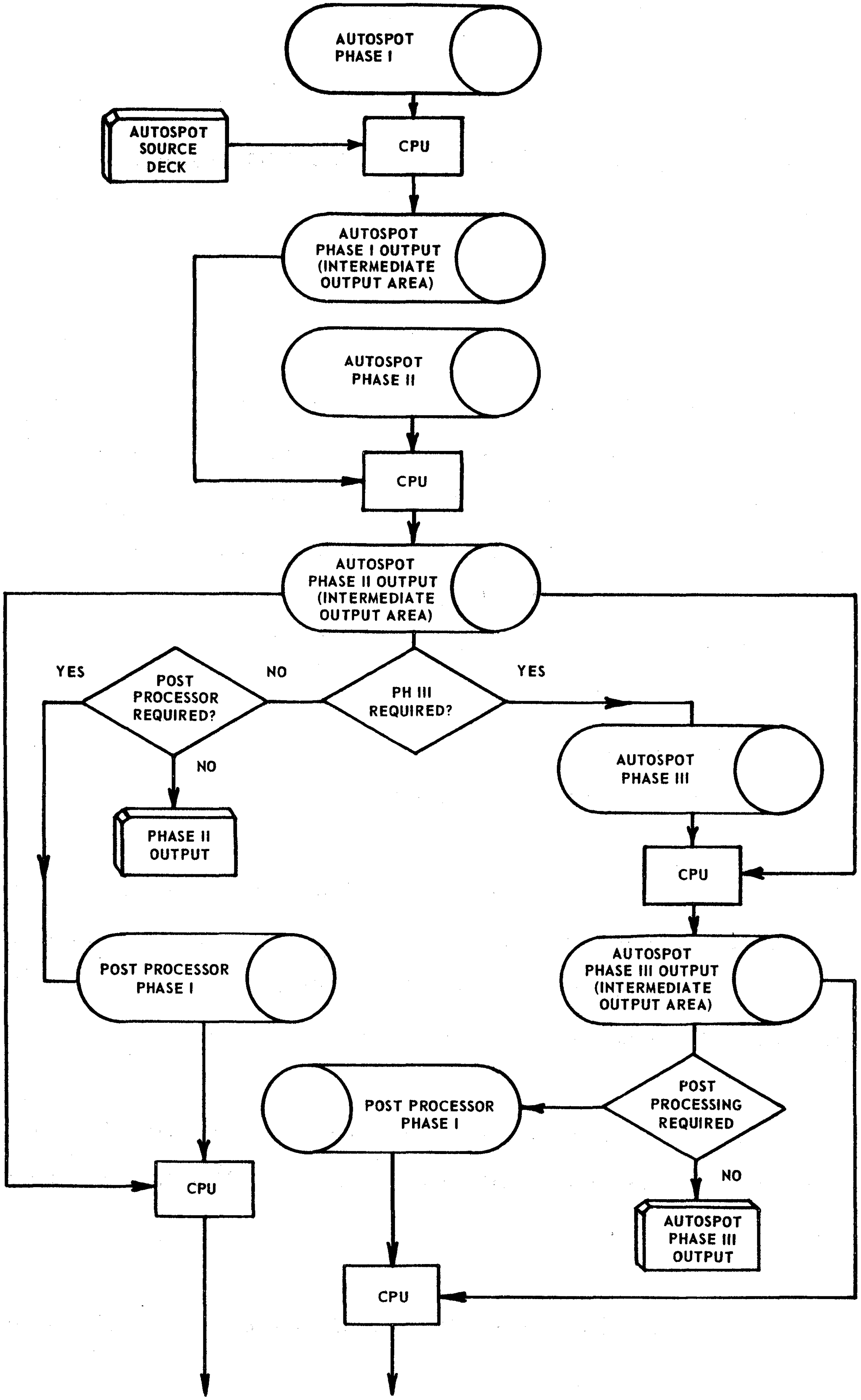
(3) Tests for end of job

(4) Returns to main line program.

Thus, in order to process an AUTOSPOT program, one simply loads a control card which calls Phase I of the general processor from disc storage to memory Phase I then reads and processes the source deck in the usual manner with the exception of punch output.

When the program encounters a FINI card (which signifies the end of job) the program branches to a routine which calls Phase II of the general processor from its disc location.

The READ commands of Phase II are modified to BRANCH the program into


AUTOSPOT PHASE I
AUTOSPOT SOURCE DECK
CPU
AUTOSPOT PHASE I OUTPUT (INTERMEDIATE OUTPUT AREA)
AUTOSPOT PHASE II
CPU
AUTOSPOT PHASE II OUTPUT (INTERMEDIATE OUTPUT AREA)
YES
POST PROCESSOR REQUIRED?
NO
PH III REQUIRED?
YES
NO
PHASE II OUTPUT
AUTOSPOT PHASE III
CPU
POST PROCESSOR PHASE I
AUTOSPOT PHASE III OUTPUT (INTERMEDIATE OUTPUT AREA)
POST PROCESSOR PHASE I
POST PROCESSING REQUIRED
NO
AUTOSPOT PHASE III OUTPUT
CPU
CPU

a routine which reads a card image from the disc, increments the sector address and returns to the main line program. In this way the intermediate output from Phase I is read: processing proceeds in the usual manner and the resulting output is written on the disc in a fashion similar to Phase I.

When the processing of Phase II is complete and a FINI code is encountered the program tests one of the console sense switches to determine if the source program requires the use of AUTOSPOT Phase III. Phase III is not required for programs which have no tool offset or arc and slope -- that is for machining programs. If the sense switch is on (indicating contouring) the program branches to a routine which loads core with Phase III of the general processor. If, however, the switch is off (indicating no contouring) the program tests another of the sense switches to determine if the program requires post processing. If post processing is required the appropriate post processing phase is loaded into core from its location on the disc. If post processing is not required then a punch tape or a card deck containing the output of Phase II is produced.

If Phase III were required, it would be loaded into core as I mentioned previously. Processing would proceed in the usual manner with the image of each resulting card being loaded in the disc intermediate output area.

Upon completion of Phase III, indicated by a FINI code, one of the console sense switches is interrogated to determine if post processing is required; if not, a card deck containing Phase III output is produced. However, if a post

422

processor is required then the appropriate phase will be called from the disc into core.

The post processors applicable to our particular machine tools have been adapted to run in a manner very similar to the one just outlined.

These post processors are:

(1) Computer Routines for The Milwaukee Matic Solid State Controlled Machining Center, User Group Library, No. 10.4.004, and

(2) MATIC, a proprietary pseudo-language developed by Kearney and Trecker Corporation for its Bendix Dynapath controlled "Milwaukee Matics"

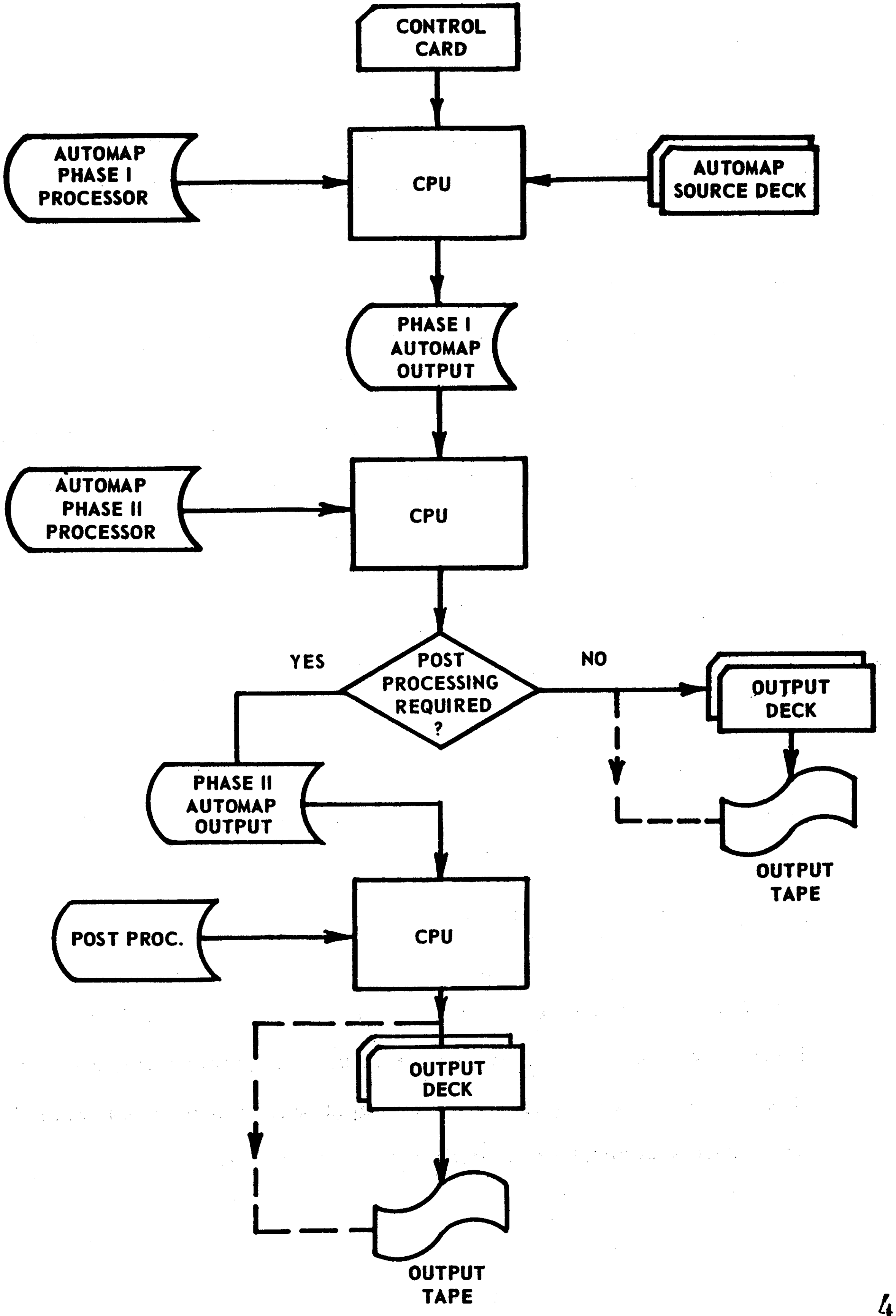
In general this is now the system operates. Let us now take a brief glance at the advantages and limitations of it.

ADVANTAGES AND LIMITATIONS

The principal advantages of this system as we see them are as follows:

(Slide #1A)

(1) Simplicity of operation: One control card calls Phase I of AUTOMAP from disc into core -- the source deck is processed and the intermediate deck is loaded on the disc. This output will then be read from the disc after Phase II has been transferred from disc to core.


CONTROL CARD
AUTOMAP PHASE I PROCESSOR
CPU
AUTOMAP SOURCE DECK
PHASE I AUTOMAP OUTPUT
AUTOMAP PHASE II PROCESSOR
CPU
YES
POST PROCESSING REQUIRED ?
NO
OUTPUT DECK
PHASE II AUTOMAP OUTPUT
OUTPUT TAPE
POST PROC.
CPU
OUTPUT DECK
OUTPUT TAPE

(2) Core Clearing: Manual clearing of core to zeros or flag zeros is not required.

(3) Card Handling: The handling of intermediate decks is eliminated thus reducing operator time and the hazard of dropped or mixed up decks.

(4) Speed of Operation: The reading of lengthy processor decks with the 1622 is eliminated, saving computer hours and operator hours. The elimination of intermediate output decks speeds up the operation from both operator time and machine time standpoints.

(5) Card Savings: If the company utilizing this technique does a large volume of numerical control program processing, the savings from decreased card consumption may be a significant percentage of the 1311 rental cost.

(6) Disc READ/WRITE checks may be written into the program with as many re-reads or re-writes as desired.

At the present time we visualize that there may be two limitations or problems with this system.

(1) Program Modifications: Program modifications which require additional core will not be possible in some phases due to the fact that the additional instructions required to implement this system cause core to be completely filled. In addition, program modifications which are desired, either as a result of program modifications published by the Users Group Library or by individual preference, must be incorporated into the deck and loaded on the disc, much in the same manner as MONITOR I patches.

(2) Debugging Source Programs: Another problem exists if the individual users desire an intermediate deck for debugging -- let's say to find the location of a failure relative to some known point in the source program. It is often desirable to have this output to determine the last successful command, thereby locating the bug. At the present time, we utilize a debugging program in the form of a series of instructions, which will dump, from the disc, images of a number of cards before and after the location of the failure.

The seriousness of this limitation depends primarily on the method of processing utilized within the users organization. It is doubtful if it would be a serious limitation in any case.

III SAVINGS

Having touched on the advantages of the system in general, let us return to the specific area of cost and time savings.

One might reasonably expect savings in three areas:

(1) Savings in elapsed time from receipt, by the operator, of a source deck to production of final tape deck. This time saving is realized by the elimination of card handling between phases, core clearing, and punch clearing operations.

Our experience indicates that, normally, two hours actual processing time requires approximately three hours elapsed time. An actual test run with an average AUTOMAP program with three phases of post processing required 60 minutes utilizing the card system exclusively but was reduced to 37.5 minutes utilizing this system.

A typical AUTOSPOT program with 6 phases of post processing ran 70.5 minutes utilizing the card system exclusively and 35.2 minutes utilizing the 1311.

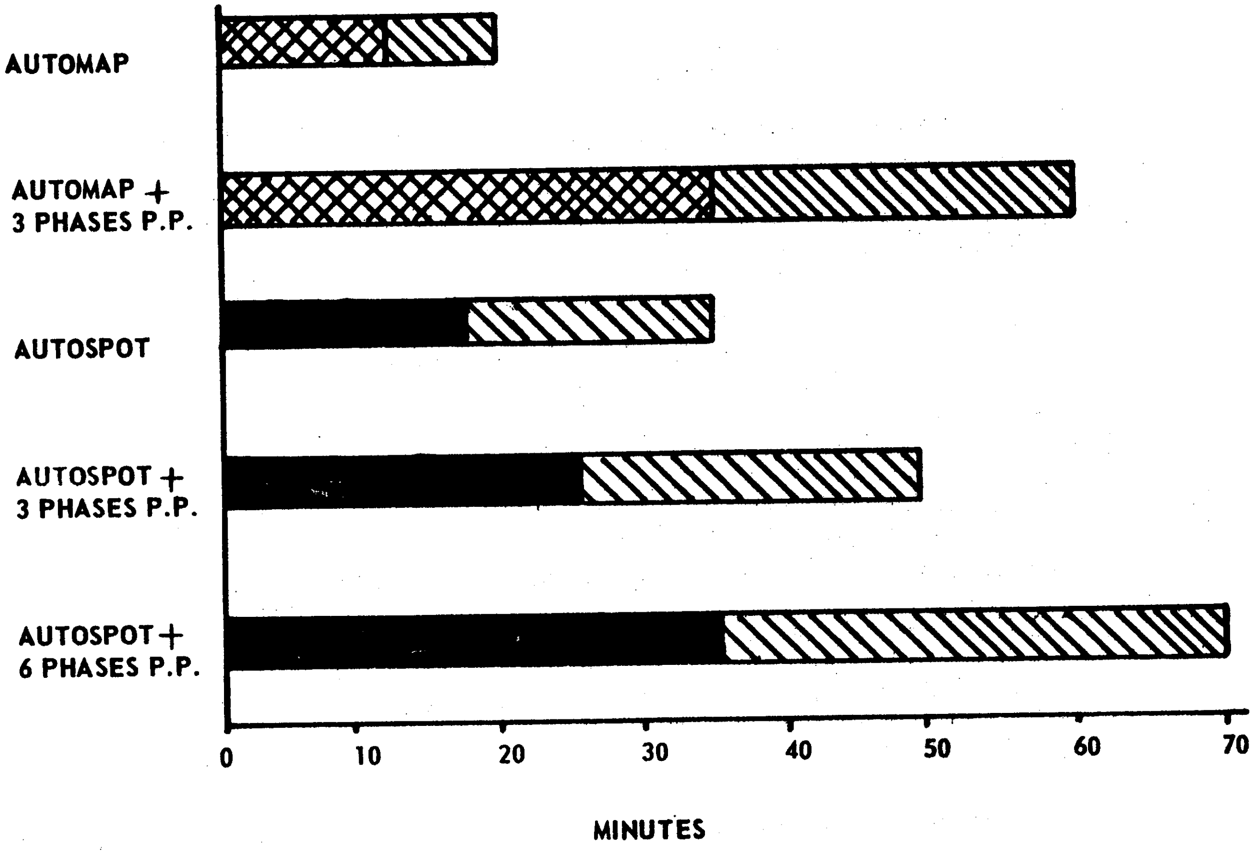
The elapsed time comparison for various AUTOSPOT and AUTOMAP plus post processor configurations are shown graphically on the next slide.

(Slide #2A)

(2) The second area of saving is in computer meter time. This saving is realized, primarily, as a result of the elimination of the 1622 as an input/output device, in all cases except the reading of source deck and punching of tape deck. (We do not utilize the on-line tape punch).

In a test utilizing what we consider an average AUTOMAP source deck plus 3 phases of post processing the meter time utilizing card I/O was 46.8 minutes and utilizing the 1311 the time was 35.7 minutes.

A similar test with an average AUTOSPOT program with six phases of post processing indicated the running time without 1311 as 56.8 minutes and with this system as 38.8 minutes.



ELAPSED TIME

A bar chart showing the comparison between the two systems for various configurations of processor plus post processor is shown on the next slide.

(Slide #3A)

(3) The third area of saving is in decreased card consumption. Obviously, this saving is realized as a result of the elimination of punched out intermediate decks.

The average AUTOMAP test deck which was utilized to determine the time comparisons previously mentioned, utilized 1600 cards with only card I/O and 450 with the help of the 1311.

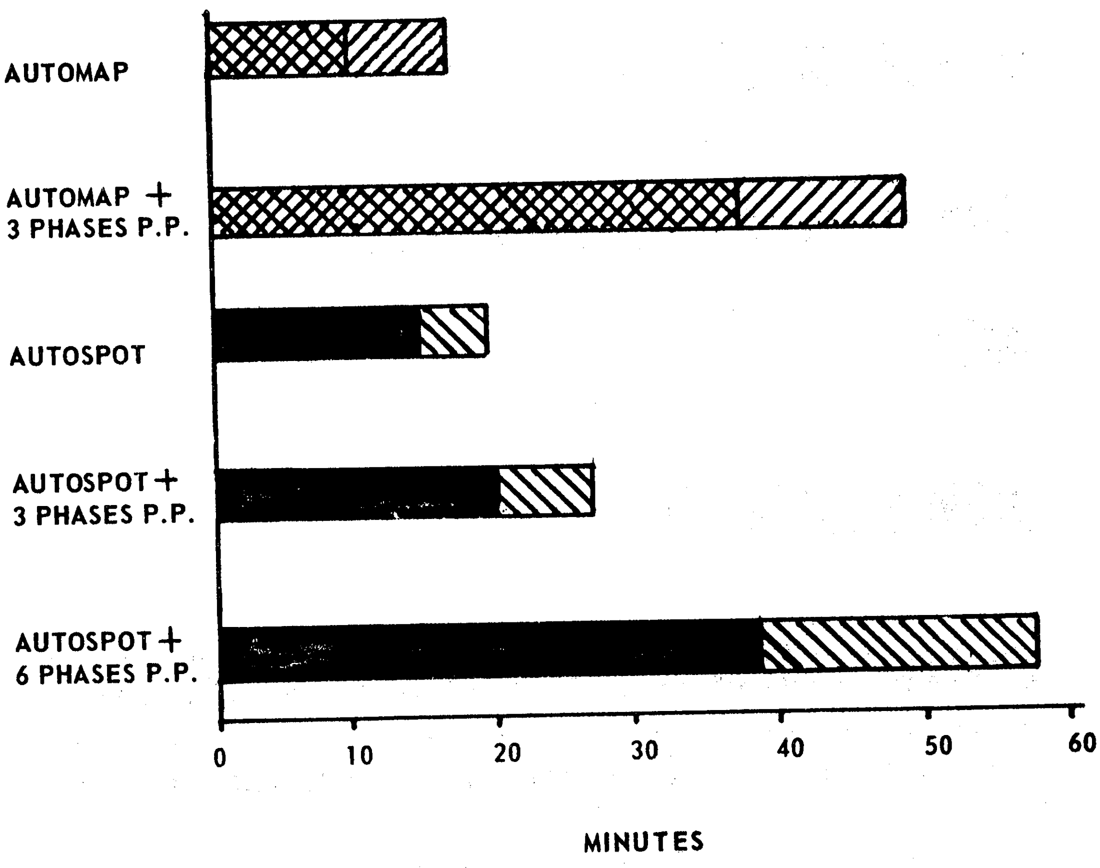
A similar test with AUTOSPOT utilized 1850 cards with card I/O and 580 cards with the 1311 as intermediate I/O medium.

The next slide illustrates the card consumption for various configurations of processor plus post processor.

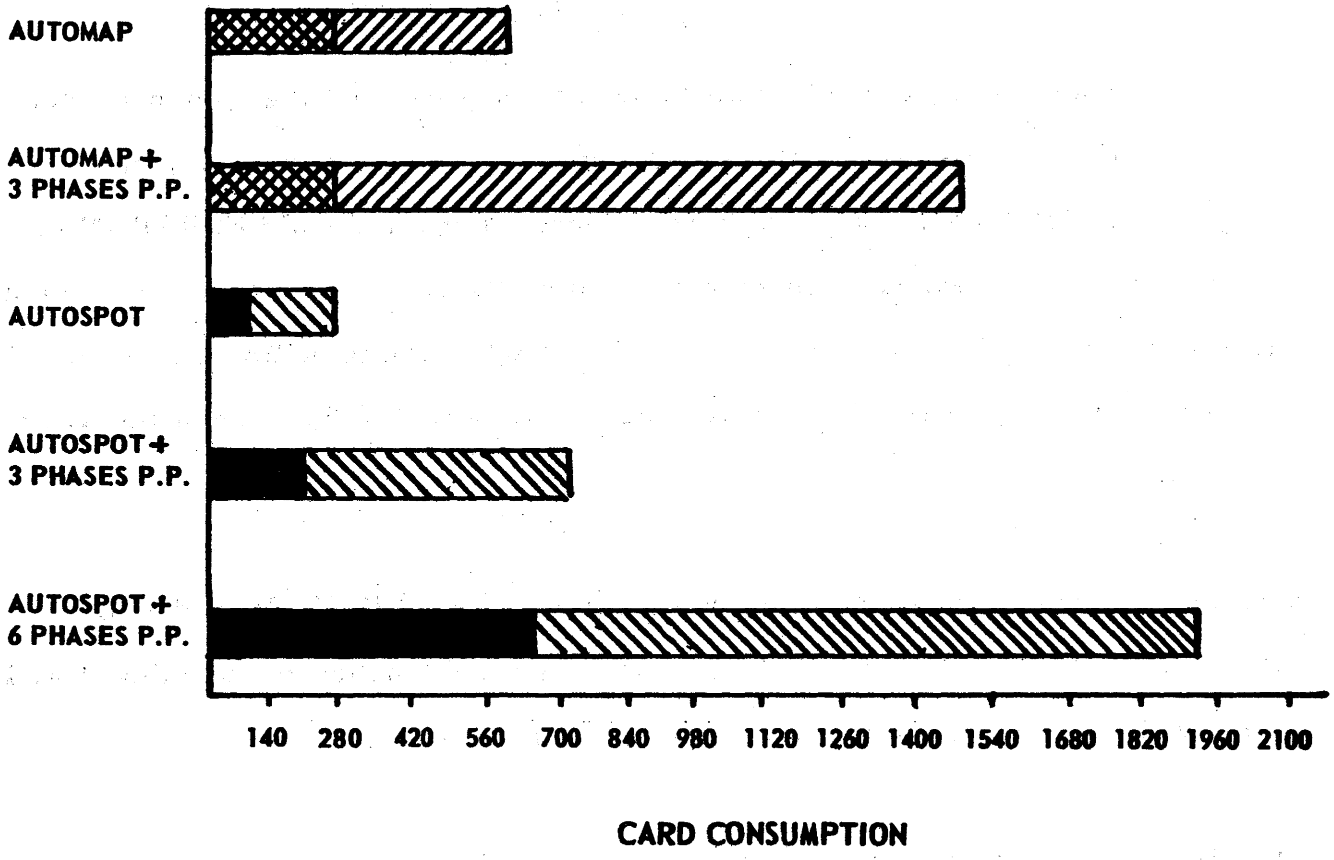
(Slide #4A)

VI. OTHER APPLICATIONS

The technique which we have been discussing in general and which will be covered in greater detail later on, is by no means limited to numerical control program processing.


AUTOMAP
AUTOMAP +
3 PHASES P.P.
AUTOSPOT
AUTOSPOT +
3 PHASES P.P.
AUTOSPOT +
6 PHASES P.P.
0
10
20
30
40
50
60
MINUTES

METER TIME


AUTOMAP
AUTOMAP +
3 PHASES P.P.
AUTOSPOT
AUTOSPOT +
3 PHASES P.P.
AUTOSPOT +
6 PHASES P.P.
140
280
420
560
700
840
980
1120
1260
1400
1540
1680
1820
1960
2100
CARD CONSUMPTION

We have implemented it for this particular use because a considerable bulk of our processing is in the numerical control area.

The same technique has been utilized successfully for processing machine language programs compiled under "FORTRAN with FORMAT" and SPS and even for linking two or more such programs together.

This application is somewhat troublesome from a programming standpoint but we feel that there exist areas wherein it would be useful. For example, companies which have already compiled programs under the old FORTRAN or SPS systems might find it inconvenient or impractical to re-write and re-compile these programs under MONITOR. Thus they might wish to utilize a method similar to this to process these programs, particularly if they are of the daily run type.

The point is this: a method by which one might bypass MONITOR and still take advantage of the disc does exist, if the time and trouble for utilizing it will yield sufficient savings in the particular case in question.

We will be happy to answer questions regarding this technique at the end of our formal discussion and to discuss the details of it during the workshop which will be held this afternoon.

At this time, I would like to present Mr. Ernie Moore, Systems Engineer from Huntsville's IBM Branch Office. Mr. Moore has done a considerable amount of programming in this effort and will discuss the system in greater detail.

For those of you who already have a 1311 Disk System, I hope to show you a procedure for placing a machine language program on the disk. For those who have not yet obtained a disk system, I hope to show you how the Monitor System can be supplemented in the case of established programs which have heavy usage and are completely contained in one core image. To date, we have modified about fifteen programs to operate from the disk. Some of these programs also write data on the disk and then call the next program or phase which then uses this data written on the disk to continue the processing, as in the case of post-processors. We have modified both FORTRAN and SPS Programs. The procedure which I am going to explain is the result of our experience on these programs.

Our approach was intentionally the simplest one possible. It was to load the established program into core and then to transfer the complete core image, with minor modifications, onto the disk. When we wish to operate the program, we simply call the complete core image back into core using a single call card containing only four instructions. By this step, we have eliminated the handling of the program decks and the time to read cards into core. We are essentially using the disk as a storage medium for our card decks.

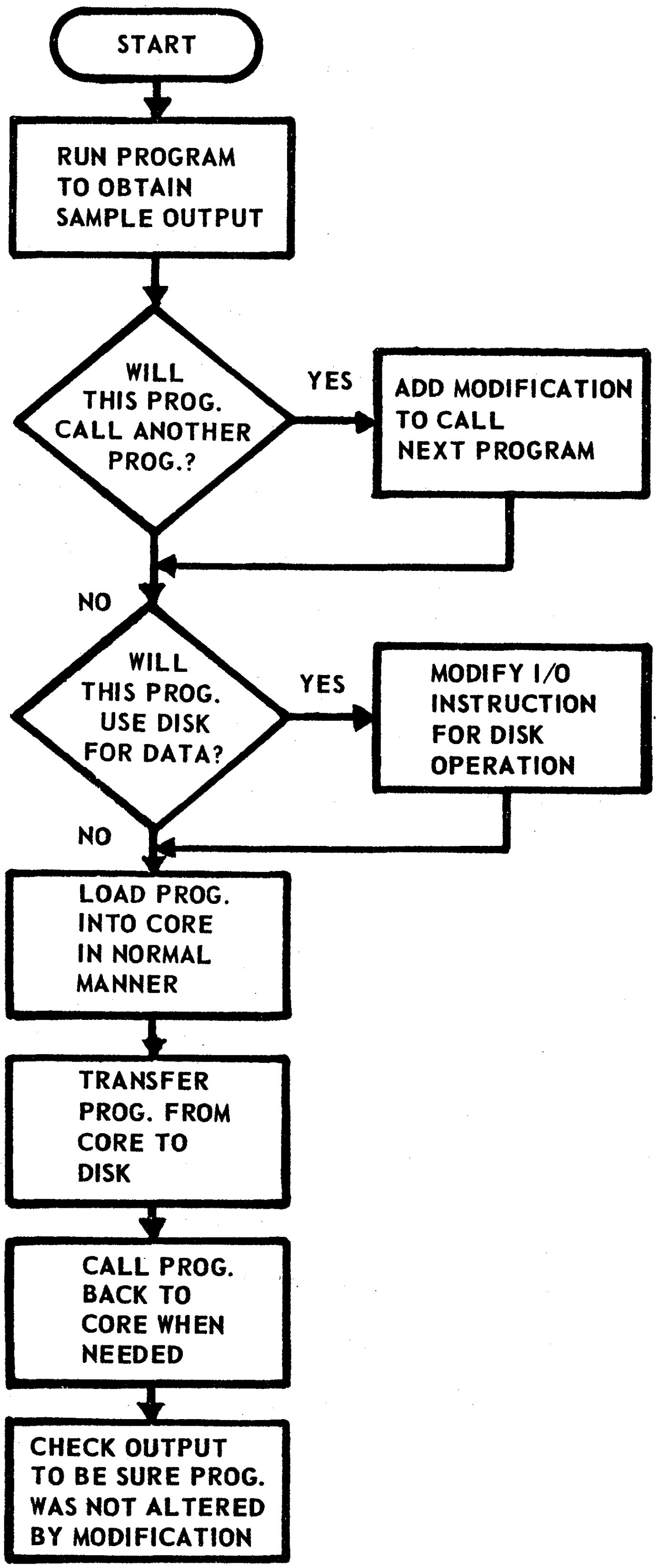
As we became more familiar with the disk, the next logical step was to have the first program in a series of programs automatically call in the second program when the first program was finished. This series of each calling the next could go on indefinitely -- the only limit being the limit of 100 core images on one disk pac. Our largest string to date is ten programs.

While we were still feeling pretty good about our first two achievements, we tackled the final step -- that of modifying the first program to place its intermediate output on the disk instead of punching cards, then modifying the second program in the string to read this intermediate data from the disk as input. The same program in turn was to write its output on the disk for the third program and so forth, on to completion. Here we have almost completely eliminated cards and their associated problems, and we have also eliminated the operator, once he has called the program in and loaded the initial data.

