PROCEEDINGS OF THE

MEETING OF THE EASTERN REGION OF THE

1620 USERS GROUP

Marriott Motor Hotel

Arlington, Virginia

May 6, 7 and 8, 1964

Martin Knebel
Regional Secretary

PROCEEDINGS OF THE MEETING OF THE

EASTERN REGION OF THE 1620 USERS GROUP

MAY 6, 7 AND 8, 1964

TABLE OF CONTENTS

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

(Continued)

SECTION 1

GENERAL

Table of Contents

EASTERN REGION 1620 USERS GROUP MEETING

MARRIOTT MOTEL, WASHINGTON, D.C. - MAY 6-8, 1964

PROGRAM AGENDA

(D. D. Williams, Program Chairman)

Tuesday, May 5th

6:00 - 8:00 P.M.	Registration
8:00	Sound-off Session (Hardware, Programming Systems)

Wednesday, May 6th

8:00 A.M. Registration

Note: Wednesday was devoted exclusively to Programming Workshops and special papers. There were four (4) concurrent meetings, labeled A, B, C, and D, during certain time periods, labeled 1 through 5. This cross reference is utilized in the Synopsis of Papers.

	A	B	C	D
9:00 - 10:30 (1)	Summary of FORTRAN P/S Summary of SPS P/S	Introd. to Monitor (Section I)		
10:30 - 10:45	Coffee			
10:45 - 12:15 (2)	DOODLE, A FORTRAN Language High-Speed SPS Assembler	Introd. to Monitor (Section II)		
12:15 - 1:45	Luncheon (Informal)			
1:45 - 3:15 (3)	Experimental Translator System PDQ FORTRAN - Paper Tape "Select - External" FORTRAN	Advanced Monitor (Section I)	Comparison of SPS Systems	FORTRAN II Workshop (Section I)

	A	B	C	D
3:15 - 3:30	Coffee			
3:30 - 5:00 (4)		Advanced Monitor (Section II)	Aids to Debugging SAMP Monitor Supervision for Open Shop Users	FORTRAN II Workshop (Section II)
8:00 - 10:00 (5)	Comparison of FORTRAN Systems			

Thursday, May 7th

8:00 A.M.	New Users Breakfast Meeting Registration	Jim Oliver
	Note: Thursday morning and Friday afternoon comprise the General Session. Program Teams meet in the interim. The Plotting and Electronics Programs Teams were formed at this meeting.	
9:00 - 10:30	General Session	
	Announcements	Don Williams
	New Equipment Announcements	Tom Wagner
	Systems Design in Electronic Industry	Arnold Spitalny, Marty Goldberg
10:30 - 10:45	Coffee	
10:45 - 12:15	Comparison in Depth of New FORTRAN Systems	Don Jardine, Jan Lee
12:15 - 1:45	Luncheon	
	Keynote Address: Consolidation through Systems Integration	Jim Oliver
1:45 - 3:15	Program Team Workshops	
	Note: See Section 4 - List of Papers - for papers presented or topics treated. Team Chairmen are listed next to workshop.	
	A. Electric Utility	Frank Wells
	B. Educational Institutions	N. Goldman
	C. Rate Engineering	K. W. Brady
	D. Chemical Engineering	C. F. Schroedel - Acting for T. H. Korelitz
	E. Plotting	Tom Scott

3:15 - 3:30	Coffee	
3:30 - 5:00	Continuation of First Four Workshops E. Electronics	Arnold Spitalny
5:30 - 6:30	Social Hour	
7:30	Movie on Type 360 System	IBM
8:00	Executive Board Meeting	Jim Davidson
<u>Friday, May 8th</u>		
9:00 - 10:30	<u>Program Team Workshops</u> A. Electric Utility B. 1710 Users C. Structural and Civil Engineering D. Education Institutions	 Frank Wells J. J. Owen Tom Scott N. Goldman
10:30 - 10:45	Coffee	
10:45 - 12:15	A. Operations Research B. Continuation of 9:00 - 10:30 C. Continuation of 9:00 - 10:30	Jim Oliver
12:15 - 1:45	Luncheon	
1:45 - 3:15	A. Electric Utility B. General Data Processing C. Statistics and Mathematics	Frank Wells Robert Soucy - Acting for Arnold Canner Frank Dickinson
3:15 - 3:30	Coffee	
3:30	<u>General Session</u> Discussion of Sound-off Session IBM Reports Concluding Remarks	 Jim Oliver

SYNOPSIS OF PAPERS

The sessions presented are summarized as follows: Cross Reference Number, Title, Author (where applicable), Synopsis, and level of paper or workshop.

Wednesday, May 6th

A-1 Summary of FORTRAN P/S and SPS P/S - D. T. Northrop, Fairchild Camera and Instrument Company and E. Sinanian, IBM Corporation

A survey was presented by IBM and User personnel of the major IBM and user designed statement and symbolic language programming systems together with the additions and modifications available from the library.
Level: Intermediate

B-1 Introduction to Monitor Programming - E. Sinanian and J. Grant, IBM Corporation

A general discussion for new users of disk drives. Section I discussed the Monitor in general and SPS II-D in particular.
Level: Elementary

A-2 DOODLE - A Do-It-Yourself Problem Solver - M. V. Farina, General Electric Company

A FORTRAN programming system for 40K which interprets and executes.
Level: Intermediate

AFIT SPS with Simulated TNFTNS and MF - D. Olson, Newark College of Engineering

Level: Intermediate

SPS High Speed Assembler for 20K - K. Germann and G. Rumrill, Newark College of Engineering

Level: Intermediate

B-2 Introduction to Monitor Programming - IBM Corporation

Continued from previous session. Emphasis on FORTRAN II-D in Section II.

A-3 An Experimental Personalized Array Translator System - H. Hellerman, IBM Corporation

Level: Advanced

A-3 (Cont.) Paper Tape PDQ FORTRAN - J. W. Trantum and P. G. Boekhoff, General American Transportation Company

Level: Intermediate

FORTRAN (Select-External) - P. G. Boekhoff, General American Transportation Company

A programming system based on PDQ FORTRAN embodying a new I/O approach.

Level: Intermediate

B-3 Advanced Monitor Programming - J. Grant, IBM Corporation

A detailed discussion of Monitor I for the advanced user.

Level: Advanced

C-3 A Comparison of SPS Systems - J. A. N. Lee and R. L. Pratt

An evaluation and comparison in depth of the first generation programming systems.

Level: Elementary to Advanced

D-3 FORTRAN II Workshop - R. D. Burgess, Mechanical Technology, Inc.

A tutorial session reported to have been highly successful at MTI.

Level: Intermediate

B-4 Advanced Monitor Programming - IBM Corporation

Continued from previous session. Monitor II was emphasized in Section II.

Level: Advanced

C-4 The Art of Debugging - E. J. Orth, Jr., Southern Services, Inc.

Level: Intermediate

Search and Memory Print (SAMP) - J. M. Wolfe, Brooklyn College

Level: Elementary

FORTRAN Symbol Table Punch for IBM, UTO and PDQ FORTRAN Systems - R. C. Irons, U.S. Naval School of Aviation Medicine

Level: Intermediate

C-4 (Cont.) Monitor Supervisor for the 1620-1311 - E. E. Newman, Massachusetts Institute of Technology

Level: Intermediate

D-4 FORTRAN II Workshop - R. D. Burgess, Mechanical Technology, Inc.

Continued from previous session. Section II stressed subroutine linkage and debugging.

A-5 Comparison of FORTRAN Systems - D. A. Jardine, J. A. N. Lee, W. A. Burrows, R. J. Robinson and R. L. Pratt

An evaluation and comparison in depth of the first generation programming systems, including IBM, AFIT, UTO and PDQ FORTRAN.

Thursday, May 7th

9:00 - 10:30 Welcome introduction and reports.

Comments on theme of meeting (systems) - D. D. Williams

Use of Computers In Design of Electronic Equipment - A. Spitalny, United Aircraft Corporation

Network Analysis on the 1620 - M. Goldberg, United Aircraft Corporation

Note: The preceding two papers support the topic "Systems Design in Electronic Industry" which is listed in the "Agenda".

10:45 - 12:15 An evaluation and comparison in depth of the second generation FORTRAN programming systems including FORTRAN II and Kingstran FORTRAN by D. A. Jardine and J. A. N. Lee.

Papers classed as elementary presuppose no background on the part of the audience. Sessions classified as intermediate presuppose some acquaintance with the subject matter.

Sessions classified as advanced are intended for persons with considerable experience in the field; questions of an elementary nature will be discouraged at these sessions.

This agenda was supplemented between sessions by half-hour programmed demonstrations on a type 1620 Model 2 computer with 40 K positions of memory, two disk units, a plotter and a card read-punch unit. The following planned list of demonstrations was altered slightly to accommodate available hardware features:

Set A: Design Automation
Orbital Trajectory Calculation
Similation of Analog Computer

Set B: Monitor I
FORTRAN II
SPS III

Other Programs Available: COGO I
Circuit Analysis
Electric Load Flow
Capital Investments
FORTRAN with Format
Plotter Subroutines

Any User desiring a specific demonstration from the above selection at times other than those indicated, or desiring to run any of his own programs was asked to submit his request to the IBM representative at the demonstration center.

Comments of the Program Chairman

The general response to your Program Committee's probe for interesting material for the May, 1964, Conference was quite gratifying.

Models of previous Proceedings, the wealth of papers offered and the availability and generous offers of speakers - expert in their fields of interest, all helped to make the program planning easier.

The directive, for the best use of the contributors' material and talents, was taken from the Group's bylaws, and is repeated here:

> "The primary object of the 1620 Users Group is to advance the effectiveness of the utilization of the IBM 1620 Data Processing System . . . . "

In order to effectively advance this basic objective, the Committee decided to build a program blended with a theme which would focus attention to the currently large stockpile of Programming Systems and Applications Programs. Thus, it was hoped to generate some constructive thought toward reduction of redundant programming efforts and finally to affect some integration of Application Programs into the larger context of Data Processing Systems.

Hence, the theme, "Consolidation through Systems Integration" is being highlighted.

The agenda was arranged so as to mold the Conference into a composite of two parts, each part having its own identity and flavored by the theme.

Part I (Wednesday morning to noon of Thursday) is mainly identified with Programming Systems. Wednesday's four sessions were each of ninety minute duration, being held concurrently and being three-deep. One of these concurrent sessions is a panel discussion of SPS Systems which support the theme relative to Part I.

Further support of the theme, related to Part I, is given in two general session panel discussions covering the first and second generation of FORTRAN Systems. An evening session on Wednesday and one on Thursday morning comprise these two sessions.

Part II (Thursday morning to Friday afternoon) is mainly identified with Applications Programs. The Programs Teams' activities are confined to this period. These meetings scheduled concurrently are four-deep on Thursday afternoon and three-deep during three sessions on Friday. The Chairmen of the various teams had been requested to limit the scope of their interest in programming to Applications Programs during their teams' sessions.

The first session of Thursday morning - an overlap of Part I and Part II - is mentioned here as an example of a tie in with the theme by integrating several Applications Programs into a Program System (i.e. Spitalny's paper, "System Design in Electronic Industry").

The panel discussions previously referred to treat, in some depth, the survey list of FORTRAN and SPS Systems which are introduced during the first session of the Conference's opening. This list with some additions is reprinted in Section 5 of these Proceedings.

Further comment on the panels' activity will be found in the article following this one entitled "Some Notes on Survey List of Programming Systems and Panel Discussions".

The Program Committee's post-conference activity is confined to the compiling of all the material in readiness for printing.

In conclusion, the Committee offers sincere thanks to the many people without whose assistance our mission might not have been completed.

John A. Rodgers
Program Secretary

Donald D. Williams
Program Chairman

1964 Spring Meeting
1620 Users Group
Eastern Region

Some Notes on
"Survey List of Programming Systems"
and Panel Discussions

Survey List of Programming Systems

David T. Northrop devoted the entire opening session of the Conference to a review of a Survey List of Programming Systems - both IBM and User compiled - with a brief commentary relating to each listed item. This session as planned served to introduce the subject matter to be treated in depth during the panel discussions which were held later.

SPS Panel Discussion

R. L. Pratt, aided by Kurt Germann, officiated at an interesting panel discussion on SPS Systems. The Program Committee is indebted to R. L. Pratt for his post-editing of the Survey List for reprinting in the Proceedings.

FORTRAN Panel Discussion

Report on Panel
User Written Compilers in Depth

D. A. Jardine

Brief mention was made of IBM FORTRAN-Without-Format and Format FORTRAN by J. A. N. Lee.

R. L. Pratt described some of the useful features of his AFIT FORTRAN, such as compile time diagnostics and batch compiling. AFIT FORTRAN, while an improvement over IBM systems, is somewhat slower than other user-written compilers.

D. A. Jardine, in the absence of the authors, E. S. Lee and J. A. Field, talked about UTO FORTRAN and the contributions made by this system to improved language and easier operation. UTO FORTRAN has few compile time or object time error messages.

W. A. Burrows, in the absence of author F. H. Maskiell, described some of the features of P.D.Q. FORTRAN. It is based on UTO FORTRAN with several major improvements. It is the fastest running program at object time for any 1620 not using hardware F.P. The compiler is essentially the same as UTO, but the object time subroutines have been rewritten almost completely.

R. L. Pratt then described Auto-Float FORTRAN, derived from AFIT FORTRAN, which compiles hardware float-point add and subtract instructions in-line (not subroutined). It is probably the fastest 20K FORTRAN (at object time) currently available. Only limited FORMAT capabilities are available.

C. H. Davidson talked about FORGO, the University of Wisconsin's load-and-go compiler for teaching use. Extensive diagnostics at both compile and object time are available. 40K memory is required.

Discussion from the floor centered around details of operation, use and construction of the various compilers.

SECTION 2

KEYNOTE ADDRESS

Table of Contents

CONSOLIDATION THROUGH SYSTEMS INTEGRATION

Dr. James R. Oliver, Dean
Graduate School and Director,
Computing Center
University of Southwestern Louisiana,
Lafayette, Louisiana

I am pleased, as President of the Eastern Region 1620 Users Group, to talk briefly with you concerning the theme of our meeting, "Consolidation Through Systems Integration". I trust that I will be able to leave you with a thought or two today that might prove of benefit to you. And perhaps in so doing I will be able to help further the aims of our organization.

Few terms have as many definitions as the word "system". And I say this even when talking to a group of people interested in computing, a field in which ordinary English words have frequently been given many new and sometimes surprising meanings! Isn't this a little strange when we think of how precise our languages are for communicating with the computer?

Although "system" has many meanings, including the people involved in computing at an installation, I think we can profitably devote these few minutes to hardware systems and software systems. Thus I will direct my remarks along these areas of interest and will try to make three major points which I believe to be of importance.

Many organizations have wide requirements as pertains to computer usage--business applications, management applications, scientific applications, to mention a few--including many requirements which may not yet be known.

Most of us would enjoy having our own private machines to work with, but this is a luxury which few can afford. I can speak from the point

of view of the not-so-affluent university in saying that <u>we</u> certainly cannot do so!

Perhaps you have heard of colleges and universities which have many computers--even many large ones. Unfortunately, these few cases are overly publicized; the vast majority of educational institutions, however, are grasping and gasping for funds to even keep a computing center going. As an example, which can be repeated over and over, in my own case I teach two courses each semester, serve as Dean of the Graduate School, as Director of all of our National Science Foundation activities on campus and serve as Director of the Computing Center assisted by one full-time unit record man and one half-time lady to help with computing activities.

Our main purpose for existence as educators in the computing field is to serve the manufacturers and the users. Relatively few of our students are being educated for our own use.

But that is the subject of another talk. Today we are here to discuss a common problem--at least common to most of us. We are faced with a problem of maximum production for minimum cost. In truth we very practically are concerned with a problem in Operations Research - should we suggest a computer solution to this problem?

Perhaps if we were able to put ourselves in the shoes of management we would be better able to visualize the overall problem. But we have to live with the fact that most of us have to put up with a management which:

(1) Cannot, or sometimes will not, comprehend computer concepts,

(2) Often is in some way afraid of computer "take over", and

(3) Must be concerned with money justification to several levels--their superiors, shareholders, other members of organization.

Having acted and acting now on both sides of the proverbial fence, I have seen some of the problems faced by both sides. And I assure you that the problems of the administrator are not trivial.

All too often we call upon the administrator to act in a capacity for which he is not equipped. Can we expect a non-computer oriented administrator to make decisions that we ourselves hesitate to make?

POINT 1: Are we making a sufficient effort in educating and in keeping ~~with~~ our administrative staff informed?

Nearly all of us sooner or later must discuss computer activities with our colleges--if you haven't, don't despair, you will! Some of these colleagues are non-computer oriented. Even worse, some of them are other-computer (particularly other-concepts) oriented.

Here is one point that is certainly a most important consideration. If the organization has a total of x dollars to spend on computers, and one of our colleagues convinces management that y dollars should be spent on a business use computer, it is somewhat obvious that x - y dollars will remain for our installation. Remember, because of the non-scientific orientation of many in management, and because of the strong arguments of persons in business applications, there will frequently be sympathy for fragmentation.

Can't indeed two or more systems be consolidated into one? In many cases, yes. We know that the 1620 is a very versatile machine. If we inform our colleagues well of its capabilities, couldn't a Model II 1620, 40K, 2 disk drives and printer do most jobs as well as or better than two systems costing more collectively?

We begin to see this idea more with the Model 360. What are we going to do about it? I can foresee all kinds of decisions to be made

with the 360. It is quite possible that we will be forced to resort to the simulation of our computer installations to better determine just how they should be organized. I know of such a study made at Oak Ridge.

In the meantime, decisions with the 1620 still must be made. Few 360's will be installed by most of us by two years from today. The 1620 is still a good machine and it has much wear left in it. Those who buy or have bought 1620's have even a more serious problem. They will likely keep in touch with the 1620 for many years to come.

POINT 2: Have we informed and communicated sufficiently well with our colleagues who are, or are likely to be, interested with computer activities?

Then we come to systems as it applies to programming systems--mostly referred to as software.

Until now we have considered two distinct divisions: IBM and Users. For some time now several people, including Jim Davidson, Charlie Davidson, Don Jardine and others have been working to get better--no, to get a--liaison between IBM and User activities in this area of endeavors.

I would like to touch on two items: (1) Consultation with Users when IBM decides to produce any kind of programming package. (2) Consultation with IBM when Users decide to produce a significant programming package.

Can't IBM use our suggestions (1) because we know what we need and (2) to utilize the "brains" of those in the Users Group? Also, why all the duplicate effort? How many FORTRANs are needed? How many can we afford? Shouldn't some of the effort by Users in attempting to produce a FORTRAN which will work as it should be devoted to experimenting with other

languages? Surely those Users who earlier today heard the presentation of the characteristics of an outstanding FORTRAN compiler produced by a group of Users wonder why this has not already been available to us, instead of some of the processors we have been forced to use!

We must not overlook the need for cooperation both ways. We recognize that this brings up ideas not often used--consultating IBM when a User produces a significant programming package. At least three reasons exist for this: (1) Perhaps IBM knowledge can help the programming effort, (2) We might avoid duplication of effort by Users or by IBM, (3) IBM could advise Users of impending changes to prevent an effort being completely negated by new hardware or software being developed. We owe our colleagues the production of worthwhile packages if the Users' contributions to the Program Library is to mean anything.

POINT 3: Are we presently communicating among ourselves, or with IBM, sufficiently well to achieve consolidation through the integration of programming systems? Or will we fragment ourselves by writing an infinite number of compilers to the exclusion of better things to do?

Some of us are teaching--some are involved in computer applications. A much smaller number is engaged in preparing software. Again I will state that we are all concerned with getting the best production possible from us and our machines.

Aren't we charged with the obligation of doing best job possible? Shouldn't we all be concerned with the theme of our meeting--consolidation through systems integration?

But I should now begin winding up this talk.

A few years from now a much better talk than I can give will be produced by a properly programmed computer. The announcement will read, "Our program for this week will be, 'Consolidation Through Systems Integration' by Multivac 70809010...".

The computer's speech will be really interesting. It will have unity, coherence, and emphasis. It will meet all of the requirements of public speaking as listeners like it. The humor and jokes contained in the speech will have a mathematical probability of at least 99% of not falling flat. Such a speech will not have any terms in it that are strange to most of the audience, so the audience will not get lost. The talk will conform to all of Parkinson's Laws and Murphy's Laws. At the same time it will take into account what the audience already knows. And it will not be guilty of unnecessary repetition of stupid cliches.

In fact, instead of a single speech to a whole audience, the computer will provide multiple simultaneous speeches for various segments of the audience. In front of each member of the audience will be several small dial switches. One will be marked "Speed Control" with positions labeled "You're going too fast" and "You're going too slow". Another switch will be marked, "Understanding Control", saying "Yes, yes, I know all that, please come to the point" and "You're way over my head, please start over and explain". A third switch will be marked "Boredom Control" with positions marked "You are being mildly interesting--keep it up" and "You are boring me to extinction--please become lively".

I look forward to the time when a computer will be making speeches at meetings. I think of all the poor speeches produced by human beings that I have listened to and realize that in the future, if I live long

enough, I will listen to uniformly good speeches produced by computers. But until that time you will be forced to listen to speakers like me. I am able to recognize, however, that those of you who are still awake are reaching for the switch marked "You've talked too long--please shut up"! So I'm going to pretend that you shut the power off.

SECTION 3

PROGRAM TEAM SESSIONS

Table of Contents

Table of Contents

(Continued)

Attendants at Sessions of Statistics and Mathematics Programs Team

Number Attending - 59

"Interest - Statistics Only"

George H. Woodruff
Luke Sparvero
Norman Goldman
Myrna Weiner
John Pate Pointer
Welborn H. Smith
Bill Thompson
A. H. Best
B. G. Wingersky
S. J. Jurnack
Paul E. Brittain
Thomas C. Teeples
F. Wang
William B. Hise

"Interest - Mathematics Only"

Wayne Meriwether
Gregson Payne
Clayton E. Jensen
Dario Bollacasa
Samuel F. Martin
Charles Weiss
John Kent
Reginald T. Harling
Richard E. Grove
M. S. Wingersky
Brother B. E. O'Neill
Patricia Lussow
L. L. Cook
Greta Larson
Pat Moorhead
David C. Pixley
K. A. Bridgeman
James J. McLaughlin
Theodore R. Sabine
Lawrence Wright
Jane Bonnette

"Interest - Statistics and Mathematics"

Sidney Kellman
Richard Guion
Martin Goldberg
Joyce Currie Little
Ralph E. Lee
Judith S. Liebman
Donald L. Flagg
John E. Alman
Mae E. Meads
Eleanor Stone
Diana Lloyd
Vincent Gangi
Charles Yackulics
Simeon P. Taylor
Richard E. Scott
L. Ong
J. W. Sawyer
Lee Hendrickson
James R. Oliver
William Heltzel
C. H. Remilen
Nancy Paquin
F. R. Henderson
Frank N. Dickinson - Team Chairman

Report of Activities of
Statistics and Mathematics Programs Team

Approximately 75 people attended the session. A sign-up sheet was passed around. Fifty-nine people signed this sheet with 21 indicating their interest was mathematics, 24 indicating statistics and mathematics and 14 indicating statistics. Only three people listed physics along with mathematics. A show of hands indicated that almost everyone in attendance at this session preferred to have statistics and mathematics sessions not held concurrently, since most people would attend both sessions if they are held separately. I assume that the number of papers submitted would be the determining factor in whether the sessions would be combined or held separately at the next meeting.

Frank N. Dickinson, Chairman
Statistics and Mathematics Programs Team

Papers Presented At This Session
Which Are Reprinted in Section 4

A General Function Subprogram - M. E. Munroe, University of New Hampshire

Meeting Assumptions of Homogeneity of Variance - R. C. Irons, U.S. Naval School of Aviation Medicine

FORTRAN Programs for Computing Elliptic Integrals and Functions - H. E. Fettis and J. C. Caslin, Aerospace Research Laboratories

The Use of Discrete Convolution in the Analysis of Diffraction Patterns - J. S. Liebman and P. N. Johnson, General Electric Company

A Computer Approach to Gamma Spectrum Analysis by the Simultaneous Equations Method - W. E. Carr, Department of Health, Education and Welfare

Attendants at Sessions of
Plotting Programs Team

Number Attending - 26

Charles Carlson
Robert Chaput
Russell Davis
Henry Fell
Vincent Gangi
Kurt Germann
Martin Goldberg
Harold Gottheim
William Heltzel
Lanny Hoffman
Robert Kenngott
James McLaughlin
Robert Meckley
Thomas Morrisson
Edward Newman
Nancy Paquin
Charles Remilen
I. Rosenbarker
Henry Scaletti
Leonard Schiffmann
Thomas Scott
Arnold Spitalny
Wanda Stacke
John Tiers
Charles Yackulics
C. Bailey - Team Chairman

Report of Activities of
Plotting Programs Team

The first meeting of those users active in plotting convened at 1:45 p.m. on Thursday, May 7. Mr. C. Bailey of IBM in Endicott, N.Y., presided over the meeting.