Slide 1 illustrates the sequence which we have just gone through. These are the steps you could take in modifying your program. You will note that before modifying the program, sample output is obtained for use later as a check to see that our modifications have not changed the logic of the original program. In the first decision block in Slide 1, we ask, "Will it be necessary for this program to call the next program in a string?" If the answer is "yes", then the program must be modified by adding a three or four instruction routine at the end. The second decision block indicates that the program must also be modified if it will use the disk instead of the cards for data storage. After the necessary changes have been made, the program is loaded into core using its own load program. The complete core image is then transferred to the disk and is available

434

MODIFYING YOUR PROGRAM FOR DISK OPERATION



for call at any time it is needed. It is suggested that the output of the test run be compared with the original test data to point out any erroneous changes which have altered the logic of the program.

We will now go back and examine the individual blocks of this flowchart (Slide 1) in more detail. If it were decided that our program, when finished, should call in a second phase or program we could simply replace the final halt with the two instructions needed to call the next program from the disk. In Slide 2 we have replaced the halt at 10588 with an Op Code 34 followed by an Op Code 36. These will transfer the next program from the disk to core. This will work fine if the first instruction of the new program is located at 10612. This is because the instruction at 10612 will be the next instruction executed. From this, we can conclude that the Op Code 34 and 36 instructions to call the second program must be located just in front of the address at which the first instruction in the second program will be loaded. This is illustrated in Slide 2 where the Op Code 26 at location 10612 would be the first instruction executed in the second program.

To further illustrate this point, we will take the case of an SPS Program which has a halt at location 0 followed by a branch, in Loc 12, to the origin or beginning of the program. Here we have replaced the halt at 10588 in Slide 3 with a 49, or branch, to 19976. At location 19976, we have placed our Op Code 34 and 36. The next instruction executed after the Op Code 36 will be the instruction brought out from location 0 which is normally the first instruction in an SPS Program.

MODIFICATION TO CALL NEXT PROGRAM

CORE LOC.	OP CODE	P & Q ADDRESS	
10588	48	00000 00000	FINAL INSTRUCTION IN UNMODIFIED PROGRAM
10588	34	19962 00701	FIRST MODIFIED INSTRUCTION
10600	36	19962 00702	
10612	26		

MODIFICATION TO CALL NEXT SPS PROGRAM

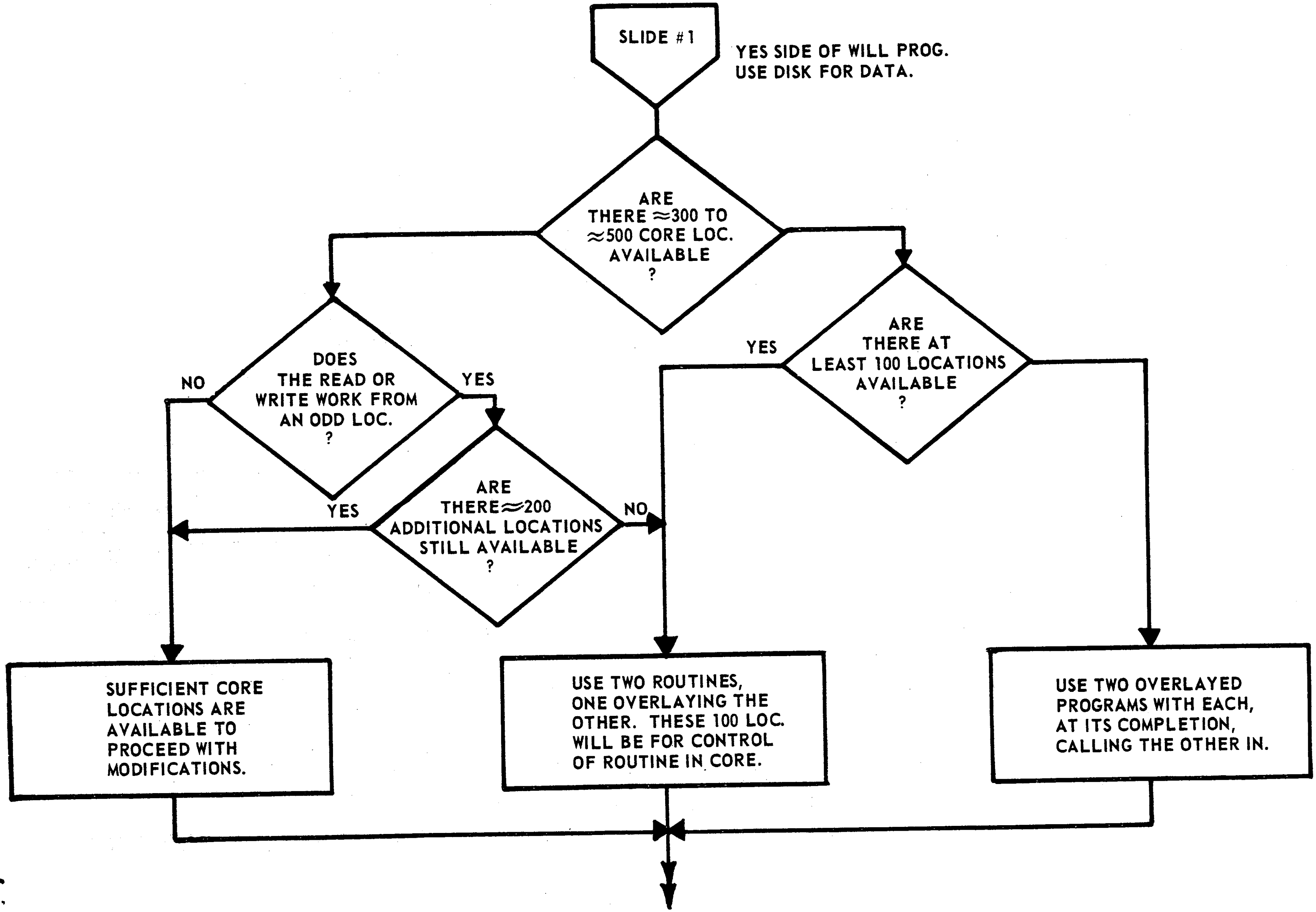
CORE LOC.	OP CODE	P & Q ADDRESS
10576	LAST	INSTRUCTION IN PROG.
10588	49	19976
19962	1000	00200 00000
19976	34	19962 00701
19988	36	19962 00702
00000	41	00000 00000
00012	49	01700

Now that the new program is in core, this is also the first instruction of our second phase or second program. This takes advantage of the wrap around feature which allows us to execute the instruction at location 0, after the instruction located at 19988 is executed. This also takes advantage of the fact that very seldom are instructions located in the upper end of core in the 19900 area. This allows us to use this area.

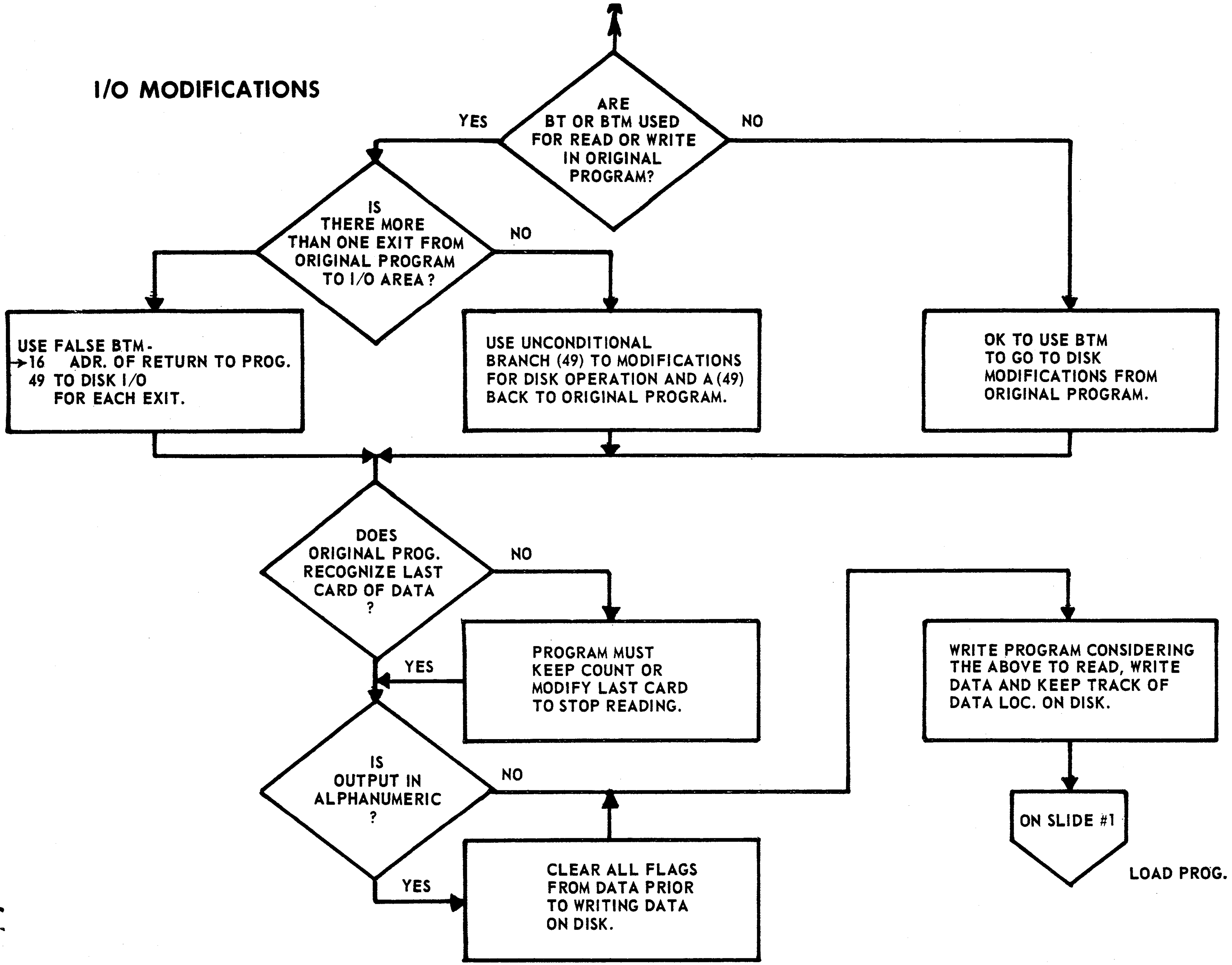
Let us now examine the block labeled "Modify I/O Instructions for Disk". Slide 4 is an enlargement of this block. The first decision block asks,"Are there 300 to 500 core locations available?" This figure, 300-500 is rather broad and varies depending on the number of read or write statements in the program which are to be modified. If we have this number of core positions available, then we go to the next decision block which asks, "Does the program read into or write from an area beginning with an odd address?" If this answer is "yes", then our modifications will require an additional 200 locations. These additional 200 locations are used to program the transfer of data so that the disk works out of or into an area beginning with an even address. If this is not done, the first or last digit of the data will be lost in a disk transfer. The transfer of data to core is made prior to a write disk or following a read disk instruction using a transmit record so that record marks and special characters would also be transferred; thus not altering the data in any respect. Once we find that sufficient core storage is available, we proceed to the writing of the modification program.

439

I/O MODIFICATIONS



I/O MODIFICATIONS



If, in the first block in Slide 4, we had found that sufficient core was not available, we would have gone out the no side to the next decision box which asks, "Are there at least 100 locations available?" These 100 locations will be necessary for a control program which will control a section of core storage calling either the I/O routine or the main program whenever either is needed. In other words, a portion of core will be shared by both the I/O program and the program which normally resides in this area. This way we can actually use a program which fills practically all of core and still have available the additional programming necessary to take care of reading and writing on the disk. There are very few cases where 100 locations cannot be found. In many cases, an output error message may be modified or abbreviated. The locations acquired in this manner may be used for control purposes. The locations from 0 - 80 in the product area may sometimes be used. By dumping the program out on the typewriter, there may be other areas which will become evident. In this way, sufficient area may be found to contain the instructions necessary to call the alternate program and control whichever program is in core at the present time. If we reach a situation where fewer than 100 cores are available, and if the logic of the program will allow, the best solution would be to have each of the two overlayed programs, at its completion, call its counterpart in on top of the existing program. In this way, you are alternating back and forth and each program, when executed, will automatically call the next.

442

The next decision block asks, "Are branch and transmit instructions used to go from the original program to its "read a card" or "punch a card" routine?" If the answer to this is "yes" we will not be able to use branch and transmit type instructions to branch our disk routines. And, in the case of this, we go out the left side of this block to the next block which asks, "Is there more than one place where the original program branches to its read or write routines?" The point here is: If branch and transmit instructions had been used by the original program to go into its routine which, after modification, we will be branching from, we may not use a branch and transmit again prior to reaching the branch back. Thus, in cases where the branch and transmit instruction is used by the original program to branch to its I/O routine, we must use a 49 type branch to branch to disk routine. But, if the original program had not used a branch and transmit instruction, we may, in turn, use branch and transmit instructions to get to and from the disk routine.

The next block says, "Does the program recognize the last card of data?" With card operation, when the last data card has been read, the card reader will stop, but the disk will continue to read sectors beyond the last data unless provisions are made to sense this last data. If the answer to this decision block is "no", then some provision must be made so that the program will not read the disk completely never knowing when it has finished the last card. This may be accomplished in two ways:

1. A count can be maintained of the number of cards stored on the disk when the data was originally written on the disk. This count can be checked as the data is read back for the last position.
2. Or, the position following the last data segment on the disk could be loaded with a special indicator which the next, or following, program will recognize as the last data area.

In the case of AUTOSPOT AND AUTOMAP, the program already made provisions for the last card by placing a fini card at the end of the data as it is written. This fini card contained a 99 and was recognized by the following program as terminating the data.

From here, we go to the next decision block which checks for input in alphanumeric form. Again, in this block we run into a uniqueness of the disk which in some cases would be an advantage,but which we must watch for. When we read the disk or write disk in alpha, flags are transferred with the data. This is contrary to reading or writing on cards. Thus, we must make provisions to remove the flags left in the data. The programs with which we are working assumed that there would be no flags in the input data and went on to set flags in the input area which were later used for data transfer. The extra flags left by the disk can cause serious errors if allowed to remain. To correct this, we used a clear flag instruction to clear all flags from the 80 positions of data prior to writing on the disk. In this way, no flags were read back from the disk into the input area.

444

Coming down through the flowchart (Slide 1), we have now reached the point which says load program into core in normal manner. The modifications which we have described up until now may be inserted into the original program deck in two ways:

1. We can modify the original deck prior to loading the program into core by repunching the necessary cards.
2. We can load the program in with no modifications and then write a "trailer program" which will load the modifications on top of the normal program.

Either method is satisfactory.

Now it becomes necessary to transfer the complete modified program onto the disk for recall at a later time. Again, there are a number of ways in which this can be done.

1. This can be done by placing the two necessary disk instructions into the input area. After we had loaded the program with the modifications, we would branch to these instructions which in turn would load the program on the disk. When the program is read back from the disk these two instructions (34 and 38) would still be in the input area but the assumption is that the first data read into this area would be read over these instructions and they would have no affect on the program.
2. If we had used the trailer program to load in our modifications after the initial program had been already loaded into core, we would have included these two

instructions. Again, either way is satisfactory. The main point is to get the program on the disk with the modifications.

Now all that remains is to have the program called in from the disk. Here, we may use the same philosophy which we had used when we had one program calling the following program. The main thing to remember here is that the Op Code 36 instruction must be located just in front of the first instruction to be executed in our next program. In SPS programs, we used the call routine illustrated on Slide 5. We have here an Op Code 41 followed by an Op Code 34, 16 and 49. The program is read into location 0. The Op Code 41 will do nothing. We go to the 34 which will seek the disk address which we specified in the control word located at 44. The 16 transmits immediately the 36 to location 0 and 1. We then branch back to 0 and execute this instruction which will now be a 36 or "read a disk". The new program will be read in and the next instruction executed, after the instruction located at location 0, will be the instruction located at 12 which, in the case of SPS program, will be a branch to the origin of the program.

In the case of non-SPS programs where the branch is not located in position 12, we may use the program similar to the one in the second part of Slide 5. Again, this program is read into location 0: the first instruction is a 34, "seek the disk". The second instruction, an Op Code 26, will transfer the Op Code 36, instruction, located at 0046, to a location just in front of where the next program will start after it is read in. The third instruction Op Code 49 will branch to and execute

TWO CALL PROGRAMS TO CALL AND START PROGRAM ON DISK

CORE LOC.	OP CODE	CONTENTS OF P & Q ADDRESS
00000	41	00044 00702
00012	34	00044 00701
00024	16	00001 00$\overline{3}$6
00036	49	00000
00044		105000 200 00000
		(DISK CONTROL WORD)
00000	34	00032 00701
00012	26	START - 1 00057
00024	49	START - 12
00032		105000 200 00000
		(DISK CONTROL WORD)
00046	$\overline{3}$6	00032 00702
00058		

the Op Code 36 instruction. The Op Code 36 instruction will be executed reading in the new program and the following instruction which will be the first instruction in the new program will be the next one executed.

The final block on Slide 1 points out the advisability of checking the finished program by comparing its output with the output from the program prior to any modifications. In this way, we can be relatively sure we've not altered the main philosophy of the program in any way.

I have included in the appendix a typical set of modifications for your reference. I might add one precaution -- DO NOT let any of your modified programs get on the Monitor Disk. Probably Monitor would have to be reloaded and your program most likely would not run any way.

In concluding, let me say that I hope I have brought to your attention an area of disk operation which has received very little publicity in the past. You must realize that there are very definite limitations to the use of the disk with programs modified in this manner. The program must be in a complete core image and if programs are linked together with each calling the next, the sequence is restricted and there can be little deviation without rewriting the modifications. But, for programs which will run in the same sequence, or for a single program that is run very often, a considerable savings can result. The resulting program is fast, economical and easy to operate.

I realize we have covered some rather technical material here in a rather short time. Therefore, I invite your questions either now or this afternoon during our workshop when we hope to sit down with you and help you modify your post-processors or any other programs which you have to modify. Please bring a copy of your program listing and find out the last location that your program uses in core. Anytime in the near future that I may be of assistance, please feel free to contact me through the Huntsville Branch Office. I hope to see many of you in our workshop this afternoon.

Now, are there any questions?

II

Appendix 1

OP Code Reference Table

and

Disc Word Explanation

CODE	MNEMONIC	TYPE OPERATION
11	AM	Add Immediate
12	SM	Subtract Immediate
13	MM	Multiply Immediate
14	CM	Compare Immediate
15	TDM	Transmit Digit Immediate
16	TFM	Transmit Field Immediate
17	BTM	Branch and Transmit Immediate
21	A	Add
22	S	Subtract
23	M	Multiply
24	C	Compare
25	TD	Transmit Digit
26	TF	Transmit Field
27	BT	Branch and Transmit
31	TR	Transmit Record
32	SF	Set Flag
33	CF	Clear Flag
34	SK	Seek (Q = x07x1)
34	K	Control
35	DN	Dump Numerically
*36	RN	Read Numerically
37	RA	Read Alphamerically
*38	WN	Write Numerically
39	WA	Write Alphamerically

CODE	MNEMONIC	TYPE OPERATION
41	NOP	No Operation
42	BB	Branch Back
43	BD	Branch Digit
44	BNF	Branch No Flag
45	BNR	Branch No Record Mark
46	BI	Branch Indicator
47	BNI	Branch No Indicator
48	H	Halt
49	B	Branch
55	BNG	Branch No Group Mark

*Read-Write disk modifiers on next page.

DISK CONTROL FIELD

In order to read from or write on the disk there are four things that must be known. These are;

(1) The disk drive number if more than one drive is attached to the system.

(2) The five position disk sector address.

(3) The number of sectors to be written or read.

(4) The starting core location.

The disk control field incorporates all four of the above items into a 14 position field. Thus:

F_0	F_1, F_2, F_3, F_4, F_5	S_6, S_7, S_8	$M_9, M_{10}, M_{11}, M_{12}, M_{13}$

The disk drive number is located in F 0. This drive code number varies with the number of drives attached to the system. For drive 0 a 1 is used. For drive 1 a 3 is used.

A sector on the disk is equal to 100 positions of core storage. There are 20,000 sectors on each disk. These sectors are numbered sequentially from 00000 - 19999. The disk control field F1 - F5 contains the sector address. This sector address determines where, on the disk, the write or read will start.

Next is the number of sectors to be read or written. This is located in S6 - S8. The maximum number of sectors that can be read or written is 200 and the minimum number is 001. The method for reading or writing fewer than 100 core locations is explained on the next page in "Read-Write Disk Modifiers".

M9 - M13 contains the core location of the leftmost position of the data transferred to or from the disk. This core location must be an even number.

In a seek, read, or write disk instruction the "P" address is the core location of the leftmost position of the disk control field. This leftmost position must be in an even location.

The "Q" address of the disk instructions contains 07 in Q8 and Q9 and a modifier in Q11. The modifier in all seek instructions is a 1. The modifier in read-write instructions is explained in "Read-Write Disk Modifiers".

READ-WRITE DISK MODIFIERS

All read-write disk instructions must have a "Q" address of "x07xM" where M is the modifier. The modifier determines whether or not a group mark (‡) will have any effect on the data being transferred.

The write disk instruction (38) with a modifier of 0 will be determined after the first group mark encountered in core has been transferred to the disk. If no group mark is encountered the instruction will be terminated when the sector count has been decremented to 000.

The read disk instruction (36) with a modifier of 0 will be terminated after the first group mark encountered on the disk has been transferred into core or, if no group mark is encountered, when the sector has been decremented to 000.

The read or write disk instruction with a modifier of 2 will treat the group mark as data and transfer data until the sector count has been decremented to 000.

III

Appendix 2

Machine Language Modifications to AUTOSPOT, AUTOMAP and Milwaukee Matic Post- Processor

Note:

The post-processor included here is Users Group Library number 10.4.004 - the "Computer Routines for the Milwaukee Matic Solid State Controlled Machining Centers".

AUTOMAP PHASE I

Statement number 1 is a five position field for the indirect address which shows from where to start the transmission of the record (statement #2). This is done because there are two write statements in the main program and each writes from a different location. As each of the two locations are odd numbered, they must be moved to an even location and, since the only locations left are 00000 - 00080, these will be used. Statements #2 and #3 move the data from the odd numbered program output area into even numbered locations. Statement #4 writes the output data on the disk.

Statements 5, 6, and 7 check indicators; address check, wrong-length record/ read back check and write check respectively and, if either the address check or write check indicator is on, a branch to a "seek" instruction (statement #10) and then a branch (statement #11) to the write instruction is made. If the WLR/RBC indicator is on a branch to the next instruction is made simply to turn off the WLR/RBC console light. This is done due to the fact that this indicator is turned on each time a record with length unequal to 100 character multiples is read or written.

Statement 8 adds one (1) to the sector address. Return to the main program is accomplished by a branch back (statement #9).

Statement #12 is the write output data disk word.

Statements 14 - 19 type the message "FINI" to indicate the end of phase 1 and to set up the calling of the next program.

Statements 20 - 25 are changes to the main program. 20 and 22 clear the disk output area. 21 and 23 branch to the write disk routine and transmit the starting address into the area reserved by statement #1. Statement 25 insures a group mark after the 80th position to terminate the write disk instruction.

Statements 26 - 31 dump the program on the disk in a core image.

Statements 32 - 39 load the modifications, read the first program loader card and branch to continue loading the main program.

AUTOMAP PH1

WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
1	19802	0$\bar{0}$	0000	
2	19808	31	00000	1980$\bar{7}$
3	19820	26	00079	15390
4	19832	38	19914	00700
5	19844	46	19894	03600
6	19856	46	19868	03700
7	19868	46	19894	00700
8	19880	11	19919	0$\bar{0}$001
9	19892	42		
10	19894	34	19914	00701
11	19906	49	19832	0
12	19914	10	$\bar{0}$0000	01000
13	19926	00		
14	19928	10	42002	00000
15	19940	00	46495	5490$\bar{7}$
16	19952	39	19943	00100
17	19964	34	19928	00701
18	19976	48	00000	00000
19	19988	36	19928	00702

AUTOMAP PH1
LOADER AND CHANGES TO MAIN PROGRAM

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
20	02504	31	00000	15982
21	02516	17	19808	$\bar{1}6063$
22	11086	31	00000	15982
23	11098	17	19808	$\bar{1}6065$
24	11182	49	19952	00000
25	16062	≠		
26	15402	34	15440	00701
27	15414	16	00004	$\bar{4}1000$
28	15426	38	15440	00702
29	15438	48		
30	15440	10	40002	00000
31	15452	00		
32	00000	36	00080	00500
33	00012	36	15402	00500
34	00024	36	19802	00500
35	00036	36	19882	00500
36	00048	36	19962	00500
37	00060	49	00080	
38	00080	36	00000	00500
39	00092	49	00000	

PHASE I SPECIFICATION SECTION OF

AUTOSPOT - MODIFICATIONS FOR DISK OPERATION

LOAD PROGRAM

```
00000  36  19522  00500
00012  36  19680  00500
00024  36  19840  00500  Load the modifications into core
00036  36  19600  00500
00048  36  19760  00500
00060  36  19920  00500
00072  49  19626  9

19522  31  19648  05819  Save  data for next phase
19534  49  19976  05914  Branch to "call next program"
19546  10500020000000    Disk control for this phase
```

FALSE BRANCH & TRANSMIT

```
19560  10520014904948    Disk control next phase
19574  15  19969  00009  Change 42 to a 49
19586  16  19821  1̅0701  Transfer "write from" address
19598  49  19822  0      Go to entry of program
19606  15  19969  0̅0002  Change 49 to 42
19618  49  10660  0      Return to program
```

MODIFICATIONS TO PROGRAM

```
19526  36  00000  00500  Reset location 0 to 80
19538  26  10654  19785  Modify unit instruction to branch to unit on disk routine
19550  26  07755  19759
19562  26  09039  19771
19574  26  06368  19778  Modify end of program
19586  15  05911  0000≠  Used to save  data for machining section
19698  32  07751  00000
19710  32  09035  00000
19722  34  19546  00701  Write program on disk
19734  38  19546  00702
19746  48
19748  2̅7  19822  02753
19760  1̅7  19822  10877  Modified instructions to be inserted in program
19772  4̅9  19976
19779  4̅9  19574

19786  11  1982̅1  0̅0001
19798  11  19816  0̅0001  Increase address and transmit digit
19810  25  19626  0̅0000
19822  14  19816  1̅9785
19834  47  1̅9878  01200  Has all data been moved?
```

461

PHASE I SPECIFICATION SECTION OF

AUTOSPOT - MODIFICATIONS FOR DISK OPERATION

```
19846  16  19840  1̅9878  Prepares this section for return pass
19858  16  19816  1̅9626
19870  49  19912  0
19878  16  19840  1̅9786  1st pass set-up
19890  49  19810  0
19898  10̅0000̅011̅9626
19912  38  19898  00702  Write data on disk
19924  47  19956  03600
19936  34  19898  00701  Rewrite if we had an address check
19948  49  19912  0
19956  11  19903  0̅0001  Increase sector address by one
19968  42  19606  0      Return to Program
19976  34  19560  00701
19988  36  19560  00702  Call next program
00000
```

462

PHASE I MACHINE SECTION

Same as Phase I specification with these exceptions

```
19522   41   00000  00000
19534   49   19976  04948   Branch to end of program
19546   10520014919746      Disk address for this phase
19560   10540020000000      Disk address for next phase

19626   41   00000  00500
19638   26   15929  19711   Modify write instruction
19650   26   07595  19719   Modify end of program
19662   32   15913  00000
19674   34   19546  00701
19686   38   19546  00702   Load this program on disk
19698   4̲8
19700   49   19574  00000
19712   4̄9   19534
19720

19746   31   05819  19648   To transmit record left by specification section of Phase I
19758   49   04984  0
```

MODIFICATION TO PHASE 2 OF AUTOSPOT TO RUN FROM DISK

PATCH PROGRAM

```
19466  26  13225̄  1203̲3
19478  11  13225  00010
19490  26  13225̄  12055
19502  11  13225  0000̄4  Program patch that was in the way moved to here
19514  49  11226  0
19522  34  19560  00701  End of this phase - seek for next phase
19534  49  19976  06850
19546  10540020000000     Disk control word this section
19560  10535001001900     Disk control word next section
19574  bbbbbbbbbb
```

MODIFICATION TO ORIGINAL PROGRAM

```
19584  36  00000  00500  Read in last cards of modification
19596  36  19680  00500
19608  26  01762  19712  Modify "read a card" instruction
19620  26  03219  19705  Modify error routine
19632  41  00000  00000  No Op
19644  26  11780  19720  Modify "write a card" instruction
19656  26  12024  19727  Modify end to call next program
19668  34  19546  00701
19680  38  19546  00702  Write these modifications all on the disk
19692  48
19694  4̄6  19466  01200
19706  4̄9  197420
19714  4̄9  19912  4̄919522
```

READ DISK DATA

```
19728  10̄00000̄0119600
19742  36  19728  00702  Read a card from disk & seek if necessary then go back
                         and read again
19754  47  19786  03600
19766  34  19728  00701
19778  49  19742  0
19786  11  19733  0̄0001  Increase sector address by one
```

TRANSFER FIELD JUST READ TO EVEN LOCATION

```
19798  25  13056  19600  Transfer data to area beginning with an even address
19810  14  19809  1̄9679
19822  46  19866  01200  Check for last transfer
19834  11  19809  0̄0001  Increase disc location by one
19846  11  19804  0̄0001
19858  49  19798  0      Repeat
```

MODIFICATIONS TO PHASE 2 of AUTOSPOT

TO RUN FROM DISK

TRANSFER FIELD JUST READ TO EVEN LOCATION

```
19866  16  19809  1̄9600  Housekeep because all data is transferred now
19878  16  19804  1̄3056
19890  49  01768  0      Return to program
```

WRITE DATA ON DISK

```
19898  1̄020000̄0113136
19912  38  19898  00702  Write data on disk
19924  47  19956  03600  If address check - seek first then write data
19936  34  19898  00701
19948  49  19912  0
19956  11  19903  0̄0001  Increase sector address by one
19968  49  11786  0      Return to program

19976  36  19560  00702  End of this phase - read in control program for next phase
19988  49  02318  00000
```

Load program to load in modifications into core

```
00000  36  19466  00500  Load modifications into core and branch to the first modificatio
00012  36  19546  00500
00024  36  19626  00500
00036  36  19760  00500
00048  36  19840  00500
00060  36  19920  00500
00072  49  19584  0
```

MODIFICATIONS TO AUTOSPOT PHASE 3 FOR DISK OPERATION

READ DATA FROM THE DISK

```
18312  36  01984  00702  Read data from disk
18324  47  18356  03600
18336  34  01984  00701  If not correct cylinder seek and go back to read
18348  49  18312  0
18356  11  01989  0̅0001  Add one to sector address
18368  25  1̅3411  1̅8626  Transfer data to data read in area which start with an odd
                         address. Use transmit digit 80 times.
18380  14  18379  18705
18392  46  18436  01200
18404  11  18374  0̅0001
18416  11  18379  0̅0001
18428  49  18368  0
18436  16  18374  1̅3411  Housekeep transfer data routine and return to program
18448  16  18379  1̅8626
18460  49  13334  00000
```

CALL OVERLAYED ROUTINE "FLOAT" & CONTROL WORDS

```
18472  16  19964  1̅0292  If program branches to FSIN or FCOS set at to call FLOAT and
                         go to 18786
18484  49  18786  00000
18496  16  19964  1̅0376
18508  49  18786  00000
18520  00000000000000
18534  41  00000  01700  Orgin of program 01700
18546  10560020000000    Disk address this program
18560  1053500100190000  Disk address of next program
```

ROUTINE TO MODIFY ORIGINAL PHASE 3 PROGRAM BEFORE IT IS PLACED ON DISK ALSO 18626 to 18705 ARE USED FOR A TRANSFER OF DATA AREA

```
18576  36  00000  00500  Read in a card and branch to it
18588  49  00000  0
18596  26  13801  00063  Modify write data instruction
18608  26  13329  00071  Modify read data instruction
18620  36  13298  00500  Read in another card
18632  36  13620  00500  Read Modification into 13620
18644  16  10322  1̅9966
18656  16  10406  0̅2590  Modification to original program to call float
18668  16  12882  0̅2570
18680  36  13272  00500  Read modifications into 13272
18692  36  00000  00500  Read original information into 00000
18704  34  18546  00701  Write complete program disk
```

MODIFICATIONS TO AUTOSPOT PHASE 3 FOR DISK OPERATION

ROUTINE TO MODIFY ORIGINAL PHASE 3 PROGRAM BEFORE IT IS PLACED ON DISK ALSO 18626 TO 18705 ARE USED FOR A TRANSFER OF DATA AREA

```
18716  38  18546  00702
18728  34  13280  00701  Write "read/write data on disk" routine on disk
18740  38  13280  00702
18752  48  20 - 0's
```

IF PROGRAM NEEDS FATN CALL IN FLOAT ROUTINE

```
18774  16  19964  1̄2852
18786  34  00000  00701  Seek float and go to 19946 which will read it in
18798  49  19946  0
```

WRITE OUTPUT DATA ON DISK FOR NEXT PROGRAM

```
18806  16  18841  1̄3847  Housekeep transmit digit
18818  16  1̲8836  1̄8626
18830  25  18626  1̄3847  Transmit digit 80 times
18842  14  18836  1̄8705  Have we transmitted digit 80 times?
18854  46  18912  01200
18862  11  18836  0̄0001  Increase count or transmit digit and write data
18878  11  18841  0̄0001
18890  49  18830  0
18902  00000000000000
18912  38  19986  00702  Write data on disk
18924  47  18956  03600
18936  34  19986  00701  If at wrong cylinder seek and rewrite
18948  49  18912  0
18956  11  19991  0̄0001  Increase sector count by one and return to program
18968  49  13806  0
18980
```

INSTRUCTION FOR LOCATION 0 TO 80

```
00000  49  01700  000    Branch to origin of program
00010  44  18312  18966
00022  36  13280  00702  If the read/write disk routine is in core go to R/W if not call it in
00034  49  18312  0
00042  44  18806  18966
00054  36  13280  00702
00066  49  18806  0≠0042M3
00080
```

INSTRUCTION FOR LOCATION 02528 TO 02610

```
02528  39  02559  00100
02540  26  17043  16793  Error message - no tool card:  "NO TC"
02552  49  02634  0
02560  55  56634  30≠
```

MODIFICATIONS TO AUTOSPOT PHASE 3 FOR DISK OPERATION

```
02570  44  18774  18966
02582  49  18668  0       Check to see is "float" routine is in core - if not call it in
02590  44  18496  18966
02602  49  17676  0
02610
```

INSTRUCTIONS FOR LOCATION 19946 to 00000

```
19946  36  13628  00602  Call in float and branch to proper location
19958  49  00000  0
19966  44  18472  18966  Check to see that float is in core if not prepare to call it in
19978  49  17708  0
19986  10̄200000118626     Control word for read data from disk
00000
```

MODIFICATIONS TO AUTOSPOT PHASE 3 FOR DISK OPERATION

Load cards for phase 3 modifications

```
00000  34  00056  00701  Load "float" on to disk
00012  38  00056  00702
00024  36  18552  00500  Load in two modification cards
00036  36  18632  00500
00048  49  18576  0      Call in next card
00056  10536700718312    Control word for float

00000  36  18312  00500  Load in six modification cards
00012  36  18392  00500
00024  36  18472  00500
00036  36  18712  00500
00048  36  18792  00500
00060  36  18872  00500
00072  49  18576  0      Call in next card
00080

00000  36  18930  00500
00012  36  19892  00500  Load in 4 modification cards
00024  36  02530  00500
00036  36  01936  00500
00048  49  18596  0      Return to Modification program
00056  4̄9  00042  0      Modification for write a card
00064  4̄9  00010  0      Modification for read a card
00072
```

TRAILER CONTROL PROGRAM FOR AUTOSPOT

DISK CONTROL WORDS

```
01900   XX̄  XXXX2 0000000 Control word to call post processor
01914   10560020000000     Disk control word to call phase 3
01928   10̄200000110000     Disk control word to punch out put
01942   10̄000000000000     Disk control word for last data written by last phase
01956   10200010010000     Disk control word for write - X per data
01970   10000010010000     Disk control word for read trace per data
```

TRANSFER DATA FROM SECTOR 100000 TO 102000 SO POST PROCESSOR WILL FIND DATA

```
01984  26  01947  19991
01996  34  01970  00701
02008  36  01970  00702  Transfer data, 10,000 location at a time
02020  34  01956  00701
02032  38  01956  00702
02044  11  01975  0̄0100  Increase sector address for transfer by 100
02056  11  01961  0̄0100
02068  24  01975  01947  If more data still - so back & transfer again
02080  47  01996  01100
```

CHECK FOR CARD OUTPUT

```
02092  34  00000  00102
02104  39  02375  00100  Type out instructions
02116  34  00000  00102
02128  48  00000  00000
02140  47  02224  00100  Check switch
```

PUNCH OUTPUT ON CARDS FROM DISK

```
02152  34  01928  00701  Read from the disk one card at a time & punch it
02164  36  01928  00702
02176  38  10000  00400
02188  11  01933  0̄0001
02200  24  01933  01947  If not finished, get next card
02212  47  02152  01100
```

CALL IN NEXT PROGRAM

```
02224  26  01927  01913  Enter here from phase 3
02236  34  00000  00102  Enter here from phase 2
02248  39  02441  00100  Set up switches for next program
02260  48  00000  00000
02272  34  01914  00701
02284  26  19999  02317  Call in next program from disk and branch to its start
02296  49  19988  0
02304  3̄6  01914  00702
02316  00
```

TRAILER CONTROL PROGRAM FOR AUTOSPOT

CHECK SWITCH 3

```
02318  34  00000  00102  Check switch 3 for either phase 3 call or card output
02330  39  02645  00100
02342  48  00000  00000
02354  46  02236  00300
02366  49  02092  0
```

PRINT AREA

```
02374  62  66007  10056  TYPE OUT AREA:              SW 1 O
02386  55  00465  65900  NbFORb
02398  43  41594  40056  CARDbO
02410  64  63215  94562  UT/RES
02422  45  63006  26341  ETbSTA
02434  59  630≠6  24563  RT≠SET
02446  64  57006  26600  UPbSWb
02458  46  56590  05545  FORbNE
02470  67  63005  75956  XTbPRO
02482  47  20594  56245  G-RESE
02494  63  00626  34159  TbSTAR
02506  63  0≠            T≠
02540  ≠0  00
```

LOAD CORE & DISK

```
02544  41  02220  00500
02556  36  02300  00500  Call the remainder of the program into core and load the
                         whole program disk with a correct halt at 02628
02568  36  02380  00500
02580  36  02460  00500
02592  36  02620  00500
02604  34  02630  00701
02616  36  02630  00702
02628  48
02630  10535001001900
02644  62  66007  30056  SWb3bO
02656  55  00465  65900  NbFORb
02668  57  48007  32159  PHb31R
02680  45  62456  30062  ESETbS
02692  63  41596  30≠    TART
```

LOAD ROUTINE - TO LOAD PROGRAM INTO CORE

```
00000  36  01900  00500
00012  36  01980  00500
00024  36  02000  00500
00036  36  02140  00500
00048  36  02220  00500
```

TRAILER CONTROL PROGRAM FOR AUTOSPOT

LOAD ROUTINE - TO LOAD PROGRAM INTO CORE

```
00060   36   02540   00500
00072   49   02544   0
```

AUTOMAP PHASE II

This phase is loaded on the disk in two sections. This is done because all core locations are taken and the "read in" area is defined as "DC" rather than "DS". Statements 1 - 14 load the first section and 15 - 23 load the second section.

Statements 24 - 31 are changes to the main program. Statement 24 branches to the read disk routine and statement 25 adds one (1) to the read sector address upon returning to the main program. Statement 26 transmits the field of numerical blanks to location 00000 rather than to the output area. Number 27 branches to the write disk routine. Number 28 changes the halt after the "END" typeout to a "no op". Statements 29 - 31 change record marks to group marks to insure termination of the write disk instruction after the transfer of 80 characters.

Statements 32 - 42 are the read disk routine. The program branches to the read instruction (#33) and if an address check or write check occurs a branch to the seek instruction (#32) is made. Upon completion of the read operation, statement 37 checks for a "FINI" code. Upon finding a "FINI", statement 39 sets up a branch to end of program routine. Statement 40 branches to the main program.

Statements 43 - 56 are the write disk instructions. The program branches to statement 44, which gives the option of either; (1) putting the output on the disk or (2) punching it in cards. Statements 45 and 46 move the output from the odd numbered core location to an even location. Statement 47 writes on the disk. Statements 48, 49, and 50 check indicators and seeks (statement 43) if necessary. Statement 51 adds one (1) to the sector address and statement 52 returns to the main program. Statement 53 in the punch statement and statement 54 returns to the main program.

The "End of Job" message is contained in statements 57, 58, and 59. Statements 62, 63, 64 and 65 type "End of Job" and call the next program.

AUTOMAP PH2
CORE TO DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
1	00000	36	19640	00500
2	00012	36	19720	00500
3	00024	36	00080	00500
4	00036	36	19900	00500
5	00048	49	00080	
6	00080	36	00000	00500
7	00092	49	00000	
R-W DISK				
CARDS				
8	19900	16	00004	$\bar{4}1000$
9	19912	16	00009	$\bar{0}0000$
10	19924	34	19950	00701
11	19936	38	19950	00702
12	19948	48		
13	19950	10	42001	98000
14	19962	00		
15	00000	36	15000	00500
16	00012	36	19800	00500
17	00024	36	19840	00500

AUTOMAP PH2

CORE TO DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
18	00036	36	19920	00500
19		49	15000	
20	15000	34	15026	00701
21	15012	38	15026	00702
22	15024	48		
23	15026	10	43980	02198
		00		
FROM R-W				
DISK CARDS				
24	01714	49	19652	00000
25	01726	11	19749	$0\bar{0}001$
26	01854	31	00000	11372
27	01866	49	19770	00000
28	07872	41		
29	11290	≠		
30	11371	≠		
31	04603	≠		
32	19640	34	19744	00701
33	19652	36	19744	00700
34	19664	46	19640	03600
35	19676	46	19688	03700

AUTOMAP PH2

CORE TO DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
36	19688	46	19640	00600
37	19700	14	11209	$0000\bar{4}$
38	19712	47	19736	01200
39	19724	16	01720	$\bar{1}9952$
40	19736	49	01726	0
41	19744	10	$\bar{0}0000$	01112
42	19756	10		
43	19758	34	19894	00700
44	19770	46	19874	00400
45	19782	31	00000	11291
46	19794	26	00079	11370
47	19806	38	19894	00700
48	19818	46	19758	03600
49	19830	46	19842	03700
50	19842	46	19758	00700
51	19854	11	19899	$0\bar{0}001$
52	19866	49	01878	0
53	19874	38	11291	00400
54	19886	49	01878	0
55	19894	10	$\bar{2}0000$	01000
56	19906	00		

AUTOMAP PH2
CORE TO DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
57	19908	00	57487	20062
58	19920	63	41596	30043
59	19932	56	5500	
60	19938	10	44000	72000
61	19950	00		
62	19952	39	19909	00100
63	19964	48	00000	00000
64	19976	34	19938	00701
65	19988	36	19938	00702

POST PROCESSOR PHASE I

Statements 1 - 7 are changes to the main program. Statement 1 branches on indicator (equal) when the "FINI" card is read to initialize the starting sector address. Statement 2 branches to the read disk routine. Statement 3 branches to the write disk routine. Statement 4 branches to set up the calling of Phase 2 rather than Phase 3. Statement 5 branches to call the next program. Statement 6 changes the message from "Reload G. P. Output" to "Starting Pass Two." Statement 7 changes another message. The old message was, "Use Phase 2, Contouring", the new message is "Calling Ph 2, Contouring".

Statements 8 - 12 load the program on the disk.

Statements 13 - 21 load the changes into core.

Statements 22 - 31 are the read disk routines. Number 22 is a two position field to receive the transmission from the BTM entry. The group mark on the disk is the 81st character and this program only has 80 positions defined for the read in area so that the first character beyond the read in area must be saved. Statement 24 accomplishes this. Statement 25 then reads disk. Statement 26 turns off WLB/RBC console light, and statement 27 returns the digit moved by statement 24. Statement 29 returns to the main program.

Statements 34 - 44 are the write disk routine. Again in order to get all 80 positions on the disk the 81st character must be moved (#36) and a group mark placed in the 81st position (#37). Statement 38 writes on the disk, statement 39 turns off the WLR/RBC light and statement 40 replaces the digit moved. Statement 41 adds one (1) to the sector address and statement 41 returns to the main program.

Statements 43 - 48 set up the program to call Phase 2 rather than Phase 3 if desired.

Statements 49 - 54 call the next program.

POST PROCESSOR PH1

READ-WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
1	11132	46	19784	01200
2	11556	17	19696	$000\bar{0}0$
3	19512	17	19806	$000\bar{0}0$
4	19596	17	19908	$000\bar{0}0$
5	19620	17	19952	$000\bar{0}0$
6	02033	STARTING PASS TWO (Alphamerically Coded)		
7	01985	CALLING PH 2 (Alphamerically Coded)		
CORE TO DISK				
8	01770	34	01808	00701
9	01782	16	00004	$\bar{4}1000$
10	01794	38	01808	00702
11	01806	48		
12	01808	10	58002	00000
		00		
LOADER				
13	00000	36	19694	00500
14	00012	36	19774	00500
15	00024	36	19854	00500

POST PROCESSOR PH1

READ-WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
16	00036	36	19934	00500
17	00048	36	01770	00500
18	00060	36	00080	00500
19	00072	49	00080	
20	00080	36	00000	00500
21	00092	49	00000	
22	19694	$\bar{0}0$		
23	19696	34	19770	00701
24	19708	25	19703	01850
25	19720	36	19770	00700
26	19732	46	19744	03700
27	19744	25	01850	19703
28	19756	11	19775	$0\bar{0}001$
29	19768	42		
30	19770	10	$\bar{2}0000$	01017
31	19782	70		
32	19784	16	19775	$\bar{0}2000$
33	19796	49	11496	0
34	19804	$\bar{0}0$		

POST PROCESSOR PH1

READ-WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
35	19806	34	19892	00701
36	19818	25	19837	10086
37	19830	15	10086	0000≠
38	19842	38	19892	00700
39	19854	46	19866	03700
40	19866	25	10086	19837
41	19878	11	19897	$0\bar{0}001$
42	19890	42		
43	19892	10	$\bar{0}0000$	01100
44	19904	06		
45	19906	$\bar{0}0$		
46	19908	16	19939	$\bar{0}6000$
47	19920	39	01985	00100
48	19932	42		
49	19934	$1\bar{0}$	62002	00000
50	19946	00	$00\bar{0}0$	
51	19952	39	01931	00100
52	19964	48	00000	00000
53	19976	34	19934	00701
54	19988	36	19934	00702

POST PROCESSOR PHASE II

Statements 1 - 16 are the read disk routine. The main program branches to the read statement (#4), then checks indicators (statements 5, 6, and 7). If the disk address check or read check indicator is on a branch to a seek (#3) is made. Following a correct transfer from disk to core the input data is transferred to the odd input address. Statements 8 - 12 are needed for this. Upon completion of transfer of the 80th character, statements 13 and 14 initialize the transmit digit instruction (#8). Statement 15 adds one (1) to the sector address and #16 returns to the main program.

Statements 17 - 29 are the write disk routine. Since there are only 80 positions defined as an output area, the 81st position must be saved (#21) in order to set a group mark (#22) to terminate the read instruction of the next program. Following this is the write disk instruction (#23), indicator checking instructions (statements 24, 25, and 26) and a branch to a seek (#19), if necessary. After the seek the digit is transmitted to the 81st position (#20) before returning to statement 21. Upon completion of the transfer from core to disk, the 81st digit is replaced (#27), one (1) is added to the sector address (#28) and statement 29 returns to the main program.

Because Phase 2 is not always used and only two areas are defined on the disk for input-output, it is necessary to move the data output by Phase 2 so that input for programs to come will be properly oriented. Statements 34 - 60 do this.

484

Statements 30 - 33 move the output exchange statements to the high end of core, out of the way of incoming data. On completion of the exchange, statements 61 - 70 set up the call of the next program.

Statements 71 - 73 are changes to the main program. #71 branches to the read routine. #72 branches to the write routine and 73 branches to end of job routine.

Statements 74 - 79 load the program on the disk.

Statements 80 - 94 load the changes into core.

POST PROCESSOR PH2

READ-WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
1	10720	$1\bar{0}$	00000	01150
2	10732	00		
3	10734	34	10720	00701
4	10746	36	10720	00700
5	10758	46	10734	03600
6	10770	46	10782	03700
7	10782	46	10734	00600
8	10794	25	$\bar{0}2365$	$\bar{1}5000$
9	10806	11	10800	$\bar{0}0001$
10	10818	11	10805	$\bar{0}0001$
11	10830	14	10805	$\bar{1}5080$
12	10842	47	10794	01200
13	10854	16	10800	$\bar{0}2365$
14	10866	16	10805	$\bar{1}5000$
15	10878	11	10725	$\bar{0}0001$
16	10890	49	01844	0
17	10898	$1\bar{0}$	20000	01024
18	10910	70		
19	10912	34	10898	00701

POST PROCESSOR PH2

READ WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
20	10924	15	02550	00000
21	10936	25	10935	02550
22	10948	15	02550	0000‡
23	10960	38	10898	00700
24	10972	46	10912	03600
25	10984	46	10996	03700
26	10996	46	10912	00700
27	11008	25	02550	10935
28	11020	11	10903	$\bar{0}0001$
29	11032	49	03264	0
30	11040	31	19618	11084
31	11052	31	19938	11404
32	11064	39	03475	00100
33	11076	49	19626	0
34	11084 (19618)	$\bar{0}0$	0	
35	11087 (19621)	$\bar{0}0$	000	
36	11092 (19626)	26	19625	10903
37	11104 (19638)	12	19625	$\bar{0}2100$
38	11116 (19650)	11	19620	$00\bar{0}01$
39	11128 (19662)	14	19625	$\bar{0}0000$

POST PROCESSOR PH2

READ WRITE DISK

STATEMENT NUMBER	CORE LOCATION		OP CODE	P ADDRESS	Q ADDRESS
40	11140	(19674)	47	19734	01300
41	11152	(19686)	12	19625	$\bar{0}0100$
42	11164	(19698)	11	19620	$00\bar{0}01$
43	11176	(19710)	14	19625	$\bar{0}0000$
44	11188	(19722)	46	19686	01100
45	11200	(19734)	34	19888	00701
46	11212	(19746)	36	19888	00702
47	11224	(19758)	46	19734	00600
48	11236	(19770)	34	19874	00701
49	11248	(19782)	38	19874	00702
50	11260	(19794)	46	19770	00700
51	11272	(19806)	11	19879	$\bar{0}0100$
52	11284	(19818)	11	19893	$\bar{0}0100$
53	11296	(19830)	12	19620	$00\bar{0}01$
54	11308	(19842)	14	19620	$00\bar{0}00$
55	11320	(19854)	47	19734	01200
56	11332	(19866)	49	19952	0
57	11340	(19874)	$1\bar{0}$	00001	00005
58	11352	(19886)	00		
59	11354	(19888)	$1\bar{0}$	20001	00005
60	11366	(19900)	00		

POST PROCESSOR PH2
READ WRITE DISK

STATEMENT NUMBER	CORE LOCATION		OP CODE	P ADDRESS	Q ADDRESS
61	11368	(19902)	00	59456	24563
62	11380	(19914)	00	62634	15963
63	11392	(19926)	23	00574	8730≠
64	11404	(19938)	10	62002	00000
65	11416	(19950)	00		
66	11418	(19952)	39	19901	00100
67	11430	(19964)	48	00000	00000
68	11442	(19976)	34	19938	00701
69	11454	(19988)	36	19938	00702
70	11466		≠		
CHANGE TO MAIN PROGRAM					
71	01832		49	10758	00000
72	03252		49	10936	00400
73	03372		49	11040	00000
CORE TO DISK					
74	15000		34	15038	00701
75	15012		16	00004	$\bar{4}1000$
76	15024		38	15038	00702
77	15036		48		

POST PROCESSOR PH2

READ WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CODE	P ADDRESS	Q ADDRESS
78	15038	10	60001	20000
79	15050	00		
80	00000	36	00080	00500
81	00012	36	00160	00500
82	00024	36	15000	00500
83	00036	36	10720	00500
84	00048	36	10800	00500
85	00060	36	10880	00500
86	00072	36	10960	00500
87	00084	36	11040	00500
88	00096	36	11120	00500
89	00108	36	11200	00500
90	00120	36	11280	00500
91	00132	36	11360	00500
92	00144	36	11440	00500
93	00156	36	00000	00500
94	00168	49	00000	

POST PROCESSOR PHASE III

Statements 1 - 18 are the read routine. The main program branches to the read instruction (#5) and transmits the starting core location into the disk word (#3 and 4). Statements 6, 7, and 8 check indicators and branch to "seek" (#1) if necessary. Statements 9 - 14 transfer the input data to the odd input address. Statements 15 and 16 initialize statement 9. Statement 18 returns to the main program.

Statements 19 - 32 are the output routine. The main program branches to #23, checks indicator (program switch 2) and if it is on punches a card (#31) then returns to the main program (#32). If program switch 2 is off the program will write the output on the disk and return to the main program (#30).

Statements 33 - 35 are "fill in zeros".

Statements 38 - 40 are the end of job message.

Statements 41 - 45 type end of job and call the next program.

Statements 46 - 51 are changes to the main program. Statement 46 "branches and transmits" to the read routine and 47, 48, 49, and 51 branch and transmit to the write routine. Statement 50 branches to the end of job routine.

Statements 52 - 57 transfer a core image to the disk.

Statements 58 - 67 load the changes into core.

POST PROCESSOR PH3

READ WRITE DISK

SATEMENT NUMBER	CORE LOCATION	OP CORE	P ADDRESS	Q ADDRESS
1	19524	34	19544	00701
2	19536	49	19558	0
3	19544	$1\bar{0}$	00000	$01\bar{0}00$
4	19556	00		
5	19558	36	19544	00700
6	19570	46	19582	03700
7	19582	46	19524	03600
8	19594	46	19524	00600
9	19606	25	$\bar{0}3653$	$\bar{0}0000$
10	19618	14	19612	$\bar{0}3732$
11	19630	46	19674	01200
12	19642	11	19612	$\bar{0}0001$
13	19654	11	19617	$\bar{0}0001$
14	19666	49	19606	0
15	19674	16	19612	$\bar{0}3653$
16	19686	16	19617	$\bar{0}0000$
17	19698	11	19549	$\bar{0}0001$
18	19710	42		
19	19712	34	19732	00701
20	19724	49	19746	0

POST PROCESSOR PH3

READ WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CORE	P ADDRESS	Q ADDRESS
21	19732	$1\bar{0}$	20000	$02\bar{0}39$
22	19744	78		
23	19746	46	19832	00200
24	19758	15	04139	00007
25	19770	38	19732	00700
26	19782	46	19794	03700
27	19794	46	19712	03600
28	19806	46	19712	00700
29	19818	11	19737	$\bar{0}0002$
30	19830	42		
31	19832	39	03979	00400
32	19844	42	00000	00000
33	19856	00	00000	00000
34	19868	00	00000	00000
35	19880	00	00000	000
36	19890	10	64001	70000
37	19902	00		
38	19904	59	45624	56300
39	19916	62	63415	96323
40	19928	00	43565	56507
41	19940	39	03085	00100
42	19952	39	19901	00100

POST PROCESSOR PH3

READ WRITE DISK

STATEMENT NUMBER	CORE LOCATION	OP CORE	P ADDRESS	Q ADDRESS
43	19964	48	00000	00000
44	19976	34	19890	00701
45	19988	36	19890	00702
CHANGE TO MAIN PROGRAM				
46	01940	17	19558	$\bar{0}0000$
47	02536	17	19746	$\bar{0}3978$
48	11754	17	19746	$\bar{0}3978$
49	12546	17	19746	$\bar{0}3978$
50	12570	49	19940	00000
51	13894	17	19746	$\bar{0}3978$
CORE TO DISK				
52	03654	34	03692	00701
53	03666	16	00004	$\bar{4}1000$
54	03678	38	03692	00702
55	03690	48		
56	03692	10	62002	00000
57	03704	00		

POST PROCESSOR PH3

LOADER

STATEMENT NUMBER	CORE LOCATION	OP CORE	P ADDRESS	Q ADDRESS
58	00000	36	00080	00500
59	00012	36	03654	00500
60	00024	36	19524	00500
61	00036	36	19604	00500
62	00048	36	19684	00500
63	00060	36	19764	00500
64	00072	36	19844	00500
65	00084	36	19924	00500
66	00096	36	00000	00500
67	00108	49	00000	