The first item discussed was the housekeeping problems involved in the isometric drawing and dimensioning of piping configurations. The problem of avoiding overlap or cross over of drawn lines is of major concern in producing such drawings.

Mr. Newman mentioned that M.I.T. has created some problem oriented plotting languages and is in the process of tying them in with COGO. Both M.I.T. and Sun Oil Company have developed very similar basic SPS routines for plotting straight lines, circles, circular arcs, and alphameric characters, all with scaling and line width factors. M.I.T. is at present working on a program to draw electrical schematics while Sun Oil Company is preparing a program to draw isometric piping drawings along with a piping bill of material.

The New York State Department of Public Works has used the plotter to sketch highway cross sections.

The remainder of the session dealt with problems arising from the hardware itself.

It was mentioned that several of the plotting units take from fifteen minutes to one half hour to warm up in the morning. During this interval the calibration device requires special attention. There seems to be a noticeable change in paper size in the direction of the carriage. The general opinion was that proper humidity control would alleviate this problem.

An end of roll sensing device was mentioned as a solution to the annoyance of running out of paper, and plotting a large portion of the drawing on the roller itself.

Ball point pens seem to be preferred on ordinary jobs, while ink pens have the edge for special jobs. Liquid lead pens have been found to provide a very dark, eraseable line.

Mr. Newman was appointed alternate chairman for the Fall meeting.

The meeting adjourned at 3:15 p.m.

Thomas J. Scott, Acting Secretary
Plotting Programs Team

3.2.2

Attendants at Sessions of
Electronics Programs Team

Number Attending - 8

Lawrence Wright

Keene A. Bridgeman

Steve Jurnack

Kurt H. Germann

Hubbard A. Seward

Herbert M. Wall

Arnold Spitalny - Team Chairman

Martin J. Goldberg

Report of Activities of
Electronics Programs Team

A meeting of the Electronics Team was held on May 7, 1964.

Arnold Spitalny was elected permanent chairman with Martin Goldberg as alternate chairman and secretary.

Herb Wall was assigned as delegate to the Share Group Meeting to be held in June in Atlantic City. He will establish contacts with the corresponding electronic team from Share.

Areas of interest discussed at the meeting were:

1. Circuit analysis programs
2. Filter design and synthesis programs
3. Application of statistics to circuit analysis
4. Programs for automatic layout of printed circuit boards and micro-electronic circuits
5. Programs for generation of circuit schematics

It was suggested that other members of the Users Group be questioned as to possible interest in the Electronics Team. This could be done in the published proceedings. Interested parties should be asked to express their areas of interest and send these to:

Martin J. Goldberg
Computer Applications Engineer
Norden Division, United Aircraft Corporation
Helen Street
Norwalk, Connecticut

Arnold Spitalny, Chairman
Electronics Programs Team

Report of Activities of Education Institutions Programs Team

In addition to the papers listed at the bottom of the page, the following paper was presented:

> Scheduling Students by Computer by Dr. Charles W. Williams, Washington and Lee University. This paper was published in Computer Applications Service.

These papers were presented in the second half of the session:

> 1620 Programs to Implement the Quine Method of Boolean Simplification by Prof. Thomas R. Hoffman, Union College, Schenectady, New York. This paper was presented at the Western Region Meeting in Phoenix.

> A New Course in Computer Programming by Prof. C. H. Davidson, University of Wisconsin. This paper was presented at the joint Mid-Western, Canadian Region Meeting in Chicago.

An extra session was called for the following day. A discussion led by Dr. J. R. Oliver occupied the entire session.

Norman Goldman, Chairman
Education Institutions
Programs Team

Papers Presented At This Session Which Are Reprinted in Section 4

Automatic Scheduling and Registration in a Small College - A. F. Jackson, Agricultural and Technical College of North Carolina

Plot Routine for NCE Load-and-Go FORTRAN - H. Seward, Newark College of Engineering

Attendants at Sessions of
<u>Structural and Civil Engineering Programs Team</u>

<u>Number Attending - 36</u>

W. Brzozowski

W. Burleigh

W. Burrows

C. Carlson

R. Chaput

R. Dabe

R. Davis

E. Dougherty

D. Flagg

H. Gelsi

H. Gottheim

W. Heltzel

L. Hendrickson

H. Hurt

C. Jensen

R. Kenngott

J. Kent

T. Latterner

V. Lippo

E. Newman

N. Ogilvie

H. Peterson

C. Remilen

R. Riley

J. Rodgers

J. Russell

F. Salek

J. Samstag

L. Schiffman

T. Scott - Team Chairman

W. Smith

A. Spitalny

R. Steinhart

W. Thompson

F. Wells

G. Woodruff

Report of Activities of Structural and Civil Engineering Programs Team

Two papers were presented at the Structural and Civil Engineering Team Meeting that convened at 9:00 a.m. on Friday, May 8, 1964. Mr. A. D. Stasi from Edwards and Kelcey Company presented his paper on the use of the small computer as a processor of O-D survey data, and Mr. E. J. Orth from Southern Services Company spoke about a newly developed pipe stress analysis program. An abstract of Mr. Orth's paper is attached to the minutes.

Thirty-six persons attended this meeting. Out of these thirty-six persons, sixteen were Civil Engineers by profession and twenty-six indicated that they would be interested in perpetuating future sessions. Four papers were tentatively promised for the Fall meeting in Buffalo, New York.

The meeting adjourned at noon.

T. J. Scott, Chairman
Structural and Civil
Engineering Programs Team

Papers Presented At This Session Which Are Reprinted in Section 4

Processing O-D Survey Data on a Small Computer - A. D. Stasi and M. B. Lipetz, Edwards and Kelcey, Inc.

1620 USERS GROUP LIBRARY
PROGRAM ABSTRACT

1. TITLE: General Pipe Stress

2. AUTHORS: E. J. Orth, Jr., Southern Services, Inc.
John Lewis, IBM, Birmingham, Alabama

DATE: 5 March 1964 USERS GROUP CODE: 1125 SUBJECT CLASSIFICATION: 9.5

3. DIRECT INQUIRIES TO: E. J. Orth, Jr.
Southern Services, Inc.
600 North 18th Street
Birmingham, Alabama
323-5341 Ext. 2339

Watch KWIC Index for availability.

4. DESCRIPTION: A four part piping stress program is presented -
1. INPUT EDIT,
2. FLEXIBILITY MATRIX CALCULATION,
3. SOLUTION FOR BRANCH POINT DEFLECTIONS, and
4. COMPUTATION OF STRESSES.

The INPUT EDIT program performs a rigorous error check on the data file. A monitor is output giving coordinates of each point. Data file consists of system topological data and, for each branch, coordinates of first point reference global origin, expansion and movement data, coordinate differences between tangent intersections, and bend radii. Data is keyed by 3 letter alphabetic codes.

The FLEXIBILITY MATRIX CALCULATION applies the algorithm described in the supplement to Kellogg's book, "The Design of Piping Systems", Second Edition.

The SOLUTION program allows for loops. Rigid stops, spring stops, and external forces may be oriented parallel to coordinate axes. The algorithm of Chen (ASME Paper No. 59-APM-24) is used, with a novel approach to solution of the matrix to take advantage of sparsity. Result is deflections at branch points and reactions each branch, all reference global origin. These are left in common storage, and Program 4 called thru a link subroutine at completion of solution.

The STRESS and DEFLECTION program computes for each point of each branch-
1. Deflections reference the point,
2. Restraining reactions reference the point, and
3. Combined stresses.

Sum of restraining reactions about each branch point is also given.

7. SPECIFICATIONS: A 60K card 1620 in FORTRAN II configuration is required. Conversion to 40K disc is possible, with major reprogramming. Peripheral equipment - reproducer, sorter, 407 E-8, keypunch.

8. PROGRAMMING TYPE: FORTRAN II mainline, with subroutine and function subprograms.

Attendants at Sessions of
Electric Utility Programs Team

Number Attending - 38

J. H. Russell	Dick Briesemeister
John H. Evans	Bob Steinhart
Chris Martin	George S. Haralampu
Arno F. Glimn	P. J. Sullivan
A. L. Lipson	J. Coburn Hubbard
G. W. Wilson	E. H. Gerrish
John F. Leuer	R. B. Riley
A. Doyle Baker	W. J. Cook
Roy W. Thomas	C. B. Scharp
Jene Y. Louis	D. D. Williams
Frank J. Wells - Team Chairman	E. H. Parker
Stanley A. Clark	J. L. Davidson
Herbert Blaicher	M. G. DeGennaro
Tom Hoke	E. E. Thompson
Robert W. Davies	B. E. Green
John E. Cromwell	J. A. Rodgers
Veikko A. Lippo	Robert R. White
Paul D. Folse	Ed Cox
Len Karr	Ed Orth

Report of Activities of
Electric Utility Programs Team

The first item discussed was a heat balance program entitled "Heat Balance I", written by Bob White and Yosh Fujimura of the Department of Water and Power, City of Los Angeles. Bob White made the presentation.

The Heat Balance program is highly subroutinized and uses a supervisory routine to control subroutine sequencing. In preparing data, the user establishes the order of calculations. This order is entered as a sequence table. The supervisory routine refers to the sequence table to find out which subroutine to use next. All subroutines use common work areas, thus cutting down on linkages.

The approach used by Gerngross on the 650 is used here. The beauty of this method is that the user can select from among the library of subroutines those necessary to describe his particular problem. Thus, theoretically, a general program results which is good for all situations. This is opposed to writing a speciallized program for each system. Note the word theoretically: while 95% of all possible conditions may be provided for, there is always the possibility that an additional subroutine will have to be written for some special condition. This is very easy to do, however, due to the building block construction of the program. The writeup is very thorough, and very well documented.

In its present form the program is written in SPS for 60K Disc Model II. However, since it is composed of three distinct parts, it may be reassembled for 20K. The three parts are input, solve, and output. The only function of the disc storage is to provide a resting place for the program. The program was conceived, tested, and used on a 40K machine. Core may be saved by omitting any subroutines not used.

When asked if IBM would be receptive to setting this up for 20K, Bob Steinhart and Chris Martin did a truly magnificent job of bouncing the ball back and forth. All came out happily, however, because they are going to look into the matter thoroughly. Whatever that means.

This program has been used both as a design tool and as a performance test tool. Results agree well with manufacturer's results and with test data, generally to within a few tenths of a per cent. Two days are required to set up data for a new steam plant: one to set up the system diagram, and one to fill out data sheets. Two or three machine tries are necessary to get valid answers due to data and setup errors. Bob emphasized that this is a qualitative tool, not a quantitative tool. Its usefulness is in providing comparisons.

In preparing data, the user cuts out a set of symbols and assembles them with glue to simulate the particular system under study. It should be noted that since this is an iterative solution, the speed and accuracy of solution is affected by the sequence of calculations. Experience will be of immense help here.

The test case in the writeup is a 330 MW unit. Running time is 15-20 minutes Model I and 3.5-4 minutes Model II. Once data is read, subsequent changes may be made without reloading the entire system. This is similar to load flow procedures.

As set up now, the program will handle up to 38 pieces of equipment, including 4 turbines and double reheat. As the situation develops, subroutines for a moisture separator and a saturated turbine will be added for nuclear systems.

The next item on the agenda was "A Program for Evaluating Alternate Generating Plant Expansion Patterns", by R. W. Davies of Baltimore Gas and Electric, in which portions of a paper presented at the 1963 American Power Conference were discussed, as indicated in the attached abstract.

George Haralampu, New England Electric Service, gave a report from the Committee on Load Flow Modifications. The committee suggests:

INPUT

1. Add one more significant digit to R, X, and capacitive susceptance fields.
2. Remove restriction that total susceptance must not exceed .999.
3. Make bus data FORTRAN compatible (minus signs).
4. Eliminate voltage entry on all non-voltage controlled buses, and add feature to allow read/punch of voltage deck.
5. Make interchange data FORTRAN compatible.
6. For interchange, use 2 cards with 40 digits.

OUTPUT

1. Since only 72 card columns are read by undoctored Format FORTRAN, move mismatch on bus card all the way over to the right.
2. Eliminate alphabetic "TAP" for compatibility purposes.
3. Change from per unit to per cent.
4. Adopt general output format other load flows: that is, bus data first, followed by line data.

Frank Wells will send out a questionnaire on these items to all of us. We will have the opportunity to accept or to reject the entire package, and to make comments.

A very interesting report on a 1401 print program for Glimn's Load Flow was given by John Evans, Southwestern Public Service Company. This program accepts data describing how the 1 line diagram is to be printed, and then takes the output from Glimn's program and proceeds to print a system 1 line diagram with all pertinent information. John showed us a sample that printed an 80 bus system on 6 pages. Parallel lines and transformers are handled, and tap settings are printed. Up to 7 buses are printed across the page. Buses are printed horizontal as the paper is in the printer and lines vertically. Note that this does not just print numeric data on a preprinted diagram, but prints both diagram and data!

A very simple procedure is followed to set up a particular system. A sketch is made to set up spacing and general arrangement, and then a set of data is recorded to specify spacing and placement to the 1401.

Present capacity is 99 lines, but this is more of a data sheet restriction than a program restriction. A 4K 1401 with 1 tape is required. John readily agreed to submit this to the library, which is to say he was not given much choice!

The next topic of discussion was Allocation of Bus Loads for peak/off-peak conditions in load flow studies. If ratios are used, var problems often occur. One company arrives at off peak by taking a percentage off of peak loads and regulating voltages with generators. Another company uses a separate var forecast. The ideal situation exists when hourly loads are sampled on each substation. A fourth voice from the back of the room said that they monitored areas, pulled out non conforming loads, and then companies make complete and separate forecasts for peak/off-peak conditions.

Arno Glimn's new B Constant Program was discussed by Roy Thomas, Southwestern Public Service Company. The entire package consists of 6 programs and calls for 20K auto divide. It is very easy to handle. A system with 13 sources took 1:20 hours after the initial load flow was completed. Al Lipson, Virginia Electric and Power Company, had worked with this program also, with a system containing 16 sources, 123 buses, and 180 lines. Time required after load flow was 16 hours. Loss on the system was matched within 0.2 MW. Overall load level was 1800 MW, loss was 23 MW. The program is being typed, and should be available from the library soon.

Len Karr, Burns and Roe, discussed work he had done on Economic Dispatch Tables since the Pittsburgh meeting.

Use of the 1620 for power system dispatch, either directly or indirectly, was the next item. Roy Thomas was of the opinion that the 1620 is too small, depending upon how accurately incremental cost data is to be represented. A straight line representation of incremental cost is fine if sufficient segments are taken. John Leuer, Niagara Mohawk, brought out the usefulness of being able to change B constants at the drop of a hat with a digital system.

An application of Linear Programming to Fuel Purchases was discussed by Ed Orth. To date, the individual companies of The Southern Company have been scheduling their purchases independently by time honored and hoary methods. When LP was applied to one company for the year 1963, savings in excess of $100,000 were indicated. Present plans are to take the much-worshipped "System Approach" and consider the entire Southern Company and all feasible sources. 100 sources, each with a maximum and minimum plus 100 other constraints plus 16 points of usage = 316 restraints. About 30% of all activities will be valid. This is obviously outside the 1620 range and in the 7094 range. At present, data is being gathered.

Certain modifications on Jose Marina's Short Circuit Program have been made by Jene Louis of LILCO. He has removed R to leave X only, increased precision from 6 to 10, gone from 2 pass to 1 pass, altered output to give current flow on all lines rather than on adjacent lines, and removed all sense switch testing. System size is 150 bus/250 lines, and 15.5K is required. Speed is equivalent to Jose Marina's program. Jene has made extensive use of network reduction to cut

down running time. This program is designed to handle the range of reactances from pipe type cable to overhead. Use of X only tends to give more conservative results.

Jene also discussed Steady State Stability by use of Glimn's load flow program.

J. C. Hubbard, Baltimore Gas and Electric Company, presented a very interesting paper on "A Distribution Feeder and Substation Load Forecasting System". Copies may be obtained by writing him at Room 1000, One Charles Center, Baltimore Gas and Electric Company, Baltimore 3, Maryland.

Transformer Load Management came in for a good deal of discussion. No one is doing it all on the 1620. George Haralampu stated that some one in Boston is testing on the 1620, but will be working on the 1401. There are two reasons for this: they will be using the billing tapes, and the amount of sorting required. Dick Briesemeister, Consolidated Edison, said that they are using data from billing on their 7080. They are achieving good results on radial, not so good on network.

On this point of networks, Tom Hoke, Oklahoma Gas and Electric, said that they have been handling their underground by using demands, with no diversity, coupled with a 1620 load flow. The big problem is data gathering on an underground network to start with a correct representation.

In answer to a question on whether the main value was restricted to large metropolitan areas, the consensus was no, the value is system wide.

There are two justifications for transformer load management: minimization of burnouts, and better loading. Better loading is probably the most valuable.

A great storm of discussion ensued when the subject of Disc Pack Applications came up, particularly on the question, disc or core? Given 20K, is it better to go 40K or 20K disc? Bob White strongly recommended core, because 20K disc, particularly under Monitor operation, is a slow situation. There is not much room left for the object program after 14K for FORTRAN II subroutines. If core images are stored, a different situation exists. When programs are stored on disc in relocatable format, it takes much, much longer to get them into core. George Wilson prefers the core image mode, wherein a simple minded monitor of sorts gets the core image from disc. Bob Steinhart made the point that although IBM is pushing Monitor, users do not have to use it!

V. A. Lippo, West Penn Power Company, gave a very interesting exposition on why they went from 40K to 20K disc. In essence, their philosophy is this: For a reduction in speed (about 10%), they will have the limits removed from their problem solving capability, in addition to being able to operate around the clock under disc supervision. They do quite a bit of work in symbolic.

The final conclusion to this discussion was that each installation has its own unique problem, and that thus, there is no general answer. One point that would point to disc is the capability to operate around the clock and thus get

the usage to the point that additional hardware is justified. That is, if management will buy that type of argument! It has been known not to.

With this, our meeting drew to a close. There were many other interesting utilities-type presentations at other team meetings, but since your secretary is not endowed with the power of bilocation, he was unable to attend.

A few items from the correspondence heap: Dick Smith, Gulf States Utilities, writes that they are increasing the size of Dr. Cooke's B constant program to 60K. Their matrix short circuit program has been corrected, but mutuals are not yet in the picture. He has again had a chance to test the bus shuffle as a part of the Mozina load flow, and the 20-30% decrease in running time was again apparent.

Has anyone done anything to convert Alcoa's 40K sag-tension program to 20K? If so, please write Stan Clark, Public Service Company of New Hampshire, 1087 Elm Street, Manchester, New Hampshire.

George Wilson of Black and Veatch, Kansas City, has written an elegant dating program for the MIS-LESS program in the library. Anyone interested may contact him directly.

Well that about rounds things up at this end. Let's keep the pipelines open, who knows, maybe somebody else will actually be interested in what you are doing!

Ed Orth, Secretary
Electric Utility Programs Team

Papers Presented At This Session
Which Are Reprinted in Section 4

A Distribution Feeder and Substation Load Forecasting System - J. C. Hubbard, Baltimore Gas and Electric Company

IBM 1620 Program Used in Study to Select Steam Conditions and Turbine Type for the No. 3 Unit Installation at Herbert A. Wagner Power Plant

Construction is now in progress for the No. 3 Unit Installation at the Baltimore Gas and Electric Company's Herbert A. Wagner Power Plant. The maximum net station sendout from this unit will be about 320,000 kw when operating at 1.5 in. Hg abs condenser pressure. Steam conditions will be 3500 psig - 1000/1000/1000 F. The turbine-generator will be a 3600/1800-rpm cross-compound unit with a double flow low pressure turbine having 40-inch last-stage blades. Net station heat rate at full load and 1.5 in. Hg abs condenser pressure is expected to be about 8500 Btu per kwhr.

Extensive studies were conducted during the spring of 1962 to examine the relative economics of different steam conditions and turbine types. A large portion of these studies were performed on the Baltimore Gas and Electric Company's type 1620 computer located in the Electric Engineers Department. This machine is a Mark I Model with a 20 K memory.

The selection of the above steam conditions and turbine type were based on the results of these studies. Additional details may be found in Volume XXV of the Proceedings of the American Power Conference dated March 26, 27 and 28, 1963 under the title "Economics of the Selection of 3500-Psig Double Reheat for a 300-Mw Unit" by R. W. Davies and G. C. Creel.

R. W. Davies
Senior Engineer
Electric Engineers Department
Baltimore Gas and Electric Company

3.6.7

Attendants at Sessions of
Chemical Engineering Programs Team

Number Attending - 19

A. H. Best

C. C. Brown

W. F. Burleigh

B. Busch-Petersen

H. R. Chancey

R. W. Ellis

H. P. Gelsi

H. Hunt

S. Jurnack

T. S. Latterner

D. A. Lloyd

T. E. Morrisson

D. J. Pearce

H. C. Peterson

I. E. Rosenbarker

C. S. Schrodel - Acting Team Chairman

M. Seewald

W. B. Thompson

L. F. Zimmerman

Report of Activities of
Chemical Engineering Programs Team

The Chemical Engineering session opened with each person present making a short statement about his area of interest. There were no formal papers given, so the session consisted of group discussion of those topics of greatest interest.

Process design calculations and statistical correlations were two areas of general interest.

The problems incurred in attempting to make generalized heat and material balances in process design were shared by many present. Convergance when recycle streams are present was one problem which was brought forth. It was also noted that the greatest complications sometimes arose in simple blending and separating operations, rather than the ostensibly more complex distillation columns, heat exchangers, reactors, etc.

In the process design the correlation of properties is important, and this led into the second area of interest - statistical correlations. There was a demonstration of one case in which multiple linear regression analysis was particularly useful as a method of correlation. In this case, equilibrium constants, normally taken from a nomograph, were computed by means of an equation found by regression analysis from the nomograph.

Enthalpy correlations were also discussed and the problem of poor data on nonideal mixtures was brought up. The concensus was that the engineer in many cases needs better numbers than he now has at his disposal.

This brought us around to the questions of accuracy of data, reproducibility of experiments, design of experiments and quality control. There was a great deal of interest in these topics but time did not permit discussion of them. It is hoped that much more in these areas can be undertaken at the Buffalo meeting in September.

Diana A. Lloyd, Acting Secretary
Chemical Engineering Programs Team

C. S. Schrodel, Acting Chairman
Chemical Engineering Programs Team

Attendants at Sessions of Rate Engineering Programs Team

Number Attending - 14

Ken Brady - Team Chairman

Harry Fitch

Jim Quigley

Charles McKelvey

Berl Springer

Jim Enterkin

Bob Nichols

Ray Van Wuyckhuyse

John Monsees

Ed Cox

Rod Dabe

Dick Davis

Francis McCue

Wanda Stacke

Report of Activities of
Rate Engineering Programs Team

The second meeting of the Rate Engineering Programs Team was held in conjunction with the 1620 Users Group Meeting in Washington, D.C. on May 6 through May 8, 1964.

Ken Brady presented copies of two programs used by Idaho Power Company. Copies of the first page of each program are attached (Attachments 1 and 2). I feel sure Mr. M. E. Byrne would furnish additional information upon request.

Ken also presented a Long Island Lighting Company program which they used in connection with an appliance survey to determine incremental appliance consumptions. Copies of this program are available upon request from Ken.

Jim Enterkin presented a New England Electric Service program. This is primarily a statistical program with the title of "Computation of Coefficient of Correlation (simple, linear correlation) by the short-cut Crude Data Method, Straight Line Regression, and Standard Error of Estimate". Attachment 3 as copied from the program gives a summary of the program. Jim said copies of the program would be available and that it would be possible to furnish a source deck to anyone really interested in the program.

Ray Van Wuyckhuyse of Rochester Gas and Electric discussed their program for the "Adjustment of Sales and Revenues for Weather and Tariff Changes". This program was written for a 1401 and Ray says it taxed a 12K memory. Copies of the paper are available and he will answer any questions regarding the program. Portions of the first pages are marked Attachment 4.

Harry Fitch of LILCO next presented a paper entitled "Crossing Points, Rate Selection and Comparison". This program compares revenues for different rates and is used to help in rate recommendations where a customer has the choice between two rates. Copies may be obtained by contacting Harry.

John Monsees, Con Ed, New York, presented the last paper. The summary of this paper is marked Attachment 5. Copies are available from John.

Three sub-committees were set up by Ken Brady. These are as follows:

I. Bill Analysis

1. Harry Fitch
2. Bob Nichols
3. Ray Van Wuyckhuyse

II. Forecasting

1. Ken Brady
2. Jim Quigley
3. Ed Cox

III. Ogive

1. Jim Enterkin
2. Ken Brady

All members were requested to select an example of the method they use in analyzing customers bills to Committee I.

Ken Brady is to prepare a glossary of rate terms used in writing programs and submit it for approval at our September meeting.

John Monsees is interested in contacting anyone using the computer to determine the Purchased Gas Adjustment.

The next meeting will be held during the Users Group Meeting in Buffalo, September 2, 3 and 4, 1964.

L. E. Cox, Jr., Secretary
Rate Engineering Programs Team

Papers Presented At This Session
Which Are Reprinted in Section 4

1620 NEES Program No. 15 - Computation of Coefficient of Correlation (Simple, Linear Correlation) by the Short-Cut Crude Data Method, Straight Line Regression, and Standard Error of Estimate - R. C. Burrowes and J. E. Enterkin, New England Power Service Company

Crossing Points, Rate Selection and Comparison - H. L. Fitch, Long Island Lighting Company

Electric Water Heating, Incremental Appliance Usage - K. W. Brady, Long Island Lighting Company

Program 421 Rate Analyzer

Summary

This program takes as input the zone report from Billing and under control of the header cards which have a proposed rate in them will recompute the revenue to yield what it would be had the new rate been in force. This program is used to study the effect of rate charges on revenue.

Input

Cards are read in this order:

(a) Title Card
(b) Minimum Card
(c) Rate by blocks (maximum of 10 blocks)
(d) Block Kwh (Maximum 10 blocks)
(e) Any number of zone cards

Cards b, c, d & e are repeated for each schedule. There is a maximum of 5 schedules permitted. There is also a limitation on the total number of blocks in all schedules and this maximum is 36.

Output

The output is a report which turns out the old and new revenue on the zone report then this is reprinted but zoned up on the new blocks. All possible percents and per units are printed out. This is done for each report's schedule. Then a per cent summary is printed out at the end when all percents are per cent of total of all schedules. The detail of the printout can be more clearly described by showing an actual printout which follows.

Attachment #1

PROGRAM 422 AUTOMATIC PLOT OF CUSTOMER KWH ZONES

Summary

This program takes the output of the program designed to accumulate zones from customer bills and sums them up into intervals, scales the sum and punches out a card which when printed 80-80 will yield a graph. There is also punched out at each interval the value of the Ogive curve as well as the actual value of the Kwh zone. A scale is printed out every 24 intervals the zone report used. As input is zoned every 10 Kwh. An interval is 40 Kwh.

Input

The input has a header card and any number of 10 Kwh zone cards. Since the zone card is put out on two cards. Each zone input reads two cards. The format is so arranged so it picks out the data needed from the two cards. In this program the input actually reads three schedules. The first schedule (non-waterheating) is to be plotted and the sum of the second and third schedule to be plotted in a second curve. (Water-heating and total electric). In selecting the scale the highest per cent encountered is considered and the scale to read in 100 divided by the maximum per cent or the number of columns per per cent. If this is written in as a multiple of 50 the scale multiplier will be on even number since the basic scale is 0-2%.

Output

When the output is punched the cards are sorted on Column 73 and the first curve will be in 1 and the second in 2. The output of each is then put on the Fortron printer board for the 407. This is wired so that a 11 punch in Col 72 will cause the card to be printed in Col 60-120 while the card without a 11 punch will print from 0-60.

Attachment #2

NEES PROGRAM NUMBER 15

TITLE: Computation of Coefficient of Correlation (simple, linear correlation) by the short-cut Crude Data Method, Straight Line Regression, and Standard Error of Estimate.

BY: R.C. Burrowes and J.E. Enterkin, New England Power Service Co.

SCOPE: In common usage, the term "correlation" is used to mean the causal effect or co-relationship which exists between variables. The degree to which variables are co-related or influence one another is represented by a computed measure called the "coefficient of correlation." Values of this coefficient range from zero (no correlation at all) to approaching +1 or -1 (approaching perfect positive or perfect negative correlation).

Positive correlation results when the movement of two series of variables is closely related in such a way that they move together. An increase in one series accompanies an increase in the other, or a decrease in one series accompanies a decrease in the other. (For example: the related movement of degree days and heating fuel consumption).

Negative correlation results when one variable increase as the other decreases, such as KWH used and average cents per KWH.

This program will concern itself only with simple linear correlation; the correlation of two variables (one of which is independent and one dependent) to be described by a straight line regression. Suffice to say that there are more elaborate methods which exist to determine the degree of correlation among more than two variables, and for computing curved regression lines.

The method outlined in this program is applicable, subject to core limitations, for any two series (one dependent and one independent) that may logically be assumed to influence one another.

Attachment #3

Rochester Gas and Electric Corporation

ADJUSTMENT OF SALES & REVENUES
FOR WEATHER AND TARIFF CHANGES

PURPOSE OF THE PROGRAM

In many companies the Rate Department is the group which provides management with the theoretical figures that are reasonable answers to the "what would happen if" questions. There is constant experiment with ideas to provide those answers from the data which are available.

A responsibility of our Rate & Economic Research Department is the preparation of forecasts of customers, sales units and revenue dollars by months for seven different classes of business by each type of gas (mixed and straight natural). The Rate Department, for proper evaluation of the forecasts, adjusts the reported figures for prior years to the same standards which are being used for the forecast periods.

The 1964 Forecast presented special problems. The heating classes were to be estimated to the monthly requirements of a so-called "New Normal" as provided by the Weather Bureau. The revenues were to be estimated on the basis of rate schedules that were not yet in effect. All of our prior years data for normal requirements were developed to meet the so-called "Old Normal". The revenue data had been produced under two different sets of prior rate schedules.

There are various ways and means of determining theoretically what revenues might have been if the weather had been normal, and there are several different ways of calculating what revenue might be on a proposed rate schedule. Our Rate Department, late in 1963, began experimenting with a mathematical method of developing these required answers monthly. Because of the tremendous amount of calculation required to test the formulas and to provide adjusted monthly data for several prior years for almost thirty different classes (by revenue code and by rate code), and the need to provide these statistics for the forecast in a very

3.8.7

short period of time, the problem was referred to the Data Processing Group. The job was completed by them from start to finish in about a month's time. The program provided not only statistics for the 1964 Forecast but also provided data for the years 1960 to date on a truly comparable basis, that is, all sales adjusted to the requirements of the "New Normal" and all revenues adjusted to theoretical on the new rate schedules. This program is now providing the Rate Department with monthly sales and revenue figures for "normalized weather."

It is interesting to note here that for the first three months of this year we have had, in total, almost normal heating requirements as related to our billing cycles. With all the gyrations of adjustments - plus and minus - the deviation between the total revenues as reported for these three months and the total revenues as developed for normal weather conditions is only 0.07% of the total. This is certainly encouraging on the adjustment for the weather aspect but we are still considering it to be an experiment. In addition, the rate schedule part must stand the tests of time and statistical methods.

THEORY

There are two theories involved in this program. One has to do with adjustment of sales units for weather. The other is concerned with adjustment of revenues based on the revised sales units and/or a change in rate schedule.

Attachment #4

OBJECTIVE OF THE PROGRAM

The objective of this program is to accumulate the monthly block analyses of the number of gas bills rendered and the associated consumption (in CCF) to an annual total within each block and to a grand total for all the blocks. A cumulative total is also calculated as each successive block is summed. When the last block is processed, the total number of bills and total consumption is computed. From the totals and the accumulated figures the consolidated factor is computed for each block. (See Appendix for details.)

To facilitate checking, the total number of bills and consumption is computed for each month and visually compared with the monthly block analysis figures already available.

The monthly data for bills and consumption is stored in 12 columns, one for each month. By summing the first figures in every column, we obtain an annual total of bills or consumption for the first analysis. (See Appendix for details.)

The present monthly gas analysis involves eight tables representing our two service classifications, our space heating customers and the irregular bills rendered. Because part of Consolidated Edison's service area is subject to a 1% utility franchise tax, separate statistics are maintained on each table for the system and the area for which the tax is not applicable. For the sixteen required tables, whose number of blocks vary from 47 to 57, the total running time on the 1620I, exclusive of listing punched output, is approximately one hour. This compares with a manual time of approximately 115 man-hours.

In the following appendix, the computer application is covered more technically.

Attachment #5

Attendants at Sessions of
Operations Research Programs Team

Number Attending - 71

A. H. Best
Jane Bonnette
Robert Burgess
William F. Burleigh
Harold Chancey
S. A. Clark
R. B. Davis
Dr. Kurt Eisemann
Jerry Eisenburg
Katherine Finnegan
Donald Flagg
Vincent Gangi
J. W. Garzanell
Hector Gelsi
E. H. Gerrish
G. S. Haralampu
F. R. Henderson
Tom Hoke
J. C. Hubbard
Samir S. Husson
Clayton Jensen
S. Kellman
John Kent
Alton Kindred
David Knight
Ted Krenzer
Thomas Latterner
Ralph Lee
Judith Liebman
Joyce C. Little
D. A. Lloyd
J. Y. Louis
Gene R. Lowrimore
Patricia Lussow
Thomas Morrison
Bennett Nunberg
Dr. James R. Oliver - Team Chairman
Brother Edward O'Neill
David Pixley
Pate Pointer
W. W. Pleines
C. H. Remilen
Theodore Sabine
John W. Sawyer
Joseph Schnecker
Maynard Shore
Luke Sparvero
Arnold Spitalny
James Stansbury
R. L. Storrer
Carlis Taylor
William Thompson
John Tiers
B. Wingersky
Joseph Wingert
George Woodruff
Lawrence Wright
Charles Yackulics
K. V. Zajic
H. E. Blaicher
Reginald Harling
Len Karr
Denise Knight
Greta Larson
A. L. Lipson
Wayne Meriwether
Norman Ogilvie
John Owens
Joan Silverman
Welborn Smith
Marilyn Wingersky

Report of Activities of Operations Research Programs Team

The Operations Research Programs Team meeting met with Dr. James R. Oliver serving as Chairman. There were approximately 75 persons in attendance.

Two papers were presented in their entirety: EMAGE - An Economic Matrix Generator for a Distribution Transformer Management Program - by W. W. Pleines, Baltimore Gas and Electric Company (abstract attached), and a Multiple-Iteration Procedure for Engineering Design - by M. L. Baxter, Jr., Gleason Works (presented by Theodore J. Krenzer of Gleason Works).

Because of a time shortage, a third paper by Dr. James R. Oliver, University of Southwestern Louisiana, Lafayette, was presented in an abbreviated form. His paper was entitled, "A Random Number Technique for the Study of Machine Breakdown and Repair". The technique used a random number generator to produce a sequence of random numbers used to determine whether or not breakdown would occur. During the study the probability of breakdown was varied over a considerable range. If breakdown occurred, a test was made to determine whether or not machines were waiting to be repaired. If they were not, the machine undergoing breakdown would begin to be serviced. If other machines were waiting, it was simply added to a line to be serviced as time became available. If no breakdown occurred, a check was made to determine whether machines were waiting to be repaired. If so, work continued on them; if not, the repairmen were then idle. In the case of machines waiting to be repaired, a summation was made of the idle machine hours. When no machines were being repaired, a summation was made of idle man hours. The values recorded by the program were the number of time units used in the experiment, number of breakdowns occurring during the time period, number of time units the repair personnel were not engaged, and the number of time units (accumulated) during which machines were not in operation.

Dr. Oliver pointed out an interesting but as yet undetermined result. When the probability, in per cent, of breakdown multiplied by the time units needed for repair equal approximately 100, a sudden discontinuity was observed in the curve representing the idle machine hours versus time and the idle man hours versus time. Also, a sudden growth of the residual line occurred under similar conditions.

The speaker made it plain that the work reported was of a preliminary nature and that additional investigation was being continued. Interested persons were encouraged to write for further information concerning this study.

There being no further papers to present or business to be transacted, the Chairman declared the meeting adjourned.

Dr. James R. Oliver, Chairman
Operations Research Programs Team

Papers Presented At This Session
Which Are Reprinted in Section 4

A Multiple-Iteration Procedure for Engineering Design - M. L. Baxter, Jr.
Gleason Works

OPTIMUM REPLACEMENT OF CAPACITY-LIMITED EQUIPMENT

W. W. PLEINES

This paper explains the development of a cost-based model for transformer replacement which can be adapted for the replacement of capacity-limited equipment in general. The model, used as a component of the Baltimore Gas and Electric Company's Transformer Load Management System, is contained in a three-part program for the IBM 1620 computer. Using fixed and demand-dependent costs for all sizes and types of transformer in place, it generates opportunity-loss matrices from which it determines, for each possible demand, whether a given unit should be left in place or changed to a larger or smaller unit of standard size.

3.9.4

Attendants at Sessions of
General Data Processing Programs Team

Number Attending - 24

W. F. Burleigh	S. Arnold
T. L. Driscoll	D. D. Williams
R. H. Davis	R. B. Riley
N. Ogilvie	R. E. Raver
G. M. Baber	D. H. MacLeod
K. R. Berger	J. P. Houlihan
E. M. Hamilton	R. A. Cooper
L. Schiffman	A. R. Kindred
W. P. Wynn	M. N. Shore
A. Spitalny	J. Stansbury
P. Lussow	E. H. Parker
S. W. Craig	R. Soucy - Acting Team Chairman

Report of Activities of
General Data Processing Programs Team

The Programs Team Meeting for General Data Processing was called to order by temporary Chairman, Robert Soucy.

Following the theme of the May Meeting, three special session program papers were delivered at the meeting. They were:

a. "Electric Load Statistic System" by E. H. Parker, Baltimore Gas and Electric Company.

b. "A Work Performance Monitoring System" by D. D. Williams, Baltimore Gas and Electric Company.

Both of these papers clearly indicate how the integration of data can be a source of valuable management reports and operating information.

A sample of the management reports produced by these programs were distributed to those attending the meeting.

c. "ZIP - A Multiple Line Keyword in Context Program" by T. J. Scott, Sun Oil Company.

Mr. Scott's program is designed to perform a Keyword (or Keyphrase) in context analysis on a multiple line or card abstract, paragraph or magazine article.

An outline of Mr. Scott's paper and a sample of "Keyword Phrases in Context" output list was distributed to those attending the meeting.

R. Soucy, Acting Chairman
General Data Processing Programs Team

Papers Presented At This Session Which Are Reprinted in Section 4

ZIP - A Multiple Line Keyword in Context Program - T. J. Scott, Sun Oil Company

A Work Performance Monitoring System Designed for an Electric Utility Meter and Installation Department - D. D. Williams, Baltimore Gas and Electric Company

An Electric Load Statistics System for an Electric Utility - E. H. Parker, Baltimore Gas and Electric Company

Attendants at Sessions of
1710 Users Sub-Group

Number Attending - 17

J. J. Owen - Team Chairman

Patricia Lussow

G. F. Hamilton

A. J. Palmieri

A. C. Bailey

F. Wang

T. E. Morrison

S. Jurnack

E. B. Svihla

Carol Braun

W. H. Beck

R. L. Storrer

J. W. Garzanelli

D. Pearce

S. Lopez

R. P. Paterniti

L. Stewart

Report of Activities of
<u>1710 Users Sub-Group</u>

Mr. J. J. Owen, Chairman for the 1710 Users Sub-Group opened the meeting with a welcome to all present, and a call for additional papers at the next meeting.

There were approximately 30 people in the meeting at various times although only 14 users and 3 IBM personnel signed the attendance roster. A list of those signing is attached. It is assumed that these are the people who are most interested in the activities of the Group.

It was announced that plans for the next meeting included a discussion of 1710 Executive systems by IBM specialists.

The program for this session consisted of two integrated presentations covering aspects of system planning. The first paper entitled "Installation Planning for an IBM 1710 System" by Mr. Lawson E. Stewart of the IBM Corporation, Baltimore, Maryland, covered System Planning and 1710 Installation Implementation. Mr. Stewart covered the general subject of project organization and planning. He began with a suggested outline of broad goals for the project:

1. Effective translation of Management plans into a working system.

2. Education for future projects.

3. Meeting of project goals and deadlines with least effort and cost.

With respect to the first point, Mr. Stewart emphasized that most of the less than successful computer control installations throughout the industry were the result of improper definition and understanding of project goals. To prevent this occurrence, Mr. Stewart recommended direct involvement of top management in the project via an initial report covering goals and intended methods of achieving them, plus periodic reports of progress and plan modifications.

Education for future projects, reports Mr. Stewart, stems from a conscious effort to man the project with this view and from a comprehensive system of documentation which covers not only programming but physical planning and all other aspects of the project.

Meeting of goals and deadlines requires careful coordination and control. The most important aspects of this would be selection of proper personnel for each job function on the project team, assurance of complete communication between team members, and use of such planning tools as critical path analysis. On this last point, Mr. Stewart dis-

played a portion of a critical path schedule prepared using the IBM LESS program. He emphasized the need to include all portions of the projects in the time estimates and job descriptions.

Following this portion of the discussion, Mr. Stewart presented outlines of the functions to be considered in any computer control application:

For Known Applications

1. Establish specific goals for this system, long range and short range. What is to be accomplished, by when? What would be the ultimate extension of this application?

2. Organize the basic project team to include these functions or abilities - Project leadership - a leader and administrator, not just a technical specialist. Process knowledge - practical experience plus sound knowledge of fundamentals. Instrumentation - knowledge of measurement techniques and signal treatment. Programming - math and system knowledge.

   Installation - field installation experience and checkout techniques. Documentation - probably a full time job for one person, to maintain documentation standards and assure the satisfactory preparation of all reports and final documentation.

3. Prepare critical path schedule for total project: Takes about three weeks to complete the first time.

4. Establish a list of reports and calculations to be performed.

5. Generate a sensor list.

6. Analyze accuracy requirements for all measurements, so that high precision instruments are used only where needed.

7. Follow a formal schedule of project meetings and benchmark progress reports.

8. Check and plan completely the physical installation.

9. Prepare program flow-special emphasis here on the use of Decision Tables for both program logic planning tool and as program documentation.

10. Program checking by simulation or use of a 1710 system.

11. Conduct electrical noise tests at the installation site.

12. Design the operator's console. This was considered a vital point from the view that poor design would reduce operator cooperation and system effectiveness.

13. Operator training in use of the console and computer program is a must for successful operation and application growth.

14. Model Building Tests are required with just about any computer control application. This phase should be under the direction of an expert in this particular work. This man normally has as a background, practical experience with the operation of the specific process, knowledge of statistical techniques and high-precision measurement.

15. Document all phases of project as you go, and prepare progress reports which serve to inform top management as well as continually selling the advantages of the project. The greatest failing in project teams is the lack of understanding that a project must be continually "sold" to top management in order to keep interest and support for the project.

<u>For New Applications</u>

List included only additional requirements above those for a known application.

1. Study the basic process. Accumulate all known information on theory and practical operation.

2. Run manual tests to establish important process parameters.

3. Establish, by data analysis, potential control areas.

4. Analyze measurement and control practicality.

5. Install system for data acquisition and reduction.

6. Develop math model of the process and verify.

7. Proceed as with known application.

Mr. Stewart concluded with a plea for more presentations in this area as more 1710 systems go into operation.

The second paper of the session entitled "Some Early Experiences in the Design of a System for an Electric Utility Power Plant by Mr. William H. Beck of the Baltimore Gas and Electric Company introduced specific material used in the documentation and planning of a 1710 System for the Herbert A. Wagner Electrical Generating Plant of the Baltimore Gas and Electric Company. Mr. Beck presented copies of the documentation and programming standards adopted by his company for the 1710 installation project. In the discussion of the attached standards, Mr. Beck emphasized the fact that a primary goal of the 1710 installation was the acquisition of knowledge and education in computer control of power plants. For this reason, additional care is going into the development and following of standards for documentation of programs, project planning, writing of reports and logging of data on effort and time required for each aspect of the entire project.

As a specific example of the planning outlined in Mr. Stewart's

discussion, Mr. Beck's presentation was considered by those present to be extremely valuable and exactly the kind of specific, practical material needed for other users about to embark upon an installation.

The meeting concluded after a question and answer period which covered the two presentations and additional material on measurements, etc.

J. J. Owen, Chairman
1710 Users Sub-Group

Note: Formal copies of the papers by Mr. Stewart and Mr. Beck are not available for reprinting within these "Proceedings".

SECTION 4

LIST OF PAPERS

Table of Contents

Table of Contents

(Continued)

Table of Contents

(Continued)

COMPUTATION OF COEFFICIENT OF CORRELATION
(SIMPLE, LINEAR CORRELATION) BY THE
SHORT-CUT CRUDE DATA METHOD,
STRAIGHT LINE REGRESSION, AND
STANDARD ERROR OF ESTIMATE

R. C. Burrowes and J. E. Enterkin
New England Power Service Company

4.1.0

1620

NEES PROGRAM NUMBER 15

I **TITLE:** Computation of Coefficient of Correlation (simple, linear correlation) by the short-cut Crude Data Method, Straight Line Regression, and Standard Error of Estimate.

II **BY:** R.C. Burrowes and J.E. Enterkin, New England Power Service Co.

III **SCOPE:** In common usage, the term "correlation" is used to mean the causal effect or co-relationship which exists between variables. The degree to which variables are co-related or influence one another is represented by a computed measure called the "coefficient of correlation." Values of this coefficient range from zero (no correlation at all) to approaching +1 or -1 (approaching perfect positive or perfect negative correlation).

Positive correlation results when the movement of two series of variables is closely related in such a way that they move together. An increase in one series accompanies an increase in the other, or a decrease in one series accompanies a decrease in the other. (For example: the related movement of degree days and heating fuel consumption).

Negative correlation results when one variable increase as the other decreases, such as KWH used and average cents per KWH.

This program will concern itself only with simple linear correlation; the correlation of two variables (one of which is independent and one dependent) to be described by a straight line regression. Suffice to say that there are more elaborate methods which exist to determine the degree of correlation among more than two variables, and for computing curved regression lines.

The method outlined in this program is applicable, subject to core limitations, for any two series (one dependent and one independent) that may logically be assumed to influence one another.

IV **TEST DATA:** The following data (taken from a sample of New England Utilities) will be utilized to illustrate the 1620 Fortran program designed to calculate the coefficient of correlation, the "Least Squares" straight line regression that best describes the relationship, and the standard error of estimate which indicates the amount of dispersion in the dependent variable, which we have failed to account for by the regression line.

X = independent variable Ult.Customers per pole mile	Y = dependent variable KVA per line transformer
12.	9.3
19.	11.1
21.	10.9
36.	16.4
43.	21.1
49.	20.5
55.	17.8
60.	28.1
65.	23.6
70.	44.5

7 0

4.1.1

105.	30.1
120.	43.4
183.	50.0

V **REFERENCES:**

A. Formulas

1. Coefficient of correlation

$$r = \frac{\sum xy}{\sqrt{\sum x^2 \cdot \sum y^2}}$$

where:

r = coeff. of correlation
$\sum$ = sum of
$\sum xy$ = sum of the individual products of each x and y deviations from their respective means.
$\sum x^2$ = sum of the squares of the x deviations from the mean of X
$\sum y^2$ = sum of the squares of the y deviations from the mean of Y

The short-cut method utilizes the "centering process" (as outlined in appendix D) to arrive at the sum of the squares of the deviations of X, Y and the sum of the products of the deviations X times the deviations Y.

2. Linear Regression Equation

$$T = a + bX, \quad b = \frac{\sum xy}{\sum x^2}, \quad a = \overline{Y} - b\,\overline{X}$$

where:

T = computed trend of Y variable
$\sum xy$ = same as defined above
$\sum x^2$ = same as defined above
$\overline{X}$ = arithmetic mean of X series
$\overline{Y}$ = arithmetic mean of Y series
X = original values of independent variable
Y = original values of dependent variable

As a further short-cut, the two normal equations

$$\sum Y = Na + b\sum X$$
$$\sum XY = a\sum X + b\sum X^2$$

can be reduced to

$$a = \overline{Y} - b\,\overline{X} \quad \text{(origin at } X = 0\text{)}$$

$$b = \frac{\sum xy}{\sum x^2}$$

where:

a = the computed or trend value of Y when X = 0

b = the amount and direction of slope at the point where X = 0

B. BIBLIOGRAPHY

Croxton, F.E., and D.J. Cowden, Applied General Statistics, Prentice-Hall Inc., New York, 1939

Davies, G.R., and D. Yoder, Business Statistics, John Wiley & Sons, Inc., New York, 1949

Mills, F.C., Statistical Methods, Henry Holt & Co., New York, 1938

VI PROGRAM FORMAT:

A. Input Data

The X and Y series may be of any length up to 366 pairs, but there must be one X value for each Y value.