IV

Appendix 3

Sample Problem

```
PG LIN LABEL  OP    OPERANDS                                                  LOCN  OP INSTRUCTIONS
                    *        AUTOSPOT SEPTEMBER 30, 1962
   001              *              AUTOSPOT PHASE2
   002              *        PHASE2 SCANS OUTPUT OF PHASE1 RECORD BY RECORD
   003              *        SW2 ON CARD INPUT/OUTPUT
   004              *        SW4 OFF PRINT OUTPUT OF PHASE2
   005              *        PUSH START TO CONTINUE ON ERROR STOPS
   006              *        ERROR1 UNIDENTIFIED REC TYPE
   007              *        ERROR2 TOOL NUMBER NOT IN TOOL TABLE
   008              *        ERROR3 PATTERN TABLE EXCEEDED
   009              *        ERROR4 PATTERN LENGTH EXCEEDED
   009              *        ERROR5 PATTERN GENERATION EXCEEDED
   009              *        ERROR6 PATTERN OPERATION ERROR
 1 126        DORG  01700                                                     01700
 1 130 NEXT   BC2     CARDS                                                   01700 46 01732 00200
 1 131        RNPT  INPUT-47,,,TAPE INPUT                                     01712 36 12025 00300
 1 132        B     RCOMP                                                     01724 49 01840 00000
 1 133        DORG  *-4                                                       01731
 1 134 CARDS  BNF   CARDX,PFLAG-1,,CARD INPUT                                 01732 44 01780 13261
 1 138        CF    PFLAG-1                                                   01744 33 13261 00000
 1 140 RCARD  RNCD  CARDI-79,,,READ FIRST CARD                                ~~01756 36 13056 00500~~  49 19742
 1 142        TFM   INLOC,CARDI-79                                            01768 16 13220 J3056
 1 144 CARDX  BNF   RCARD,INLOC,11,NEXT CARD                                  01780 44 01756 1322-
 1 146        TR    INPUT-47,INLOC,11                                         01792 31 12025 1322-
 1 148        AM    INLOC,1,10,INDEX INPUT POINTER                            01804 11 13220 000-1
 1 150        BNR   *-12,INLOC,11                                             01816 45 01804 1322-
 1 152        AM    INLOC,1,10                                                01828 11 13220 000-1
 1 181 RCOMP  TFM   COMP1,COMP1&2                                             01840 16 12752 J2754
 1 182 SERCH1 CM    COMP1,COMP1&20                                            01852 14 12752 J2772
 1 183        BH    TRY26                                                     01864 46 01988 01100
 1 185        C     INPUT-46,COMP1,11                                         01876 24 12026 1275K
 1 190        BE    CFIND                                                     01888 46 01920 01200
 1 200        AM    COMP1,2,10                                                01900 11 12752 000-2
 2 010        B     SERCH1                                                    01912 49 01852 00000
 2 020        DORG  *-4                                                       01919
 2 021 CFIND  TR    OUTPUT-47,INPUT-47                                        01920 31 12916 12025
 2 022        BNF   *&36,CFIND                                                01932 44 01968 01920
 2 023        SF    OUTPUT-46                                                 01944 32 12917 00000
 2 024        CF    CFIND                                                     01956 33 01920 00000
 2 025        BTM   OPUT,,7                                                   01968 17 11484 -0000
 2 026        B     NEXT                                                      01980 49 01700 00000
 2 028        DORG  *-4                                                       01987
 2 051 TRY26  CM    INPUT-46,26,10,TEST FOR GO TO                             01988 14 12026 000K6
 2 052        BE    GOTO                                                      02000 46 02036 01200
 2 053        CM    INPUT-46,39,10,TEST FOR STOP                              02012 14 12026 000L9
 2 054        BNE   CONT                                                      02024 47 02232 01200
 2 055 GOTO   BNF   MAJCK,PFLAG&11,,TEST FOR PATTERN                          02036 44 02104 13273
 2 056        BD    CFIND,INPUT-45                                            02048 43 01920 12027
```

```
PG LIN LABEL  OP    OPERANDS                                                  LOCN  OP INSTRUCTIONS

 2 057        BTM   SP,INPUT-47,7,STO IN PAT TABLE                            02060 17 11288 J2025
 2 058        CM    RTYPE,1,10                                                02072 14 12127 000-1
 2 058        BE    NEXT,,,READ NEXT RECORD                                   02084 46 01700 01200
 2 058        B     CFIND                                                     02096 49 01920 00000
 2 058        DORG  *-4                                                       02103
 2 058 MAJCK  CM    INPUT-44,,10                                              02104 14 12028 000-0
 2 058        BNE   CFIND                                                     02116 47 01920 01200
 2 058        CM    INPUT-42,11,10,TEST FOR Z COORD                           02128 14 12030 000J1
 2 058        BE    CFIND                                                     02140 46 01920 01200
 2 058        FA    INPUT-32,TXTY-11                                          02152 16 00469 -2187
 2 058                                                                        02164 16 00445 J2040
 2 058                                                                        02176 49 00422 J2114
 2 058        FA    INPUT-22,TXTY-1                                           02188 16 00469 -2223
 2 058                                                                        02200 16 00445 J2050
 2 058                                                                        02212 49 00422 J2124
 2 058        B     CFIND                                                     02224 49 01920 00000
 2 058        DORG  *-4                                                       02231
 2 058 CONT   CM    INPUT-46,10,10,TEST FOR NEW MINOR SECT                    02232 14 12026 000J0
 2 069        BE    PATCK                                                     02244 46 02752 01200
 2 080        BH    CODE                                                      02256 46 02868 01100
 2 030        BNF   NEXT,PFLAG                                                02268 44 01700 13262
 2 030        CF    PFLAG                                                     02280 33 13262 00000
 2 030        BNF   *&36,INPUT-46                                             02292 44 02328 12026
 2 080        CF    INPUT-46                                                  02304 33 12026 00000
 2 080        SF    CFIND                                                     02316 32 01920 00000
 2 080        TF    RTYPE,INPUT-46,,SAVE REC TYPE                             02328 26 12127 12026
 2 081        BNF   CKODD,PFLAG&11                                            02340 44 02436 13273
 2 082        CF    PFLAG&11                                                  02352 33 13273 00000
 2 083        BD    PLONG,PFLAG&1,,TEST PAT LENGTH                            02364 43 11172 13263
 2 084        SF    PFLAG&2                                                   02376 32 13264 00000
 2 085        TF    SCNCT,PFLAG&4,6,RESET CNTR                                02388 26 1325N 13266
 2 085        TFM   PFLAG&4,,8                                                02400 16 13266 0-000
 2 088        TF    TYLOC,PLEV,6,SET PAT GENERATION INDICATOR                 02412 26 1279- 13277
 2 089        TFM   PLEV,,10                                                  02424 16 13277 000-0
 2 120 CKODD  MM    INPUT-46,05,10,TEST FOR NEW PAT                           02436 13 12026 000-5
 2 130        BD    PATN,99                                                   02448 43 02600 00099
 2 140 EVEN   CM    INPUT-46,,10                                              02460 14 12026 000-0
 2 144        BE    NEXT                                                      02472 46 01700 01200
 2 146        TF    TYPE,INPUT-43,,SAVE OPERATION CODE                        02484 26 12130 12029
 2 147        CM    INPUT-46,02,10                                            02496 14 12026 000-2
 2 148        BE    CFIND                                                     02508 46 01920 01200
 2 149        CM    INPUT-46,06,10                                            02520 14 12026 000-6
 2 150        BE    SIX                                                       02532 46 02580 01200
 2 153        TF    TOLNO,INPUT-39,,SAVE TOOL NUMBER                          02544 26 12134 12033
 2 154        CM    INPUT-46,04,10                                            02556 14 12026 000-4
 2 155        BE    CFIND                                                     02568 46 01920 01200
 2 158 SIX    TF    TOLMOD,INPUT-36,,SAV TOOL MOD                             02580 26 12177 12036
 2 159        B     CFIND                                                     02592 49 01920 00000
```

```
PG LIN LABEL  OP    OPERANDS                                                   LOCN  OP INSTRUCTIONS

 2 160        DORG  *-4                                                        02599
 2 171 PAIN   SF    PFLAG&11,,,PATTERN OPERATION                               02600 32 13273 00000
 2 171        SF    PFLAG&10,,,FOR SAVE                                        02612 32 13272 00000
 2 172        SM    INPUT-46,01,10,SET RECORD TYPE EVEN                        02624 12 12026 000-1
 2 173        AM    PTAB,01,10                                                 02636 11 13260 000-1
 2 177        AM    PCNT,01,10                                                 02648 11 13275 000-1
 2 178        TF    PTAB,PCNT,6,TRANSFER PATTERN NUMBER                        02660 26 1326- 13275
 2 178        AM    PTAB,02,10                                                 02672 11 13260 000-2
 2 178        TF    TYLOC,PTAB                                                 02684 26 12790 13260
 2 179        AM    PTAB,03,10                                                 02696 11 13260 000-3
 2 180        TF    SCNCT,PTAB,,ADDRESS OF SCAN COUNT FOR THIS PATTERN         02708 26 13255 13260
 2 181        AM    PTAB,01,10                                                 02720 11 13260 000-1
 2 182        AM    PFLAG&4,7,10                                               02732 11 13266 000-7
 2 184        B     EVEN                                                       02744 49 02460 00000
 2 186        DORG  *-4                                                        02751
 2 186 PATCK  SF    PFLAG                                                      02752 32 13262 00000
 2 187        BNF   MINOR,PFLAG&11                                             02764 44 02812 13273
 2 188        BTM   SP,INPUT-45,7,STO IN PAT TABLE                             02776 17 11288 J2027
 2 193        CM    RTYPE,01,10                                                02788 14 12127 000-1
 2 194        BE    NEXT,,,01 REC TYPE-NO OPUT                                 02800 46 01700 01200
 2 198 MINOR  TR    TXTY-20,INPUT-43,,SAVE TX AND TY                           02812 31 12105 12029
 3 010        TR    OUTPUT-47,INPUT-47                                         02824 31 12916 12025
 3 020        TR    OUTPUT-43,AUXIN,,SET CONTROL REC MARK                      02836 31 12920 12104
 3 022        BTM   OPUT,,,OUTPUT THIS RECORD                                  02848 17 11484 -0000
 3 040        B     NEXT                                                       02860 49 01700 00000
 3 050        DORG  *-4                                                        02867
10 010 CODE   TFM   COMP2,COMP2&2                                              02868 16 12777 J2779
10 012 FIND2  CM    COMP2,COMP2&8                                              02880 14 12777 J2785
10 014        BH    TRY13                                                      02892 46 02948 01100
10 016        C     INPUT-46,COMP2,11,TEST FOR PATTERN OPERATION               02904 24 12026 1277P
10 018        BE    PATOP                                                      02916 46 03748 01200
10 020        AM    COMP2,02,10                                                02928 11 12777 000-2
10 022        B     FIND2                                                      02940 49 02880 00000
10 014        DORG  *-4                                                        02947
 3 060 TRY13  CM    INPUT-46,16,10                                             02948 14 12026 000J6
 3 070        BH    AUX,,,TEST FOR AUXILLIARY SECTION                          02960 46 03124 01100
 3 072        BNF   TRANS,PFLAG&11,,TEST FOR PATTERN                           02972 44 03020 13273
 3 074        BTM   SP,INPUT-47,7,STO IN PAT TABLE                             02984 17 11288 J2025
 3 075        CM    RTYPE,1,10,TEST FOR DEFINE STATEMENT                       02996 14 12127 000-1
 3 088        BE    NEXT,,,01 REC TYPE                                         03008 46 01700 01200
 3 089 TRANS  CM    INPUT-46,11,10                                             03020 14 12026 000J1
 3 090        BE    CFIND                                                      03032 46 01920 01200
 3 100        FA    INPUT-36,TXTY-11                                           03044 16 00469 -3079
 3 100                                                                         03056 16 00445 J2036
 3 100                                                                         03068 49 00422 J2114
 3 110        FA    INPUT-26,TXTY-1                                            03080 16 00469 -3115
 3 110                                                                         03092 16 00445 J2046
 3 110                                                                         03104 49 00422 J2124
```

```
PG LIN LABEL  OP    OPERANDS                                                   LOCN  OP INSTRUCTIONS

 3 120        B     CFIND                                                      03116 49 01920 00000
 3 130        DORG  *-4                                                        03123
 3 140 AUX    CM    INPUT-46,49,10                                             03124 14 12026 000M9
 3 142        BL    TRY30                                                      03136 47 03232 01300
 3 144        CM    INPUT-46,51,10,TEST FOR DAIMETER                           03148 14 12026 000N1
 3 146        BE    EQU51                                                      03160 46 03264 01200
 3 150        CM    INPUT-46,65,10                                             03172 14 12026 00005
 3 152        BL    CFIND                                                      03184 47 01920 01300
 3 154        CM    INPUT-46,91,10,TEST FOR TOOL CARD                          03196 14 12026 000R1
 3 156        BE    EQU91                                                      ~~03208 46 19942 01200~~   46 19466 01200
 3 158 ERR1   BTM   ERSUB,7100,8,WRITE ERROR MESSG                             03220 17 11190 0P100
 3 167        DORG  19942                                                      19942
 3 168 EQU91  TF    TABPT,INPUT-39,6,SAVE TOOL NO                              19942 26 1322N 12033
 3 174        AM    TABPT,10,10                                                19954 11 13225 000J0
 3 176        TF    TABPT,INPUT-17,6,SAVE TIP ANGLE                            19966 26 1322N 12055
 3 178        AM    TABPT,04,10                                                19978 11 13225 000-4
 3 180        B     ERSUB&36,,,OUTPUT TOOL CARD                                19990 49 11226 00000
 3 182        DORG  ERR1&12                                                    03232
 3 184 TRY30  CM    INPUT-46,30,10,TEST FOR REMARK STAT                        03232 14 12026 000L0
 3 186        BE    ERSUB&36                                                   03244 46 11226 01200
 3 188        B     ERR1,,,ILLEGAL CODE                                        03256 49 03220 00000
 3 190        DORG  *-4                                                        03263
 4 040 EQU51  CM    TYPE,7,9,TEST FOR CSINK OR SPDRILL                         03264 14 12130 00-07
 4 050        BE    ANGLE                                                      03276 46 03312 01200
 4 060        CM    TYPE,009,9                                                 03288 14 12130 00-09
 4 070        BNE   CFIND                                                      03300 47 01920 01200
 4 072 ANGLE  TFM   ADRES,TBLE-556                                             03312 16 13230 J2191
 4 078 SERCH2 CM    ADRES,TBLE-19,,SCAN FOR TOOL NO                            03324 14 13230 J2728
 4 080        BL    *&24                                                       03336 47 03360 01300
 4 081        BTM   ERSUB,7200,8,TOOL NO MISSING                               03348 17 11190 0P200
 4 086        C     TOLNO,ADRES,11                                             03360 24 12134 1323-
 4 088        BE    ANGLOC                                                     03372 46 03404 01200
 4 090        AM    ADRES,14,10                                                03384 11 13230 000J4
 4 092        B     SERCH2                                                     03396 49 03324 00000
 4 094        DORG  *-4                                                        03403
 4 096 ANGLOC AM    ADRES,10,10,COMPUTE DEPTH                                  03404 11 13230 000J0
 4 120        FM    -ADRES,RCONV                                               03416 16 00469 -3451
 4 120                                                                         03428 26 01260 1323-
 4 120                                                                         03440 49 01262 J3409
 4 125        TF    THETA,99,,SAVE CONVERTED ANGLE                             03452 26 12154 00099
 4 130        FD    THETA,TWO                                                  03464 16 00469 -3499
 4 130                                                                         03476 26 01260 12154
 4 130                                                                         03488 49 01422 J2144
 4 134        TF    THETA,00099                                                03500 26 12154 00099
 4 140        FSIN  BUFFN,THETA                                                03512 16 14270 -3547
 4 140                                                                         03524 16 14258 J2164
 4 140                                                                         03536 49 13548 J2154
 4 150        FCOS  BUFFD,THETA                                                03548 16 14270 -3583
```

```
PG LIN LABEL  OP    OPERANDS                                            LOCN  OP INSTRUCTIONS

 4 150                                                                  03560 16 14258 J2174
 4 150                                                                  03572 49 13516 J2154
 4 160        FD    BUFFN,BUFFD                                         03584 16 00469 -3619
 4 160                                                                  03596 26 01260 12164
 4 160                                                                  03608 49 01422 J2174
 4 170        TF    THETA,99                                            03620 26 12154 00099
 4 190        FD    INPUT-36,TWO                                        03632 16 00469 -3667
 4 190                                                                  03644 26 01260 12036
 4 190                                                                  03656 49 01422 J2144
 4 200        TF    INPUT-36,00099                                      03668 26 12036 00099
 5 010        FD    INPUT-36,THETA                                      03680 16 00469 -3715
 5 010                                                                  03692 26 01260 12036
 5 010                                                                  03704 49 01422 J2154
 5 020        TF    INPUT-36,99,,COMPUTED DEPTH TO BUFFER               03716 26 12036 00099
 5 040        TDM   INPUT-46,,,SET DEPTH CODE                           03728 15 12026 00000
 5 050        B     CFIND,,,OUTPUT DEPTH                                03740 49 01920 00000
 5 060        DORG  *-4                                                 03747
10 019 PATOP  SF    PATOP,,,PATTERN OPERATION                           03748 32 03748 00000
10 020        SF    PFLAG                                               03760 32 13262 00000
10 021        CM    RTYPE,5,10,TEST FOR HIGHER GENERATION PAT           03772 13 12127 000-5
10 040        BD    ODD40,00099                                         03784 43 03804 00099
10 050        B     PDUMP,,,OUTPUT THIS PATTERN                         03796 49 04000 00000
10 055        DORG  *-4                                                 03803
10 060 ODD40  BNF   STORE,PFLAG&10                                      03804 44 03864 13272
10 070        CF    PFLAG&10                                            03816 33 13272 00000
10 080        SM    PTAB,04,10,INDEX POINTER BACKWARDS                  03828 12 13260 000-4
10 090        TF    TYLOC,PTAB,,SAVE ADD OF GENERATION INDICATOR        03840 26 12790 13260
10 100        AM    PTAB,04,10                                          03852 11 13260 000-4
10 110 STORE  BTM   SP,INPUT-47,7,STORE HIGHER GENERATION PATTERN       03864 17 11288 J2025
10 111        BD    LEVOK,PLEV-1                                        03876 43 03976 13276
10 113        TF    PATNO,INPUT-42,,SAVE PATTERN NO                     03888 26 12890 12030
10 114        BT    PATLOC,PATNO,,LOCATE THIS PAT IN  PAT TABLE         03900 27 11396 12890
10 115        AM    COMP9&6,2,10,INDEX FST LEV POINTER                  03912 11 11414 000-2
10 116        BD    *&20,COMP9&6,11,TEST PAT LEVEL                      03924 43 03944 1141M
10 117        B     TFM1                                                03936 49 03964 00000
10 118        DORG  *-4                                                 03943
10 118        TFM   PLEV,11,10,SET PAT LEV INDICATOR                    03944 16 13277 000J1
10 119        B     LEVOK                                               03956 49 03976 00000
10 120        DORG  *-4                                                 03963
10 121 TFM1   TFM   PLEV,1,10                                           03964 16 13277 000-1
10 125 LEVOK  CM    RTYPE,1,10,TEST FOR OUTPUT                          03976 14 12127 000-1
10 150        BE    NEXT                                                03988 46 01700 01200
15 030 PDUMP  TF    PATNO,INPUT-42                                      04000 26 12890 12030
15 040        BT    PATLOC,PATNO,,LOCATE THIS PATTERN                   04012 27 11396 12890
15 050        AM    COMP9&6,02,10,INDEX POINTER FORWARD                 04024 11 11414 000-2
15 052        BNF   *&36,PATOP,,SWITCH FOR PATTERN GENERATION           04036 44 04072 03748
15 054        CF    PATOP                                               04048 33 03748 00000
15 056        TF    GNO,COMP9&6,11,SAVE GENERATION LEV INDICATOR        04060 26 13515 1141M
```

```
PG LIN LABEL  OP    OPERANDS                                            LOCN  OP INSTRUCTIONS

15 060        BNF   CKON2,PFLAG&7,,TEST 3RD LEV FLAG                    04072 44 04268 13269
15 070        BD    *&20,COMP9&6,11                                     04084 43 04104 1141M
15 080        B     RING1                                               04096 49 04344 00000
15 090        DORG  *-4                                                 04103
15 100        BNF   LOAD2,PFLAG&8,,IS A PAT DEFINED ON A PAT            04104 44 04128 13270
15 105        BTM   ERSUB,7500,8,TEST FOR PAT GEN ERROR                 04116 17 11190 0P500
15 120 LOAD2  SF    PFLAG&8,,,SEC LEV SWITCH                            04128 32 13270 00000
15 130        TF    PLOC2,COMP9&6,,LOAD SEC GEN OPERATION BUFFERS       04140 26 12795 11414
15 140        AM    PLOC2,03,10                                         04152 11 12795 000-3
15 150        TF    PCNT2,PLOC2                                         04164 26 12800 12795
15 160        A     PCNT2,PLOC2,11                                      04176 21 12800 1279N
15 170        SM    PCNT2,08,10,SET LENGTH OF THIS PATTERN              04188 12 12800 000-8
15 172        AM    PLOC2,1,10                                          04200 11 12795 000-1
15 173        TR    OPER2-1,PLOC2,11,SAVE OPER,SURF,PATNO, AND COORD TRANS 04212 31 12801 1279N
15 174        TF    PATNO,SURF2&2                                       04224 26 12890 12806
15 175        AM    PLOC2,1,10                                          04236 11 12795 000-1
15 176        BNR   *-12,PLOC2,11,INDEX SCAN POINTER                    04248 45 04236 1279N
15 330        B     PDUMP&12                                            04260 49 04012 00000
15 340        DORG  *-4                                                 04267
15 350 CKON2  BNF   CKSIM,PFLAG&8,,TEST FOR 2ND GEN PAT                 04268 44 04332 13270
15 360        BD    *&20,COMP9&6,11                                     04280 43 04300 1141M
15 370        B     RING1                                               04292 49 04344 00000
15 380        DORG  *-4                                                 04299
16 390        TR    PLOC3-4,PLOC2-4,,SET 3RD LEV OPER BUFFERS           04300 31 12840 12791
15 430        SF    PFLAG&7,,,SET 3RD GEN PAT SWITCH                    04312 32 13269 00000
15 440        B     LOAD2&12,,,RELOAD 2ND LEV BUFFS                     04324 49 04140 00000
15 450        DORG  *-4                                                 04331
15 460 CKSIM  BD    LOAD2,COMP9&6,11,TEST FOR SIMPLE PAT                04332 43 04128 1141M
15 470 RING1  AM    COMP9&6,03,10                                       04344 11 11414 000-3
15 480        TF    PTCNT,COMP9&6                                       04356 26 12182 11414
15 490        A     PTCNT,COMP9&6,11                                    04368 21 12182 1141M
15 500        SM    PTCNT,08,10,SET LENGTH OF PAT                       04380 12 12182 000-8
15 505        TF    RTCNT,PTCNT,,SET PAT LIMIT FOR REVERSE OPERATION    04392 26 12187 12182
15 510        AM    COMP9&6,02,10                                       04404 11 11414 000-2
15 520        TF    OUTPUT-44,COMP9&6,11,SURF TO OUTPUT BUFF            04416 26 12919 1141M
15 530        BNF   FLAG7,PFLAG&8                                       04428 44 04472 13270
15 540        BD    *&20,SURF2-1                                        04440 43 04460 12803
15 545        B     FLAG7                                               04452 49 04472 00000
15 550        DORG  *-4                                                 04459
15 560        TF    OUTPUT-44,SURF2                                     04460 26 12919 12804
15 600 FLAG7  BNF   PRIOR,PFLAG&7                                       04472 44 04516 13269
15 620        BD    *&20,SURF3-1                                        04484 43 04504 12852
15 625        B     PRIOR                                               04496 49 04516 00000
15 630        DORG  *-4                                                 04503
15 640        TF    OUTPUT-44,SURF3                                     04504 26 12919 12853
15 650 PRIOR  BD    *&20,INPUT-45,,TEST FOR NEW SURF                    04516 43 04536 12027
15 655        B     TRA10                                               04528 49 04548 00000
15 660        DORG  *-4                                                 04535
```

```
PG LIN LABEL  OP    OPERANDS                                                  LOCN  OP INSTRUCTIONS

15 670        TF    OUTPUT-44,INPUT-44                                        04536 26 12919 12028
15 680 TRAIO  TFM   OUTPUT-46,10,10                                           04548 16 12917 000J0
15 690        TR    OUTPUT-43,AUXIN                                           04560 31 12920 12104
15 691        BTM   OPUT,,,OUTPUT DASH SURF                                   04572 17 11484 -0000
17 030 SETUP  AM    COMP9&6,01,10                                             04584 11 11414 000-1
17 032        TR    ST-20,COMP9&6,11,SAVE COORD TRANSFORM                     04596 31 12891 1141M
17 036        AM    COMP9&6,21,10                                             04608 11 11414 000K1
17 038 XMAIN  TR    OUTPUT-47,COMP9&6,11,FST MACHINE POINT TO OPUT BUFF       04620 31 12916 1141M
17 039        AM    COMP9&6,1,10                                              04632 11 11414 000-1
17 040        CM    OUTPUT-46,26,10,TEST FOR GOTO OR STOP                     04644 14 12917 000K6
17 042        BL    *&24                                                      04656 47 04680 01300
17 048        TR    OUTPUT-51,OUTPUT-47,,SHIFT REC IN OPUT BUFF               04668 31 12912 12916
15 900        BD    ER36,GNO,,TEST FOR 1ST GEN PAT                            04680 43 04744 13515
15 901        TFM   LAST1&54,*&20,,SET RETURN ADDRESS                         04692 16 06206 -4712
15 902        B     EXEC                                                      04704 49 05896 00000
15 903        DORG  *-4                                                       04711
15 904        SF    TRAN                                                      04712 32 08516 00000
15 904        SF    TRIGF                                                     04724 32 13267 00000
15 905        B     ANYMO                                                     04736 49 06894 00000
15 906        DORG  *-4                                                       04743
15 907 ER36   CM    INPUT-46,36,10                                            04744 14 12026 000L6
15 908        BNE   TRYG2                                                     04756 47 04780 01200
15 909 ERR2   BTM    ERSUB,7600,d,PAT OPER ERROR                              04768 17 11190 0P600
15 914 TRYG2  BD    G3,GNO-1                                                  04780 43 04856 13514
15 915        TFM   NXT11-2,*&20                                              04792 16 06734 -4812
15 916        B     EXEC2,,,SECOND GEN PAT                                    04804 49 06264 00000
15 917        DORG  *-4                                                       04811
15 918        TFM   OPER2,,10                                                 04812 16 12802 000-0
15 919        CF    DTRAN                                                     04824 33 09120 00000
15 920        SF    DI                                                        04836 32 09348 00000
15 923        B     ER36-32                                                   04848 49 04712 00000
15 924        DORG  *-4                                                       04855
15 925 G3     CM    OPER3,36,0110,3RD GEN PAT                                 04856 JM 12851 000L6
15 925        BE    ERR2                                                      04868 46 04768 01200
15 925        BNF   EXEC3,G3                                                  04880 44 05028 04856
15 926        CF    G3                                                        04892 33 04856 00000
15 927        CM    TRAN3-31,12,10,TEST TO SAVE 1ST PT OF PAT                 04904 14 12857 000J2
15 928        BL    LOC1                                                      04916 47 04960 01300
15 931        TF    FG3X,TRAN3-21                                             04928 26 13481 12867
15 932        TF    FG3Y,TRAN3-11                                             04940 26 13491 12877
15 993        B     EXEC3                                                     04952 49 05028 00000
15 934        DORG  *-4                                                       04959
15 936 LOC1   CM    TRAN2-31,12,10                                            04960 14 12808 000J2
15 938        BL    LOC3                                                      04972 47 05016 01300
15 940        TF    FG3X,TRAN2-21                                             04984 26 13481 12818
15 941        TF    FG3Y,TRAN2-11                                             04996 26 13491 12828
15 942        B     EXEC3                                                     05008 49 05028 00000
15 943        DORG  *-4                                                       05015
```

```
PG LIN LABEL  OP    OPERANDS                                                  LOCN  OP INSTRUCTIONS

15 944 LOC3   TR    FG3X-9,ST-20,,SAVE 1ST PT OF PAT                          05016 31 13472 12891
15 694 EXEC3  CM    TYPE-1,2,10,TEST FOR LINE MILL                            05028 14 12129 000-2
15 695        BNE   EX2&24                                                    05040 47 05164 01200
15 966        BD    EX2,INPUT-46,,MILL INSTR-CHECK FOR INVERT                 05052 43 05140 12026
15 967        BD    EX2-20,OPER3,,TEST FOR 3RD GEN INVERT                     05064 43 05120 12851
15 968        CM    OPER2,36,10,,TEST FOR 2ND GEN INVERT                      05076 14 12802 000L6
15 969        BNL   LOC7                                                      05088 46 05208 01300
15 970        TFM   EX2&23,OPER2                                              05100 16 05163 J2802
15 971        B     *&40                                                      05112 49 05152 00000
15 972        DORG  *-4                                                       05119
15 973        TFM   EX2&23,OPER3                                              05120 16 05163 J2851
15 974        B     *&20                                                      05132 49 05152 00000
15 975        DORG  *-4                                                       05139
15 975 EX2    TFM   *&23,INPUT-46,,MILL INVERSION                             05140 16 05163 J2026
15 975        BTM   MILLR                                                     05152 17 10236 -0000
15 975        CM    OPER2,34,10,TEST FOR 2ND GEN INVERSION                    05164 14 12802 000L4
15 975        BH    LOC7                                                      05176 46 05208 01100
15 975        BT    INVRT,OPER2                                               05188 27 07544 12802
15 976        B     LOC6                                                      05200 49 05252 00000
15 976        DORG  *-4                                                       05207
15 977 LOC7   CM    OPER2,36,10,TEST FOR 2ND GEN REVERSAL                     05208 14 12802 000L6
15 978        BNE   LOC6                                                      05220 47 05252 01200
15 979        BTM   REVRS,36,10                                               05232 17 08184 000L6
15 980        B     LOC8                                                      05244 49 05376 00000
15 981        DORG  *-4                                                       05251
15 982 LOC6   CM    TRAN2-31,11,10,TEST FOR 2ND GEN ROTAT W/TRANS             05252 14 12808 000J3
15 983        BNE   LOC9                                                      05264 47 05320 01200
15 984        BNF   LOC10,TRIGF,,SWITCH FOR ANGLE CONVERSION                  05276 44 05300 13267
15 985        BT    SICO,TRAN2-1,,COMPUTE SINE AND COSINE OF ROTAT ANGLE      05288 27 10096 12838
15 990 LOC10  BT    ROTD,SICO-1,,ROTATE THIS POINT                            05300 27 07682 10095
15 991        B     LOC8                                                      05312 49 05376 00000
15 992        DORG  *-4                                                       05319
15 993 LOC9   CM    TRAN2-31,11,10,TEST FOR 2ND GEN ROT W/O TRANS             05320 14 12808 000J1
15 994        BNE   LOC8                                                      05332 47 05376 01200
15 995        BNF   LOC10,TRIGF,,HAS ANGLE BEEN CONVERTED                     05344 44 05300 13267
15 996        BT    SICO,TRAN2-21                                             05356 27 10096 12818
15 998        B     LOC10                                                     05368 49 05300 00000
15 999        DORG  *-4                                                       05375
16 100 LOC8   CM    OPER3,40,10,TEST FOR 2ND GEN DIFFFRENCE TRANS             05376 14 12851 000M0
16 101        BNE   *&36                                                      05388 47 05424 01200
16 102        CM    TRAN3-31,10,10                                            05400 14 12857 000J0
16 103        BE    LOC11                                                     05412 46 05596 01200
16 104        BTM   DTRAN,,10,COMPUTE 2ND GEN DIFF TRANS                      05424 17 09120 000-0
16 105        CM    OPER3,34,10,TEST FOR 3RD GEN INVERSION                    05436 14 12851 000L4
16 107        BH    LOC13                                                     05448 46 05472 01100
16 109        BT    INVRT,OPER3                                               05460 27 07544 12851
16 115 LOC13  CM    TRAN3-31,13,10,TEST FOR 3RD GEN ROTATION                  05472 14 12857 000J3
16 116        BNE   LOC14                                                     05484 47 05540 01200
```

```
PG LIN LABEL  OP    OPERANDS                                              LOCN  OP INSTRUCTIONS

16 117        BNF   LOC15,TRIGF,,ANGLE TEST                               05496 44 05520 13267
16 118        BT    SICO,TRAN3-1                                          05508 27 10096 12087
16 123 LOC15  BT    ROTO,SICO-1                                           05520 27 07682 10095
16 124        B     LOC11                                                 05532 49 05596 00000
16 125        DORG  *-4                                                   05539
16 126 LOC14  C4    TRAN3-11,11,10,TEST FOR 3RD GEN ROT W/O TRANS         05540 14 12857 000J1
16 127        BNE   LOC11                                                 05552 47 05596 01200
16 128        BNF   LOC15,TRIGF                                           05564 44 05520 13267
16 128        BT    SICO,TRAN3-21                                         05576 27 10096 12867
16 130        B     LOC15                                                 05588 49 05520 00000
16 131        DORG  *-4                                                   05595
16 132 LOC11  CM    INPUT-46,40,10,TEST FOR 3RD GEN DIFF TRANS            05596 14 12026 000M0
16 133        BNE   *&36                                                  05608 47 05644 01200
16 134        CM    INPUT-40,10,10                                        05620 14 12032 000J0
16 135        BE    LOC18                                                 05632 46 05816 01200
16 136        BTM   DTRAN2,,10,COMPUTE 3RD GEN DIFF TRANS                 05644 17 09614 000-0
16 137        CM    INPUT-46,34,10,TEST FOR INVERSION                     05656 14 12026 000L4
16 139        BH    LOC16                                                 05668 46 05692 01100
16 141        BT    INVRT,INPUT-46                                        05680 27 07544 12026
16 145 LOC16  CM    INPUT-40,13,10,TEST FOR ROT W/TRANS                   05692 14 12032 000J3
16 146        BNE   LOC20                                                 05704 47 05760 01200
16 147        BNF   LOC19,TRIGF                                           05716 44 05740 13267
16 148        BT    SICO,INPUT-10                                         05728 27 10096 12062
16 153 LOC19  BT    ROTO,SICO-1                                           05740 27 07682 10095
16 154        B     LOC18                                                 05752 49 05816 00000
16 155        DORG  *-4                                                   05759
16 156 LOC20  CM    INPUT-40,11,10,TEST FOR ROT W/O TRANS                 05760 14 12032 000J1
16 157        BNE   LOC18                                                 05772 47 05816 01200
16 158        BNF   LOC19,TRIGF                                           05784 44 05740 13267
16 159        BT    SICO,INPUT-30                                         05796 27 10096 12042
16 161        B     LOC19                                                 05808 49 05740 00000
16 162        DORG  *-4                                                   05815
16 193 LOC18  BTM   TRAN,,10,FINAL TRANSFORMATION                         05816 17 08516 000-0
16 165        BTM   OPUT,,,OUTPUT THIS POINT                              05828 17 11484 -0000
16 164        C     COMP9&6,RTCNT                                         05840 24 11414 12187
16 165        BL    XMAIN                                                 05852 47 04620 01300
16 166        SF    DI,,,RESET SWITCHES ON LAST PT OF PAT                 05864 32 09348 00000
16 167        CF    DTRAN                                                 05876 33 09120 00000
16 168        B     ER36-32                                               05888 49 04712 00000
16 171        DORG  *-4                                                   05895
17 201 EXEC   CM    TYPE-1,2,10,TEST FOR LINE MILL                        05896 14 12129 000-2
17 202        BNE   *&48                                                  05908 47 05956 01200
17 203        CM    INPUT-46,34,10,TEST FOR MILL INVERSION                05920 14 12026 000L4
17 204        BH    TRYRV                                                 05932 46 06000 01100
17 204        BTM   MILLR,INPUT-46                                        05944 17 10236 J2026
17 204        CM    INPUT-46,34,10,INVERSION TEST                         05956 14 12026 000L4
17 204        BH    *&32                                                  05968 46 06000 01100
17 204        BT    INVRT,INPUT-46                                        05980 27 07544 12026
```

```
PG LIN LABEL  OP    OPERANDS                                              LOCN  OP INSTRUCTIONS

17 205        B     TRYRO                                                 05992 49 06092 00000
17 206        DORG  *-4                                                   05999
17 208 TRYRV  CM    INPUT-46,36,10,TEST FOR REVERSAL                      06000 14 12026 000L6
17 209        BNE   TRYRO                                                 06012 47 06092 01200
17 210        BT    REVRS,INPUT-46,,36                                    06024 27 08184 12026
17 212        BTM   OPUT,,,OUTPUT THIS POINT                              06036 17 11484 -0000
17 212        TFM   OUTPUT-50,,10,RESET GOTO OR STOP SWITCH               06048 16 12913 000-0
17 213        C     COMP9&6,PTCNT,,TEST FOR LAST POINT                    06060 24 11414 12182
17 214        BH    ER36-32                                               06072 46 04712 01100
17 215        B     TRYRV&24,,,GET NEXT POINT                             06084 49 06024 00000
10 216        DORG  *-4                                                   06091
17 217 TRYRO  CM    INPUT-40,13,10, TEST FOR ROT W/TRANS                  06092 14 12032 000J3
17 218        BNE   TRYR1                                                 06104 47 06208 01200
17 219        BNF   LAST1-12,TRIGF,,ANGLE CHECK                           06116 44 06140 13267
17 220        BT    SICO,INPUT-10                                         06128 27 10096 12062
17 225        BT    ROTO,SICO-1                                           06140 27 07682 10095
17 226 LAST1  BT    TRAN,INPUT-40,,FINAL TRANS                            06152 27 08516 12032
17 227        BTM   OPUT,,,OUTPUT THIS POINT                              06164 17 11484 -0000
17 229        C     COMP9&6,RTCNT,,TEST FOR LAST PT THIS PAT              06176 24 11414 12187
17 230        BL    XMAIN,,,NEXT POINT                                    06188 47 04620 01300
17 231        B     ,,,RETURN TO MAIN ROUTINE                             06200 49 00000 00000
17 232        DORG  *-4                                                   06207
17 233 TRYR1  CM    INPUT-40,11,10,TEST ROT W/O TRANS                     06208 14 12032 000J1
17 234        BNE   LAST1                                                 06220 47 06152 01200
17 235        BNF   LAST1-12,TRIGF                                        06232 44 06140 13267
17 236        BT    SICO,INPUT-30                                         06244 27 10096 12042
17 238        B     LAST1-12                                              06256 49 06140 00000
17 239        DORG  *-4                                                   06263
17 240 EXEC2  CM    TYPE-1,2,10,TEST FOR LINE MILL                        06264 14 12129 000-2
17 241        BNE   E2                                                    06276 47 06376 01200
17 242        BD    E3,INPUT-46,,TEST FOR MILL INVERSION                  06288 43 06356 12026
17 243        CM    OPER2,34,10,TEST FOR 2ND GEN MILL INVERT              06300 14 12802 000L4
17 244        BH    NOW36                                                 06312 46 06420 01100
17 245        TFM   *&23,OPER2                                            06324 16 06347 J2802
17 246 E4     BTM   MILLR,OPER2                                           06336 17 10236 J2802
17 247        B     *&28                                                  06348 49 06376 00000
17 248        DORG  *-4                                                   06355
17 249 E3     TFM   E4&11,INPUT-46                                        06356 16 06347 J2026
17 250        B     E4                                                    06368 49 06336 00000
17 251        DORG  *-4                                                   06375
17 251 E2     CM    OPER2,34,10,TEST FOR 2ND GEN INVERT                   06376 14 12802 000L4
17 251        BH    *&32                                                  06388 46 06420 01100
17 251        BT    INVRT,OPER2                                           06400 27 07544 12802
17 251        B     NOW13                                                 06412 49 06464 00000
17 251        DORG  *-4                                                   06419
17 251 NOW36  CM    OPER2,36,10,TEST INVERSION                            06420 14 12802 000L6
17 251        BNE   NOW13                                                 06432 47 06464 01200
17 251        BTM   REVRS,,10                                             06444 17 08184 000-0
```

```
PG LIN LABEL  OP    OPERANDS                                                  LOCN  OP INSTRUCTIONS

17 252        B     NOT1&12                                                   06456 49 06524 00000
17 254        DORG  *-4                                                       06463
17 853 NOW13  CM    TRAN2-31,13,10,TEST FOR 2ND GEN ROTAT W/TRANS             06464 14 12808 000J3
17 854        BNE   NOW11                                                     06476 47 06792 01200
17 855        BNF   NOT1,TRIGF                                                06488 44 06512 13267
17 856        BT    SICO,TRAN2-1                                              06500 27 10096 12838
17 861 NOT1   BT    ROTO,SICO-1                                               06512 27 07682 10095
17 861        CM    INPUT-46,40,10,TEST FOR 2ND LEV DIFF TRAN                 06524 14 12026 000M0
17 861        BNE   *&36                                                      06536 47 06572 01200
17 861        CM    INPUT-40,10,10                                            06548 14 12032 000J0
17 861        BE    NOT2&12                                                   06560 46 06680 01200
17 862        BTM   DTRAN,,10                                                 06572 17 09120 000-0
17 863        CM    INPUT-46,34,10,TEST FOR INVERT                            06584 14 12026 000L4
17 864        BH    NXT13                                                     06596 46 06620 01100
17 865        BT    INVRT,INPUT-46                                            06608 27 07544 12026
17 866 NXT13  CM    INPUT-40,13,10,TEST FOR ROT W/TRANS                       06620 14 12032 000J3
17 867        BNF   NXT11                                                     06632 47 06736 01200
17 868        BNF   NOT2,TRIGF                                                06644 44 06668 13267
17 869        BT    SICO,INPUT-10                                             06656 27 10096 12062
17 874 NOT2   BT    ROTO,SICO-1                                               06668 27 07682 10095
17 875        BTM   TRAN,,10,FINAL TRANS                                      06680 17 08516 000-0
17 876        BTM   OPUT,,,OUTPUT THIS POINT                                  06692 17 11484 -0000
17 876        C     COMP9&6,RTCNT,,TEST FOR LAST PT OF PAT                    06704 24 11414 12187
17 877        BL    XMAIN                                                     06716 47 04620 01300
17 878                                                                        06728 49 06728 00000
17 879        DORG  *-4                                                       06735
17 880 NXT11  CM    INPUT-40,11,10,TEST FOR ROT W/O TRANS                     06736 14 12032 000J1
17 881        BNE   NOT2&12                                                   06748 47 06680 01200
17 882        BNF   NOT2,TRIGF                                                06760 44 06668 13267
17 883        BT    SICO,INPUT-30                                             06772 27 10096 12042
17 885        B     NOT2                                                      06784 49 06668 00000
17 886        DORG  *-4                                                       06791
17 891 NOW11  CM    TRAN2-31,11,10,TEST FOR 2ND GEN ROT W/OTRANS              06792 14 12808 000J1
17 892        BNE   NOT1&12                                                   06804 47 06524 01200
17 893        BNF   NOT1,TRIGF                                                06816 44 06512 13267
17 893        BT    SICO,TRAN2-21                                             06828 27 10096 12818
17 895        B     NOT1                                                      06840 49 06512 00000.
17 090        DORG  *-2                                                       06849
17 217 INIT   RCTY                                                            06850 34 00000 00102
17 217        TFM   OUTPUT-50,00,10                                           06862 16 12913 000-0
17 217        TR    TABPT-4,TBLE1-4,,REINITIALIZE PHASE2                      06874 31 13221 13279
17 217        B     NEXT                                                      06886 49 01700 00000
17 217        DORG  *-4                                                       06893
15 715 ANYMO  BNF   NOBFL,PFLAG&8,,TEST FOR 2ND GEN PAT                       06894 44 07522 13270
15 720        C     PLOC2,PCNT2,,TEST FOR LAST SUBPATTERN                     06906 24 12795 12800
15 730        BH    OFF2                                                      06918 46 07278 01100
15 733        AM    PLOC2,02,10,INDEX POINTER FOR 2ND GEN PAT                 06930 11 12795 000-2
15 734 TRYGO  CM    PLOC2,26,610,TEST FOR GOTO                                06942 14 1279N 000K6
```

```
PG LIN LABEL  OP    OPERANDS                                                  LOCN  OP INSTRUCTIONS

15 734        BE    *&36                                                      06954 46 06990 01200
15 734        CM    PLOC2,39,10,TEST FOR STOP                                 06966 14 12795 000L9
15 734        BNE   NOGO                                                      06978 47 07210 01200
15 734        SM    PLOC2,01,10                                               06990 12 12795 000-1
15 734        TR    OUTPUT-47,PLOC2,11                                        07002 31 12916 1279N
15 734        AM    PLOC2,1,10                                                07014 11 12795 000-1
15 734        FS    OUTPUT-32,FG2X                                            07026 16 00469 -7061
15 734                                                                        07038 16 00445 J2931