Two F 10.4 formats permit use of X and Y data of six whole numbers and four decimals. However, machine storage capacity and output formats may restrict the use of long series of large digit numbers.

The first entry card must be an "Item" card showing the number of pairs of X and Y variables in the series to follow. This item number must be punched with its units position in Col. 3. (see appendix A).

X, Y data cards follow, and are entered in the following sequence:

X 1	Y 1	Card 1
X 2	Y 2	Card 2
		"
——	——	"
——	——	"
X n	Y n	Card n

with the X 1 value punched in cols. 1 through 10 (the cols. 7,8,9,10 representing four places to the right of the decimal point) and the corresponding Y 1 value punched in cols. 11 through 20 (the cols. 17,18, 19,20 representing four places to the right of the decimal point). Each card contains an X value and its corresponding Y value.

An attempt has been made to make the formats as broad as possible, but on occasion we have experienced an overflow in printing the sum of the squares of the y deviations from the mean of Y (symbolically Σy^2) coded as SOYSQ and printed in output as SUM DEV Y SQ. When this overflow occured, the number was printed in "floating point." This "F-8 overflow error" did not invalidate the computations, it merely shifted the mode of output for this particular value from fixed point to floating point.

B. Symbols Used in the Program are: (See Appendix B)

ITEM = the number of pairs of X and Y items to be correlated
SUMX = ΣX = total of X series
SUMY = ΣY = total of Y series
SMSQX = ΣX^2 = sum of squares of each X value
SMSQY = ΣY^2 = sum of squares of each Y value
SMPRO = ΣXY = sum of the product of each X value multiplied by its corresponding Y value
AVGX = $\overline{X}$ = arithmetic mean of the X series
AVGY = $\overline{Y}$ = arithmetic mean of the Y series
SOXSQ = Σx^2 = sum of the squares of the deviations of each X value from the mean of the X series.
SOYSQ = Σy^2 = sum of the squares of the deviations of each Y value from the mean of the Y series.
SOXY = Σxy = sum of the cross products of the deviations of each X value from the mean of X times the deviation of each Y value from the mean of Y.
R = the coefficient of correlation
T = the computed trend values of Y
A = the Y intercept = $\overline{Y} - B\,\overline{X}$
B = the amount and direction of slope at the point where $X = 0$; or $\Sigma xy \div \Sigma x^2$
D = deviations of actual Y values from the computed trend values of Y.
SMDSQ = ΣD^2 = sum of the squares of the deviations of actual Y values from the computed trend values of Y.
SUMT = sum of the trend values of Y

STDEE = $\sqrt{\dfrac{\Sigma D^2}{\text{COUNT}}}$ = standard error of estimate

SQRTF = square root
READ = read via punched card
PUNCH = punch output on card
PRINT = print output on typewriter
COUNT = ITEM

C. OUTPUT DATA (See Appendix C)

During the course of the program execution, the value of the coefficient of correlation R, the trend equation constants A and B, and the count of the number of pairs of X and Y data are printed on the typewriter as soon as they are computed to provide immediate answers.

Complete statistics are punched on cards for separate listing, and these include the following:

R = coef. of correlation
A = regression constant } Equation, Y = A + BX
B = regression constant }
Count = no. of pairs of X and Y data
List of X, Y, Trend Values, Deviations from Trend
Std. Error of Estimate
Mean of X series
Mean of Y series
Sum of Dev. of X, squared
Sum of Dev. of Y, squared
Sum of Dev. of XY
Sum of Dev. from Trend, squared

NOTE:
The standard error of estimate is a measure, in absolute terms, of the variation of the actual values of the dependent variable (Y) from their estimated values. It would be expected that 68% of the actual Y values would fall within $\pm$ 1 std. error est. of the regression line, 95% within $\pm$ 2 std. error est. and 99.7% $\pm$ 3 std. error of est.

ANALYSIS OF VARIANCE

The std. error of estimate, when squared, represents the amount of variance unexplained by the regression line.

The Sum DEV Y SQ divided by Count or $\left(\frac{\sum y^2}{N}\right)$ represents total variance.

Total variance minus unexplained variance = explained variance.

The explained and unexplained variance can be turned into % of total variance for quantitative analysis of the success of predicting from the regression equation.

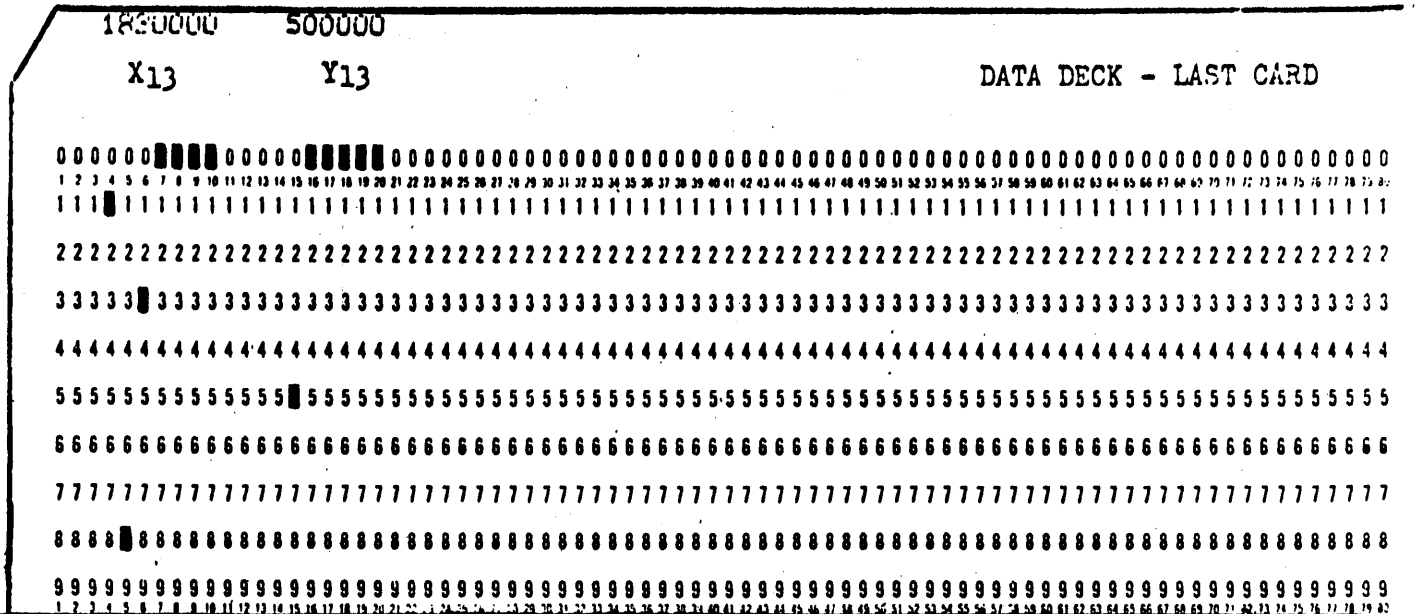
The square of the coeff. of correlation also equals the % of explained variance.

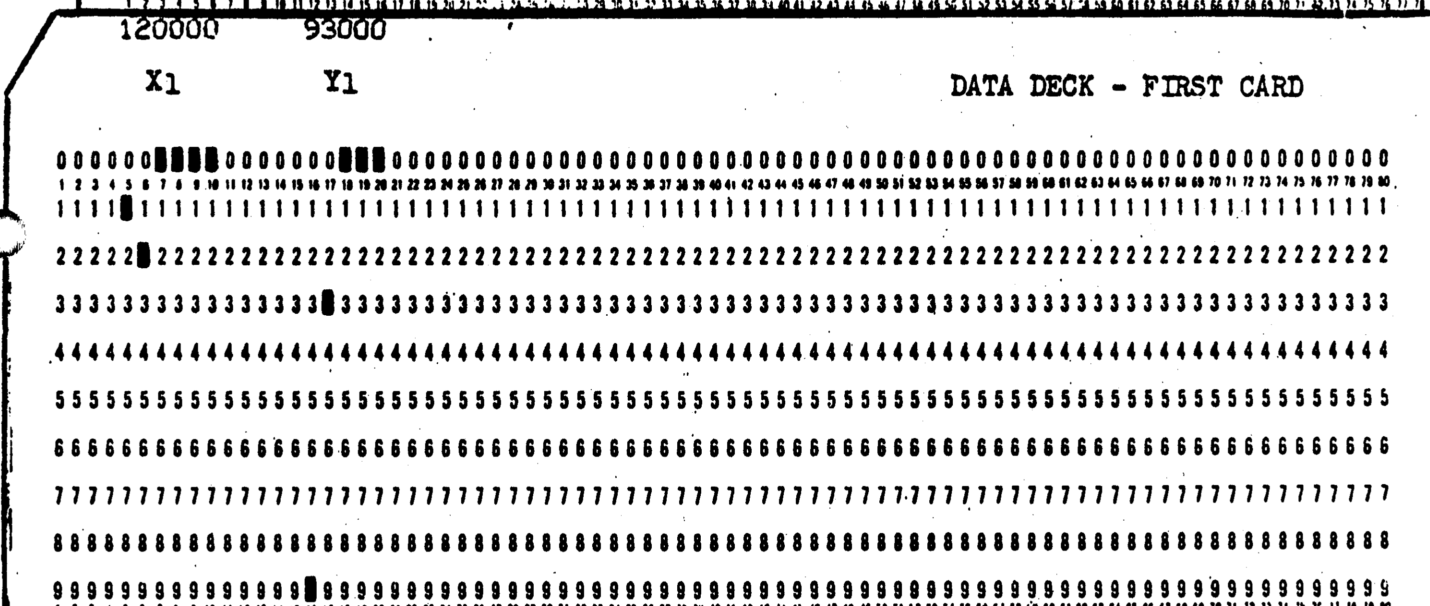
Although the analysis of variance has not been programmed, the necessary components are readily available and the hand calculation is so minor that it was thought to be worth including to round out the correlation analysis.

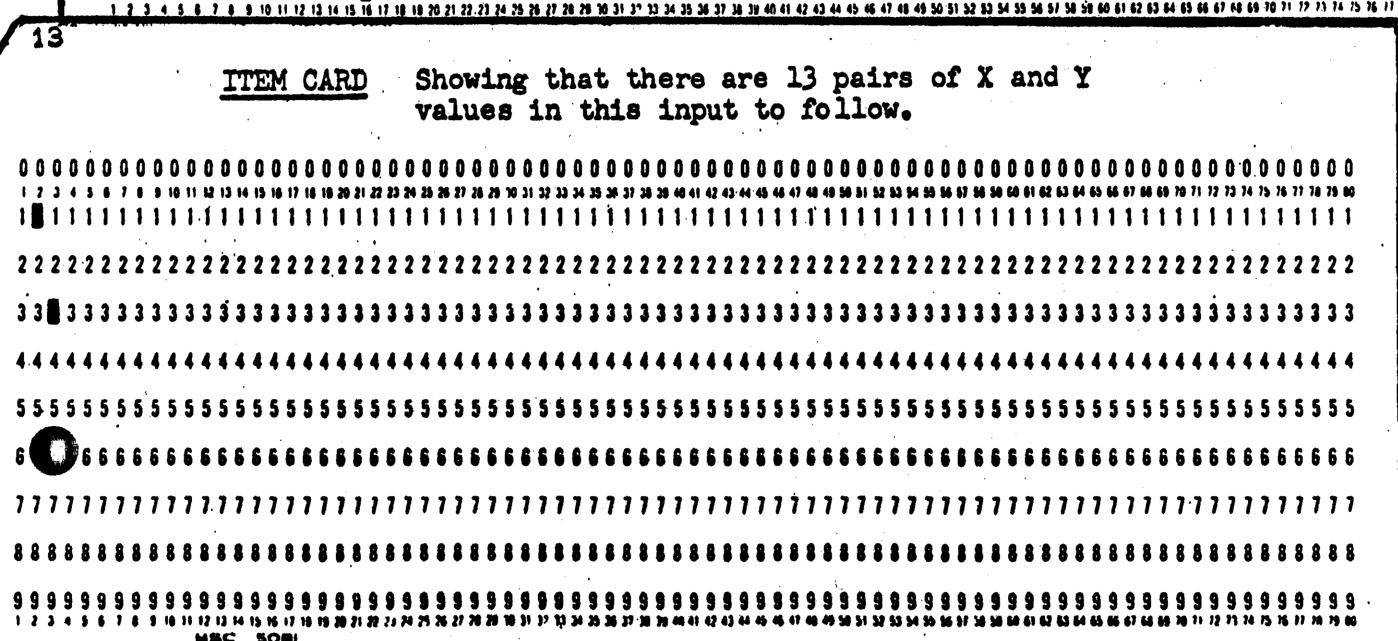
JEE/mib
3-11-64

APPENDIX A

ILLUSTRATION OF INPUT CARDS







4-1.6

```
08300 C      COEF. OF CORRELATION, STRAIGHT LINE REGRESSION AND STD.ERROR ES
08300 C      R.BURROWES AND J.ENTERKIN     JANUARY 1964
08300        DIMENSION X(366), Y(366)
08300     10 SUMX = 0.
08324        SUMY = 0.
08348        SMSQX =0.
08372        SMSQY = 0.
08396        SMPRO = 0.
08420        READ1,ITEM
08444      1 FORMAT (I3)
08466        COUNT = ITEM
08502        DO100  I=1,ITEM
08514    100 READ 2, X(I), Y(I)
08634      2 FORMAT (2F10.4)
08662        DO101  I=1,ITEM
08674        SUMX = SUMX + X(I)
08734        SUMY = SUMY + Y(I)
08794        SMSQX = SMSQX + X(I)**2
08866        SMSQY = SMSQY + Y(I)**2
08938    101 SMPRO = SMPRO + X(I)*Y(I)
09070        AVGX = SUMX/COUNT
09106        AVGY = SUMY/COUNT
09142        SOXSQ =SMSQX - AVGX * SUMX
09202        SOYSQ =SMSQY - AVGY * SUMY
09262        SOXY  =SMPRO - AVGX * SUMY
09322        R = SOXY/(SQRTF(SOXSQ * SOYSQ))
09382        B = SOXY/SOXSQ
09418        A = AVGY - B * AVGX
09478        PRINT145,R,ITEM
09514        PUNCH145,R,ITEM
09550    145 FORMAT(1X14HR                =,F14.4/1X14HCOUNT        =,I9/)
09676        PRINT147,A,B
09712        PUNCH147,A,B
09748    147 FORMAT(1X14HEQUATION,  Y =,F17.7,3H + ,F17.7,4H * X/)
09852        PUNCH148
09876    148 FORMAT(10X1HX,16X1HY,14X5HTREND,13X4HDEV./)
10082        SUMT =0.
10106        SMDSQ =0.
10130        SMSQX=0.
10154        DO102 I =1,ITEM
10166        T = A + B * X(I)
10238        D= Y(I)- T
10298        SMSQX=SMSQX+D
10334        SMDSQ = SMDSQ+D**2
10382        SUMT = SUMT + T
10418    102 PUNCH 48, X(I), Y(I), T,D
10562     48 FORMAT(1X4F17.7)
10608        STDEE = SQRTF (SMDSQ/COUNT)
10656        PUNCH 48, SUMX, SUMY, SUMT,SMSQX
10716        PUNCH47,STDEE,AVGX
10752     47 FORMAT(///1X14HSTD ERR.     =,F17.7/1X14HMEAN X        =,F17.7)
10888        PUNCH50,AVGY,SOXSQ
10924     50 FORMAT(1X14HMEAN Y        =,F17.7/1X14HSUM DEV X SQ.=,F17.7)
11044        PUNCH52,SOYSQ,SOXY
```

```
T1080     52 FORMAT(1X14HSUM DEV Y SQ.=,F17.7/1X14HSUM DEV XY   =,F17.7)
T1200        PUNCH54,SMDSQ
T1224     54 FORMAT(1X14HSUM DEV D SQ.=,F17.7)
T1290        GO TO 10
T1298        END

PROG SW1 OFFFOR SYMBOL TABLE, PUSH START
T9989 SINF        T9979 COS         T9969 COSF              T9999 SIN
T9949 ATANF       T9939 EXP         T9929 EXPF              T9959 ATAN
T9909 LOGF        T9899 SQRT        T9889 SQRTF             T9919 LOG
T9869 ABSF        T9859 X      T6209                        T9879 ABS
T2539 0010        T2529 SUMX        T2519 0000000099        T6199 Y      T2549
T2509 SUMY        T2499 SMSQX       T2489 SMSQY             T2479 SMPRO
T2469 0001        T2459 0001        T2449 ITEM              T2439 COUNT
T2429 0100        T2419 I           T2409 0002              T2399 0002
T2389 0101        T2379 000         T2369 0002              T2359 AVGX
T2349 AVGY        T2339 SOXSQ       T2329 SOYSQ             T2319 SOXY
T2309 R           T2299 001         T2289 B                 T2279 A
T2269 0145        T2259 0145        T2249 0147              T2239 0147
T2229 0148        T2219 0148        T2209 SUMT              T2199 SMDSQ
T2189 0102        T2179 T           T2169 D                 T2159 0048
T2149 0048        T2139 STDEE       T2129 0047              T2119 0047
T2109 0050        T2099 0050        T2089 0052              T2079 0052
T2069 0054        T2059 0054
SW 1 OFF TO IGNORE SUBROUTINES, PUSH START

PROCESSING COMPLETE
```

APPENDIX C

R	#	.8836
COUNT	#	13

EQUATION, Y # 8.8943910 & .2507991 * X

	X	Y	TREND	DEV.
	12.0000000	9.3000000	11.9039810	-2.6039810
	19.0000000	11.1000000	13.6595750	-2.5595750
	21.0000000	10.9000000	14.1611730	-3.2611730
	36.0000000	16.4000000	17.9231610	-1.5231610
	43.0000000	21.1000000	19.6787560	1.4212440
	49.0000000	20.5000000	21.1835510	-.6835510
	59.0000000	17.8000000	23.6915430	-5.8915430
	60.0000000	28.1000000	23.9423420	4.1576580
	65.0000000	23.6000000	25.1963380	-1.5963380
	70.0000000	44.5000000	26.4503340	18.0496660
	105.0000000	30.1000000	35.2283050	-5.1283050
	120.0000000	43.4000000	38.9902930	4.4097070
	183.0000000	50.0000000	54.7906410	-4.7906410
column total →	842.0000000	326.8000000	326.7999700	.0000070

STD ERR.	#	6.0625265
MEAN X	#	64.7692300
MEAN Y	#	25.1384610
SUM DEV X SQ.	#	27076.3080000
SUM DEV Y SQ.	#	2180.9110000
SUM DEV XY	#	6790.7160000
SUM DEV D SQ.	#	477.8049700

COMPUTATION OF COEFFICIENT OF CORRELATION

(Short-cut crude data method)

	(1) X INDEPENDENT VARIABLE ULT. CUSTOMER PER POLE MILE	(2) Y DEPENDENT VARIABLE KVA PER LINE TRANSF.	(3) X^2	(4) XY	(5) Y^2	(6) REGRESSION Yc VALUES COMPUTED	(7) DEVIATIONS OF ACTUAL Y VALUES FROM REGRESSION OR COMPUTED Y VALUES
			SQUARES & PRODUCTS				
	12	9.3	144	111.6	86.49	11.9	- 2.6
	19	11.1	361	210.9	123.21	13.7	- 2.6
	21	10.9	441	228.9	118.81	14.2	- 3.3
	36	16.4	1,296	590.4	268.96	17.9	- 1.5
	43	21.1	1,849	907.3	445.21	19.7	+ 1.4
	49	20.5	2,401	1,004.5	420.25	21.2	- .7
	59	17.8	3,481	1,050.2	316.84	23.7	- 5.9
	60	28.1	3,600	1,686.0	789.61	23.9	+ 4.2
	65	23.6	4,225	1,534.0	556.96	25.2	- 1.6
	70	44.5	4,900	3,115.0	1,980.25	26.5	+18.0
	105	30.1	11,025	3,160.5	906.01	35.2	- 5.1
	120	43.4	14,400	5,208.0	1,883.56	39.0	+ 4.4
	183	50.0	33,489	9,150.0	2,500.00	54.8	- 4.8
TOTALS	842	326.8	81,612	27,957.3	10,396.16	326.9	Σ + deviations=28.0 Σ - deviations=28.1
MEANS	$\bar{X}$ = 64.77	$\bar{Y}$ = 25.14					
CORRECTIONS*			54,536	21,166.8	8,215.75		
CENTERED			27,076 Σx^2	6,790.5 Σxy	2,180.41 Σy^2		

$\bar{X}$ = arithmetic mean of X

$\bar{Y}$ = arithmetic mean of Y

Σ = sum of

* Corrections

64.77 x 842 = 54,536
64.77 x 326.8 = 21,166.8
25.14 x 326.8 = 8,215.75

$$r = \frac{\Sigma xy}{\sqrt{\Sigma x^2 \cdot \Sigma y^2}} = \frac{6{,}790.5}{\sqrt{(27{,}076)\cdot(2{,}180.41)}}$$

$$= \frac{6{,}790.5}{\sqrt{59{,}036{,}781.16}} = \frac{6{,}790.5}{7{,}683.54} = +0.8838$$

Coefficient of Correlation

COMPUTATION OF REGRESSION LINE

$$b = \frac{\Sigma xy}{\Sigma x^2} = \frac{6,790.5}{27,076}$$
$$= .2508$$

$$a = \bar{Y} - b\bar{X} = 25.14 - (.2508)(64.77)$$
$$= 25.14 - 16.24$$
$$= 8.90$$

Trend Line or $Y_c = a + bX$ Origin at $X = 0$
$$= 8.90 + .2508\ X$$

Equation for straight line $Y_c = a + bX$, where:
- X = independent variable
- Y_c = computed Y value or trend value of the dependent variable
- a = the Y_c value when $X=0$
- b = the amount and direction of slope at the point where $X=0$

Substituting the original values of X in the above equation $Y_c = a+bX$, the estimated Yc values of the regression line are computed. For example:

$$Y_c = a+bX = 8.9 + (.2508)(12) = 11.9$$

$$Y_c = a+bX = 8.9 + (.2508)(183) = 54.8$$ (See Column 6

At the point where original X value = 12 the computed or regression value of Y = 11.9, and where original X value = 183 the computed regression value of Y = 54.8. Connecting these two computed Y values by a straight line gives the regression line.

Note that the sum of the regression or computed Yc values approximates the sum of the given Y values (See column 6). Also, the sum of the deviations of the dots above the regression line approximates the sum of the deviations of the dots below the regression line (See column 7). Thus, the regression line calculated by the "method of least-squares" is termed the line of best fit.

The coefficient of correlation has been computed and found to be +.8838. Using the knowledge that +1.00 is perfect positive correlation and -1.00 represents perfect negative correlation, it might be interpreted that +.8838 represents fairly high positive correlation between ultimate customers per pole mile and KVA per line transformer. However, there remains a final computation to test the significance of the coefficient of correlation determined from these two series and appraise its value -- whether it be real and significant correlation or merely due to chance.

The test of significance is based upon the "null hypothesis" which assumes that no correlation exists between the series analyzed. Chance values of "r" the coefficient of correlation or related measures (F) are calculated and evaluated in a special table of F values. From this table (found in Davies and Yoder, BUSINESS STATISTICS, pp. 586-7, or Croxton and Cowden, APPLIED GENERAL STATISTICS, pp. 878-9 -- see reference sources) the value of F computed from the coefficient of correlation derived from any correlation analysis can be compared with F values expected due to "chance" (chance operating 5 times out of 100) and due to "significant co-relation" (chance operating 1 or less time out of 100). If our computed F value disproves the "null hypothesis", then we have real correlation.

The F table lists values for .05 or chance level, and .01 or significant level. Note that the F values are in inverse relationship to the probability of chance or the chance level. The greater the opportunity of chance the smaller the F value and vice versa. The table can be interpreted as follows:

1) If the computed F value for a particular problem is less than the .05 value of F, then the correlation can be said to be due to chance.

2) If the computed F value for a particular problem is greater than the .05 value of F but less than the .01 value, then the correlation may be termed "significant".

3) If the computed F value for a particular problem is greater than the .01 value of F, then the correlation may be interpreted as being "highly significant".

TEST OF SIGNIFICANCE

Computation and Interpretation of F value for our problem:

$$F = \frac{r^2}{1-r^2} \cdot \frac{(N-m)}{(m-1)}$$

$$= \frac{.8838^2}{1-.8838^2} \cdot \frac{(13-2)}{(2-1)}$$

$$= \frac{.78110244}{1-.78110244} \cdot (11)$$

$$= \frac{.78110244}{.21889756} \cdot (11)$$

$$= 3.5683 \cdot (11)$$

$$= 39.25$$

Symbol explanation

r^2 = coefficient of correlation squared

N = the number of pairs of items correlated (13 in our problem)

m = the number of series of variables correlated (2 in our problem)

From F table where columns are (m-1) and lines are (N-m), we find the F values for our problem in Column #1 (2-1) and Line #11 (13-2). Each line number has two values of F, the .05 level and the .01 level (in Davies and Yoder, pp. 586-587). In Croxton and Cowden (Appendix G-2) the .001 level is shown in addition to the .05 and .01 F values.

CHANCE LEVEL	F VALUE FOR 2 SERIES OF 13 PAIRS OF ITEMS (DAVIES AND YODER)
.05 (chance operating 5 times out of 100) or chance level of F	= 4.84
.01 (chance operating 1 time out of 100) or significant level of F	= 9.65

Since our computed value of F = 39.25 and is a value greater than 9.65, the F value of the .01 or significant level, chance is operating less than 1 time out of 100. Therefore, it must be concluded that the coefficient of correlation of +.8838 determined from 13 pair of items of Ultimate Customers per Pole Mile and KVA per Line Transformer taken from the 1957 T & D Study is highly significant. In other words, the items correlated definitely prove that a real cause and effect relationship exists. The increase in KVA per Line Transformer is influenced by the increase in the number of Ultimate Customers per Pole Mile. The F test further indicates that our sample of 13 pair of items, while perhaps a small sample, was large enough to prove conclusively that real correlation exists between these two series.

On the other hand, if the computed value of F had been smaller than the .05 level of F, then it would have indicated that there was no real correlation present, and either the coefficient of correlation would have had to have been larger than .8838 in order to be significant or that our sample of 13 pair of items was not large enough to prove that real correlation existed.

The foregoing hasbeen a brief attempt to acquaint the reader with some of the statistical terms associated with correlation analysis, to illustrate a method of computing the coefficient of correlation and regression line, to test the significance of the results, and to point out some of the limitations in basing one's interpretation of the degree of correlation solely upon the computed value of "r" (the coefficient of correlation).

81a

4.1.13

CROSSING POINTS

RATE SELECTION AND COMPARISON

H. L. Fitch

Long Island Lighting Company

CROSSING POINTS

Rate Selection and Comparison

H. L. Fitch - LILCO

Rate design and simplification is our business. We are constantly reviewing our structure searching for improved and reasonable rates, not only to reduce our work load and expense, but also for clarity and understanding by our consumers. LILCO is no exception.

I would like to refer to the N. Y. S. Public Service Commission's memorandum adopted January 31, 1955 quote - "--After thorough consideration of the various methods of bringing about the desired rate uniformity and tariff simplification, it is the judgement of the Commission that, with the two exceptions of service from underground lines and seasonal service, there should be one tariff schedule for the entire territory for each of the major categories of service, residential, general or commercial, and industrial."

Since that time our department has been endeavoring to attain this goal. Combining or cancelling rates, because of the severe effect on customer's bills, must be done only after a complete review of each account affected and with steps of justification.

Faced with this problem we have for the past five years spent endless days of clerical computations examining our conflicting rate with the ultimate goal of eliminating it from the schedule. Gradually through careful rate selection and reductions, utilizing fully our IBM 1620 digital computer, we are on the door step of attaining complete rate uniformity and tariff simplification.

This program written in FORMAT FORTRAN for card input into an IBM 1620 digital computer, will enable you to compare rates by producing crossing points and bill tables, if desired, in a matter of minutes.

Exhibit I (two pages) is a self-explanatory flow diagram.

Exhibit II (three pages) is the program written in FORMAT FORTRAN

Exhibit III is a sample of the input data card and a glossary of terms and switches used.

Exhibit IV is a sample of the header cards.

Exhibit V is a sample of the output data generated by the program.

Exhibit VI is an example of printed output data in steps of 50 KW Demand

Exhibit VII is a like example of printed output data in steps of 1 KW Demand

Exhibit VIII is a sample of a bill table generated by the program if desired.

This program uses as input data, a single card designed by you containing a demand punch and a time punch. Together with the utilization of switches this offers unlimited flexibitity as to starting point and output desired.

Crossing Points (cont'd)

Naturally the complexity of the rates to be compared will determine the input scope. In explaining our Crossing Points program to you, allow me to refer to the specific examples exhibited. Please feel free to ask any questions.

Program Utilization

With reference to exhibit II, this program was initiated to compare S.C. No 2 with S.C. No. 4 by calculating the difference in the billing of a specific demand in steps of hours use. Incorporating switch 1 for Primary or Secondary service, switch 2 to increment the demand after the 730 hour cycle and switch 3 as an option to obtain a bill table, we were able to produce Crossing points to graphically show the advantages of our rate 2 vs rate 4. Actually we are comparing four rates:

Rate 2 secondary vs. Rate 4 scondary

Rate 2 primary vs. Rate 4 primary

A complete cycle of 730 hours (avg. hours in avg. month), selecting each time the proper rate formula (9 - rate 2, 4 - rate 4), and generating output cards on either Crossing Points or bill table, requires approximately eight minutes machine time. A review of both Exhibits I (flow diagram) together with Exhibit II (the program) will clearly detail the sequence of a single cycle.

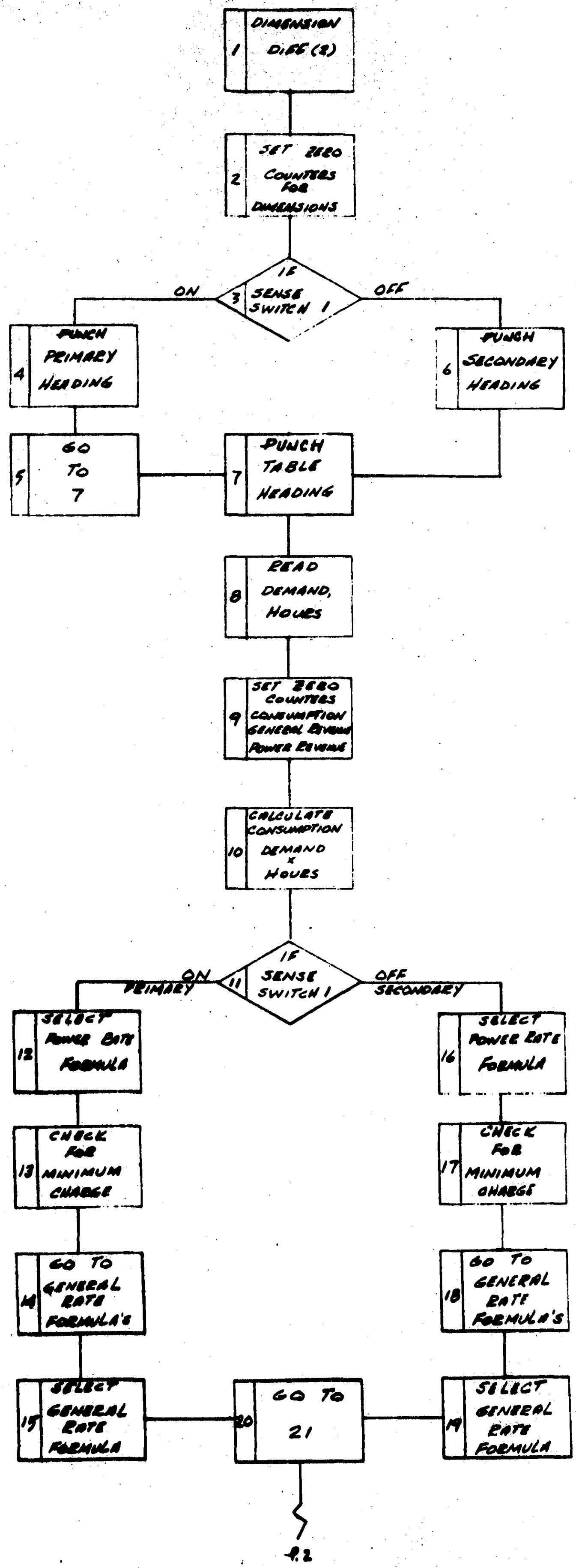
The output data, printed examples (VI, VII, VIII) was generated from one input card (Exhibit III), manually operating the switches at will, in approximately 40 minutes.

Let us examine this output. - (Blackboard Exhibit)

Conclusion:

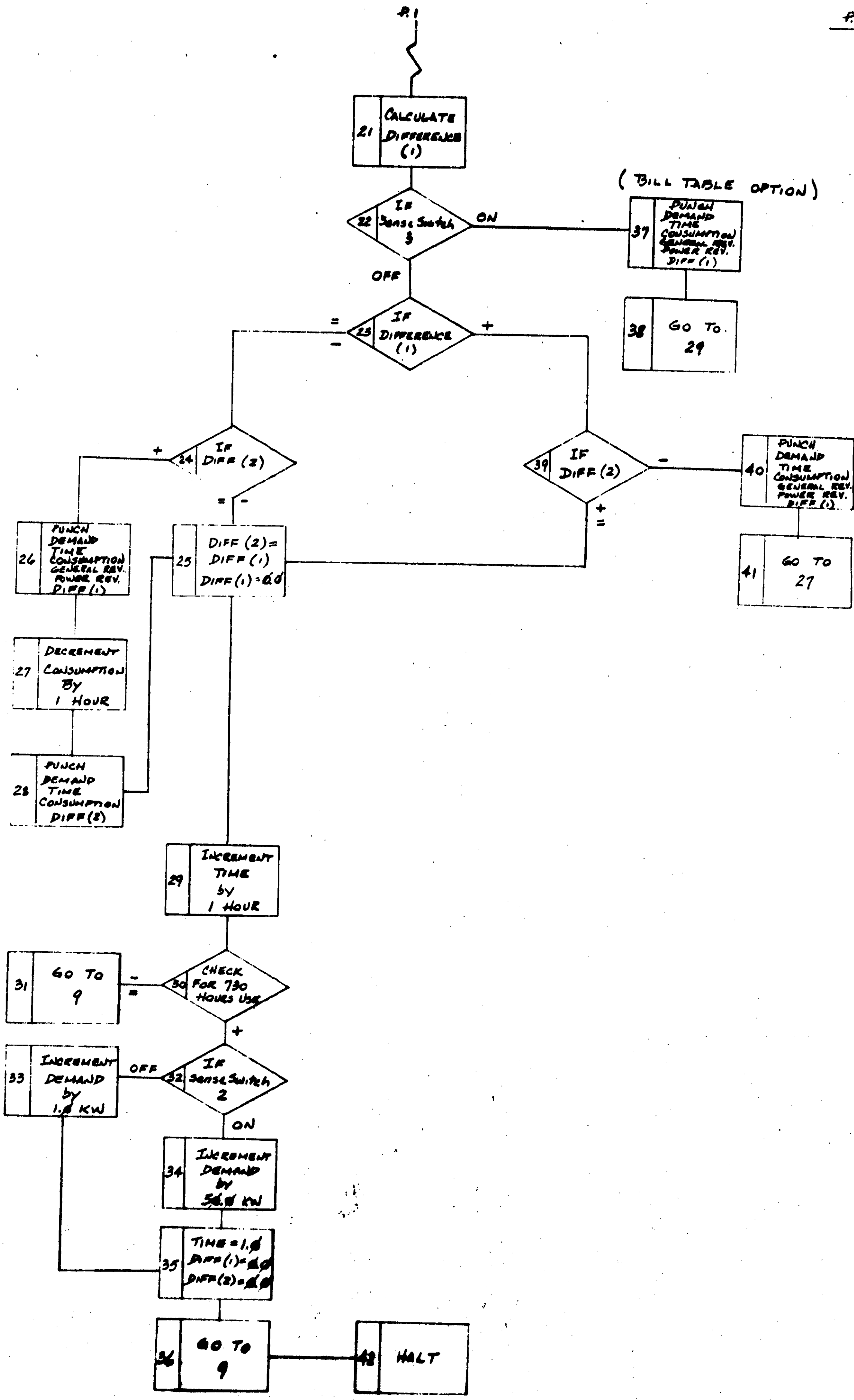
This demonstration, however brief, indicates the advantage of 1620 digital computer application in rate comparison or rate selection work. Possibly some of you are presently confronted with this problem. Although this program was designed to satisfy our demand, only slight modification would be necessary to solve innumerable comparison problems. The general FORMAT is here for general use by any member.

CROSSING POINTS 2 in 4
EXHIBIT I



4.2.3

85


P.1
21 CALCULATE DIFFERENCE (1)
22 IF Sense Switch 3
ON
OFF
(BILL TABLE OPTION)
37 PUNCH DEMAND TIME CONSUMPTION GENERAL REV. POWER REV. DIFF (1)
38 GO TO 29
23 IF DIFFERENCE (1)
=
−
+
24 IF DIFF (2)
39 IF DIFF (2)
40 PUNCH DEMAND TIME CONSUMPTION GENERAL REV. POWER REV. DIFF (1)
41 GO TO 27
26 PUNCH DEMAND TIME CONSUMPTION GENERAL REV. POWER REV. DIFF (1)
25 DIFF (2) = DIFF (1) DIFF (1) = 0.0
27 DECREMENT CONSUMPTION BY 1 HOUR
28 PUNCH DEMAND TIME CONSUMPTION DIFF (2)
29 INCREMENT TIME BY 1 HOUR
30 CHECK FOR 730 HOURS USE
31 GO TO 9
32 IF Sense Switch 2
33 INCREMENT DEMAND BY 1.0 KW
34 INCREMENT DEMAND BY 50.0 KW
35 TIME = 1.0 DIFF (1) = 0.0 DIFF (2) = 0.0
36 GO TO 9
42 HALT

4.2.4

EXHIBIT II
P. 1 of 3

```
C   CROSSING POINTS S.C.NO.2 VS S.C.NO.4 EFF 6/1/64        H.L.FITCH 3/64
C   SWITCH 1 ON FOR PRIMARY, OFF FOR SECONDARY
C   SWITCH 2 ON FOR INCREMENTS OF 50KW, OFF FOR INCREMENTS OF 1KW
C   SWITCH 3 ON - BILL TABLE, OFF BY-PASS
      DIMENSION DIFF(2)
      DIFF(1)=0.0
      DIFF(2)=0.0
      IF(SENSE SWITCH 1)906,907
  906 PUNCH 921
      GO TO 904
  907 PUNCH 920
  904 PUNCH 1
      PUNCH 4
      READ 3,D,T
  104 C=0.0
      GR=0.0
      PR=0.0
      C=D*T
      IF(SENSE SWITCH 1)5,6
    6 IF(D-100.)7,7,8
    7 IF(C-D*30.)9,9,10
    9 PR=1.800*D
      GO TO 500
   10 IF(C-D*120.)11,11,12
   11 PR=0.851*D+0.0233*C
  620 IF(PR-1.800*D)9,9,14
   14 PR=PR
      GO TO 500
   12 IF(C-72000.)15,15,16
   15 PR=2.027*D+0.0135*C
      GO TO 620
   16 PR=2.027*D+0.0085*C+360.000
      GO TO 620
    8 IF(C-D*30.)18,18,19
   18 PR=1.800*D
      GO TO 500
   19 IF(C-D*120.)20,20,21
   20 PR=0.801*D+0.0233*C+5.000
   27 IF(PR-1.800*D)18,18,22
   22 PR=PR
      GO TO 500
   21 IF(C-72000.)23,23,24
   24 IF(D-200.)25,25,26
   23 PR=1.977*D+0.0135*C+5.000
      GO TO 27
   25 PR=1.977*D+0.0085*C+365.000
      GO TO 27
   26 IF(C-360.*D)23,23,29
   29 PR=3.777*D+0.0085*C+5.000
      GO TO 27
  500 IF(C-90.)501,501,502
  501 GR=1.500*D+0.039*C-2.588
      GO TO 1000
  502 IF(C-210.)503,503,504
  503 GR=1.500*D+0.031*C-1.868
      GO TO 1000
  504 IF(C-1500.)505,505,506
  505 GR=1.500*D+0.025*C-0.608
```


4.25

P' 2 of 3

```
      GO TO 1000
  506 IF(C-12000.)507,507,508
  507 GR=1.500*D+0.019*C+8.392
      GO TO 1000
  508 IF(C-36000.)509,509,512
  509 GR=1.500*D+0.013*C+80.392
      GO TO 1000
  512 IF(D-100.)513,513,514
  513 GR=1.500*D+0.008*C+260.392
      GO TO 1000
  514 IF(C-360.*D)509,509,515
  515 GR=3.300*D+0.008*C+80.392
      GO TO 1000
    5 IF(D-100.)600,600,601
  600 IF(C-D*30.)602,602,603
  602 PR=1.800*D
      GO TO 700
  603 IF(C-D*120.)604,604,605
  604 PR=0.551*D+0.0233*C
  609 IF(PR-1.800*D)602,602,606
  606 PR=PR
      GO TO 700
  605 IF(C-72000.)607,607,608
  607 PR=1.727*D+0.0135*C
      GO TO 609
  608 PR=1.727*D+0.0085*C+360.000
      GO TO 609
  601 IF(C-D*0.)611,611,612
  611 PR=1.800*D
      GO TO 700
  612 IF(C-D*120.)613,613,614
  613 PR=0.501*D+0.0233*C+5.000
  610 IF(PR-1.800*D)611,611,615
  615 PR=PR
      GO TO 700
  614 IF(C-72000.)616,616,617
  617 IF(D-200.)618,618,619
  616 PR=1.677*D+0.0135*C+5.000
      GO TO 610
  618 PR=1.677*D+0.0085*C+365.000
      GO TO 610
  619 IF(C-360.*D)616,616,622
  622 PR=3.477*D+0.0085*C+5.000
      GO TO 610
  700 IF(D-100.)701,701,702
  701 IF(C-90.)703,703,704
  703 GR=1.350*D+0.039*C+4.912
      GO TO 1000
  704 IF(C-210.)705,705,706
  705 GR=1.350*D+0.031*C+5.632
      GO TO 1000
  706 IF(C-1500.)707,707,708
  707 GR=1.350*D+0.025*C+6.892
      GO TO 1000
  708 IF(C-12000.)709,709,710
  709 GR=1.350*D+0.019*C+15.892
      GO TO 1000
  710 IF(C-36000.)711,711,714
  711 GR=1.350*D+0.013*C+87.892
      GO TO 1000
  714 GR=1.350*D+0.008*C+267.892
      GO TO 1000
  702 IF(C-90.)803,803,804
  803 GR=1.200*D+0.039*C+19.912
      GO TO 1000
  804 IF(C-210.)805,805,806
```


4.2.6

P. 3 of 3

```
  805 GR=1.200*D+0.031*C+20.632
      GO TO 1000
  806 IF(C-1500.)807,807,808
  807 GR=1.200*D+0.025*C+21.892
      GO TO 1000
  808 IF(C-12000.)809,809,810
  809 GR=1.200*D+0.019*C+30.892
      GO TO 1000
  810 IF(C-36000.)811,811,814
  811 GR=1.200*D+0.013*C+102.892
      GO TO 1000
  814 IF(D-00.)815,815,816
  815 GR=1.200*D+0.008*C+282.892
      GO TO 1000
  816 IF(C-360.*D)811,811,817
  817 GR=3.000*D+0.008*C+102.892
 1000 DIFF(1)=GR-PR
      IF(SENSE SWITCH 3)2006,1500
 1500 IF(DIFF(1))2000,2000,2001
 2000 IF(DIFF(2))2005,2005,2006
 2006 PUNCH 101,D,T,,GR,PR,DIFF(1)
      IF(SENSE SWITCH 3)903,2500
 2500 CC=D*(T-1.0)
      TT=T-1.0
      PUNCH 5005,D,TT,CC,DIFF(2)
 2005 DIFF(2)=DIFF(1)
      DIFF(1)=0.0
      GO TO 903
 2001 IF(DIFF(2))5000,2005,2005
 5000 PUNCH 101,D,T,C,GR,PR,DIFF(1)
      GO TO 2500
  903 T=T+1.0
      IF(T-730.)104,104,105
  105 IF(SENSE SWITCH 2)41,42
   42 D=D+1.0
      GO TO 43
   41 D=D+50.0
   43 T=1.0
      IF(SENSE SWITCH 3)104,1501
 1501 DIFF(1)=0.0
      DIFF(2)=0.0
      GO TO 104
    1 FORMAT(41H   KW   HRS    KWH   S.C.NO.2    S.C.NO.4  ,10HDIFFERENCE)
  101 FORMAT(F5.0,F5.0,F8.0,F10.2,F10.2,F8.2)
    3 FORMAT(F5.0,F5.0)
  920 FORMAT(18H SECONDARY SERVICE)
  921 FORMAT(16H PRIMARY SERVICE)
    4 FORMAT(1X)
 5005 FORMAT(F5.0,F5.0,F8.0,20XF8.2)
      END
```


4.-7

CROSSING POINTS S.C. No. 2 vs. 4

GLOSSARY

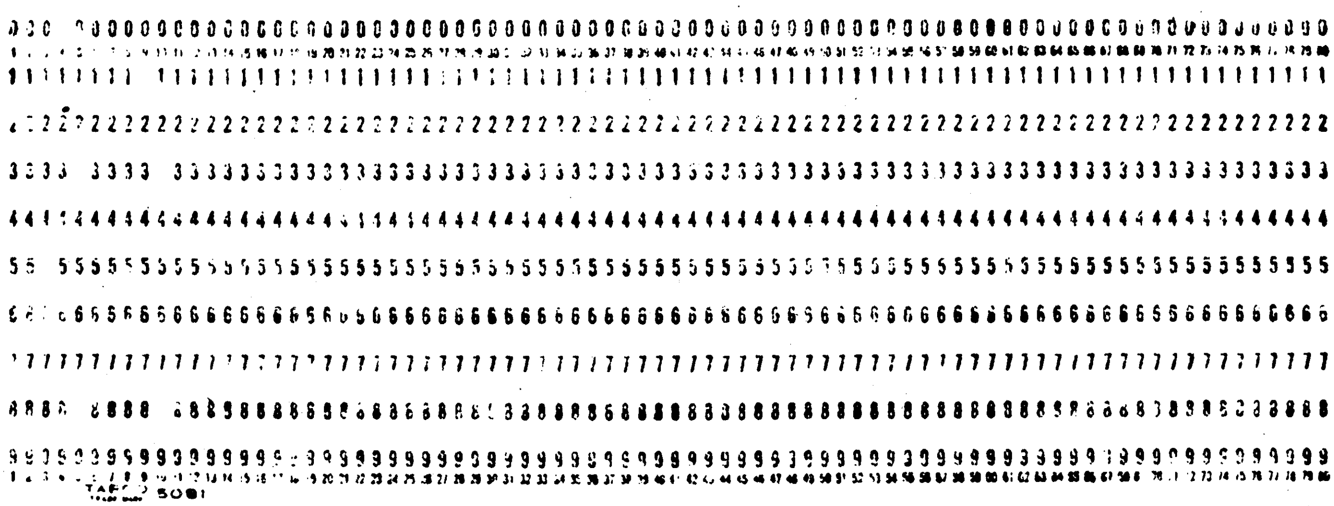
a. DIFF - Difference
b. C - Consumption
c. D - Demand
d. T - Time (hours)
e. PR - Power Revenue
f. GR - General Revenue
g. CC - Previous Consumption
h. TT - Time decremented by one hour

SWITCHES

1 ON - Primary Rates
1 OFF - Secondary Rates
2 ON - Increments of 50 KW
2 OFF - Increments of 1 KW
3 ON - Bill Table
3 OFF - By-Pass

INPUT DATA

50. 1.