15 734                                                                        07050 49 00402 J3439
15 734        FS    OUTPUT-22,FG2Y                                            07062 16 00469 -7097
15 734                                                                        07074 16 00445 J2941
15 734                                                                        07086 49 00402 J3449
15 734        TR    ST-20,FG2X-9                                              07098 31 12891 13430
15 734        TFM   PTCNT,,, THIS PAT INCLUDES A GOTO OR A STOP               07110 16 12182 -0000
15 734        TF    COMP9&6,PLOC2                                             07122 26 11414 12795
15 734        TFM   LAST1&54,*&20,,OPERATE ON A GOTO OR STOP                  07134 16 06206 -7154
15 724        B     EXEC                                                      07146 49 05896 00000
15 724        DORG  *-4                                                       07153
15 734        SF    TRIGF                                                     07154 32 13267 00000
15 734        SF    TRAN                                                      07166 32 08516 00000
15 734        AM    PLOC2,1,10                                                07178 11 12795 000-1
15 734        BNR   *-12,PLOC2,11,INDEX 2ND GEN POINTER                       07190 45 07178 1279N
15 734        B     TRYGO-12                                                  07202 49 06930 00000
15 734        DORG  *-4                                                       07209
15 736 NOGO   SM    PLOC2,1,10                                                07210 12 12795 000-1
15 738        TR    OPER2-1,PLOC2,11,RELOAD 2ND GEN BUFFERS                   07222 31 12801 1279N
15 740        TF    PATNO,SURF2&2                                             07234 26 12890 12806
15 742        AM    PLOC2,1,10                                                07246 11 12795 000-1
15 744        BNR   *-12,PLOC2,11                                             07258 45 07246 1279N
15 850        B     PDUMP&12,,OUTPUT THIS PAT                                 07270 49 04012 00000
15 860        DORG  *-4                                                       07277
15 900 OFF2   CF    PFLAG&8,,END OF 2ND GEN PAT                               07278 33 13270 00000
15 905        SF    DTRAN,,RESET 2ND GEN CONTROL SWITCHES                     07290 32 09120 00000
15 907        SF    D1&1                                                      07302 32 09349 00000
15 908        CF    D4                                                        07314 33 09752 00000
15 909        SF    D4&1                                                      07326 32 09753 00000
15 908        TFM   FG2X-7,500,9                                              07338 16 13432 00N00
15 909        TFM   FG2Y-7,500,9                                              07350 16 13442 00N00
15 915        BNF   LEAVE,PFLAG&7,,TEST FOR 3RD GEN PAT                       07362 44 07514 13269
15 920        C     PLOC3,PCNT3,,TEST FOR LAST SUBPATTERN                     07374 24 12844 12849
15 930        BH    OFF3                                                      07386 46 07466 01100
15 932        AM    PLOC3,1,10,INDEX 3RD GEN POINTER                          07398 11 12844 000-1
15 935        TR    OPER3-1,PLOC3,11,RELOAD 3RD GEN OPERATION BUFFERS         07410 31 12850 1284M
15 937        TF    PATNO,SURF3&2                                             07422 26 12890 12855
15 939        AM    PLOC3,1,10                                                07434 11 12844 000-1
15 941        BNR   *-12,PLOC3,11,INDEX 3RD GEN POINTER                       07446 45 07434 1284M
16 060        B     PDUMP&12,,OUTPUT NEXT PATTERN                             07458 49 04012 00000
16 070        DORG  *-4                                                       07465
```

```
PG LIN LABEL  OP    OPERANDS                                                LOCN  OP INSTRUCTIONS

16 080 OFF3   CF    PFLAG67,,,RESET 3RD GEN SWITCHES                        07466 33 13269 00000
16 081        SF    G3                                                      07478 32 04856 00000
16 084        SF    DTRAN2                                                  07490 32 09614 00000
16 086        SF    D4                                                      07502 32 09752 00000
16 090 LEAVE  B     NEXT,,,READ NEXT INPUT RECORD                           07514 49 01700 00000
16 100        DORG  *-4                                                     07521
16 110 NOBFL  BNF   LEAVE,PFLAG67,,TEST FOR END OF ENTIRE PAT               07522 44 07514 13269
16 020        B     OFF2&12,,,OUTPUT NEXT PAT                               07534 49 07290 00000
16 130        DORG  *-2                                                     07543
17 300              *           INVERSION SUBROUTINE
17 302 INVRT  CM    OUTPUT-46,11,10,TEST FOR Z COORD                        07544 14 12917 000J1
17 303        BE    PTOMX&12                                                07556 46 07630 01200
17 310        CM    INVRT-1,32,10                                           07568 14 07543 000L2
17 311        BNF   FLIPY                                                   07580 47 07632 01200
17 312        BNF   PTOMX,OUTPUT-36,,TEST FOR INVERT RIGHT OR LEFT          07592 44 07618 12927
17 320        CF    OUTPUT-36,,,MINUS TO PLUS                               07604 33 12927 00000
17 322        BB                                                            07616 42 00000 00000
17 324        DORG  *-10                                                    07617
17 326 PTOMX  SF    OUTPUT-36,,,PLUS TO MINUS                               07618 32 12927 00000
17 328        BB                                                            07630 42 00000 00000
17 330        DORG  *-10                                                    07631
17 332 FLIPY  BNF   PTOMY,OUTPUT-26,,TEST FOR INVERT UP OR DOWN             07632 44 07658 12937
17 334        CF    OUTPUT-26                                               07644 33 12937 00000
17 336        BB                                                            07656 42 00000 00000
17 338        DORG  *-10                                                    07657
17 340 PTOMY  SF    OUTPUT-26                                               07658 32 12937 00000
17 342        BB                                                            07670 42 00000 00000
17 400              *           ROTATION SUBROUTINE
17 402 ROTO   CM    OPER2,36,10,TEST FOR REVERSAL                           07682 14 12802 000L6
17 402        BNE   R4                                                      07694 47 07776 01200
17 402        AM    PTCNT,03,10                                             07706 11 12182 000-3
17 402        CM    PTCNT,11,610                                            07718 14 1218K 000J1
17 402        BE    *&32                                                    07730 46 07762 01200
17 402        SM    PTCNT,03,10                                             07742 12 12182 000-3
17 402        B     R4&24                                                   07754 49 07800 00000
17 402        DORG  *-4                                                     07761
17 402        SM    PTCNT,03,10                                             07762 12 12182 000-3
17 402        BB                                                            07774 42 00000 00000
17 402        DORG  *-10                                                    07775
17 402 R4     CM    OUTPUT-46,11,10,TEST FOR Z COORD ONLY                   07776 14 12917 000J1
17 403        BE    REVRS-4                                                 07788 46 08180 01200
17 414        FM    OUTPUT-36,THCOS                                         07800 16 00469 -7835
17 414                                                                      07812 26 01260 12927
17 414                                                                      07824 49 01262 J3419
17 416        TF    BUFFN,99,,   X.COS&THETA¤  TO BUFFER                    07836 26 12164 00099
17 418        FM    OUTPUT-26,THSIN                                         07848 16 00469 -7883
17 418                                                                      07860 26 01260 12937
17 418                                                                      07872 49 01262 J3429
```

```
PG LIN LABEL  OP    OPERANDS                                                LOCN  OP INSTRUCTIONS

17 420        FS    BUFFN,99                                                07884 16 00469 -7919
17 420                                                                      07896 16 00445 J2164
17 420                                                                      07908 49 00402 -0099
17 422        FM    OUTPUT-26,THCOS                                         07920 16 00469 -7955
17 422                                                                      07932 26 01260 12937
17 422                                                                      07944 49 01262 J3419
17 424        TF    BUFFD,99,,   Y.COS%THETA¤                               07956 26 12174 00099
17 426        FM    OUTPUT-36,THSIN                                         07968 16 00469 -8003
17 426                                                                      07980 26 01260 12927
17 426                                                                      07992 49 01262 J3429
17 428        FA    BUFFD,99                                                08004 16 00469 -8039
17 428                                                                      08016 16 00445 J2174
17 428                                                                      08028 49 00422 -0099
17 430        TF    OUTPUT-36,BUFFN,,   X.COS%THETA¤-Y.SIN%THETA¤           08040 26 12927 12164
17 432        TF    OUTPUT-26,BUFFD,,   Y.COS%THETA¤&X.SIN%THETA¤           08052 26 12937 12174
17 434        CM    OUTPUT-46,16,10,TEST FOR A SLOPE                        08064 14 12917 000J6
17 436        BNE   *&62                                                    08076 47 08138 01200
17 438        TFM   *&23,OUTPUT-16,,ALGEBRAICALLY SUM ANGLES                08088 16 08111 J2947
17 440        FA    OUTPUT-16,ROTO-1                                        08100 16 00469 -8135
17 440                                                                      08112 16 00445 J2947
17 440                                                                      08124 49 00422 -7681
17 442        BB                                                            08136 42 00000 00000
17 444        DORG  *-10                                                    08137
17 446        CM    OUTPUT-46,14,10,TEST FOR AN ARC OR A SLOPE              08138 14 12917 000J4
17 448        BL    *-14                                                    08150 47 08136 01300
17 450        TFM   *-39,OUTPUT-6                                           08162 16 08123 J2957
17 452        B     *-74                                                    08174 49 08100 00000
17 454        DORG  *-2                                                     08183
17 500              *REVERSE SUBROUTINE
17 504 REVRS  SM    PTCNT,1,10                                              08184 12 12182 000-1
17 505        BNR   *-12,PTCNT,11,INDEX POINTER BACKWARDS                   08196 45 08184 1218K
17 506        AM    PTCNT,2,10                                              08208 11 12182 000-2
17 507        CM    PTCNT,26,610,TEST FOR A GOTO OR A STOP                  08220 14 1218K 000K6
17 508        BL    ND39                                                    08232 47 08482 01300
17 509        BD    *&32,OUTPUT-50                                          08244 43 08276 12913
17 509        SF    NOREV                                                   08256 32 08688 00000
17 509        B     *&20                                                    08268 49 08288 00000
17 509        DORG  *-4                                                     08275
17 509        SF    NOREV&1                                                 08276 32 08689 00000
17 509        SM    PTCNT,1,10                                              08288 12 12182 000-1
17 510        TR    OUTPUT-51,PTCNT,11,SHIFT  4 LOCATIONS LEFT              08300 31 12912 1218K
17 511        BD    DEIN2,GNO,,TEST FOR FIRST GEN PAT                       08312 43 08468 13515
17 512 GOST   CM    OUTPUT-46,11,10,TEST FOR Z COORD                        08324 14 12917 000J1
17 513        BE    DEIN2                                                   08336 46 08468 01200
17 514        FA    OUTPUT-36,ST-11                                         08348 16 00469 -8383
17 514                                                                      08360 16 00445 J2927
17 514                                                                      08372 49 00422 J2900
17 515        FA    OUTPUT-26,ST-1                                          08384 16 00469 -8419
```

```
PG LIN LABEL  OP    OPERANDS                                              LOCN  OP INSTRUCTIONS

17 515                                                                    08396 16 00445 J2937
17 515                                                                    08408 49 00422 J2910
17 516        CM    OUTPUT-50,,10                                         08420 14 12913 000-0
17 517        BE    DEIN2                                                 08432 46 08468 01200
17 518        TR    48,OUTPUT-51,,SHIFT 4 LOCATIONS RIGHT                 08444 31 00048 12912
17 519        TR    OUTPUT-47,48                                          08456 31 12916 00048
17 520 DEIN2  SM    PTCNT,2,10,INDEX POINTER                              08468 12 12182 000-2
17 521        BB                                                          08480 42 00000 00000
17 522        DORG  *-10                                                  08481
17 523 NO39   SM    PTCNT,1,10                                            08482 12 12182 000-1
17 524        TR    OUTPUT-47,PTCNT,11,THIS POINT TO OUTPUT BUFFER        08494 31 12916 1218K
17 525        B     GOST-12                                               08506 49 08312 00000
17 556        DORG  *-2                                                   08515
17 598              *     SUBROUTINE TRAN  PERFORMS FINAL TRANSFORMATION
17 600 TRAN   BNF   TRAN1,TRAN,0                                          08516 M4 08588 08516
17 601        CF    TRAN,,,FIRST TIME SWITCH                              08528 33 08516 00000
17 602        CM    INPUT-40,12,10,TEST FOR NEW FIRST POINT               08540 14 12032 000J2
17 603        BL    TRA3                                                  08552 47 08958 01300
17 612        TF    ST-11,INPUT-30,,SAVE FIRST POINT                      08564 26 12900 12042
17 614        TF    ST-1,INPUT-20                                         08576 26 12910 12052
17 604 TRAN1  CM    OPER2,36,10,TEST FOR REVERSAL                         08588 14 12802 000L6
17 604        BNE   NOREV                                                 08600 47 08688 01200
17 604        AM    PTCNT,03,10                                           08612 11 12182 000-3
17 604        CM    PTCNT,11,610                                          08624 14 1218K 000J1
17 604        BNE   *&32                                                  08636 47 08668 01200
17 604        SM    PTCNT,03,10                                           08648 12 12182 000-3
17 604        B     POINT                                                 08660 49 08784 00000
17 604        DORG  *-4                                                   08667
17 604        SM    PTCNT,03,10                                           08668 12 12182 000-3
17 604        B     NOREV&24                                              08680 49 08712 00000
17 604        DORG  *-4                                                   08687
17 605 NOREV  CM    OUTPUT-46,11,10,TEST FOR Z COORD                      08688 14 12917 000J1
17 608        BE    POINT                                                 08700 46 08784 01200
17 617        FA    OUTPUT-36,ST-11                                       08712 16 00469 -8747
17 617                                                                    08724 16 00445 J2927
17 617                                                                    08736 49 00422 J2900
17 618        FA    OUTPUT-26,ST-1                                        08748 16 00469 -8783
17 618                                                                    08760 16 00445 J2937
17 618                                                                    08772 49 00422 J2910
17 622 POINT  BNF   *&20,NOREV                                            08784 44 08804 08688
17 623        B     *&44                                                  08796 49 08840 00000
17 623        DORG  *-4                                                   08803
17 623        CM    OUTPUT-50,,10,TEST FOR GOTO OR STOP                   08804 14 12913 000-0
17 624        BE    TRAN4                                                 08816 46 08884 01200
17 625        AM    COMP966,04,10                                         08828 11 11414 000-4
17 625        BNF   *&20,NOREV&1                                          08840 44 08860 08689
17 625        B     TRAN4                                                 08852 49 08884 00000
17 625        DORG  *-4                                                   08859
```

```
PG LIN LABEL  OP    OPERANDS                                              LOCN  OP INSTRUCTIONS

17 626        TR    48,OUTPUT-51                                          08860 31 00048 12912
17 628        TR    OUTPUT-47,48,,RIGHT SHIFT 4 LOCATIONS                 08872 31 12916 00048
17 639 TRAN4  AM    COMP966,1,10                                          08884 11 11414 000-1
17 640        BNR   *-12,COMP966,11,INDEX FIRST GEN POINTER               08896 45 08894 1141M
17 941        AM    COMP966,1,10                                          08908 11 11414 000-1
17 630        TFM   OUTPUT-50,,10,RESET SPECIAL SWITCHES                  08920 16 12913 000-0
17 632        CF    NOREV                                                 08932 33 08688 00000
17 634        CF    NOREV&1                                               08944 33 08689 00000
17 664        BB                                                          08956 42 00000 00000
17 645        DORG  *-10                                                  08957
17 648 TRA3   BNF   TRA8,PFLAG&7,,TEST FOR 3RD GEN PATTERN                08958 44 09098 13269
17 649        BNF   TRA7,DTRAN2,, ARE 2ND GEN PATTERNS LINKED TOGETHER    08970 44 09078 09614
17 649 TR10   BNF   *&20,D1&1,,ARE 1ST GEN PATS LINKED                    08982 44 09002 09349
17 650        B     TRA6                                                  08994 49 09022 00000
17 650        DORG  *-4                                                   09001
17 650        TR    ST-20,FG2X-9,,SAVE 1ST POINT OF 2ND GEN PAT           09002 31 12891 13430
17 651        B     TRAN1                                                 09014 49 08588 00000
17 652        DORG  *-4                                                   09021
17 653 TRA6   CM    TRAN2-31,12,10,TEST FOR NEW 1ST PT OF THIS 1ST GEN PAT 09022 14 12808 000J2
17 654        BL    TRAN1                                                 09034 47 08588 01300
17 657        TF    ST-11,TRAN2-21,,SAVE FIRST POINT                      09046 26 12900 12818
17 658        TF    ST-1,TRAN2-11                                         09058 26 12910 12828
17 659        B     TRAN1                                                 09070 49 08588 00000
17 650        DORG  *-4                                                   09077
17 651 TRA7   TR    ST-20,FG3X-9,,SAVE FIRST PT OF 3RD GEN PAT            09078 31 12891 13472
17 663        B     TRAN1                                                 09090 49 08588 00000
17 664        DORG  *-4                                                   09097
17 665 TRA8   BNF   TRAN1,PFLAG&8,,TEST FOR 2ND GEN PAT                   09098 44 08588 13270
17 666        B     TR10                                                  09110 49 08982 00000
17 667        DORG  *-2                                                   09119
17 900              *  DTRAN SUBROUTINE COMPUTES DIFF TRANSFORMATION WHICH LINKS 1ST GEN O
17 904 DTRAN  BNF   D1,DTRAN,0,1ST TIME SWITCH                            09120 M4 09348 09120
17 905        BNF   OUT1,D1&1                                             09132 44 09290 09349
17 906        CF    D1&1                                                  09144 33 09349 00000
17 907        CM    TRAN2-31,12,10,TEST FOR NEW 1ST FOR 1ST GEN PAT       09156 14 12808 000J2
17 908        BNL   OUT1&2                                                09168 46 09292 01300
17 912        TR    FG2X-9,ST-20                                          09180 31 13430 12891
17 912 TRB1   BNF   OUT1,PFLAG&7,,TEST FOR 3RD GEN PAT                    09192 44 09290 13269
17 912        CM    INPUT-40,10,10,TEST FOR PAT OPERATION                 09204 14 12032 000J0
17 912        BNE   OUT1                                                  09216 47 09290 01200
17 912        CM    TRAN3-31,12,10,NO NEW 1ST PT FOR ENTIRE PAT           09228 14 12857 000J2
17 912        BL    SHIFT                                                 09240 47 09278 01300
17 912        TF    FG3X,TRAN3-21                                         09252 26 13481 12867
17 912        TF    FG3Y,TRAN3-11                                         09264 26 13491 12877
17 912        BB                                                          09276 42 00000 00000
17 912        DORG  *-10                                                  09277
17 912 SHIFT  TR    FG3X-9,FG2X-9,,RESET 1ST PT FOR PAT                   09278 31 13472 13430
17 913 OUT1   BB                                                          09290 42 00000 00000
```

```
PG LIN LABEL  OP    OPERANDS                                                    LOCN  OP INSTRUCTIONS

17 914        DORG *-10                                                         09291
17 915        TF    FG2X,TRAN2-21,,RESET 1ST PT                                 09292 26 13439 12818
17 916        TF    FG2Y,TRAN2-11                                               09304 26 13449 12828
17 918        TF    ST-11,TRAN2-21                                              09316 26 12900 12818
17 918        TF    ST-1,TRAN2-11                                               09328 26 12910 12828
17 919        B     TRB1                                                        09340 49 09192 00000
17 920        DORG *-4                                                          09347
17 921 D1     BNF   FAD,D1,01, 1ST TIME SWITCH                                  09348 MM 09480 09348
17 922        CF    D1                                                          09360 33 09348 00000
17 923        CM    TRAN2-31,12,10,TEST FOR NEW 1ST POINT                       09372 14 12808 000J2
17 924        BNL   NEWT                                                        09384 46 09580 01300
17 927        TR    NXG2X-9,ST-20,,SAVE  1ST PT OF NEXT 1ST GEN PAT             09396 31 13451 12891
17 929        FS    NXG2X,FG2X                                                  09408 16 00469 -9443
17 929                                                                          09420 16 00445 J3460
17 929                                                                          09432 49 00402 J3439
17 930        FS    NXG2Y,FG2Y                                                  09444 16 00469 -9479
17 930                                                                          09456 16 00445 J3470
17 930                                                                          09468 49 00402 J3449
17 939 FAD    CM    OUTPUT-46,11,10,TEST FOR Z COORD                            09480 14 12917 000J1
17 939        BNE   *&14                                                        09492 47 09506 01200
17 939        BB                                                                09504 42 00000 00000
17 939        DORG *-10                                                         09505
17 939        FA    OUTPUT-36,NXG2X                                             09506 16 00469 -9541
17 939                                                                          09518 16 00445 J2927
17 939                                                                          09530 49 00422 J3460
17 940        FA    OUTPUT-26,NXG2Y                                             09542 16 00469 -9577
17 940                                                                          09554 16 00445 J2937
17 940                                                                          09566 49 00422 J3470
17 941        BB                                                                09578 42 00000 00000
17 942        DORG *-10                                                         09579
17 943 NEWT   TF    NXG2X,TRAN2-21,,SAVE NEW TRANSFORM                          09580 26 13460 12818
17 944        TF    NXG2Y,TRAN2-11                                              09592 26 13470 12828
17 945        B     FAD-72                                                      09604 49 09408 00000
17 946        DORG *-2                                                          09613
17 948              *  DTRAN2 SUBROUTINE COMPUTES DIFF TRAN BETWEEN 2ND GEN PATS
17 950 DTRAN2 BNF   D4,DTRAN2,0, 1ST TIME SWITCH                                09614 M4 09752 09614
17 951        CF    DTRAN2                                                      09626 33 09614 00000
17 952        CM    TRAN3-31,12,10                                              09638 14 12857 000J2
17 953        BL    DT2                                                         09650 47 09688 01300
17 956        TF    FG3X,TRAN3-21,,SAVE NEW 2ND GEN PAT 1ST POINT               09662 26 13481 12867
17 957        TF    FG3Y,TRAN3-11                                               09674 26 13491 12877
17 958        BB                                                                09686 42 00000 00000
17 959        DORG *-10                                                         09687
17 960 DT2    CM    TRAN2-31,12,10                                              09688 14 12808 000J2
17 961        BL    DT3                                                         09700 47 09738 01300
17 964        TF    FG3X,TRAN2-21,,SAVE NEW 1ST GEN START POINT                 09712 26 13481 12818
17 965        TF    FG3Y,TRAN2-11                                               09724 26 13491 12828
17 966        BB                                                                09736 42 00000 00000
```

```
PG LIN LABEL  OP    OPERANDS                                                    LOCN  OP INSTRUCTIONS

17 967        DORG *-10                                                         09737
17 968 DT3    TR    FG3X-9,ST-20,,TRANSFORM EQUALS 1ST POINT OF PAT             09738 31 13472 12891
17 970        BB                                                                09750 42 00000 00000
17 971        DORG *-10                                                         09751
17 972 D4     BNF   *&14,D4,0                                                   09752 M4 09766 09752
17 974        BB                                                                09764 42 00000 00000
17 975        DORG *-10                                                         09765
17 976        BNF   D5,D4&1                                                     09766 44 09910 09753
17 977        CM    TRAN3-31,12,10                                              09778 14 12857 000J2
17 978        BL    D6                                                          09790 47 10010 01300
17 981        TF    NXG3X,TRAN3-21,,SAVE 1ST POINT OF NEXT 2ND GEN PAT          09802 26 13502 12867
17 982        TF    NXG3Y,TRAN3-11                                              09814 26 13512 12877
17 983 D7     FS    NXG3X,FG3X                                                  09826 16 00469 -9861
17 983                                                                          09838 16 00445 J3502
17 983                                                                          09850 49 00402 J3481
17 984        FS    NXG3Y,FG3Y                                                  09862 16 00469 -9897
17 984                                                                          09874 16 00445 J3512
17 984                                                                          09886 49 00402 J3491
17 985        CF    D4&1                                                        09898 33 09753 00000
17 922 D5     CM    OUTPUT-46,11,10,TEST FOR Z COORD                            09910 14 12917 000J1
17 993        BNE   *&14                                                        09922 47 09936 01200
17 994        BB                                                                09934 42 00000 00000
17 995        DORG *-10                                                         09935
17 996        FA    OUTPUT-36,NXG3X                                             09936 16 00469 -9971
17 996                                                                          09948 16 00445 J2927
17 996                                                                          09960 49 00422 J3502
17 997        FA    OUTPUT-26,NXG3Y                                             09972 16 00469 J0007
17 997                                                                          09984 16 00445 J2937
17 997                                                                          09996 49 00422 J3512
17 998        BB                                                                10008 42 00000 00000
17 999        DORG *-10                                                         10009
18 000 D6     CM    TRAN2-31,12,10                                              10010 14 12808 000J2
18 001        BL    D8                                                          10022 47 10066 01300
18 004        TF    NXG3X,TRAN2-21                                              10034 26 13502 12818
18 005        TF    NXG3Y,TRAN2-11                                              10046 26 13512 12828
18 006        B     D7                                                          10058 49 09826 00000
18 007        DORG *-4                                                          10065
18 008 D8     TR    NXG3X-9,ST-20                                               10066 31 13493 12691
18 010        B     D7                                                          10078 49 09826 00000
18 040        DORG *&6                                                          10095
18 040              *  SICO SUBROUTINE - COMPUTES SINE AND COSINE OF ROT ANGLE
18 041 SICO   FM    RCONV,SICO-1                                                10096 16 00469 J0131
18 041                                                                          10108 26 01260 13409
18 041                                                                          10120 49 01262 J0095
18 042        TF    9,99,,ANGLE IN RADIANS                                      10132 26 00009 00099
18 043        FSIN THSIN,9                                                      10144 16 14270 J0179
18 043                                                                          10156 16 14258 J3429
18 043                                                                          10168 49 13548 -0009
```

```
PG LIN LABEL  OP    OPERANDS                                                          LOCN  OP INSTRUCTIONS

18 044        FCOS  THCOS,9                                                           10180 16 14270 J0215
18 044                                                                                10192 16 14258 J3419
18 044                                                                                10204 49 13516 -0009
18 045        CF    TRIGF                                                             10216 33 13267 00000
18 046        BB                                                                      10228 42 00000 00000
18 200        DORG  *-4                                                               10235
18 200              *   MILLR SUBROUTINE  PERFORMS INVERSIONS FOR MILLING OPERATIONS
18 201 MILLR  CM    TYPE,20,9,TEST FOR MILL OR FACE MILL                              10236 14 12130 00-20
18 202        BH    FACE                                                              10248 46 10980 01100
18 203        TR    OUTPUT-47,AID-42,,NEXT POINT TO OUTPUT BUFFER                     10260 31 12916 13012
18 204        SM    PTCNT,1,10,INDEX POINTER BACKWARDS                                10272 12 12182 000-1
18 206        BNR   *-12,PTCNT,11                                                     10284 45 10272 1218K
18 208        AM    PTCNT,1,10                                                        10296 11 12182 000-1
18 210        BNF   M1,AORS,,TEST FOR ARC OR SLOPE                                    10308 44 10340 10484
18 212        TR    AID-42,PTCNT,11,THIS POINT TO AUX BUFF                            10320 31 13012 1218K
18 214        B     AORS                                                              10332 49 10484 00000
18 216        DORG  *-4                                                               10339
18 218 M1     TR    OUTPUT-47,PTCNT,11, 1ST POINT TO OUTPUT BUFF                      10340 31 12916 1218K
18 220        CM    OUTPUT-46,26,10,TEST FOR GOTO OR STOP                             10352 14 12917 000K6
18 222        BL    M2                                                                10364 47 10434 01300
18 224        BD    *&32,OUTPUT-50,,TEST FOR GOTO OR STOP                             10376 43 10408 12913
18 226        SF    NOREV                                                             10388 32 08688 00000
18 228        B     *&20                                                              10400 49 10420 00000
18 230        DORG  *-4                                                               10407
18 232        SF    NOREV&1                                                           10408 32 08689 00000
18 234        TR    OUTPUT-51,OUTPUT-47,,LEFT SHIFT 4 LOCATIONS                       10420 31 12912 12916
18 236        BB                                                                      10432 42 00000 00000
18 238        DORG  *-10                                                              10433
18 240 M2     CM    OUTPUT-46,14,10,SLOPE OR ARC CHECK                                10434 14 12917 000J4
18 242        BNL   *&26                                                              10446 46 10472 01300
18 244        SM    PTCNT,1,10                                                        10458 12 12182 000-1
18 246        BB                                                                      10470 42 00000 00000
18 248        DORG  *-10                                                              10471
18 250        BNF   M6,AORS,,WAS LAST POINT A SLOPE OR AN ARC                         10472 44 10924 10484
18 252 AORS   TF    OUTPUT-36,AID-31,,X AND Y TO OUTPUT BUFF                          10484 26 12927 13023
18 254        TF    OUTPUT-26,AID-21                                                  10496 26 12937 13033
18 256        CM    OUTPUT-46,16,10,WAS LAST PT A SLOPE                               10508 14 12917 000J6
18 258        BNE   MFAD&36                                                           10520 47 10732 01200
18 260        CM    OUTPUT-21,53360,,TEST SIZE OF ANGLE                               10532 14 12942 N3360
18 262        BL    *&56                                                              10544 47 10600 01300
18 264        FS    OUTPUT-16,D360                                                    10556 16 00469 J0391
18 264                                                                                10568 16 00445 J2947
18 264                                                                                10580 49 00402 J3399
18 266        B     *-60                                                              10592 49 10532 00000
18 268        DORG  *-4                                                               10599
18 270        TFM   MFAD&35,090                                                       10600 16 10731 J3369
18 272        TFM   MFAD&30,00422                                                     10612 16 10726 -0422
18 274        CM    OUTPUT-21,53270,,TEST FOR ANGLE GREATER THAN 270 DEGREES          10624 14 12942 N3270
```

```
PG LIN LABEL  OP    OPERANDS                                                          LOCN  OP INSTRUCTIONS

18 276        BNH   M3                                                                10636 47 10800 01100
18 278        TFM   MFAD&30,00402                                                     10648 16 10726 -0402
18 280 M4     CM    MILLR-1,32,610,TEST FOR INVERT RIGHT                              10660 14 1023N 000L2
18 282        BNE   *&24                                                              10672 47 10696 01200
18 284        TFM   MFAD&35,D270                                                      10684 16 10731 J3389
18 186 MFAD   FA    OUTPUT-16,D90                                                     10696 16 00469 J0731
18 186                                                                                10708 16 00445 J2947
18 186                                                                                10720 49 00422 J3369
18 287        CM    AID-41,14,10,TEST FOR CW ARC                                      10732 14 13013 000J4
18 288        BL    *&36                                                              10744 47 10780 01300
18 289        CM    AID-41,26,10,TEST FOR GOTO OR STOP                                10756 14 13013 000K6
18 290        BL    M2&24                                                             10768 47 10458 01300
18 191        CF    AORS,,,RESET ARC OR SLOPE SWITCH                                  10780 33 10484 00000
18 292        B     M2&24                                                             10792 49 10458 00000
18 293        DORG  *-4                                                               10799
18 294 M3     CM    OUTPUT-21,52900,7,TEST FOR ANGLE LESS THAN 90 DEG                 10800 14 12942 N2900
18 295        BL    M4                                                                10812 47 10660 01300
18 296        CM    OUTPUT-21,53180,7,TEST FOR ANGLE GREATER THAN 180 DEG             10824 14 12942 N3180
18 298        BH    M5                                                                10836 46 10892 01100
18 300        CM    MILLR-1,32,610,TEST FOR INVERT RIGHT OR LEFT                      10848 14 1023N 000L2
18 302        BE    MFAD                                                              10860 46 10696 01200
18 304        TFM   MFAD&30,00402                                                     10872 16 10726 -0402
18 306        B     MFAD                                                              10884 49 10696 00000
18 308        DORG  *-4                                                               10891
18 310 M5     CM    MILLR-1,34,610,TEST FOR INVERT UP OR DOWN                         10892 14 1023N 000L4
18 312        BE    MFAD                                                              10904 46 10696 01200
18 314        B     M5-20                                                             10916 49 10872 00000
18 316        DORG  *-4                                                               10923
18 318 M6     SF    AORS,,,SET ARC OR SLOPE SWITCH                                    10924 32 10484 00000
18 320        TR    AID-42,OUTPUT-47,,NEXT PT TO AUX BUFF                             10936 31 13012 12916
18 322        TR    OUTPUT-25,AUXIN,,SET CONTROL REC MARK                             10948 31 12938 12104
18 324        TFM   OUTPUT-46,12,10,SET COORD CODE                                    10960 16 12917 000J2
18 326        B     M2&24                                                             10972 49 10458 00000
18 328        DORG  *-4                                                               10979
18 320 FACE   BNF   *&38,FACE,,TEST FOR FACE MILL                                     10980 44 11018 10980
18 322        TR    OUTPUT-47,AID-42,,THIS POINT TO OUTPUT BUFF                       10992 31 12916 13012
18 324        CF    FACE                                                              11004 33 10980 00000
18 326        BB                                                                      11016 42 00000 00000
18 328        DORG  *-10                                                              11017
18 330        TR    AID-42,OUTPUT-47,,SHIFT PT TO AUX BUFF                            11018 31 13012 12916
18 332        TF    9,COMP9&6                                                         11030 26 00009 11414
18 332        AM    9,1,10                                                            11042 11 00009 000-1
18 333        BNR   *-12,9,11,INDEX 1ST GEN POINTER                                   11054 45 11042 0000R
18 334        AM    9,1,10                                                            11066 11 00009 000-1
18 335        TR    OUTPUT-47,9,11,NEXT POINT TO OUTPUT BUFF                          11078 31 12916 0000R
18 336        BNR   *&20,AID-20,,TEST FOR DEPTH                                       11090 45 11110 13034
18 337        B     *&56                                                              11102 49 11158 00000
18 338        DORG  *-4                                                               11109
```

```
PG LIN LABEL  OP    OPERANDS                                                       LOCN  OP INSTRUCTIONS

18 339        TR    OUTPUT-25,AID-20,,SHIFT DEPTH TO 1ST PT OF PAT                 11110 31 12938 13034
18 340        TR    AID-20,AUXO,,SET CONTROL REC MARK                              11122 31 13034 13011
18 341        TDM   AID-41,2,, XCHANGE COORD CODES                                 11134 15 13013 00002
18 342        TDM   OUTPUT-46,3                                                    11146 15 12917 00003
18 343        SF    FAC                                                            11158 32 10980 00000
18 344        BB                                                                   11170 42 00000 00000
18 345        DORG  *-10                                                           11171
16 170 PLONG  BTM   CRSUB,7400,8                                                   11172 17 11190 0P400
16 171 MESSG  DAC   3,                                                             11135          3 3,E
16 172 CRSUB  TR    *-1,AUXIN                                                      11190 31 11189 12104
16 173        WATY  MESSG                                                          11202 39 11185 00100
16 174        H     ,,,PUSH START TO CONTINUE                                      11214 48 00000 00000
16 175        TR    OUTPUT-47,INPUT-47                                             11226 31 12916 12025
16 176        BTM   OPUT,,,OUTPUT THIS REC                                         11238 17 11484 -0000
16 177        TR    INPUT,AUXIN                                                    11250 31 12072 12104
16 178        TR    OUTPUT,AUXO                                                    11262 31 12963 13011
16 179        B     NEXT                                                           11274 49 01700 00000
19 010        DORG  *62                                                            11287
19 014              *SUBROUTINE SP-STORES VARIABLE REC IN PAT TABLE
19 015 SP     TR    PTAP,*-1,611,STORE THIS RECORD IN PATTERN TABLE                11288 31 1326- 1128P
19 020        AM    PTAP,1,10                                                      11300 11 13260 000-1
19 025        AM    PFLAG64,1,10                                                   11312 11 13266 000-1
19 030        CM    PTAB,19999,7,TEST FOR END OF PAT STORAGE                       11324 14 13260 J9999
19 035        BH    PATEX                                                          11336 46 11382 01100
19 040        BNR   *614,SP-1,11                                                   11348 45 11362 1128P
19 045        BB                                                                   11360 42 00000 00000
19 059        DORG  *-10                                                           11361
19 056        AM    SP-1,1,10,INDEX LENGTH OF REC POINTER                          11362 11 11287 000-1
19 060        B     SP612                                                          11374 49 11300 00000
19 062        DORG  *-4                                                            11381
19 065 PATEX  BTM   ERSUB,7300,8,ALLOWABLE PAT STORAGE EXCEEDED                    11382 17 11190 0P300
19 120        DORG  *52                                                            11395
20 000              *  SUBROUTINE PATLOC-SCANS PAT TABLE TO LOCATE PAT TO BE OUTPUT
20 030 PATLOC TFM   COMP966,GNO6994,,RESET TO START OF PAT TABLE                   11396 16 11414 J4509
20 040 COMP9  C     00000,PATLOC-1,,TEST FOR PAT NO                                11408 24 00000 11395
20 050        BNE   *614                                                           11420 47 11434 01200
20 060        BB                                                                   11432 42 00000 00000
20 062        DORG  *-10                                                           11433
20 070        TF    ADD9611,COMP966                                                11434 26 11469 11414
20 080        AM    ADD9611,05,10                                                  11446 11 11469 000-5
20 090 ADD9   A     COMP966,00000,,MOVE POINTER TO TEST NEXT PAT IN PAT TABLE      11458 21 11414 00000
20 100        B     COMP9                                                          11470 49 11408 00000
21 012        DORG  *62                                                            11483
21 013              *SUBROUTINE OPUT---OUTPUTS CARDS OR TAPE W/ OR W/O TYPED LISTING
21 040 OPUT   BC2   PNCHO,,,TEST FOR CARD OUTPUT                                   11484 46 11586 00200
21 016        WNPT  OUTPUT-47,,,PAPER TAPE OUTPUT                                  11496 38 12916 00200
21 017        BC4   *620                                                           11508 46 11528 00400
21 018        B     PRNTO                                                          11520 49 11554 00000
```

```
PG LIN LABEL  OP    OPERANDS                                                       LOCN  OP INSTRUCTIONS

21 019        DORG  *-4                                                            11527
21 020 CK99   CM    OUTPUT-46,99,10,TEST FOR END OF PROGRAM                        11528 14 12917 000R9
21 021        BE    FINI                                                           11540 46 12006 01200
21 022 TWOUT  BB                                                                   11552 42 00000 00000
21 023        DORG  *-10                                                           11553
21 024 PRNTO  WATY  OUTPUT-47,,,OUTPUT LISTING ON TYPEWRITTER                      11554 38 12916 00100
21 026        TBTY                                                                 11566 34 00000 00108
21 028        B     CK99                                                           11578 49 11528 00000
21 030        DORG  *-4                                                            11585
21 032 PNCHO  AM    RSIZE,01,10,CARD OUTPUT                                        11586 11 13245 000-1
21 034        BNR   PNCHO,RSIZE,11,COUNT LENGTH OF THIS RECORD                     11598 45 11586 1324N
21 036        AM    RSIZE,01,10                                                    11610 11 13245 000-1
21 038        SM    RSIZE,OUTPUT-47                                                11622 12 13245 J2916
21 040        A     OUTLOC,RSIZE                                                   11634 21 13235 13245
21 042        CM    OUTLOC,CARDO-5,,TEST FOR FULL OUTPUT BUFFER                    11646 14 13235 J3210
21 044        BH    NOFIT                                                          11658 46 11690 01100
21 046        S     OUTLOC,RSIZE                                                   11670 22 13235 13245
21 048        B     FILL                                                           11682 49 11914 00000
21 050        DORG  *-4                                                            11689
21 052 NOFIT  S     OUTLOC,RSIZE                                                   11690 22 13235 13245
21 054 DIGO   TDM   OUTLOC,00000,6,RESET REMAINDER OF CARD TO ZERO                 11702 15 1323N 00000
21 056        AM    OUTLOC,01,10                                                   11714 11 13235 000-1
21 058        CM    OUTLOC,CARDO-4                                                 11726 14 13235 J3211
21 053        BL    DIGO                                                           11733 47 11702 01300
21 062        AM    SEQNO,01,10                                                    11750 11 13250 000-1
21 064        TF    CARDO,SEQNO,,SET SEQUENCE NO                                   11762 26 13215 13250
21 066        WNCD  CARDO-79,,,PUNCH ONE CARD                                      11774 38 13136 00400   49 19972
21 067 NOEND  BC4   REFEO,,,TEST FOR LISTING                                       11786 46 11890 00400
21 071 PRNOT  BNF   REFIT,PRLOC,11                                                 11798 44 11878 1324-
21 072        WNTY  PRLOC,,6,OUTPUT THIS REC ON TYPEWRITTER                        11810 38 1324- 00100
21 072        TBTY                                                                 11822 34 00000 00108
21 073        AM    PRLOC,01,10                                                    11834 11 13240 000-1
21 074        BNR   *-12,PRLOC,11                                                  11846 45 11834 1324-
21 075        AM    PRLOC,01,10                                                    11858 11 13240 000-1
21 076        B     PRNOT                                                          11870 49 11798 00000
21 077        DORG  *-4                                                            11877
21 078 REFIT  TFM   PRLOC,CARDO-79,,RESET POINTERS                                 11878 16 13240 J3136
21 081 REFEO  TFM   OUTLOC,CARDO-79                                                11890 16 13235 J3136
21 082        BNF   REFIX,TRIGFE1                                                  11902 44 11938 13268
21 083 FILL   TR    OUTLOC,OUTPUT-47,6,THIS REC TO OUTPUT BUFFER                   11914 31 1323N 12916
21 084        A     OUTLOC,RSIZE,,INDEX OUTPUT BUFF REGISTER                       11926 21 13235 13245
21 085 REFIX  TFM   RSIZE,OUTPUT-47                                                11938 16 13245 J2916
21 086        CM    OUTPUT-46,99,10,TEST FOR END OF PROGRAM                        11950 14 12917 00089
21 088        BNE   TWOUT                                                          11962 47 11552 01200
21 090        BNF   FINI,TRIGFE1                                                   11974 44 12006 13268
21 092        CF    TRIGFE1                                                        11986 33 13268 00000
21 094        B     DIGO,,,PUNCH LAST CARD                                         11998 49 11702 00000
21 096        DORG  *-4                                                            12005
```