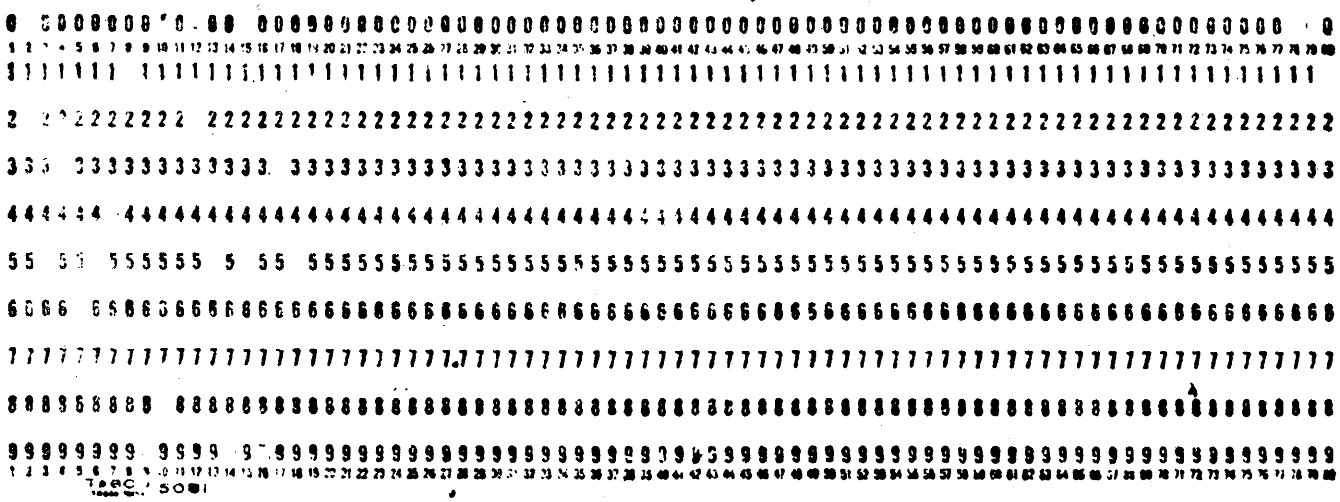
4.2.8

EXHIBIT III

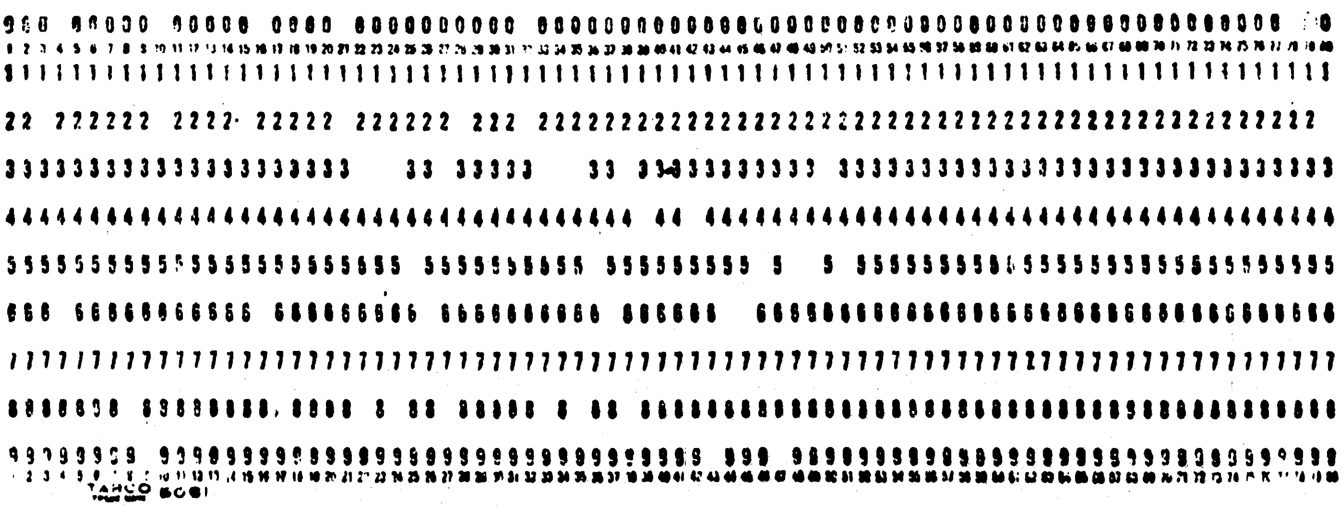
CROSSING POINTS - HEADERS

(PRIMARY SERVICE)

[illegible] PRIMARY SERVICE 001



KW HRS KWH S.C.NO.2 S.C.NO.4 DIFFERENCE 002



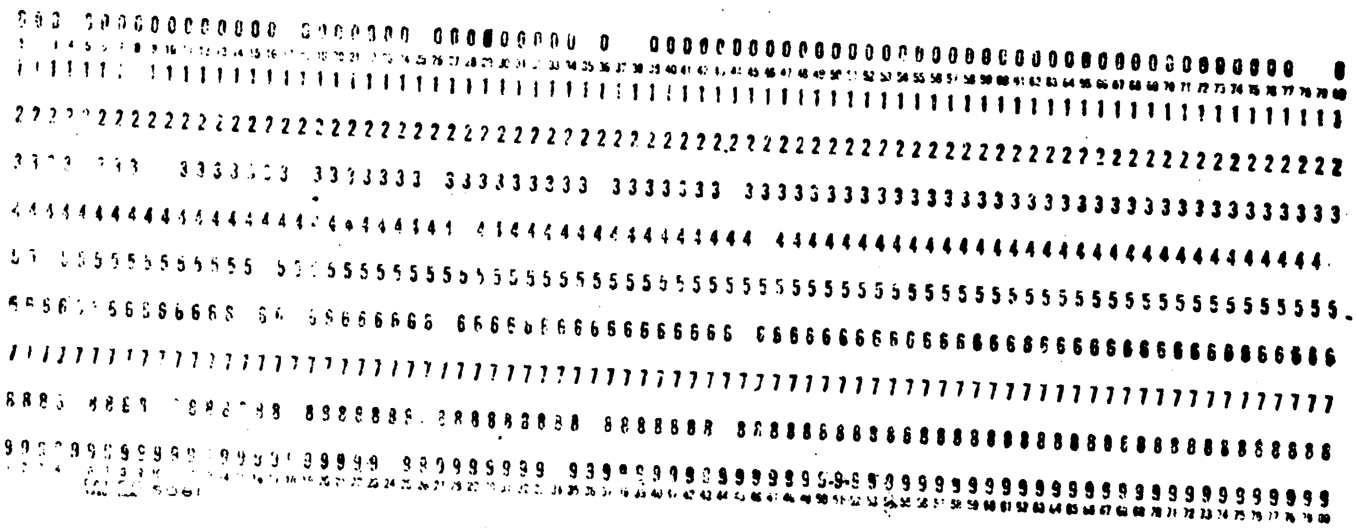
4.2.9

EXHIBIT IV

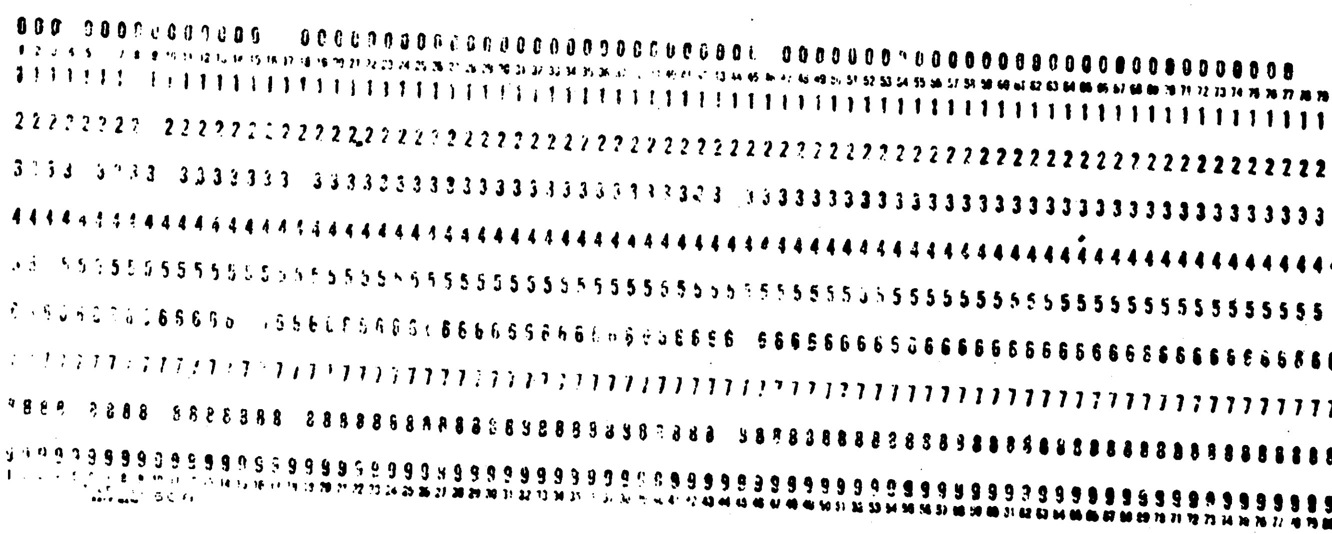
CROSSING POINTS - OUTPUT

DEMAND KW / HOURS / KWH / S.C. NO. 2 / S.C. NO. 4 / DIFF

004



005



92

4.2.10

EXHIBIT V

SECONDARY SERVICE

KW	HRS	KWH	S.C.NO.2	S.C.NO.4	DIFFERENCE
50.	13.	650.	90.64	90.00	.64
50.	12.	600.			-.60
100.	13.	1300.	181.89	180.00	1.89
100.	12.	1200.			-.60
100.	378.	37800.	712.79	713.00	-.20
100.	377.	37700.			.34
150.	13.	1950.	270.44	270.00	.44
150.	12.	1800.			-2.40
150.	117.	17550.	533.54	534.06	-.52
150.	116.	17400.			1.02
200.	14.	2800.	361.59	360.00	1.59
200.	13.	2600.			-2.20
200.	105.	21000.	653.39	654.50	-1.10
200.	104.	20800.			.95
250.	15.	3750.	454.64	450.00	4.64
250.	14.	3500.			-.10
250.	98.	24500.	773.89	776.10	-2.20
250.	97.	24250.			.36
300.	15.	4500.	543.89	540.00	3.89
300.	14.	4200.			-1.30
300.	93.	27900.	893.09	895.37	-2.27
300.	92.	27600.			.81
350.	15.	5250.	633.14	630.00	3.14
350.	14.	4900.			-3.50
350.	89.	31150.	1010.34	1011.14	-.80

CROSSING POINTS - STEPS OF 50 KW

EXHIBIT VI

4.2.11

SECONDARY KW	HRS	SERVICE KWH	S.C.NO.2	S.C.NO.4	DIFFERENCE
350.	297.	103950.			-98.13
351.	15.	5265.	634.92	631.80	3.12
351.	14.	4914.			-3.54
351.	89.	31239.	1012.99	1014.01	-1.02
351.	88.	30883.			2.59
352.	15.	5280.	636.71	633.60	3.11
352.	14.	4928.			-3.57
352.	89.	31328.	1015.65	1016.89	-1.23
352.	88.	30976.			2.38
353.	15.	5295.	638.49	635.40	3.09
353.	14.	4942.			-3.61
353.	89.	31417.	1018.31	1019.76	-1.45
353.	88.	31064.			2.17
354.	15.	5310.	640.28	637.20	3.08
354.	14.	4956.			-3.64
354.	89.	31506.	1020.97	1022.64	-1.67
354.	88.	31152.			1.97
355.	15.	5325.	642.06	639.00	3.06
355.	14.	4970.			-3.67
355.	89.	31595.	1023.62	1025.51	-1.89

CROSSING POINTS - STEPS OF 1KW

EXHIBIT VII

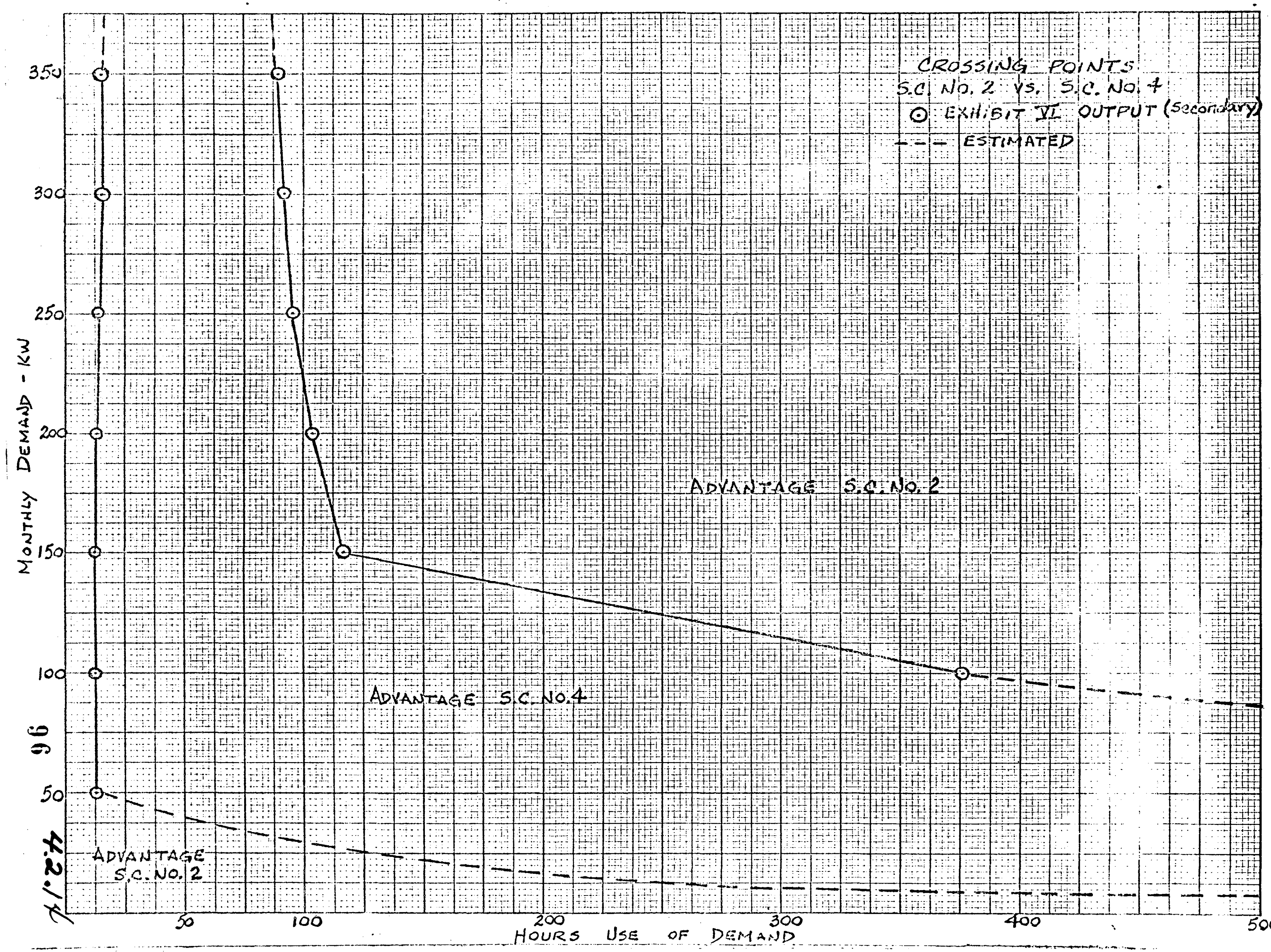
SECONDARY SERVICE

KW	HRS	KWH	S.C.NO.2	S.C.NO.4	DIFFERENCE
350.	138.	48300.	1233.29	1349.00	-115.70
350.	139.	48650.	1237.84	1353.72	-115.88
350.	140.	49000.	1242.39	1358.45	-116.05
350.	141.	49350.	1246.94	1363.17	-116.23
350.	142.	49700.	1251.49	1367.90	-116.40
350.	143.	50050.	1256.04	1372.62	-116.58
350.	144.	50400.	1260.59	1377.35	-116.75
350.	145.	50750.	1265.14	1382.07	-116.93
350.	146.	51100.	1269.69	1386.80	-117.10
350.	147.	51450.	1274.24	1391.52	-117.28
350.	148.	51800.	1278.79	1396.25	-117.45
350.	149.	52150.	1283.34	1400.97	-117.63
350.	150.	52500.	1287.89	1405.70	-117.80
350.	151.	52850.	1292.44	1410.42	-117.98
350.	152.	53200.	1296.99	1415.15	-118.15
350.	153.	53550.	1301.54	1419.87	-118.33
350.	154.	53900.	1306.09	1424.60	-118.50
350.	155.	54250.	1310.64	1429.32	-118.68
350.	156.	54600.	1315.19	1434.05	-118.85
350.	157.	54950.	1319.74	1438.77	-119.03
350.	158.	55300.	1324.29	1443.50	-119.20
350.	159.	55650.	1328.84	1448.22	-119.38
350.	160.	56000.	1333.39	1452.95	-119.55
350.	161.	56350.	1337.94	1457.67	-119.73
350.	162.	56700.	1342.49	1462.40	-119.90
350.	163.	57050.	1347.04	1467.12	-120.08
350.	164.	57400.	1351.59	1471.85	-120.25
350.	165.	57750.	1356.14	1476.57	-120.43
350.	166.	58100.	1360.69	1481.30	-120.60

BILL TABLE

EXHIBIT VIII

4.2.13


CROSSING POINTS
S.C. NO. 2 VS. S.C. NO. 4
⊙ EXHIBIT VI OUTPUT (Secondary)
--- ESTIMATED
ADVANTAGE S.C. NO. 2
ADVANTAGE S.C. NO. 4
ADVANTAGE S.C. NO. 2
MONTHLY DEMAND - KW
350
300
250
200
150
100
50
HOURS USE OF DEMAND
50
100
200
300
400
500

96

4.2.14

ELECTRIC WATER HEATING

INCREMENTAL APPLIANCE USAGE

K. W. Brady

Long Island Lighting Company

ELECTRIC WATER HEATING

Incremental Appliance Usage

K. W. Brady - LILCO

This program is written in FORMAT FORTRAN for card input into an IBM 1620 digital computer.

The initial phase of this program began with a mail survey of our electric residential customers. LILCO has better than 600,000 such customers; all were not surveyed. Some 250,000 of these accounts were selected; I believe this was accomplished through the use of random numbers. Included in this survey were all our electric water heating customers, approximately 29,000. We were going to attempt to determine the usage of an electric water heater as a sole major appliance as well as in conjunction with various other major appliances.

The program was scheduled to run for one year at which time results would be tabulated to determine the necessary as well as the reliability of the data. This final phase has not been completed as yet, however, the 1620 section has, and that is what I would like to demonstrate to you today.

On page one (1) of the exhibits there is a copy of the original survey card which was mailed to the customer. Many departments in the company had their fingers in this survey so that there is some information on the card which is superfluous. This is the reason we designed a "condensed card", also shown on page 1, which contained information pertinent to the water heating study. This card was key punched directly from the original survey card and verified. This deck of condensed cards, one for each customer who replied to the survey, then became our master control deck. Incidentally, with regard to returns we received approximately a 52% return from the water heating customers.

This master deck was then broken down by types of major appliances in conjunction with electric water heating to determine the population of each appliance group. Where it was felt that the size of the group was too small to give significant answers, the group was dropped and put into an overall non-conforming group. Page 2 shows the various appliance groups we retained as well as the code given to each group. You will note the groups are split between yes and no. We have some honest customers, who, though receiving the benefit of the water heating reduction in our electric rate, indicated that they no longer had an electric water heater.

The next step in the program was to match detail billing cards with this master deck by appliance groups. Page 3 of the exhibits shows a flow diagram which illustrates the method used for this match. LILCO has both monthly and bimonthly billing so it was necessary to run four passes through a collator to obtain the proper match. These passes were:

1. Yes group vs. 30 day billing.
2. Yes group vs. 60 day billing.
3. No group vs. 30 day billing.
4. No group vs. 60 day billing.

When all the detail bill cards were collated with the master deck, those that matched had their proper appliance code number punched into them;those that did not match had the non-conforming group code punched into them. This gave us a code number for all of the customers who replied to the survey. For those who did not reply, a code, (30), was given and they were handled as a separate item in the program.

The detail bill cards containing the code number were then sorted by appliance codes into a series of predetermined frequency steps. These steps were actually an assumed class interval of 50KWH. We merely counted the number of items for each class interval and produced a summary card for the interval or frequency step. We set up 51 steps; an illustration of this is on pages 12 and 13 in the rear of the exhibits.

Getting back to the flow diagram, once the detail bill cards were summarized into frequency steps, they were retained in account number order as we wanted to obtain a print-out of each customer's account for the test year. The frequency decks, 14 for the 'yes' group (the flow diagram shows 13, one was added at the last munute), 10 for the 'no' group, and 1 each for the non-conforming groups were then processed on the 1620. These decks of cards were white; the two non-conforming groups were merged with their respective yes or no group and an overall summary group produced (for yes and no). These groups were coded 10 for 'yes' and 20 for 'no'; the color of the cards, pink. We now had detail by appliance groups as well as summary, both 'yes' and 'no' for all the customers who replied to the survey.

A pink deck was also produced for the no-reply customers in the same frequency steps. We had all the detail bill cards for our electric water heating rate, so it was decided to include the non-replies as a single group. The three pink decks, summary decks, were then merged into a single overall summary deck (blue). This deck gave us a check on the number of customers billed for each month to make certain we had all of the detail cards. It would also tell us if we had to many.

The next phase was the 1620 program. I am not a statistician by any means, so that in producing the program I worked with the statistician in the department. He supplied the formulae necessary to obtain the answers, and he incidentally is the one who is presently analyzing the results. My sole function was to produce a machine method which would provide answers, thereby relieving extended clerical calculations.

The statistician set down the formulae in statistical form, which I converted to my own language. These are shown on page 4 of the exhibits; one look at some of the symbols and you will understand why I used my own language.

Page 5 contains a flow diagram of the program, 6 and 7 a print-out and 8 a glossary of the terms used in the program. I believe they are self-explanatory and suffice it to say the program utilizing the frequency distribution data set up in a class interval of 50KWH, an assumed mean of 225, provides the arithmetic mean, standard deviation, standard error of mean, and co-efficient of variation for the various appliance groupings.

Page 8-A illustrates a sample output of the program. Column 1 is the frequency steps (class interval); column 2, the deviation from the assumed mean (225); column 3, the deviation squared; column 4, the frequency (number of items by step); column 5, the frequency times the deviation and column 6 the frequency times the deviation squared. Columns 5 and 6 are summarized by the program as they are used in the formula calculations.

Page 9 is an illustration of the four header cards used in the program. A set of these cards is necessary before each code being processed by the computer.

Pages 10 and 11 are card layout forms illustrating the fields used in the input data cards and the output data cards.

I am presenting this program to you to demonstrate one of the many ways rate men can make use of the 1620 computer. This program was originally intended to be a one-shot set-up, however, with certain modifications, I believe it could be written in a way to be of general use to all of us.

1

SURVEY CARD

PLEASE ANSWER QUESTIONS BELOW BY MAKING A CHECK IN THE BOX AND FILLING IN NUMBERS ON THE LINES

YOU CAN HELP US SERVE YOU BETTER!

DWELLING

21☐ SINGLE FAMILY 22☐ TWO FAMILY 23☐ APARTMENT

24-25 ______ TOTAL ROOMS 26-27 ______ NO. IN FAMILY

28☐ OWN 29☐ RENT

30☐ SUMMER HOME 31☐ YEAR ROUND

WAS YOUR HOME BUILT

32☐ BEFORE 1956 33☐ 1956 OR AFTER

KITCHEN APPLIANCES

RANGE

39☐ ELECTRIC
40☐ LILCO GAS
41☐ BOTTLED GAS

REFRIGERATOR

42☐ ELECTRIC
43☐ LILCO GAS
44☐ OTHER ______
45☐ SEPARATE FREEZER
46☐ DISHWASHER

LAUNDRY APPLIANCES

47☐ CLOTHES WASHER

CLOTHES DRYER.

48☐ GAS
49☐ ELECTRIC

HOME HEATING-CENTRAL SYSTEM

FUEL

50☐ LILCO GAS
51☐ BOTTLED GAS
52☐ OIL
53☐ TOTAL ELECTRIC
54☐ NONE

TYPE SYSTEM

55☐ STEAM
56☐ HOT WATER
57☐ WARM AIR
58☐ OTHER ______

SUPPLEMENTARY HEATERS

59☐ LILCO GAS
60☐ BOTTLED GAS
61☐ OIL
62☐ ELECTRIC

DOMESTIC WATER HEATING

63☐ FROM HEATING SYSTEM

SEPARATE WATER HEATING TANK

64☐ OIL
65☐ LILCO GAS
66☐ BOTTLED GAS
67☐ ELECTRIC
68☐ COAL
69☐ NONE

AIR CONDITIONING

70 ______ WINDOW UNITS, TOTAL
71☐ CENTRAL, ELECTRIC
72☐ SEPARATE DEHUMIDIFIER
73☐ ATTIC FAN

TELEVISION

74 ------ BLACK & WHITE
75 ------ COLOR

WATER PUMPS

76☐ DRINKING SUPPLY
77☐ LAWN WATERING
78☐ OTHER ______

79 WHAT APPLIANCE DO YOU PLAN TO ADD TO YOUR HOME ?

______

THANK YOU FOR YOUR HELP.

PLEASE DO NOT BEND OR MUTILATE THIS CARD

FC-5964.1

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 78 79 80

IBM N19109-0

CONDENSED CARD

Rate 38 "NEW DECK"

CYCLE | DIST. | Book | Folio | SUFF. C | RATE | No. Family | No. Rooms | E. Range | G. Range | REF. Refrig | REF. FREEZ | E. FREEZER | WASHER | WASH-DRY | E. DRY | GAS DRY | E.W. Heat | Gas W.H. | Oil W.H. | Boiler W.H. | GAS HEAT | Oil | El. | Hot Water | Steam | Warm Air | Other | CENT. A/C | Unit A/C

00000000000000000000000000000000000000000000000000000000000000000000000000000000
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 78 79 80
11111111111111111111111111111111111111111111111111111111111111111111111111111111
22222222222222222222222222222222222222222222222222222222222222222222222222222222
33333333333333333333333333333333333333333333333333333333333333333333333333333333
44444444444444444444444444444444444444444444444444444444444444444444444444444444
55555555555555555555555555555555555555555555555555555555555555555555555555555555
66666666666666666666666666666666666666666666666666666666666666666666666666666666
77777777777777777777777777777777777777777777777777777777777777777777777777777777
88888888888888888888888888888888888888888888888888888888888888888888888888888888
99999999999999999999999999999999999999999999999999999999999999999999999999999999
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 78 79 80

TABCO 5081

4.3.4

SAMPLING PROGRAM - APPLIANCE GROUP CODE

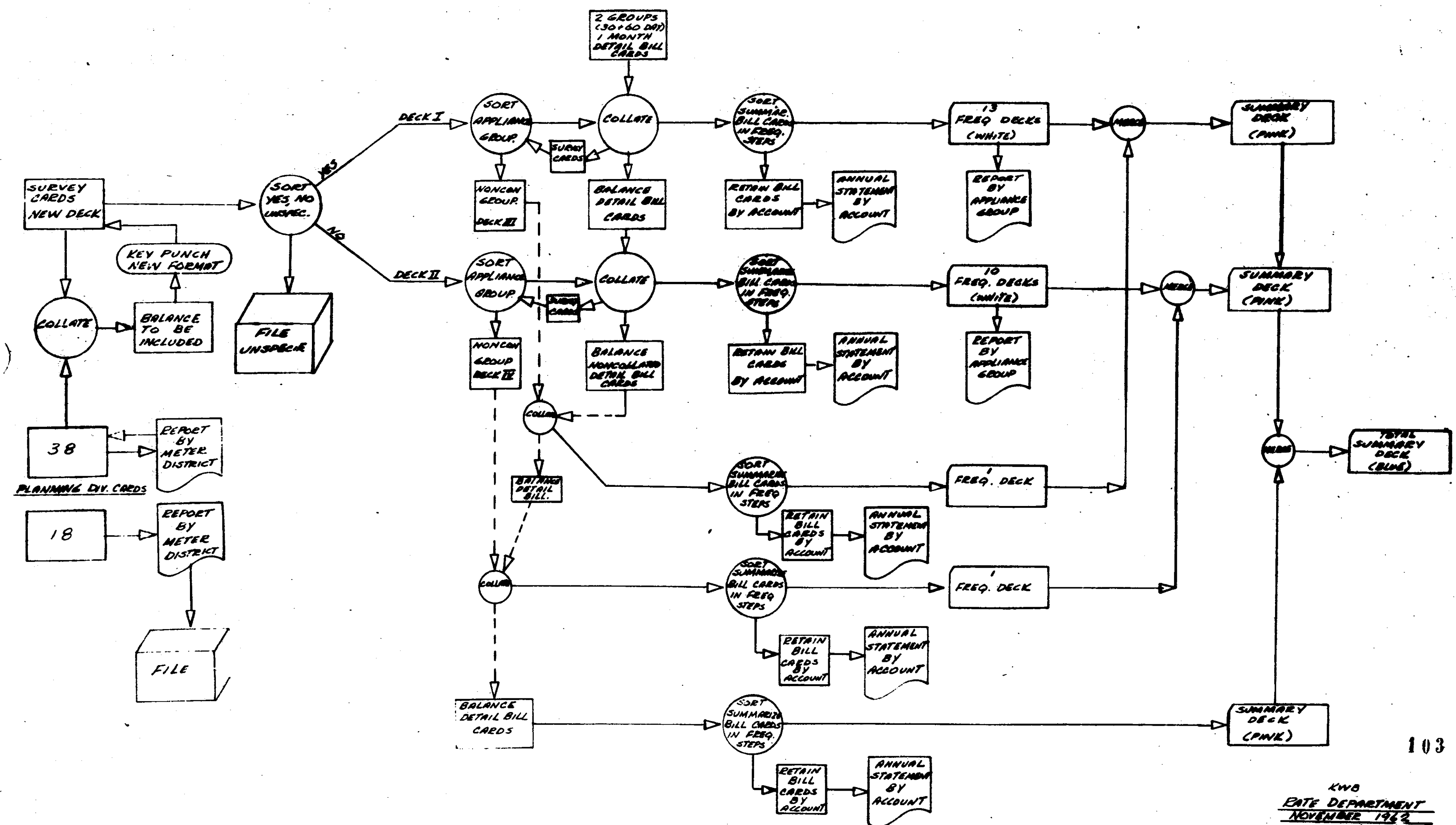
From Survey of Rate Code 38

Sample No. & Code No.	Appliance Groups
	"Yes" Group
1	Base Load & Water Htg.
2	Water Htg. & Range
3	Water Htg. & Dryer
4	Water Htg. & Freezer
5	Water Htg. & Air Cond.
6	Water Htg. Range & Dryer
7	Water Htg. Range & Freezer
8	Water Htg. Range & Air Cond.
9	Water Htg. Dryer & Freezer
11	Water Htg. Range, Dryer, Air Cond.
12	Water Htg. Range, Dryer, Freezer, Air Cond.
13	Water Htg. Range, Dryer, Freezer
24	Non - Conforming Group
10	Summary "Yes Group"
	"No" Group
14	Base Load
15	Base Load & Range
16	Base Load & Dryer
17	Base Load & Freezer
18	Base Load & Air Cond.
19	Base Load, Range & Air Cond.
22	Base Load, Range, Dryer & Air Cond.
23	Base Load, Range, Dryer, Freezer, Air Cond.
25	Non - Conforming Group
20	Summary "No Group"
30	Residual Group (Non - Survey)
31	Total - All Groups

Notes:

1. One set for Monthly Billing
   One set for Bimonthly Billing

RATE CODE 38 - SURVEY ANALYSIS



KWB
RATE DEPARTMENT
NOVEMBER 1962
4.3.6

Appliance Increment Consumptions — Sampling Program

Statistical Term	Description	Symbol	IBM Equivalent	
Class	Consumption Step	C	STEP	
Class Interval	Size of Class	Ci	CI	(constant)
Number of Classes	Number of Steps in Distribution		max	
Frequency	Number of Cases in a Class	f	F	
Sum of Frequencies	Total number of Cases	N	TF	
Assumed Mean	Arbitrary Origin in Series	X'	ASM	(constant)
Deviation from X'	in Class Intervals	d'	D	
Deviations Squared		$(d')^2$	DS	
Frequency x Deviation		fd'	FD	
Frequency x Deviations Squared		$f(d')^2$	FDS	
Total of Columns	Summation Directive	Σ	T	
Mean	Arithmetic Average	$\bar{X}$	AV	
Standard Deviation		σ	SD	
Standard Error of Mean		σ_μ	SEM	
Coefficient of Variation	(in per cent)	V	VAR	

FORMULAS

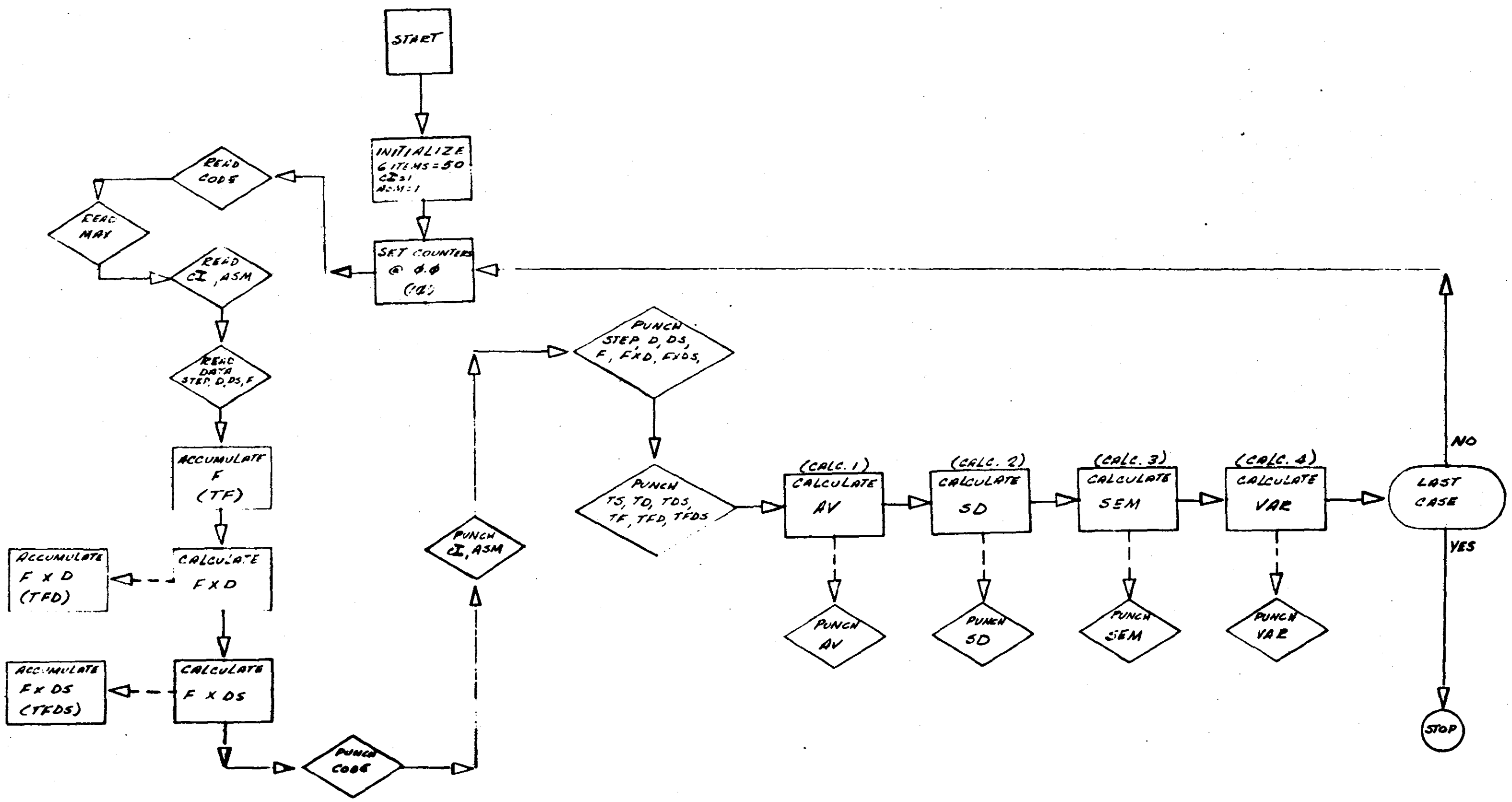
1. $\bar{X} = X' + \frac{\Sigma fd}{N} Ci$ $\qquad AV = ASM + \frac{TFD}{TF} CI$

2. $\sigma = ci\sqrt{\frac{\Sigma f(d')^2}{N} - \left(\frac{\Sigma fd'}{N}\right)^2}$ $\qquad SD = CI\sqrt{\frac{TFDS}{TF} - \left(\frac{TFD}{TF}\right)^2}$

3. $\sigma_\mu = \frac{\sigma}{\sqrt{N}}$ $\qquad SEM = \frac{SD}{\sqrt{TF}}$

4. $V = \frac{100\sigma}{\bar{X}}$ $\qquad VAR = \frac{100\ SD}{AV}$

SAMPLING PROGRAM 9/12/62



4.3.8

6.

```
260000800009RS
LOAD DATA
TOP
ENTER SOURCE PROGRAM, PUSH START
08300 C   SAMPLING PROGRAM - DETERMINATION OF INCREMENTAL APPLIANCE
08300 C   CONSUMPTIONS    BY K.BRADY AND J.O CONNOR  SEPTEMBER 1962
08300          DIMENSION STEP(50),D(50),DS(50),F(50),FD(50),FDS(50)
08300        1 TS=0.0
08324          TD=0.0
08348          TDS=0.0
08372          TF=0.0
08396          TFD=0.0
08420          TFDS=0.0
08444          AV=0.0
08468          SD=0.0
08492          SEM=0.0
08516          VAR=0.0
08540       10 FORMAT(F5.0)
08562          READ 10,CODE
08586       20 FORMAT(I2)
08608          READ 20,MAX
08632       30 FORMAT(F5.0)
08654          READ 30,CI
08678       40 FORMAT(F5.0)
08700          READ 40,ASM
08724       50 FORMAT(F7.0,F11.0,F11.0,F11.0)
08762          DO 2 K=1,MAX
08774          READ 50,STEP(K),D(K),DS(K),F(K)
08930        2 TF=F(K)+TF
09026          FD(MAX)=0.0
09074          DO 3 K=1,MAX
09086          FD(K)=D(K)*F(K)
09194        3 TFD=FD(K)+TFD
09290          FDS(MAX)=0.0
09338          DO 4 K=1,MAX
09350          FDS(K)=DS(K)*F(K)
09458        4 TFDS=FDS(K)+TFDS
09554       60 FORMAT(6H CODE F5.0)
09596          PUNCH 60,CODE
09620       70 FORMAT(16H CLASS INTERVAL F5.0)
09682          PUNCH 70,CI
09706       80 FORMAT(14H ASSUMED MEAN F5.0)
09764          PUNCH 80,ASM
09788       90 FORMAT(F7.0,F11.0,F11.0,F11.0,F11.0,F11.0)
09836          DO 5 K=1,MAX
09848        5 PUNCH 90,STEP(K),D(K),DS(K),F(K),FD(K),FDS(K)
10112          PUNCH 90,TS,TD,TDS,TF,TFD,TFDS
10196          AB=0.0
10220          AC=0.0
10244          AB=TFD/TF
10280          AC=AB*CI
10316          AV=ASM+AC
10352      100 FORMAT(6H MEAN F10.3)
10394          PUNCH 100,AV
10418          Z=0.0
10442          W=0.0
10466          Y=0.0
10490          R=0.0
10514          Z=TFDS/TF
10550          W=TFD/TF*TFD/TF
10610          Y=Z-W
10646          R=SQRT(Y)
10670          SD=CI*R
```

106

4.3.9

7.

```
10706     200 FORMAT(20H STANDARD DEVIATION F8.3)
10776         PUNCH 200,SD
10800         V=0.0
10824         V=SQRT(TF)
10848         SEM=SD/V
10884     300 FORMAT(24H STANDARD ERROR OF MEAN F8.3)
10962         PUNCH 300,SEM
1   6         VAR=100.*SD/AV
11034     400 FORMAT(26H COEFFICIENT OF VARIATION F8.3)
11116         PUNCH 400,VAR
11140         GO TO 1
11148         END

PROG SW 1 ONFOR SYMBOL TABLE, PUSH START
SW 1 OFF TO IGNORE SUBROUTINES, PUSH START
1620 FORTRAN SUBR.   9/30/61
PROCESSING COMPLETE26000080000926000080000935
LOAD DATA26000080000935
LOAD DATA
```

107

4.3.10

GLOSSARY

Step	-	Number of Consumption Steps
D	-	Deviation from Assumed Mean
DS	-	Deviation Squared
F	-	Frequency (Number of Customers)
FD	-	Frequency X Deviation
FDS	-	Frequency X Deviation Squared
TS	-	Total Steps
TD	-	Total Deviation
TDS	-	Total Deviation Squared
TF	-	Total Frequency
TFD	-	Total Frequency X Deviation
TFDS	-	Total Frequency X Deviation Squared
AV	-	Arithmetic Mean
SD	-	Standard Deviation
SEM	-	Standard Error of Mean
VAR	-	Coefficient of Variation

CODE 10.
CLASS INTERVAL 50. 8-A.
*ASSUMED MEAN 225.

*	50.	-4.	16.	8.	-32.	128.	*
*	100.	-3.	9.	3.	-9.	27.	*
*	150.	-2.	4.	8.	-16.	32.	*
*	200.	-1.	1.	13.	-13.	13.	*
	250.	.	.	22.	.	.	
*	300.	1.	1.	14.	14.	14.	*
*	350.	2.	4.	21.	42.	84.	*
*	400.	3.	9.	28.	84.	252.	*
*	450.	4.	16.	41.	164.	656.	*
*	500.	5.	25.	30.	150.	750.	*
*	550.	6.	36.	39.	234.	1404.	*
*	600.	7.	49.	39.	273.	1911.	*
*	650.	8.	64.	55.	440.	3520.	*
*	700.	9.	81.	40.	360.	3240.	*
*	750.	10.	100.	49.	490.	4900.	*
*	800.	11.	121.	43.	473.	5203.	*
*	850.	12.	144.	37.	444.	5328.	*
*	900.	13.	169.	48.	624.	8112.	*
*	950.	14.	196.	54.	756.	10584.	*
*	1000.	15.	225.	54.	810.	12150.	*
*	1050.	16.	256.	51.	816.	13056.	*
*	1100.	17.	289.	42.	714.	12138.	*
*	1150.	18.	324.	36.	648.	11664.	*
*	1200.	19.	361.	36.	684.	12996.	*
*	1250.	20.	400.	44.	880.	17600.	*
*	1300.	21.	441.	33.	693.	14553.	*
*	1350.	22.	484.	27.	594.	13068.	*
*	1400.	23.	529.	29.	667.	15341.	*
*	1450.	24.	576.	22.	528.	12672.	*
*	1500.	25.	625.	22.	550.	13750.	*
*	1550.	26.	676.	16.	416.	10816.	*
*	1600.	27.	729.	15.	405.	10935.	*
*	1650.	28.	784.	10.	280.	7840.	*
*	1700.	29.	841.	11.	319.	9251.	*
*	1750.	30.	900.	8.	240.	7200.	*
*	1800.	31.	961.	15.	465.	14415.	*
*	1850.	32.	1024.	4.	128.	4096.	*
*	1900.	33.	1089.	9.	297.	9801.	*
*	1950.	34.	1156.	5.	170.	5780.	*
*	2000.	35.	1225.	9.	315.	11025.	*
*	2050.	36.	1296.	3.	108.	3888.	*
*	2100.	37.	1369.	1.	37.	1369.	*
*	2150.	38.	1444.	3.	114.	4332.	*
*	2200.	39.	1521.	4.	156.	6084.	*
*	2250.	40.	1600.	4.	160.	6400.	*
*	2300.	41.	1681.	2.	82.	3362.	*
*	2350.	42.	1764.	.	.	.	*
*	2400.	43.	1849.	2.	86.	3698.	*
*	2450.	44.	1936.	.	.	.	*
*	2500.	45.	2025.	4.	180.	8100.	*
	.	.	.	1113.	16020.	323538.	

MEAN 944.676
*STANDARD DEVIATION 456.936 *
*STANDARD ERROR OF MEAN 13.696 *
*COEFFICIENT OF VARIATION 48.369 *

109

40.12

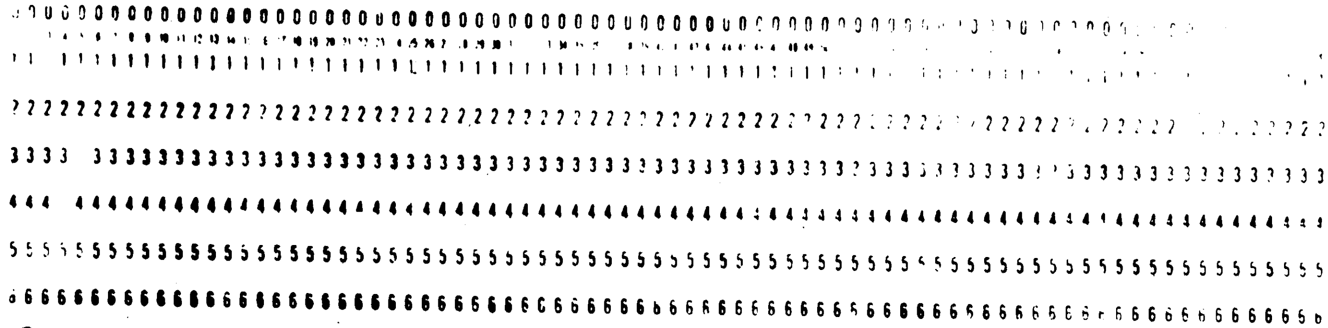
PROGRAM HEADER CARDS

S

14.

APPLIANCE CODE CARD

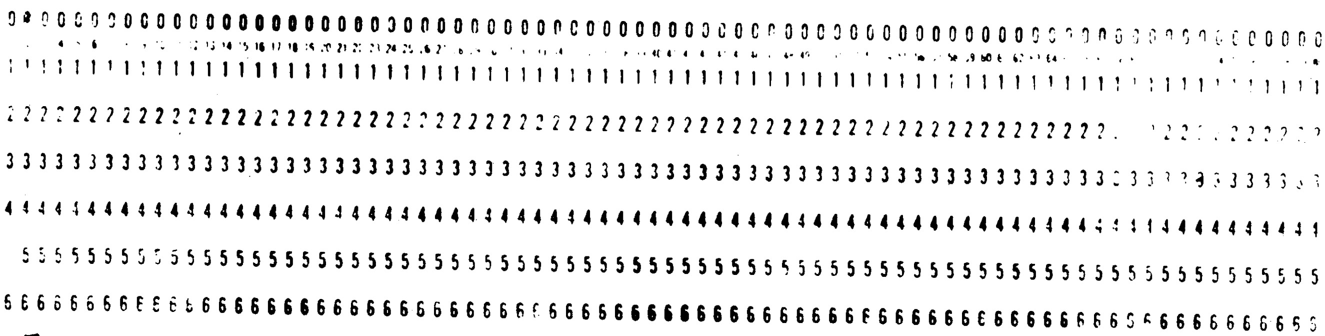
1



50

NUMBER OF STEPS IN DISTRIBUTION

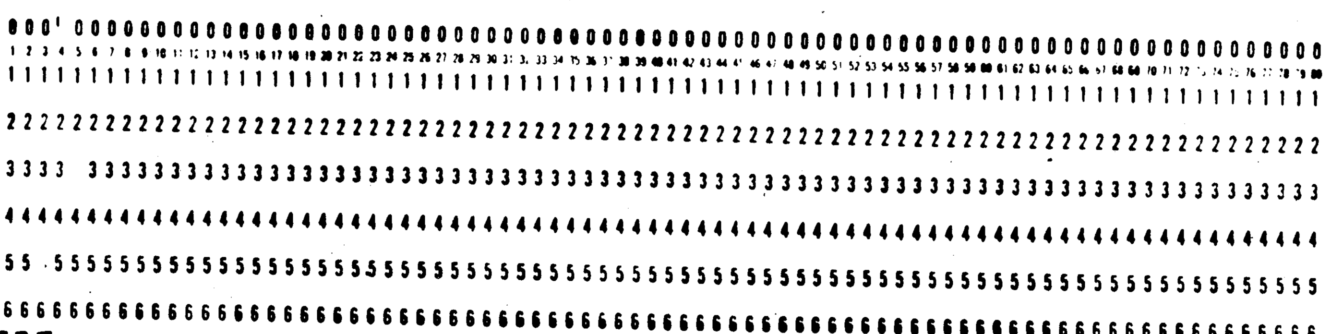
2



50.

ASSUMED CLASS INTERVAL

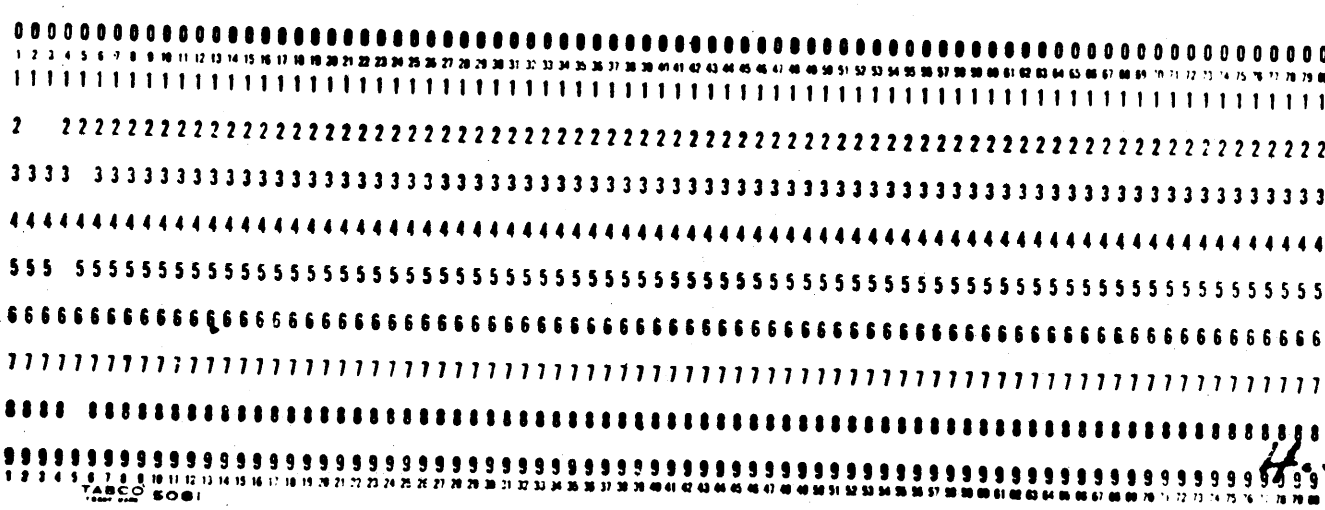
3.