49 19522

```
PG LIN LABEL  OP    OPERANDS                                  LOCN  OP INSTRUCTIONS

21 098 FINI   WATY   MESS1,,,END OF PHASE2                    12006 39 13339 00100
21 100        B     INIT,,,INITIALIZE FOR NEXT JOB            ~~12018 49 06050 00000~~
21 101        DORG  *-4                                       12025
 1 010 INPUT  DS    48                                        12072       48
 1 015        DC    1,@,INPUT                                 12072        1
 1 016 AUXIN  DS    32                                        12104       32
 1 017        DC    1,@,AUXIN                                 12104        1
 1 020 TXTY   DS    21                                        12125       21
 1 029 RTYPE  DC    2,00                                      12127        2 -0
 1 030 TYPE   DS    3                                         12130        3
 1 040 TOLNO  DS    4                                         12134        4
 1 041 TWO    DC    10,5120000000                             12144       10 N120000000
 1 042 THETA  DS    10                                        12154       10
 1 043 BUFFN  DS    10                                        12164       10
 1 044 BUFFD  DS    10                                        12174       10
 1 046 TOLMOD DS    3                                         12177        3
 1 046 PTCNT  DC    5,00000                                   12182        5 -0000
 1 046 RTCNT  DC    5,00000                                   12187        5 -0000
 1 047 TBLE   DS    560                                       12747      560
 1 050 COMP1  DSA   *67                                       12752        5 J2754
 1 055        DC    2,17                                      12754        2 J7
 1 060        DC    2,18                                      12756        2 J8
 1 070        DC    2,19                                      12758        2 J9
 1 080        DC    2,21                                      12760        2 K1
 1 090        DC    2,22                                      12762        2 K2
 1 095        DC    2,27                                      12764        2 K7
 1 099        DC    2,28                                      12766        2 K8
 1 099        DC    2,29                                      12768        2 K9
 1 100        DC    2,90                                      12770        2 R0
 1 104        DC    2,99                                      12772        2 R9
 1 110 COMP2  DSA   *67                                       12777        5 J2779
 1 113        DC    2,32                                      12779        2 L2
 1 116        DC    2,34                                      12781        2 L4
 1 120        DC    2,36                                      12783        2 L6
 1 124        DC    2,40                                      12785        2 M0
16 139 TYLOC  DC    5,00000                                   12790        5 -0000
16 141 PLOC2  DC    5,00000                                   12795        5 -0000
16 141 PCNT2  DC    5,00000                                   12800        5 -0000
16 142 OPER2  DC    2,00                                      12802        2 -0
16 143 SURF2  DC    2,00                                      12804        2 -0
16 143        DC    2,00                                      12806        2 -0
16 144 TRAN2  DS    33                                        12839       33
16 145 PLOC3  DC    5,00000                                   12844        5 -0000
16 146 PCNT3  DC    5,00000                                   12849        5 -0000
16 147 OPER3  DC    2,00                                      12851        2 -0
16 148 SURF3  DC    2,00                                      12853        2 -0
16 148        DC    2,00                                      12855        2 -0
16 149 TRAN3  DS    33                                        12888       33
```

```
PG LIN LABEL  OP    OPERANDS                                  LOCN  OP INSTRUCTIONS

16 145 PATNO  DC    2,00                                      12890        2 -0
16 146 ST     DS    21                                        12911       21
16 148        DC    2,00                                      12913        2 -0
16 149        DC    2,00                                      12915        2 -0
16 150 OUTPUT DS    48                                        12963       48
16 151        DC    1,@,OUTPUT                                12963        1
16 152 AUXO   DS    48                                        13011       48
16 153        DC    1,@,AUXO                                  13011        1
16 153 AIO    DS    43                                        13054       43
16 153        DC    1,@                                       13055        1
16 154 CARDI  DS    80                                        13135       80
16 155 CARDO  DS    80                                        13215       80
16 156 INLOC  DSA   CARDI-79                                  13220        5 J3056
16 157 TABPT  DSA   TBLE-556                                  13225        5 J2191
16 159 ADRES  DSA   TBLE-556                                  13230        5 J2191
16 160 OUTLOC DSA   CARDO-79                                  13235        5 J3136
16 162 PRLOC  DSA   CARDO-79                                  13240        5 J3136
16 164 PSIZE  DSA   OUTPUT-47                                 13245        5 J2916
16 166 SEQNO  DC    5,00000                                   13250        5 -0000
16 167 SCNCT  DC    5,00000                                   13255        5 -0000
16 168 PTAB   DSA   GN06993                                   13260        5 J4508
16 168 PFLAG  DC    2,-00                                     13262        2 --
16 168 TRIGF  DC    5,-00000                                  13267        5 -000-
16 168        DC    6,000000                                  13273        6 -00000
16 168 PCNT   DC    2,00                                      13275        2 -0
16 168 PLEV   DC    2,00                                      13277        2 -0
16 168        DC    1,@                                       13278        1
16 169 TELET  DSA   TBLE-556,TBLE-556,CARDO-79,CARDO-79,OUTPUT-47,,,GN06993   13283   5 J2191
16 169                                                        13293        5 J3136
16 169                                                        13303        5 J2916
16 169                                                        13313        5 -0000
16 169                                                              C   2# -004
16 169        DC    5,-00000                                  13325        5 -000-
16 169        DC    6,000000                                  13331        6 -00000
16 169        DC    2,00                                      13333        2 -0
16 169        DC    3,00@                                     13336        3 -0
16 170 MESS1  TAG   11,END PHASE2@                            13339       11 11,END PHASF2@
16 172 D30    DC    10,5290000000                             13369       10 N290000000
16 174 E180   DC    10,5318000000                             13379       10 N318000000
16 176 E270   DC    10,5327000000                             13389       10 N327000000
16 178 D360   DC    10,5336000000                             13399       10 N336000000
18 810 RCELV  DC    10,4917453293                             13409       10 M917453293
18 885 [illegible] DC    10                                        13419       10
18 890 [illegible] DC    10                                        13429       10
18 892 [illegible] DC    10                                        13439       10
18 894 [illegible] DC    10                                        13449       10
18 895        DC    1,                                        13450        1
18 896 [illegible] DC    10                                        13460       10
```

```
 PG LIN LABEL  OP    OPERANDS                                              LOCN  OP INSTRUCTIONS

 18 898 NXG2Y  DS    10                                                    13470        10
 18 899        DC    1,‡                                                   13471         1
 18 900 FG3X   DS    10                                                    13481        10
 18 901 FG3Y   DS    10                                                    13491        10
 18 902        DC    1,‡                                                   13492         1
 18 902 NXG3X  DS    10                                                    13502        10
 18 903 NXG3Y  DS    10                                                    13512        10
 18 904        DC    1,‡                                                   13513         1
 18 905 GND    DC    2,00                                                  13515         2 -0
  6 110        DEND  NEXT                                                  01700
```

MODIFICATION TO PHASE 2 OF AUTOSPOT TO RUN FROM DISK

PATCH PROGRAM

```
19466   26   13225̅   1203̲3
19478   11   13225   00010
19490   26   13225̅   12055
19502   11   13225   0000̅4   Program patch that was in the way moved to here
19514   49   11226   0
19522   34   19560   00701   End of this phase - seek for next phase
19534   49   19976   06850
19546   10540020000000        Disk control word this section
19560   10535001001900        Disk control word next section
19574   bbbbbbbbb
```

MODIFICATION TO ORIGINAL PROGRAM

```
19584   36   00000   00500   Read in last cards of modification
19596   36   19680   00500
19608   26   01762   19712   Modify "read a card" instruction
19620   26   03219   19705   Modify error routine
19632   41   00000   00000   No Op
19644   26   11780   19720   Modify "write a card" instruction
19656   26   12024   19727   Modify end to call next program
19668   34   19546   00701
19680   38   19546   00702   Write these modifications all on the disk
19692   48
19694   4̅6   19466   01200
19706   4̅9   197420
19714   4̅9   19912   4̅919522
```

READ DISK DATA

```
19728   10̅00000̅0119600
19742   36   19728   00702   Read a card from disk & seek if necessary then go back
                             and read again
19754   47   19786   03600
19766   34   19728   00701
19778   49   19742   0
19786   11   19733   0̅0001   Increase sector address by one
```

TRANSFER FIELD JUST READ TO EVEN LOCATION

```
19798   25   13056   19600   Transfer data to area beginning with an even address
19810   14   19809   1̅9679
19822   46   19866   01200   Check for last transfer
19834   11   19809   0̅0001   Increase disc location by one
19846   11   19804   0̅0001
19858   49   19798   0        Repeat
```

MODIFICATIONS TO PHASE 2 of AUTOSPOT

TO RUN FROM DISK

```
TRANSFER  FIELD JUST READ TO EVEN LOCATION
19866  16  19809  1̄9600  Housekeep because all data is transferred now
19878  16  19804  1̄3056
19890  49  01768  0      Return to program

WRITE DATA ON DISK
19898  1̄020000̄0113136
19912  38  19898  00702  Write data on disk
19924  47  19956  03600  If address check - seek first then write data
19936  34  19898  00701
19948  49  19912  0
19956  11  19903  0̄0001  Increase sector address by one
19968  49  11786  0      Return to program

19976  36  19560  00702  End of this phase - read in control program for next phase
19988  49  02318  00000
```

Load program to load in modifications into core

```
00000  36  19466  00500  Load modifications into core and branch to the first modificati
00012  36  19546  00500
00024  36  19626  00500
00036  36  19760  00500
00048  36  19840  00500
00060  36  19920  00500
00072  49  19584  0
```

GOODYEAR

GOODYEAR AEROSPACE
CORPORATION

ARIZONA DIVISION
LITCHFIELD PARK, ARIZONA

GENERAL RAY TRACE PROGRAM

Presented to the 1620 Users Group
(Western Region) Meeting at Tempe,
Arizona, December 12, 1963.

Presented by D. H. O'Herren

AAP-18375 December 12, 1963

514

INDEX

GENERAL RAY TRACE PROGRAM

This program is based upon a paper by Gordon H. Spencer titled "A General Ray Tracing Procedure", IBM Research Paper RC-549. Spencer's paper applies, with a few extensions, to the problem of tracing a light ray thru surfaces which may be rotationally symmetric, cylindrical, or conic and may have arbitrary orientations with respect to a reference system. Certain adjustments and additions have been included in the program. This paper does not attempt to delve deeply into the mathematics behind the program. It is felt that a brief description of some of its advantageous points along with a sample ray trace problem would be of more interest. The sample will attempt to illustrate the value of the program as an optics system evaluation.tool.

In general, surfaces are described in a local coordinate system. This local system is then positioned in the optical system thru translation and rotation of the local system in relation to the basic coordinate system. It is thereby possible to position surfaces off the optical axis and "tilted" to the desired degree.

One feature the program possesses which is not common to some other ray trace programs is the capability to trace rays thru cylinders and prisms. These types of lenses are fairly common in modern optics systems making this an important feature.

The basic surface equation suggested by Spencer which is used in the program is:

$$F(X, Y, Z) = AX^2 + BY^2 + CZ^2 + Z = 0.$$

This represents the surface obtained by revolving a conic section with vertex at the origin about the Z axis. In general, a ray intersects such surfaces at two points. By using vertex equations set up so that the first iteration point lies on the plane Z = o thru the vertex, the iteration starts closer to the desired intersection point than to the extraneous point.

For cylinders with conic cross-section in the XZ plane, the surface equation becomes:

$$F = AX^2 + CZ^2 + Z = 0$$

The rulings of this cylinder are parallel to the Y axis. A similar equation will describe a cylinder with rulings parallel to the X axis.

The equation:

$$\frac{X^2}{(aY)^2} + \frac{Z^2}{(bY)^2} = 1$$

describes a cone with apex at the origin with the Y axis as the principal axis. Cross-sections parallel to the XZ plane are ellipses. This equation may be rewritten in the form:

$$F = Z + D\sqrt{EX^2 + Y^2} = 0$$

To iterate with this equation, the following rule holds:

$$Z \geq 0 \quad \text{for } D < 0$$

$$Z \leq 0 \quad \text{for } D > 0$$

Therefore it is possible to represent three types of surfaces with the single equation:

$$F = AX^2 + BY^2 + CZ^2 + Z + D\sqrt{EX^2 + Y^2} = 0$$

The constants A, B, C, D, and E are program input data.

GENERAL RAY TRACE

INPUT

A header card and 3 cards for each surface are required input. Ray information may be entered in 2 ways, either one ray per card or as a fan of rays. To enter single rays, switch No. 1 must be on.

CODE	HEADER CARD:	
REFR (1)	Cols. 1 - 11	Initial index of refraction (may appear anywhere in Cols. 1 - 11). Ex ∴ If the initial medium is air, the number 1.0 can be punched in Cols. 1 - 3.
	Col. 12	Blank
NOSUR	Cols. 13 - 14	Number of surfaces. If the number of surfaces is 9 or less, punch it in Col. 14 and leave Col. 13 blank.
TOL	Cols. 15 - 29	Iteration tolerance in form + 0.10000000E-YY. This number is used to establish a criterion for convergence of the iteration process. Convergence is assumed when the increment magnitude is less than the tolerance. It is recommended that +0.10000000E-06 ($.1 \times 10^{-6} = 10^{-7}$) be used, as only 8 places are carried in computation. It may sometimes be necessary to relax the tolerance to + 0.10000000E-05, either to obtain convergence or to speed up the program.
	Cols. 30 - 80	Blank

SURFACE CARDS:

Card A:

XO(I)	Col. 1 - 11	X - coordinate of origin of local system, punched in same fashion as the initial index of refraction on the header card.
YO(I)	Cols. 12 - 22	Y - coordinate of origin of local system, punched in same fashion as X - coordinate
ZO(I)	Cols. 23- 33	Z - coordinate of origin of local system, punched in same fashion as X - coordinate.
Alpha(I)	Cols. 34 - 44	Y axis euler angle (α) in decimal degrees, punched in same fashion as X - coordinate.
Beta(I)	Cols. 45 - 55	X axis euler angle (β) in decimal degrees, punched in same fashion as X - coordinate.
Gamma(I)	Cols. 56 - 66	Z axis euler angle (γ) in decimal degrees, punched in same fashion as X - coordinate.
	Cols. 67 - 77	Blank.
	Cols. 78 - 80	It is suggested, but not required, that the surface cards be punched 01A, 01B, 01C, 02A, 02B, etc., in Cols. 78 - 80 to ensure that they are kept in the correct order.

Card B:

The first five fields give the coefficients in the surface equation $F = AX^2 + BY^2 + CZ^2 + Z + D \quad EX^2 + Y^2 = 0$, as follows:

A(I)	Cols. 1 - 11, A (punched in same fashion as X - coordinate on Card A)
B(I)	Cols. 12 - 22B (punched in same fashion as A).
C(I)	Cols. 23 - 33C (punched in same fashion as A).
D(I)	Cols. 34 - 44D (punched in same fashion as A).
E(I)	Cols. 45 - 55E (punched in same fashion as A).

Refr(I)	Cols. 56 - 66	Index of refraction of medium following the surface, except in the case of reflection, when the negative of the index of refraction for the previous surface is used. (Field is punched in same fashion as A).
	Cols. 67 - 77	Blank
	Cols. 78 - 80	May be punched as suggested for Card A.

Card C:

AP1(I)	Cols. 1 - 11	X - coordinate of center of circular-annular aperture or coordinate of center of hyperbolic aperture or X lower bound of base of rectangular-trapezoidal aperture.
AP2(I)	Cols. 12 - 22	Y - coordinate of center of circular-annular aperture or coordinate of center of hyperbolic aperture or X upper bound of base of rectangular-trapezoidal aperture.
AP3(I)	Cols. 23 - 33	Inner radius of circular-annular aperture or length of semi-major axis of hyperbolic aperture or Y lower bound for rectangular-trapezoidal aperture.
AP4(I)	Cols. 34 - 44	Outer radius of circular-annular aperture or length of semi-minor axis of hyperbolic aperture or Y upper bound for rectangular-trapezoidal aperture.
AP5(I)	Cols. 45 - 55	Y lower bound for hyperbolic aperture or reciprocal slope of left hand side of rectangular-trapezoidal aperture. Enter 0 for circular-annular aperture.

AP6(I)	Cols. 56 - 66	Y upper bound for hyperbolic aperture or reciprocal slope of right hand side of rectanular-trapezoidal aperture. Enter 0, for circular-annular aperture.
	Cols. 67	Blank
NAP(I)	Col. 68	Aperture code Blank: circular - annular 1: rectangular-trapezoidal 2: hyperbolic
	Col. 69	Blank
NOUT(I)	Col. 70	Output code Blank: no output at surface 1: output at surface
	Cols. 71 - 77	Blank

RAY INPUT (Single ray per card, switch 1 on)

One record, either typed or punched, is used for each ray. NOTE: Coordinates with a bar over them are system coordinates. Those without a bar are local coordinates.

XA	Cols. 1 - 11	$\bar{X}$ - coordinate of 1st point on ray.
YA	Cols. 12 - 22	$\bar{Y}$ - coordinate of 1st point on ray.
ZA	Cols. 23 - 33	$\bar{Z}$ - coordinate of 1st point on ray.
RPAR 1	Cols. 34 - 44	$\bar{X}$ - coordinate of 2nd point on ray. or $\bar{X}$ direction cosine of ray at 1st point.
RPAR 2	Cols. 45 - 55	$\bar{Y}$ - coordinate of 2nd point on ray or $\bar{Y}$ direction cosine of ray at 1st point.
RPAR 3	Cols. 56 - 66	$\bar{Z}$ - coordinate of 2nd point on ray or $\bar{Z}$ direction cosine of ray at 1st point.
	Col. 67	Blank

NIN	Col. 68	Input code Blank: 2 points 1: 1 point and direction cosine
	Col. 69	Blank
NAXIN	Col. 70	Optical axis intersection computation code Blank: bypass computation of intersection of image ray with optical axis. 1: Perform above computation
IRAY	Cols. 71 - 73	Ray identification number (right justified)
	Cols. 74 - 80	Blank.

<u>RAY INPUT</u> (Fans of rays on 2 cards)

<u>Card 1</u>

Cols. 1 - 11	FXU1, the maximum $\overline{X}$ coordinate at the 1st point.
Cols. 12 - 22	FZ1, the $\overline{Z}$ coordinate at the 1st point.
Cols. 23 - 33	XGAP1, the $\overline{X}$ spacing between fans at 1st point.
Cols. 34 - 44	FXU2, the maximum $\overline{X}$ coordinate of each fan at the 2nd point.
Cols. 45 - 55	FZ2, the $\overline{Z}$ coordinate at the 2nd point.
Cols. 56 - 66	XGAP2, the $\overline{X}$ spacing between rays at the 2nd point.

<u>Card 2</u>

Cols. 1 - 11	FYU2, the maximum $\overline{Y}$ coordinate at the 2nd point.
Cols. 12 - 22	YGAP2, the $\overline{Y}$ spacing between rays at the 2nd point.

NOTE: All fans are assumed to originate at $Y = 0$ at the 1st point.

The following sample ray trace problem will better illustrate the useage of the program. The surface system to be traced is as follows:

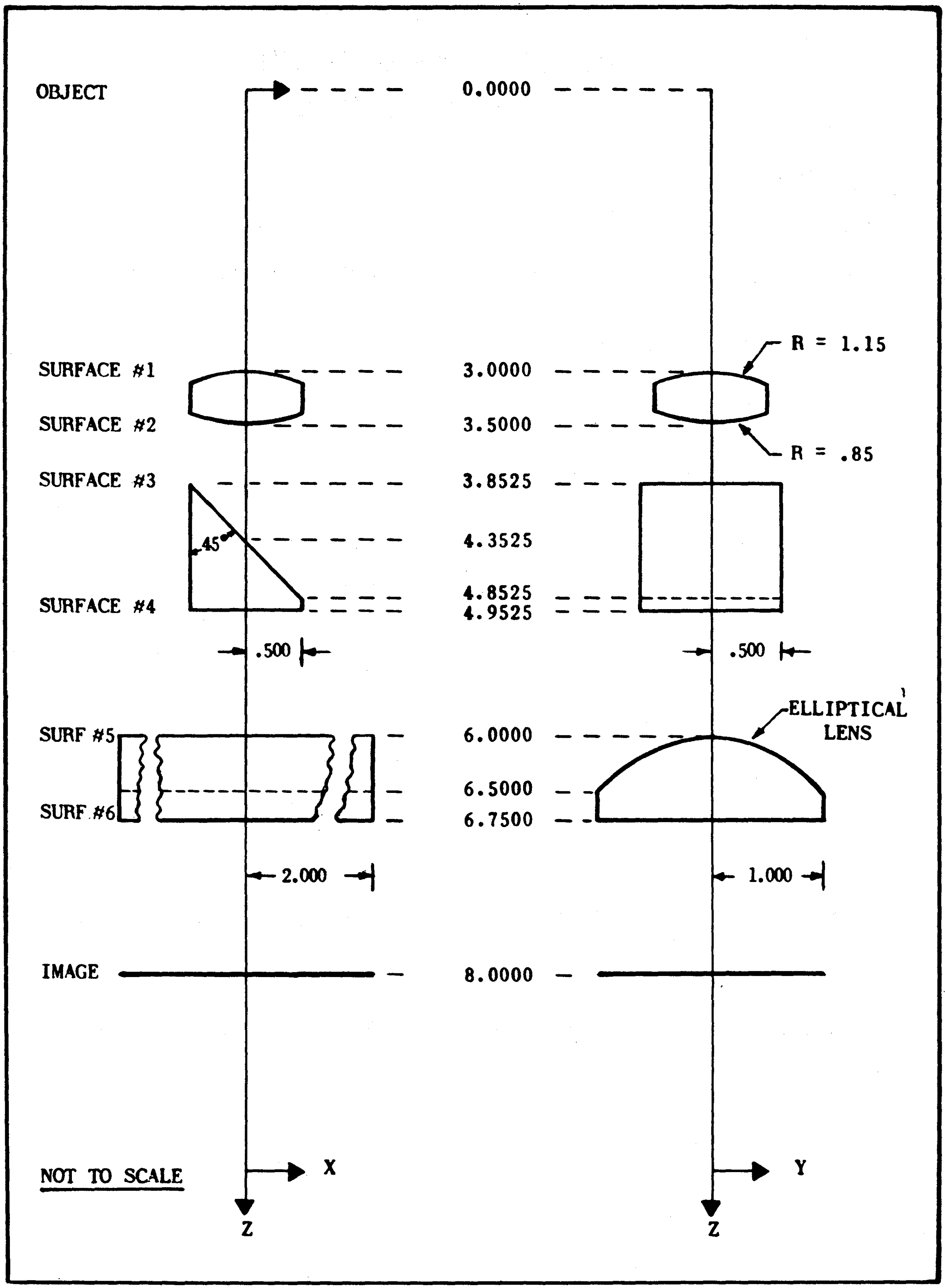


FIGURE - SAMPLE RAY TRACE SYSTEM

DATA INPUT TO GENERAL RAY TRACE
PROGRAM FOR SAMPLE PROBLEM

SURFACE DATA

SURFACES 1 & 2

These are both spherical surfaces. The surface equation becomes:

$$F = AX^2 + BY^2 + CZ^2 + Z = 0$$

$$= \frac{1}{2R} X^2 + \frac{1}{2R} Y^2 + \frac{1}{2R} Z^2 + Z = 0$$

$$= -.435 X^2 - .435 Y^2 - .435 Z^2 + Z = 0 \quad \text{(Surface \#1)}$$

$$= .588 X^2 + .588 Y^2 + .588 Z^2 + Z = 0 \quad \text{(Surface \#2)}$$

These are not cones. Therefore $D = E = 0$

The vertex planes for these surfaces are perpendicular to the Z (optical) axis. Therefore α, β, and γ are zero.

The vertex coordinates are:

Surface #1	(0, 0, 3.00)
Surface #2	(0, 0, 3.50)

The index of refraction following Surface #1 is 1.523, that following Surface #2 is 1.00.

The apertature on both surfaces is a circle of 0.5 inches diameter.

Surfaces 3 & 4

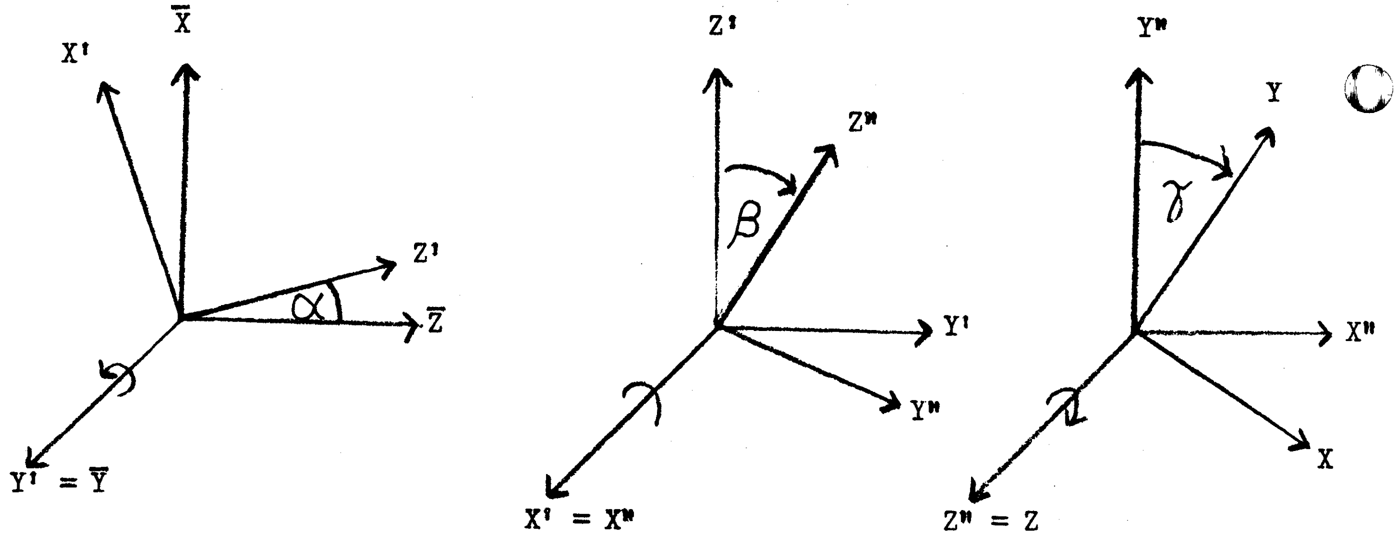
Both surfaces are planes arranged to constitute a prism. The first, No. 3, is arranged so that rays are deflected in the negative X direction. The second surface, No. 4, is perpendicular to the optic axis.

The surface equation for both surfaces is:

$$F = Z = 0$$

$$A = B = C = D = E = 0$$

Surface No. 3 is rotated 45 degrees when positioned in the system. To accomplish this rotation, it is necessary to specify the angle alpha equal to 45 degrees. The rotation angles, alpha, beta, and gamma are defined:



The specified apertature is a square 1" x 1". Therefore the minimum and maximum allowable x and y values are (0.5, 0.5) and (-0.5, 0.5) respectively. These values apply to the surfaces before rotation. After rotation of surface No. 3, the actual minimum and maximum x apertature values will be corrected by the program to $(-0.5/\sqrt{2}, 0.5/\sqrt{2})$. AP5 and AP6 are the inverse slopes of the left and right edges of the apertature as viewed from the object point (positive Y up). In the sample, these slopes are reciprocal infinity or zero. NAP is set equal to 1 to specify a rectangular apertature. NOUT is set equal to 1 to obtain output data at the surfaces. The index of refraction of the prism is 1.6.

Surface 5 & 6

Surface 5 is an elliptical cylinder. Surface 6 is a simple plane with input similar to surface No. 4. The equation of the ellipse in the YZ plane is:

$$\frac{Y^2}{a^2} + \frac{Z^2}{b^2} = 1 \qquad a = 1 \qquad b = 0.5$$

The surface equation, adjusted so that the origin is at the point nearest the object point, is:

$$F = -0.25\,Y^2 - Z^2 + Z = 0$$

Therefore $A = D = E = 0$, $B = -0.25$, $C = -1.00$. Note that surface No. 5 is a cylinder with rulings in the X direction. If it had been desired to translate this surface off the optic axis in the Y direction, then YO would be specified accordingly.

This program was originally written by William Webb of Goodyear Aerospace Corporation, Akron, Ohio and to that gentlemen goes the credit for this significant contribution to the lens designer's kit of tools. The program has been altered slightly to suit current needs and is being maintained by the writer.

SAMPLE PROBLEM DATA INPUT

```
SAMPLE PROBLEM. PROGRAM NO. 143A-63,   12/02/63

1.0          7+0.10000000E-06

0.0         0.0         3.0         0.0         0.0         0.0                   1A

-.435       -.435       -.435       0.0         0.0         1.523                 1B

0.0         0.0         0.0         0.5         0.0         0.0         0 1       1C

0.          0.          3.5         0.0         0.0         0.0                   2A

0.588       0.588       0.588       0.0         0.0         1.0                   2B

0.0         0.0         0.0         0.5         0.0         0.0         0 1       2C

0.0         0.0         4.3525      -45.0       0.0         0.0                   3A

0.0         0.0         0.0         0.0         0.0         1.6                   3B

-0.5        0.5         -0.5        0.5         0.0         0.0         1 1       3C

0.0         0.0         4.9525      0.0         0.0         0.0                   4A

0.0         0.0         0.0         0.0         0.0         1.0                   4B

-0.5        0.5         -0.5        0.5         0.0         0.0         1 1       4C

0.          0.0         6.0         0.0         0.0         0.0                   5A

0.0         -0.25       -1.0        0.0         0.0         1.5                   5B

-2.0        2.0         -1.0        1.0         0.0         0.0         1 1       5C

0.0         0.0         6.75        0.0         0.0         0.0                   6A

0.0         0.0         0.0         0.0         0.0         1.0                   6B

-2.0        2.0         -1.0        1.0         0.0         0.0         1 1       6C

0.0         0.0         8.0         0.0         0.0         0.0                   7A

0.0         0.0         0.0         0.0         0.0         1.0                   7B

0.0         0.0         0.0         10.0        0.0         0.0         0 1       7C

0.1         0.0         0.1         0.3         3.0         0.3

0.3         0.3
```

```
SAMPLE PROBLEM, PROGRAM NO. 143A-63,    12/02/63
RAY SURF                                    OBJ PT     2D PT/DC       INT PT       DIR COS


 1   1                       X OR K         .10000      .30000        .30568       -.05589
                             Y OR L         .00000      .30000        .30852       -.03509
                             Z OR M         .00000     3.00000       3.08521        .99781
 1   2                       X OR K         .10000      .30000        .28848       -.32813
                             Y OR L         .00000      .30000        .29772       -.30423
                             Z OR M         .00000     3.00000       3.39209        .89429
 1   3                       X OR K         .10000      .30000       -.04675       -.49435
                             Y OR L         .00000      .30000       -.01308       -.19014
                             Z OR M         .00000     3.00000       4.30574        .84820
 1   4                       X OR K         .10000      .30000       -.42369       -.79097
                             Y OR L         .00000      .30000       -.15807       -.30423
                             Z OR M         .00000     3.00000       4.95250        .53085
 1   5 NO INCIDENCE
 2   1                       X OR K         .10000      .00000       -.00136       -.02133
                             Y OR L         .00000      .30000        .30409       -.02971
                             Z OR M         .00000     3.00000       3.04095        .99933
 2   2                       X OR K         .10000      .00000       -.01006       -.02552
                             Y OR L         .00000      .30000        .29198       -.24762
                             Z OR M         .00000     3.00000       3.44823        .96851
 2   3                       X OR K         .10000      .00000       -.03302       -.33870
                             Y OR L         .00000      .30000        .06923       -.15476
                             Z OR M         .00000     3.00000       4.31947        .92807
 2   4                       X OR K         .10000      .00000       -.26404       -.54192
                             Y OR L         .00000      .30000       -.03633       -.24762
                             Z OR M         .00000     3.00000       4.95250        .80311
 2   5                       X OR K         .10000      .00000       -.99486       -.36128
                             Y OR L         .00000      .30000       -.37026       -.08675
                             Z OR M         .00000     3.00000       6.03553        .92840
 2   6                       X OR K         .10000      .00000      -1.27289       -.54192
                             Y OR L         .00000      .30000       -.43702       -.13013
                             Z OR M         .00000     3.00000       6.75000        .83028
 2   7                       X OR K         .10000      .00000      -2.08877       -.54192
                             Y OR L         .00000      .30000       -.63295       -.13013
                             Z OR M         .00000     3.00000       8.00000        .83028
 2     NO INT W/OPT AX

 3   1                       X OR K         .10000     -.30000       -.31159        .01673
                             Y OR L         .00000      .30000        .30869       -.03736
                             Z OR M         .00000     3.00000       3.08697        .99916
 3   2                       X OR K         .10000     -.30000       -.30660        .28045
                             Y OR L         .00000      .30000        .29755       -.30434
                             Z OR M         .00000     3.00000       3.38486        .91034
 3   3                       X OR K         .10000     -.30000       -.01229       -.21390
                             Y OR L         .00000      .30000       -.02183       -.19021
                             Z OR M         .00000     3.00000       4.34020        .95815
 3   4                       X OR K         .10000     -.30000       -.14898       -.34225
                             Y OR L         .00000      .30000       -.14338       -.30434
                             Z OR M         .00000     3.00000       4.95250        .88895
 3   5                       X OR K         .10000     -.30000       -.58127       -.22816
                             Y OR L         .00000      .30000       -.52780       -.08569
                             Z OR M         .00000     3.00000       6.07531        .96984
 3   6                       X OR K         .10000     -.30000       -.74000       -.34225
```

```
                      Y OR L       .00000      .30000     -.58741     -.12854
                      Z OR M       .00000     3.00000     6.75000      .93077
3  7                  X OR K       .10000     -.30000    -1.19964     -.34225
                      Y OR L       .00000      .30000     -.76005     -.12854
                      Z OR M       .00000     3.00000     8.00000      .93077
3    NO INT W/OPT AX

4  1                  X OR K       .00000      .30000      .30860     -.03614
                      Y OR L       .00000      .30000      .30860     -.03614
                      Z OR M       .00000     3.00000     3.08608      .99869
4  2                  X OR K       .00000      .30000      .29766     -.30412
                      Y OR L       .00000      .30000      .29766     -.30412
                      Z OR M       .00000     3.00000     3.38849      .90278
4  3                  X OR K       .00000      .30000     -.02026     -.48145
                      Y OR L       .00000      .30000     -.02026     -.19007
                      Z OR M       .00000     3.00000     4.35223      .85561
4  4                  X OR K       .00000      .30000     -.36928     -.77032
                      Y OR L       .00000      .30000     -.15805     -.30412
                      Z OR M       .00000     3.00000     4.95250      .56045
4  5 APERTURE STOP
5  1                  X OR K       .00000      .00000      .00000      .00000
                      Y OR L       .00000      .30000      .30409     -.02964
                      Z OR M       .00000     3.00000     3.04095      .99956
5  2                  X OR K       .00000      .00000      .00000      .00000
                      Y OR L       .00000      .30000      .29201     -.24751
                      Z OR M       .00000     3.00000     3.44828      .96888
5  3                  X OR K       .00000      .00000      .00000     -.32679
                      Y OR L       .00000      .30000      .06101     -.15469
                      Z OR M       .00000     3.00000     4.35250      .93234
5  4                  X OR K       .00000      .00000     -.21030     -.52287
                      Y OR L       .00000      .30000     -.03853     -.24751
                      Z OR M       .00000     3.00000     4.95250      .81568
5  5                  X OR K       .00000      .00000     -.90414     -.34858
                      Y OR L       .00000      .30000     -.36698     -.08821
                      Z OR M       .00000     3.00000     6.03488      .93311
5  6                  X OR K       .00000      .00000    -1.17128     -.52287
                      Y OR L       .00000      .30000     -.43459     -.13232
                      Z OR M       .00000     3.00000     6.75000      .84207
5  7                  X OR K       .00000      .00000    -1.94745     -.52287
                      Y OR L       .00000      .30000     -.63101     -.13232
                      Z OR M       .00000     3.00000     8.00000      .84207
5    NO INT W/OPT AX

6  1                  X OR K       .00000     -.30000     -.30860      .03614
                      Y OR L       .00000      .30000      .30860     -.03614
                      Z OR M       .00000     3.00000     3.08608      .99869
6  2                  X OR K       .00000     -.30000     -.29766      .30412
                      Y OR L       .00000      .30000      .29766     -.30412
                      Z OR M       .00000     3.00000     3.38849      .90278
6  3                  X OR K       .00000     -.30000      .04085     -.20566
                      Y OR L       .00000      .30000     -.04085     -.19007
                      Z OR M       .00000     3.00000     4.39335      .95998
6  4                  X OR K       .00000     -.30000     -.07893     -.32906
                      Y OR L       .00000      .30000     -.15156     -.30412
                      Z OR M       .00000     3.00000     4.95250      .89399
6  5                  X OR K       .00000     -.30000     -.49297     -.21937
                      Y OR L       .00000      .30000     -.53421     -.08393
```

```
                     Z OR M      .00000    3.00000    6.07732     .97202
6  6                 X OR K      .00000    -.30000    -.64479    -.32906
                     Y OR L      .00000     .30000    -.59230    -.12590
                     Z OR M      .00000    3.00000    6.75000     .93587
6  7                 X OR K      .00000    -.30000   -1.08431    -.32906
                     Y OR L      .00000     .30000    -.76046    -.12590
                     Z OR M      .00000    3.00000    8.00000     .93587
6    NO INT W/OPT AX

7  1                 X OR K     -.10000     .30000     .31159    -.01673
                     Y OR L      .00000     .30000     .30869    -.03736
                     Z OR M      .00000    3.00000    3.08697     .99916
7  2                 X OR K     -.10000     .30000     .30660    -.28045
                     Y OR L      .00000     .30000     .29755    -.30434
                     Z OR M      .00000    3.00000    3.38486     .91034
7  3                 X OR K     -.10000     .30000     .00650    -.46886
                     Y OR L      .00000     .30000    -.02811    -.19021
                     Z OR M      .00000    3.00000    4.35900     .86254
7  4                 X OR K     -.10000     .30000    -.31611    -.75018
                     Y OR L      .00000     .30000    -.15900    -.30434
                     Z OR M      .00000    3.00000    4.95250     .58702
7  5                 X OR K     -.10000     .30000   -1.91805    -.50012
                     Y OR L      .00000     .30000    -.80890     .11802
                     Z OR M      .00000    3.00000    6.20602     .85787
7  6 APERTURE STOP
8  1                 X OR K     -.10000     .00000     .00136     .02133
                     Y OR L      .00000     .30000     .30409    -.02971
                     Z OR M      .00000    3.00000    3.04095     .99933
8  2                 X OR K     -.10000     .00000     .01006     .02552
                     Y OR L      .00000     .30000     .29198    -.24762
                     Z OR M      .00000    3.00000    3.44823     .96851
8  3                 X OR K     -.10000     .00000     .03481    -.31508
                     Y OR L      .00000     .30000     .05188    -.15476
                     Z OR M      .00000    3.00000    4.38731     .93636
8  4                 X OR K     -.10000     .00000    -.15537    -.50413
                     Y OR L      .00000     .30000    -.04152    -.24762
                     Z OR M      .00000    3.00000    4.95250     .82736
8  5                 X OR K     -.10000     .00000    -.81469    -.33608
                     Y OR L      .00000     .30000    -.36538    -.08932
                     Z OR M      .00000    3.00000    6.03457     .93758
8  6                 X OR K     -.10000     .00000   -1.07115    -.50413
                     Y OR L      .00000     .30000    -.43354    -.13398
                     Z OR M      .00000    3.00000    6.75000     .85317
8  7                 X OR K     -.10000     .00000   -1.80976    -.50413
                     Y OR L      .00000     .30000    -.62984    -.13998
                     Z OR M      .00000    3.00000    8.00000     .85317
8    NO INT W/OPT AX

9  1                 X OR K     -.10000    -.30000    -.30568     .05589
                     Y OR L      .00000     .30000     .30852    -.03509
                     Z OR M      .00000    3.00000    3.08521     .99781
9  2                 X OR K     -.10000    -.30000    -.28848     .32813
                     Y OR L      .00000     .30000     .29772    -.30423
                     Z OR M      .00000    3.00000    3.39209     .89429
9  3                 X OR K     -.10000    -.30000     .10094    -.19763
                     Y OR L      .00000     .30000    -.06333    -.19014
                     Z OR M      .00000    3.00000    4.45344     .96165
```

```
 9  4                          X OR K       -.10000       -.30000       -.00162       -.31621
                               Y OR L        .00000        .30000       -.16200       -.30423
                               Z OR M        .00000       3.00000       4.95250        .89858
 9  5                          X OR K       -.10000       -.30000       -.39854       -.21081
                               Y OR L        .00000        .30000       -.54388       -.08126
                               Z OR M        .00000       3.00000       6.08042        .97414
 9  6                          X OR K       -.10000       -.30000       -.54344       -.31621
                               Y OR L        .00000        .30000       -.59974       -.12190
                               Z OR M        .00000       3.00000       6.75000        .94082
 9  7                          X OR K       -.10000       -.30000       -.96357       -.31621
                               Y OR L        .00000        .30000       -.76170       -.12190
                               Z OR M        .00000       3.00000       8.00000        .94082
 9    NO INT W/OPT AX

10  1                          X OR K        .10000        .30000        .30270       -.05012
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       3.04057        .99874
10  2                          X OR K        .10000        .30000        .28206       -.27290
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       3.45185        .96204
10  3                          X OR K        .10000        .30000        .02070       -.45816
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       4.37320        .88886
10  4                          X OR K        .10000        .30000       -.27789       -.73305
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       4.95250        .68016
10  5                          X OR K        .10000        .30000      -1.40684       -.48870
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       6.00000        .87244
10  6                          X OR K        .10000        .30000      -1.82695       -.73305
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       6.75000        .68016
10  7                          X OR K        .10000        .30000      -3.17415       -.73305
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       8.00000        .68016
10    OPT AX INTERSECTION Z =    5.05485

11  1                          X OR K        .10000        .00000        .00000       -.02187
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       3.00000        .99976
11  2                          X OR K        .10000        .00000       -.01093       -.02658
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       3.49992        .99964
11  3                          X OR K        .10000        .00000       -.03274       -.33430
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       4.31975        .94246
11  4                          X OR K        .10000        .00000       -.25718       -.53488
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       4.95250        .84492
11  5                          X OR K        .10000        .00000       -.92031       -.35659
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       6.00000        .93426
11  6                          X OR K        .10000        .00000      -1.20657       -.53488
                               Y OR L        .00000        .00000        .00000        .00000
                               Z OR M        .00000       3.00000       6.75000        .84492
11  7                          X OR K        .10000        .00000      -1.99789       -.53488
                               Y OR L        .00000        .00000        .00000        .00000
```

```
                          Z OR M      .00000     3.00000     8.00000      .84492
11    OPT AX INTERSECTION Z =       4.84405

12  1                     X OR K      .10000     -.10000     -.30591      .00947
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     3.04134      .99995
12  2                     X OR K      .10000     -.30000     -.30168      .22237
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     3.44468      .97496
12  3                     X OR K      .10000     -.30000     -.12259     -.22583
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     4.22990      .97416
12  4                     X OR K      .10000     -.30000     -.29010     -.36133
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     4.95250      .93243
12  5                     X OR K      .10000     -.30000     -.69602     -.24088
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     6.00000      .97059
12  6                     X OR K      .10000     -.30000     -.88216     -.36133
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     6.75000      .93243
12  7                     X OR K      .10000     -.30000    -1.36655     -.36133
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     8.00000      .93243
12    OPT AX INTERSECTION Z =       4.47350

13  1                     X OR K      .00000      .30000      .30409     -.02964
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     3.04095      .99956
13  2                     X OR K      .00000      .30000      .29201     -.24751
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     3.44828      .96888
13  3                     X OR K      .00000      .30000      .04060     -.44478
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     4.40110      .89563
13  4                     X OR K      .00000      .30000     -.22523     -.71165
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     4.95250      .70252
13  5                     X OR K      .00000      .30000    -1.20633     -.47443
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     6.00000      .88028
13  6                     X OR K      .00000      .30000    -1.69055     -.71165
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     6.75000      .70252
13  7                     X OR K      .00000      .30000    -2.95679     -.71165
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     8.00000      .70252
13    OPT AX INTERSECTION Z =       5.08112

14  1                     X OR K      .00000      .00000      .00000      .00000
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     3.00000     1.00000
14  2                     X OR K      .00000      .00000      .00000      .00000
                          Y OR L      .00000      .00000      .00000      .00000
                          Z OR M      .00000     3.00000     3.50000     1.00000
14  3                     X OR K      .00000      .00000     -.00000     -.32100
                          Y OR L      .00000      .00000      .00000      .00000
```

```
                          Z OR M      .00000      3.00000      4.35250      .94680
14   4                    X OR K      .00000       .00000      -.20393     -.51488
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      4.95250      .85725
14   5                    X OR K      .00000       .00000      -.83508     -.34325
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      6.00000      .93924
14   6                    X OR K      .00000       .00000     -1.10718     -.51488
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      6.75000      .85725
14   7                    X OR K      .00000       .00000     -1.85796     -.51488
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      8.00000      .85725
14     OPT AX INTERSECTION Z =     4.90661

15   1                    X OR K      .00000      -.30000      -.30409      .02964
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      3.04095      .99956
15   2                    X OR K      .00000      -.30000      -.29201      .24751
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      3.44828      .96888
15   3                    X OR K      .00000      -.30000      -.08195     -.21611
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      4.27054      .97636
15   4                    X OR K      .00000      -.30000      -.23290     -.34578
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      4.95250      .93831
15   5                    X OR K      .00000      -.30000      -.61892     -.23052
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      6.00000      .97306
15   6                    X OR K      .00000      -.30000      -.79660     -.34578
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      6.75000      .93831
15   7                    X OR K      .00000      -.30000     -1.25725     -.34578
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      8.00000      .93831
15     OPT AX INTERSECTION Z =     4.58836

16   1                    X OR K     -.10000       .30000       .30551     -.00947
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      3.04134      .99995
16   2                    X OR K     -.10000       .30000       .30168     -.22237
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      3.44460      .97496
16   3                    X OR K     -.10000       .30000       .07705     -.43166
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      4.42955      .90203
16   4                    X OR K     -.10000       .30000      -.17519     -.69066
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      4.95250      .72317
16   5                    X OR K     -.10000       .30000     -1.17361     -.46044
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      6.00000      .88760
16   6                    X OR K     -.10000       .30000     -1.56263     -.69066
                          Y OR L      .00000       .00000       .00000      .00000
                          Z OR M      .00000      3.00000      6.75000      .72317
16   7                    X OR K     -.10000       .30000     -2.75644     -.69066
```

```
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       8.00000        .72317
16    OPT AX INTERSECTION Z =          5.11381

17   1                    X OR K      -.10000         .00000        .00000        .02187
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       3.00000        .99976
17   2                    X OR K      -.10000         .00000        .01093        .02658
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       3.49992        .99964
17   3                    X OR K      -.10000         .00000        .03453      -.30950
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       4.38703        .95089
17   4                    X OR K      -.10000         .00000      -.14951       -.49520
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       4.95250        .86877
17   5                    X OR K      -.10000         .00000      -.74659       -.33013
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       6.00000        .94393
17   6                    X OR K      -.10000         .00000     -1.00890       -.49520
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       6.75000        .86877
17   7                    X OR K      -.10000         .00000     -1.72140       -.49520
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       8.00000        .86877
17    OPT AX INTERSECTION Z =          4.97999

18   1                    X OR K      -.10000       -.30000       -.30270        .05012
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       3.04057        .99874
18   2                    X OR K      -.10000       -.30000       -.28206        .27290
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       3.45105        .96204
18   3                    X OR K      -.10000       -.30000       -.03709       -.20658
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       4.31540        .97842
18   4                    X OR K      -.10000       -.30000       -.17161       -.33053
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       4.95250        .94379
18   5                    X OR K      -.10000       -.30000       -.53847       -.22035
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       6.00000        .97541
18   6                    X OR K      -.10000       -.30000       -.70791       -.33053
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       6.75000        .94379
18   7                    X OR K      -.10000       -.30000      -1.14569       -.33053
                          Y OR L        .00000        .00000        .00000        .00000
                          Z OR M        .00000       3.00000       8.00000        .94379
18    OPT AX INTERSECTION Z =          4.72868

19   1                    X OR K        .10000        .30000        .30568      -.05589
                          Y OR L        .00000      -.30000       -.30852        .03509
                          Z OR M        .00000       3.00000       3.08521        .99781
19   2                    X OR K        .10000        .30000        .28848      -.32013
                          Y OR L        .00000      -.30000       -.29772        .30423
                          Z OR M        .00000       3.00000       3.39289        .89429
19   3                    X OR K        .10000        .30000       -.04675      -.49435
```

```
                         Y OR L     .00000    -.30000      .01308     .19014
                         Z OR M     .00000    3.00000     4.30574     .84820
19   4                   X OR K     .10000     .30000     -.42369    -.79097
                         Y OR L     .00000    -.30000      .15807     .30423
                         Z OR M     .00000    3.00000     4.95250     .53085
19   5 NO INCIDENCE
20   1                   X OR K     .10000     .00000     -.00136    -.02133
                         Y OR L     .00000    -.30000     -.30409     .02971
                         Z OR M     .00000    3.00000     3.04095     .99933
20   2                   X OR K     .10000     .00000     -.01006    -.02552
                         Y OR L     .00000    -.30000     -.29198     .24762
                         Z OR M     .00000    3.00000     3.44823     .96851
20   3                   X OR K     .10000     .00000     -.03302    -.33870
                         Y OR L     .00000    -.30000     -.06923     .15476
                         Z OR M     .00000    3.00000     4.31947     .92807
20   4                   X OR K     .10000     .00000     -.26404    -.54192
                         Y OR L     .00000    -.30000      .03633     .24762
                         Z OR M     .00000    3.00000     4.95250     .80311
20   5                   X OR K     .10000     .00000     -.99486    -.36128
                         Y OR L     .00000    -.30000      .37026     .08675
                         Z OR M     .00000    3.00000     6.03553     .92840
20   6                   X OR K     .10000     .00000    -1.27289    -.54192
                         Y OR L     .00000    -.30000      .43702     .13013
                         Z OR M     .00000    3.00000     6.75000     .83028
20   7                   X OR K     .10000     .00000    -2.08877    -.54192
                         Y OR L     .00000    -.30000      .63295     .13013
                         Z OR M     .00000    3.00000     8.00000     .83028
20     NO INT W/OPT AX

21   1                   X OR K     .10000    -.30000     -.31159     .01673
                         Y OR L     .00000    -.30000     -.30869     .03736
                         Z OR M     .00000    3.00000     3.08697     .99916
21   2                   X OR K     .10000    -.30000     -.30660     .28045
                         Y OR L     .00000    -.30000     -.29755     .30434
                         Z OR M     .00000    3.00000     3.38486     .91034
21   3                   X OR K     .10000    -.30000     -.01229    -.21390
                         Y OR L     .00000    -.30000      .02183     .19021
                         Z OR M     .00000    3.00000     4.34020     .95815
21   4                   X OR K     .10000    -.30000     -.14898    -.34225
                         Y OR L     .00000    -.30000      .14338     .30434
                         Z OR M     .00000    3.00000     4.95250     .88895
21   5                   X OR K     .10000    -.30000     -.58127    -.22816
                         Y OR L     .00000    -.30000      .52780     .08569
                         Z OR M     .00000    3.00000     6.07531     .96984
21   6                   X OR K     .10000    -.30000     -.74000    -.34225
                         Y OR L     .00000    -.30000      .58741     .12854
                         Z OR M     .00000    3.00000     6.75000     .93077
21   7                   X OR K     .10000    -.30000    -1.19964    -.34225
                         Y OR L     .00000    -.30000      .76005     .12854
                         Z OR M     .00000    3.00000     8.00000     .93077
21     NO INT W/OPT AX

22   1                   X OR K     .00000     .30000      .30860    -.03614
                         Y OR L     .00000    -.30000     -.30860     .03614
                         Z OR M     .00000    3.00000     3.08608     .99869
22   2                   X OR K     .00000     .30000      .29766    -.30412
                         Y OR L     .00000    -.30000     -.29766     .30412
```

```
                       Z OR M     .00000     3.00000     3.38849     .90278
22  3                  X OR K     .00000     -.30000     -.02026    -.48145
                       Y OR L     .00000     -.30000      .02026     .19007
                       Z OR M     .00000     3.00000     4.33223     .85561
22  4                  X OR K     .00000     -.30000     -.36928    -.77032
                       Y OR L     .00000     -.30000      .15805     .30412
                       Z OR M     .00000     3.00000     4.95250     .56045
22  5 APERTURE STOP
23  1                  X OR K     .00000      .00000      .00000     .00000
                       Y OR L     .00000     -.30000     -.30409     .02964
                       Z OR M     .00000     3.00000     3.04095     .99956
23  2                  X OR K     .00000      .00000      .00000     .00000
                       Y OR L     .00000     -.30000     -.29201     .24751
                       Z OR M     .00000     3.00000     3.44828     .96888
23  3                  X OR K     .00000      .00000      .00000    -.32679
                       Y OR L     .00000     -.30000     -.06101     .15469
                       Z OR M     .00000     3.00000     4.35250     .93234
23  4                  X OR K     .00000      .00000     -.21030    -.52287
                       Y OR L     .00000     -.30000      .03853     .24751
                       Z OR M     .00000     3.00000     4.95250     .81568
23  5                  X OR K     .00000      .00000     -.90414    -.34858
                       Y OR L     .00000     -.30000      .36698     .08821
                       Z OR M     .00000     3.00000     6.03488     .93311
23  6                  X OR K     .00000      .00000    -1.17128    -.52287
                       Y OR L     .00000     -.30000      .43459     .13232
                       Z OR M     .00000     3.00000     6.75000     .84207
23  7                  X OR K     .00000      .00000    -1.94745    -.52287
                       Y OR L     .00000     -.30000      .63101     .13232
                       Z OR M     .00000     3.00000     8.00000     .84207
23    NO INT W/OPT AX

24  1                  X OR K     .00000     -.30000     -.30860     .03614
                       Y OR L     .00000     -.30000     -.30860     .03614
                       Z OR M     .00000     3.00000     3.08608     .99869
24  2                  X OR K     .00000     -.30000     -.29766     .30412
                       Y OR L     .00000     -.30000     -.29766     .30412
                       Z OR M     .00000     3.00000     3.38849     .90278
24  3                  X OR K     .00000     -.30000      .04085    -.20566
                       Y OR L     .00000     -.30000      .04085     .19007
                       Z OR M     .00000     3.00000     4.39335     .95998
24  4                  X OR K     .00000     -.30000     -.07893    -.32906
                       Y OR L     .00000     -.30000      .15156     .30412
                       Z OR M     .00000     3.00000     4.95250     .89399
24  5                  X OR K     .00000     -.30000     -.49297    -.21937
                       Y OR L     .00000     -.30000      .53421     .08393
                       Z OR M     .00000     3.00000     6.07732     .97202
24  6                  X OR K     .00000     -.30000     -.64479    -.32906
                       Y OR L     .00000     -.30000      .59230     .12590
                       Z OR M     .00000     3.00000     6.75000     .93587
24  7                  X OR K     .00000     -.30000    -1.08431    -.32906
                       Y OR L     .00000     -.30000      .76046     .12590
                       Z OR M     .00000     3.00000     8.00000     .93587
24    NO INT W/OPT AX

25  1                  X OR K     .10000      .30000      .31159    -.01673
                       Y OR L     .00000     -.30000     -.30869     .03736
                       Z OR M     .00000     3.00000     3.08697     .99916
```

```
25  2                      X OR K    -.10000     .30000     .30660    -.28045
                           Y OR L     .00000    -.30000    -.29755     .30434
                           Z OR M     .00000    3.00000    3.38486     .91034
25  3                      X OR K    -.10000     .30000     .00650    -.46886
                           Y OR L     .00000    -.30000     .02811     .19021
                           Z OR M     .00000    3.00000    4.35900     .86254
25  4                      X OR K    -.10000     .30000    -.31611    -.75018
                           Y OR L     .00000    -.30000     .15900     .30434
                           Z OR M     .00000    3.00000    4.95250     .58702
25  5                      X OR K    -.10000     .30000   -1.91805    -.50012
                           Y OR L     .00000    -.30000     .80890    -.11802
                           Z OR M     .00000    3.00000    6.20602     .85787
25  6 APERTURE STOP
26  1                      X OR K    -.10000     .00000     .00136     .02133
                           Y OR L     .00000    -.30000    -.30409     .02971
                           Z OR M     .00000    3.00000    3.04095     .99935
26  2                      X OR K    -.10000     .00000     .01006     .02552
                           Y OR L     .00000    -.30000    -.29198     .24762
                           Z OR M     .00000    3.00000    3.44823     .96851
26  3                      X OR K    -.10000     .00000     .03481    -.31508
                           Y OR L     .00000    -.30000    -.05188     .15476
                           Z OR M     .00000    3.00000    4.38731     .93636
26  4                      X OR K    -.10000     .00000    -.15537    -.50413
                           Y OR L     .00000    -.30000     .04152     .24762
                           Z OR M     .00000    3.00000    4.95250     .82736
26  5                      X OR K    -.10000     .00000    -.81469    -.33608
                           Y OR L     .00000    -.30000     .36538     .08932
                           Z OR M     .00000    3.00000    6.03457     .93758
26  6                      X OR K    -.10000     .00000   -1.07115    -.50413
                           Y OR L     .00000    -.30000     .43354     .13398
                           Z OR M     .00000    3.00000    6.75000     .85317
26  7                      X OR K    -.10000     .00000   -1.80976    -.50413
                           Y OR L     .00000    -.30000     .62984     .13398
                           Z OR M     .00000    3.00000    8.00000     .85317
26    NO INT W/OPT AX

27  1                      X OR K    -.10000    -.30000    -.30568     .05589
                           Y OR L     .00000    -.30000    -.30852     .03509
                           Z OR M     .00000    3.00000    3.08521     .99781
27  2                      X OR K    -.10000    -.30000    -.28848     .32813
                           Y OR L     .00000    -.30000    -.29772     .30423
                           Z OR M     .00000    3.00000    3.39209     .89429
27  3                      X OR K    -.10000    -.30000     .10094    -.19763
                           Y OR L     .00000    -.30000     .06333     .19014
                           Z OR M     .00000    3.00000    4.45344     .96165
27  4                      X OR K    -.10000    -.30000    -.00162    -.31621
                           Y OR L     .00000    -.30000     .16200     .30423
                           Z OR M     .00000    3.00000    4.95250     .89858
27  5                      X OR K    -.10000    -.30000    -.39854    -.21081
                           Y OR L     .00000    -.30000     .54388     .08126
                           Z OR M     .00000    3.00000    6.08042     .97414
27  6                      X OR K    -.10000    -.30000    -.94344    -.31621
                           Y OR L     .00000    -.30000     .59974     .12190
                           Z OR M     .00000    3.00000    6.75000     .94082
27  7                      X OR K    -.10000    -.30000    -.96357    -.31621
                           Y OR L     .00000    -.30000     .76170     .12190
                           Z OR M     .00000    3.00000    8.00000     .94082
```

27 NO INT W/OPT AX

```
C     GENERAL RAY TRACE PROGRAM                             CARD I/O
      DIMENSION XO(35),YO(35),ZO(35),ALPHA(35),BETA(35),GAMMA(35),A(35)
      DIMENSION REFR(36),AP1(35),AP2(35),AP3(35),AP4(35),AP5(35),AP6(35)
      DIMENSION NAP(35),NOUT(35),CKBAR(35),CLBAR(35),CMBAR(35),X0BAR(35)
      DIMENSION Y0BAR(35),Z0BAR(35),B(35),C(35),D(35),E(35)
      DIMENSION  R(35),KSUR(35)
      KKK = 1
      GO TO (813, 803), KKK
  803 PRINT 115
  115 FORMAT (20HLOAD DATA,PUSH START)
      PAUSE
  813 KKK = 2
C     READ TITLE CARD
      READ 2
2FORMAT(49H                                                 ,8H        )
      PUNCH 2
      PUNCH 100
      JB = 50
      PUNCH 1
      PUNCH 100
      PUNCH 100
      PI = 3.14159265
      CONV = PI/180.
C     READ HEADER CARD
  102 FORMAT (F11.5,I3,E15.8)
      READ 102, REFR(1),NOSUR,TOL
C     REFR = INITIAL INDEX OF REFR  , NOSUR = NO. OF SURFACES
C     TOL = MIN. LIMIT ON ITERATION INCREMENT
      NOSUR = NOSUR+1
```

```
      DO 203 I = 2, NOSUR
C     READ SURFACE DATA
  103 FORMAT (F11.5, F11.5, F11.5, F11.5, F11.5, F11.5,I2,I2,I3)
      READ 103, XO(I),YO(I),ZO(I),ALPHA(I),BETA(I),GAMMA(I)
C     XO, YO, AND ZO ARE LOCAL SYSTEM SURFACE VERTEX COORDINATES.
C     ALPHA, BETA, AND GAMMA ARE EULER ROTATIONAL ANGLES.
      READ 103,A(I),B(I),C(I),D(I),E(I),REFR(I)
C     A, B, C, D, AND E ARE SURFACE EQUATION COEFFICIENTS.
C     REFR = INDEX OF REFRACTION FOLLOWING SURFACE.
  203 READ 103,AP1(I),AP2(I),AP3(I),AP4(I),AP5(I),AP6(I),NAP(I),NOUT(I)
C     AP1, AP2, AP3, AP4, AP5, AND AP6 SPECIFY APERTURE DIMENSIONS. SEE
C             INPUT WRITE UP.
C     NAP = 0 FOR CIRCULAR-ANNULAR APERTURE, = 1 FOR RECT-TRAPEZOIDAL
C        APERTURE, = 2 FOR HYPERBOLIC APERTURE.
C     NOUT = 0 FOR NO OUTPUT AT SURFACE, = 1 OTHERWISE.
C     READ RAY DATA
C     SWITCH 1 ON TO USE SINGLE RAY CARD INPUT
      IF(SENSE SWITCH 1) 205, 201
  205 READ 103, XA,YA,ZA,RPAR1,RPAR2,RPAR3,NIN,NAXIN,IRAY
C     XA, YA, AND ZA ARE RAY COORDINATES AT FIRST POINT
C     RPAR1, RPAR2, RPAR3 ARE RAY COORDINATES AT SECOND POINT
C     OR RAY DIRECTION COSINES AT FIRST POINT
C     NIN = 0 IF RAYS ARE SPEC. BY 2 POINTS
C     NIN = 1 IF RAYS ARE SPEC. BY 1 POINT AND DIR. COSINES.
C     NAXIN = 1 FOR COMPUTATION OF INTERSECTION OF RAY WITH OPTICAL
C         AXIS, = 0 OTHERWISE IRAY = RAY NUMBER.
      GO TO 206
  201 READ 116, FXU1, FZ1, XGAP1, FXU2, FZ2, XGAP2
      READ 116, FYU2, YGAP2
```

```
C      RAY INPUT DATA, FXU1 = MAX.XR, FZ1 = ZR, FXU2 = MAX. X AT POINT 2
C      XGAP1 = X SPACING AT POINT 1, XGAP2 = X SPACING AT POINT 2
C      FZ2 = Z COORDINATE AT POINT 2
C      FYU2 = MAX Y COORDINATE AT 2ND POINT, YGAP2 = Y SPACING AT POINT 2
   116 FORMAT(F11.5,F11.5,F11.5,F11.5,F11.5,F11.5)
       IRAY = 0
       GAP3=0.
       RPAR2=FYU2
   812 RPAR2=RPAR2-GAP3
       IF(RPAR2+FYU2)803,814,814
   814 GAP3=YGAP2
       GAP1 = 0.
       XA = FXU1
     4 XA = XA-GAP1
       IF(XA+FXU1)812,804,804
   804 GAP1 = XGAP1
       GAP2 = 0.
       RPAR1 = FXU2
     3 RPAR1 = RPAR1-GAP2
       IF(RPAR1+FXU2)4,805,805
   805 GAP2 = XGAP2
       YA = 0.0
       ZA = FZ1
       RPAR3 = FZ2
       NIN = 0
       NAXIN = 1
       IRAY = IRAY+1
   206 NIN = NIN+1
       GO TO (207,208), NIN
```

```
208 CKBAR(1) = RPAR1
    CLBAR(1) = RPAR2
    CMBAR(1) = RPAR3
    GO TO 209
207 XD = RPAR1-XA
    YD = RPAR2-YA
    ZD = RPAR3-ZA
    RALEN = SQRT(XD*XD+YD*YD+ZD*ZD)
    CKBAR(1) = XD/RALEN
    CLBAR(1) = YD/RALEN
    CMBAR(1) = ZD/RALEN
 209 XØBAR(1) = XA
    YØBAR(1) = YA
    ZØBAR(1) = ZA
    DO 15  I = 2, NOSUR
    IM1 = I-1
    SA = SIN(CONV*ALPHA(I))
    CA = COS(CONV*ALPHA(I))
    SB = SIN(CONV*BETA(I))
    CB = COS(CONV*BETA(I))
    SG = SIN(CONV*GAMMA(I))
    CG = COS(CONV*GAMMA(I))
    X1 = XØBAR(IM1)-XO(I)
    Y1 = YØBAR(IM1)-YO(I)
    Z1 = ZØBAR(IM1)-ZO(I)
    R11 = CA*CG+SA*SB*SG
    R12 = -CB*SG
    R13 = CA*SB*SG-SA*CG
    R21 = CA*SG-SA*SB*CG
```

```
      R22 = CB*CG
      R23 = -(SA*SG+CA*SB*CG)
      R31 = SA*CB
      R32 = SB
      R33 = CA*CB
      X0 = X1*R11+Y1*R12+Z1*R13
      Y0 = X1*R21+Y1*R22+Z1*R23
      Z0 = X1*R31+Y1*R32+Z1*R33
      CK = CKBAR(IM1)*R11+CLBAR(IM1)*R12+CMBAR(IM1)*R13
      CL = CKBAR(IM1)*R21+CLBAR(IM1)*R22+CMBAR(IM1)*R23
      CM = CKBAR(IM1)*R31+CLBAR(IM1)*R32+CMBAR(IM1)*R33
      J = 0
      IF(D(I)) 310,311,302
  310 Z2 = -Z0
      IF(CM) 600,10,600
  600 X0 = X0-2.*CK*Z0/CM
      Y0 = Y0-2.*CL*Z0/CM
      GO TO 211
  311 IF(CM) 210,302,210
  210 X0 = X0-CK*Z0/CM
      Y0 = Y0-CL*Z0/CM
      Z2 = 0.
      GO TO 211
  302 Z2 = Z0
  211 S = 0.
    5 J = J+1
      X = X0+CK*S
      Y = Y0+CL*S
      Z = Z2+CM*S
```

```
      IF(D(I)) 212,213,214
  212 IF (Z) 7,6,6
    7 Z = -Z
      IF (CM) 400  10, 400
  400 X = X0+CK*(Z-Z2)/CM
      Y = Y0+CL*.Z-Z2)/CM
    6 IF(Y) 215,216,216
  215 Y = -Y
      IF(CL) 401, 10, 401
  401 X = X0+CK*(Y-Y0)/CL
      Z = Z2+CM*(Y-Y0)/CL
  216 DD = D(I)*D(I)
      F = E(I)*DD*X*X+DD*Y*Y-Z*Z
      FX = 2.*X*E(I)*DD
      FY = 2.*Y*DD
      FZ = -2.*Z
      GO TO 8
  214 IF(Z) 6,6,7
  213 F = A(I)*X*X+B(I)*Y*Y+C(I)*Z*Z+Z
      FX = 2.*A(I)*X
      FY = 2.*B(I)*Y
      FZ = 2.*C(I)*Z+1.
    8 DETMT = CK*FX+CL*FY+CM*FZ
      IF(DETMT) 218,217,218
  217 IF(F) 10,9,10
   10 PUNCH 104, IRAY, IM1
   11 CONTINUE
      IF (SENSE SWITCH 1)205,3
  218 DELS = -F/DETMT
```

```
      DELS2 = DELS*DELS
      TOL2 = TOL*TOL
      IF(DELS2-TOL2) 9, 9, 219
  219 IF(J-JB) 220,10,10
  220 S = S+DELS
      GO TO 5
    9 IF(D(I)) 320,321,301
  320 IF(S-2.*Z0/CM) 221, 222, 222
  321 IF(CM) 300,301,300
  301 IF(S) 221,222,222
  300 IF(S-Z0/CM) 221,222,222
  221 PUNCH 105, IRAY, IM1
      GO TO 11
  222 KAP = NAP(I)+1
      GO TO (223,224,225),KAP
  223 RHSQ = (X-AP1(I))*(X-AP1(I))+(Y-AP2(I))*(Y-AP2(I))
      IF(RHSQ-AP3(I)*AP3(I)) 12, 226, 226
   12 PUNCH 106, IRAY, IM1
      GO TO 11
  226 IF(RHSQ-AP4(I)*AP4(I)) 13, 12, 12
  224 IF(Y-AP3(I)) 12,12,227
  227 IF(Y-AP4(I)) 228,12,12
  228 IF(X-(AP5(I)*(Y-AP3(I))+AP1(I))) 12,12,229
  229 IF(X-(AP6(I)*(Y-AP3(I))+AP2(I))) 13,12,12
  225 IF(Y-AP5(I)) 12,12,230
  230 IF(Y-AP6(I)) 231,12,12
  231 FFF = (X-AP1(I))/AP3(I)
      FFF2 = FFF*FFF
      GGG = (Y-AP2(I))/AP4(I)
```

```
      GGG2 = GGG*GGG
      IF(FFF2-GGG2-1.) 13, 12, 12
   13 RAT = REFR(IM1)/REFR(I)
      ALC = (RAT*DETMT)/(FX*FX+FY*FY+FZ*FZ)
      IF (REFR(I)+REFR(IM1)) 232,233,232
  233 GAMUC = 2.*ALC
      RAT = 1.
      GO TO 14
  232 BLC = (RAT*RAT-1.)/(FX*FX+FY*FY+FZ*FZ)
      DISC = ALC*ALC-BLC
      IF(DISC) 234,235,235
  234 PUNCH 107, IRAY, IM1
      GO TO 11
  235 DISC = SQRT(DISC)
      IF(ALC) 236,10,237
  236 DISC = -DISC
  237 GAMUC = DISC-ALC
   14 CK = RAT*CK+GAMUC*FX
      CL = RAT*CL+GAMUC*FY
      CM = RAT*CM+GAMUC*FZ
      X0BAR(I) = R11*X+R21*Y+R31*Z+XO(I)
      Y0BAR(I) = R12*X+R22*Y+R32*Z+YO(I)
      Z0BAR(I) = R13*X+R23*Y+R33*Z+ZO(I)
      CKBAR(I) = R11*CK+R21*CL+R31*CM
      CLBAR(I) = R12*CK+R22*CL+R32*CM
      CMBAR(I) = R13*CK+R23*CL+R33*CM
      KOUT = NOUT(I)+1
      GO TO (15,16),KOUT
   16 PUNCH 108,IRAY,IM1,XA,RPAR1,X0BAR(I),CKBAR(I)
```

```
      PUNCH 109,YA,RPAR2,Y0BAR(I),CLBAR(I)
      PUNCH 110,ZA,RPAR3,Z0BAR(I),CMBAR(I)
   15 CONTINUE
      NAXIN = NAXIN+1
      GO TO (17,239),NAXIN
  239 IF(CKBAR(NOSUR)) 240,241,240
  241 IF(X0BAR(NOSUR)) 18,242,18
   18 PUNCH 111, IRAY
      GO TO 17
  242 S = -Y0BAR(NOSUR)
      IF(CLBAR(NOSUR)) 243,244,243
  244 IF(Y0BAR(NOSUR)) 18,245,18
  243 S = S/CLBAR(NOSUR)
      GO TO 245
  240 S = -X0BAR(NOSUR)/CKBAR(NOSUR)
      IF(CLBAR(NOSUR)) 246,244,246
  246 IF(S+Y0BAR(NOSUR)/CLBAR(NOSUR)) 18,245,18
  245 AXIN = Z0BAR(NOSUR)+S*CMBAR(NOSUR)
      PUNCH 112,IRAY,AXIN
   17 PUNCH 100
      IF(SENSE SWITCH 1) 205, 3
    1 FORMAT(8HRAY SURF,27X6HOBJ PT,4X8H2D PT/DC,5X6HINT PT,6X7HDIR COS)
  100 FORMAT (1X)
  104 FORMAT (I3,I3,13H NO INCIDENCE)
  105 FORMAT (I3,I3,13H VIRTUAL PATH)
  106 FORMAT (I3,I3,14H APERTURE STOP)
  107 FORMAT (I3,I3,14H NO REFRACTION)
  108 FORMAT (I3,I3,16X,8HX OR K  ,F11.5,1X,F11.5,1X,F11.5,1X,F11.5)
  109 FORMAT (22X,8HY OR L  ,F11.5,1X,F11.5,1X,F11.5,1X,F11.5)
```

547

```
110 FORMAT (22X,8HZ OR M  ,F11.5,1X,F11.5,1X,F11.5,1X,F11.5)
111 FORMAT (I3,4X,15HNO INT W/OPT AX)
112 FORMAT (I3,4X,23HOPT AX INTERSECTION Z =,F11.5)
    END
```