225.

ASSUMED ARITHMETIC MEAN

4.



110

4.3.13

IBM SUPPLIES DIVISION

IBM CARD LAYOUT FORM 1

FORM X74-4049-12

SAMPLING PROGRAM - FREQUENCY INPUT DATA CARD

Field	Card columns
STEP	1–7
DEVIATION (D)	8–18
DEVIATION² (D²)	19–29
FREQUENCY (F)	30–40
(unused)	41–72
MONTH	73–74
3 OR 6	75
GROUP CODE	76–77
CARD NUMBER	78–80

COMPLETE THESE SECTIONS FOR ALL LAYOUTS

BRANCH OFFICE NAME — TRANSMITTAL NO.

CUSTOMER NAME — CUSTOMER NO.

CUSTOMER REFERENCE AND DATE — PROOFS REQUIRED? YES ☐ NO ☐

MAIL PROOFS TO

B. W. Brady — LAYOUT PREPARED BY — DATE

Is this a revision of, or replacement for, an existing IBM card form? YES ☐ NO ☐

If (YES), indicate present IBM card form no.

May the existing printing plate be scrapped? YES ☐ NO ☐

If (YES), indicate authority for scrapping:

AUTHORITY — DATE

CORNERS TO BE CUT: NONE ☐ UPPER ☐ ☐ LOWER ☐ ☐ (LEFT, RIGHT). STANDARD LOWER LEFT CORNER CUT IS NOT RECOMMENDED FOR CARDS TO BE USED WITH IBM 24 CARD PUNCH AND MACHINES WITH SIMILAR TYPE FEEDS. (C-3 IS RECOMMENDED.)

INTERPRETATION: NONE ☐; 45 POS. INT. & PRINTERS ☐; 60 ALPHA-MERICAL ☐; PRINTING PUNCHES ☐; 10-TO-INCH PRINTERS ☐; 552 ☐ 557 ☐; BILL FEED ___ LINES TO INCH ☐; END PRINTING STND. ☐ SPEC. ☐; TYPEWRITER SPACING ___ LINES TO INCH ☐

COMPLETE THESE SECTIONS WHEN APPLICABLE

REQUIREMENTS: SCORING ☐ CONSECUTIVE PREPUNCHING ☐; PRENUMBERING ☐ CONSECUTIVE PRENUMBERING ☐ (INDICATE ON THE LAYOUT THE LOCATION OF EACH FEATURE SPECIFIED ABOVE.) STUB ☐ PAD ☐ CONTINUOUS ☐; IF CARD IS TO BE PRINTED ON BOTH SIDES, INDICATE STYLE. BOOK ☐ TUMBLE ☐

FOR MARK SENSING CARDS, INDICATE STYLE

POINTERS FOR SHORT CARD LAYOUTS:
1) DETERMINE SHORT CARD COLUMN NUMBERS BY REFERRING TO GUIDE AND COLUMNS PRINTED BELOW. USABLE COLUMN RANGE IS ESTABLISHED BY THE CAPACITY LOCATION ON EACH SIDE OF GUIDE. E.G., ON A 51-COLUMN CARD, USUABLE COLUMNS ARE 15-65.
2) CUT OUT THE STRIP OF COLUMN NUMBERS.
3) PASTE THE STRIP ON THE LAYOUT FORM IN DESIRED LOCATION FOR THE SHORT CARD.
4) RULE ENDS OF SHORT CARD 2 1/2 COLUMNS BEYOND THE STRIP OF COLUMN NUMBERS.
5) DETERMINE MARK SENSE POSITIONS, IF ANY, BY ASSOCIATING STRIP NUMBERS WITH CARD LAYOUT POSITIONS. E.G., ON A 51-COLUMN CARD, M.S. POSITIONS 6-22 ARE USABLE.

SHORT CARD COLUMN GUIDE — CARD CAPACITY (COLUMNS)

SCALE: APPROX. DOUBLE SIZE. ACTUAL CARD SIZE: 3 1/4" X 7 3/8"

CUT-OUTS FOR PASTING

4.3.14

IBM SUPPLIES DIVISION

IBM CARD LAYOUT FORM 2

FORM X74-4049-12

SAMPLING PROGRAM — FREQUENCY TABLE OUTPUT CARD

Columns	Field
1–6	STEP
7–17	DEVIATION (D)
18–28	DEVIATION^2 (D^2)
29–39	FREQUENCY (F)
40–50	FREQUENCY / DEVIATION (FD)
51–61	FREQUENCY × DEVIATION^2 (FDS)
62–80	

Card rows: 0 1 2 3 4 5 6 7 8 9 (columns 1–80)

TYPE SPACING: 10-TO-INCH PRINTERS; PRINTING PUNCHES; 60-POSITION INTERPRETERS; 45-POS. PRINTERS & INT.

COLUMN GUIDE FOR MARK SENSING → 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

COMPLETE THESE SECTIONS FOR ALL LAYOUTS

BRANCH OFFICE NAME ____ TRANSMITTAL NO. ____

CUSTOMER NAME ____ CUSTOMER NO. ____

PROOFS REQUIRED? YES ☐ NO ☐

CUSTOMER REFERENCE AND DATE ____

MAIL PROOFS TO ____

R. W. Moody 2/3/62 — LAYOUT PREPARED BY · DATE

Is this a revision of, or replacement for, an existing IBM card form? YES ☐ NO ☐

If (YES), indicate present IBM card form no. ____

May the existing printing plate be scrapped? YES ☐ NO ☐

If (YES), indicate authority for scrapping: ____

AUTHORITY ____ DATE ____

CORNERS TO BE CUT: NONE ☐ UPPER ☐ ☐ LOWER ☐ ☐ (LEFT, RIGHT). STANDARD LOWER LEFT CORNER CUT IS NOT RECOMMENDED FOR CARDS TO BE USED WITH IBM 24 CARD PUNCH AND MACHINES WITH SIMILAR TYPE FEEDS. (C-3 IS RECOMMENDED.)

INTERPRETATION: NONE ☐ PRINTING PUNCHES ☐; 45 POS. INT. & PRINTERS ☐ 10-TO-INCH PRINTERS ☐; 60 ALPHA-MERICAL 552 ☐ 557 ☐; BILL FEED ____ LINES TO INCH ☐; END PRINTING STND. ☐ SPEC. ☐; TYPEWRITER SPACING ____ LINES TO INCH ☐

COMPLETE THESE SECTIONS WHEN APPLICABLE

REQUIREMENTS: SCORING ☐ CONSECUTIVE PREPUNCHING ☐; PRENUMBERING ☐ CONSECUTIVE PRENUMBERING ☐ (INDICATE ON THE LAYOUT THE LOCATION OF EACH FEATURE SPECIFIED ABOVE.); STUB ☐ PAD ☐ CONTINUOUS ☐; IF CARD IS TO BE PRINTED ON BOTH SIDES, INDICATE STYLE. BOOK ☐ TUMBLE ☐

FOR MARK SENSING CARDS, INDICATE STYLE: Use for maximum writing or printing space between M. S. positions. Not recommended for cards requiring writing or printing between M. S. pos.

POINTERS FOR SHORT CARD LAYOUTS: 1) DETERMINE SHORT CARD COLUMN NUMBERS BY REFERRING TO GUIDE AND COLUMNS PRINTED BELOW. USABLE COLUMN RANGE IS ESTABLISHED BY THE CAPACITY LOCATION ON EACH SIDE OF GUIDE. E. G., ON A 51-COLUMN CARD, USUABLE COLUMNS ARE 15-65. 2) CUT OUT THE STRIP OF COLUMN NUMBERS. 3) PASTE THE STRIP ON THE LAYOUT FORM IN DESIRED LOCATION FOR THE SHORT CARD. 4) RULE ENDS OF SHORT CARD 2 1/2 COLUMNS BEYOND THE STRIP OF COLUMN NUMBERS. 5) DETERMINE MARK SENSE POSITIONS, IF ANY, BY ASSOCIATING STRIP NUMBERS WITH CARD LAYOUT POSITIONS. E. G., ON A 51-COLUMN CARD, M. S. POSITIONS 6-22 ARE USABLE.

SHORT CARD COLUMN GUIDE — CARD CAPACITY (COLUMNS)

SCALE: APPROX. DOUBLE SIZE. ACTUAL CARD SIZE: 3 1/4" X 7 3/8"

CUT-OUTS FOR PASTING: 1–80

4.5.15

MEETING ASSUMPTIONS OF HOMOGENEITY OF VARIANCE: NONLINEAR DATA TRANSFORMS AND BARTLETT'S TEST

By

Richard C. Irons

U.S. NAVAL SCHOOL OF AVIATION MEDICINE
U.S. NAVAL AVIATION MEDICAL CENTER
PENSACOLA, FLORIDA

USERS CODE: 1159

INTRODUCTION

When working with various statistical routines, an investigator may have reason to suspect that there are substantial differences in variance among the sets of scores or observations. Under these conditions he may wish to perform Bartlett's Test for Homogeneity of Variance.

This test is usually performed on the original data, and if the variance is found to be heterogeneous, then a nonlinear transformation on the data can be accomplished, and Bartlett's test done a second time. The process is usually repeated until a transform is found which will produce variance in the sets of data which does not depart significantly from homogeneity.

In programming the above operations on the IBM 1620 computer the slowest operation is in reading the data into the computer. Consequently, in the present work it has been found advantageous to read in the original data, perform several nonlinear transformations, and to punch out a card containing the original score and the transformed scores. In addition, summary cards which contain the number of cases, the sums of the scores, and the sums of the squared scores are punched out for the original data and for each transformation of each set of scores. The four transforms used are discussed by Edwards (1). They are: 1) $X + .5$, 2) $X + \quad X + 1$, 3) Log of (X), and 4) the reciprocal of (X) where (X) is the original score.

The input for the Bartlett test is the summary cards for the sets of scores, and the test is performed for the original data and each transformation. If the variance of sets of scores is found to be homogeneous for the original data, the investigator need not concern himself with the tests for the four transformations. If, however, Bartlett's test

114

44.1

on the original data shows the variance of the groups to be heterogeneous, the investigator may inspect the significance levels of the four transformations to find the most acceptable transformation of the data.

DESCRIPTIONS OF PROGRAMS

THE TRANSFORMATION PROGRAM

This program transforms the raw scores, obtains the sums of (X) and the sums of (X) squared of raw and transformed scores, punching these into six decks of cards. These may be separated by machine sorting. Five of the decks are for input to the Bartlett program, the sixth is a table which gives the group identification, subject identification, and original and transformed scores for each subject.

FORMAT FOR INPUT CARDS

Card	Columns	Information	Example
EDIT	1- 4	Number of groups	0012
GROUP	1- 4	Group number	0001
	6- 9	Number of subjects	0009
DATA	1-10	Raw score	.000416.892
	12-15	Subject's number	2043

Blanks may be used in place of leading zeros in the data fields. The input consists of the EDIT card, the GROUP card for the first group, the DATA cards for the first group with one card per subject, the GROUP card for the second group, the DATA cards for the second group on its corresponding number of cards, et cetera until all groups have been accounted for.

OPERATING THE PROGRAM

1. Place the program in the card reader.
2. Place data cards on top of the program deck.
3. Press the RESET button on the console, the LOAD button on the card reader, and the START button on the card punch.
4. When LOAD DATA is typed, the program will automatically read the data and process the problem.
5. When END OF JOB message is typed and instructions for indicating a new run are typed, NON PROCESS RUNOUT ON THE CARD PUNCH must be pressed to obtain all the output cards. The program is completed at this time.
6. The output cards must be sorted numerically on column 65. The cards in hoppers one, two, three, four, and five are the input data for the Bartlett program. The remaining cards (in the reject hopper) when listed with an 80-80 board on a 407 machine will yield the table of transformed scores.

PROCESSING TIME

The approximate computer time necessary to accomplish the Transformation Program is given by the following equation:

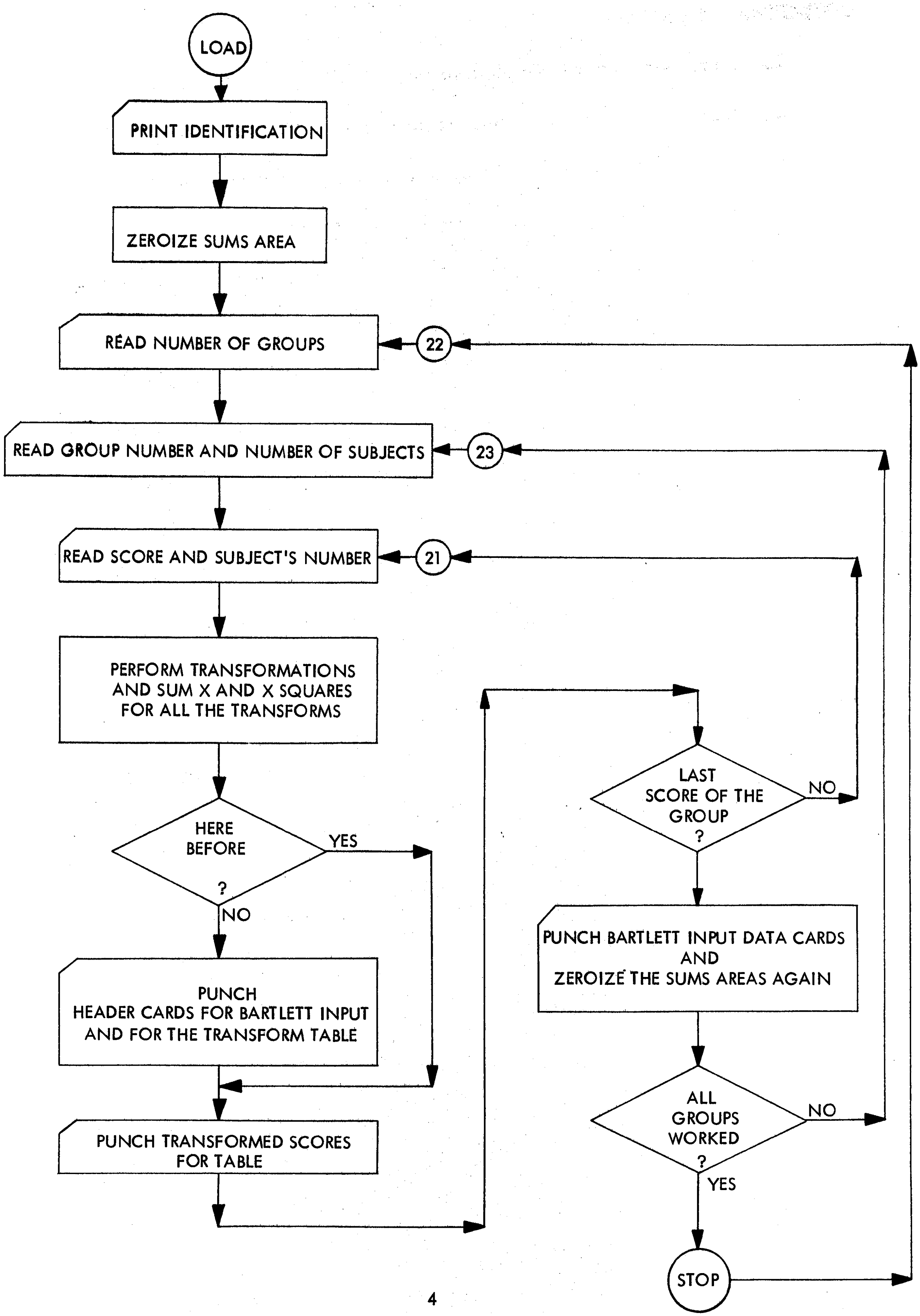
Execution Time = 2N + 50 seconds

where N = total number of cases in all of the groups.

116

4.4.3

FLOW DIAGRAM TRANSFORMATION PROGRAM



TRANSFORMATION PROGRAM LISTING

```
C      SYMBOL DEFINITIONS FOLLOW.                                      A07070
C                                                                      A07070
C      LAST=LAST CARD OF GROUP                                         A07070
C      NOGRP= NUMBER OF GROUPS                                         A07070
C      NUMGP= SUBGROUP NUMBER                                          A07070
C      NOCAS= NUMBER OF CASES                                          A07070
C      XRAW= RAW SCORE                                                 A07070
C      KASNO= CASE OR SUBJECT NUMBER                                   A07070
C      SUMX= SUM OF RAW SCORES                                         A07070
C      SUMX2= SUM OF RAW SCORES SQUARED.                               A07070
C      XA2= SQRT OF X+.5 SQUARED WHERE X IS THE RAW SCORE              A07070
C      XA= SQRT OF X+.5                                                A07070
C      SMXA2= SUM OF SQRT OF X+.5 SQUARED                              A07070
C      SMXA= SUM OF SQRT OF X+.5                                       A07070
C      XB= SQRT OF X + SQRT OF X+1                                     A07070
C      XB2= THE SQUARE OF THE SQRT OF X + SQRT OF X+1                  A07070
C      SMXB= SUM OF SQRT OF X + SQRT OF X+1                            A07070
C      SMXB2= SUM OF THE SQUARE OF THE SQRT OF X + SQRT OF X+1         A07070
C      XC= LOG OF X                                                    A07070
C      XC2= LOG OF X SQUARED                                           A07070
C      SMXC= SUM OF LOG OF X                                           A07070
C      SMXC2= SUM OF LOG OF X SQUARED                                  A07070
C      XD= RECIPROCAL OF X                                             A07070
C      XD2= SQUARE OF THE RECIPROCAL OF X                              A07070
C      SMXD= SUM OF THE RECIPROCALS OF X                               A07070
C      SMXD2= SUM OF THE SQUARES OF THE RECIPROCALS OF X               A07070
C      KOUNT= COUNTER TO DETERMINE FIRST CARD                          A07070
C      NUMTR= IDENTIFICATION NUMBER USED WITH EACH TRANSFORM.          A07070
C                                                                      A07070
C      PROGRAMMER IDENTIFICATION CARD PUNCHED.                         A07070
       PRINT 909                                                       A07070
       CON=.43429448                                                   A07082
       SUMX=0.                                                         A07106
       SUMX2=0.                                                        A07130
       SMXA2=0.                                                        A07154
       SMXA=0.                                                         A07178
       SMXB=0.                                                         A07202
       SMXB2=0.                                                        A07226
       SMXC=0.                                                         A07250
       SMXC2=0.                                                        A07274
       SMXD=0.                                                         A07298
       SMXD2=0.                                                        A07322
C      PROGRAM BEGINS.                                                 A07346
C      READ CARD WITH NUMBER OF GROUPS.                                A07346
   22  READ1,NOGRP                                                     A07346
       KOUNT=1                                                         A07370
C      READ CARD WITH GROUP NUMBER AND NUMBER OF CASES IN GROUP.       A07394
   23  READ2,NUMGP,NOCAS                                               A07394
       LAST=0                                                          A07430
C      READ CARD WITH RAW SCORE AND SUBJECT NUMBER.                    A07454
   21  READ3,XRAW,KASNO                                                A07454
C      SUM RAW SCORE AND RAW SCORE SQUARED.                            A07490
       SUMX=SUMX+XRAW                                                  A07490
       SUMX2=SUMX2+(XRAW*XRAW)                                         A07526
C      BEGIN SQRT(X+.5) TRANSFORMATION.                                A07574
```

```
      XA2=XRAW+0.5                                                      A07574
      XA=SQRTF(XA2)                                                     A07610
      SMXA2=SMXA2+XA2                                                   A07634
      SMXA=SMXA+XA                                                      A07670
C     BEGIN SQRT(X)+SQRT(X+1) TRANSFORMATION.                           A07706
      XB=SQRTF(XRAW)+SQRTF(XRAW+1.)                                     A07706
      XB2=XB*XB                                                         A07790
      SMXB=SMXB+XB                                                      A07826
      SMXB2=SMXB2+XB2                                                   A07862
C     BEGIN LOG(X) TRANSFORMATION.                                      A07898
C     IF SCORE USED IS ZERO (0),LOG OF ONE (1) IS USED.                 A07898
      IF(XRAW)4,5,4                                                     A07898
    5 XC=(LOGF(1.))*CON                                                 A07954
      XC2=XC*XC                                                         A07990
      GO TO 6                                                           A08026
    4 XC=(LOGF(XRAW))*CON                                               A08034
      XC2=XC*XC                                                         A08070
    6 SMXC=SMXC+XC                                                      A08106
      SMXC2=SMXC2+XC2                                                   A08142
C     BEGIN RECIPROCAL OF X TRANSFORMATION.                             A08178
      XD=1./XRAW                                                        A08178
      XD2=XD*XD                                                         A08214
      SMXD=SMXD+XD                                                      A08250
      SMXD2=SMXD2+XD2                                                   A08286
C     IF FIRST TIME HERE PUNCH BARTLETT HEADER CARDS,STATEMENT NO. 7.   A08322
      IF(KOUNT-1)13,7,13                                                A08322
C     BARTLETT HEADER CARDS.                                            A08390
    7 NUMTR=1                                                           A08390
      PUNCH9,NOGRP,NUMTR                                                A08414
      NUMTR=2                                                           A08450
      PUNCH 41,NOGRP,NUMTR                                              A08474
      NUMTR=3                                                           A08510
      PUNCH 42,NOGRP,NUMTR                                              A08534
      NUMTR=4                                                           A08570
      PUNCH 43,NOGRP,NUMTR                                              A08594
      NUMTR=5                                                           A08630
      PUNCH 44,NOGRP,NUMTR                                              A08654
C     IF FIRST TIME HERE PUNCH TRANSFORM TABLE HEADER CARDS,STATMENT 14. A08690
   13 KOUNT=KOUNT+1                                                     A08690
C     TRANSFORM HEADER CARDS.                                           A08726
      IF(KOUNT-2)15,14,15                                               A08726
   14 PUNCH26                                                           A08794
      PUNCH27                                                           A08806
      PUNCH28                                                           A08818
      PUNCH29                                                           A08830
      PUNCH31                                                           A08842
      PUNCH32                                                           A08854
      PUNCH33                                                           A08866
      PUNCH16                                                           A08878
C     PUNCH TABLE VALUES.                                               A08890
   15 PUNCH17,NUMGP,KASNO,XRAW,XA,XB,XC,XD                              A08890
      LAST=LAST+1                                                       A08986Z
      IF(NOCAS-LAST)19,19,21                                            A09034
   19 CASES=NOCAS                                                       A09102
      NUMTR=1                                                           A09138
```

TRANSFORMATION PROGRAM LISTING (Cont'd)

```
C     PUNCH BARTLETT INPUT DATA CARDS.                                  A09162
      PUNCH 18,CASES,SUMX,SUMX2,NUMTR                                   A09162
      NUMTR=2                                                           A09222
      PUNCH 18,CASES,SMXA,SMXA2,NUMTR                                   A09246
      NUMTR=3                                                           A09306
      PUNCH 18,CASES,SMXB,SMXB2,NUMTR                                   A09330
      NUMTR=4                                                           A09390
      PUNCH 18,CASES,SMXC,SMXC2,NUMTR                                   A09414
      NUMTR=5                                                           A09474
      PUNCH 45,CASES,SMXD,SMXD2,NUMTR                                   A09498
C     ZERO SUMMATION AREAS                                              A09558
      SUMX=0.                                                           A09558
      SUMX2=0.                                                          A09582
      SMXA=0.                                                           A09606
      SMXA2=0.                                                          A09630
      SMXB=0.                                                           A09654
      SMXB2=0.                                                          A09678
      SMXC=0.                                                           A09702
      SMXC2=0.                                                          A09726
      SMXD=0.                                                           A09750
      SMXD2=0.                                                          A09774
C     IF ALL GROUPS WORKED STOP WITH STATEMENT NO. 24.                  A09798
      IF(NOGRP-NUMGP)24,24,23                                           A09798
   24 PRINT25                                                           A09866
      PAUSE                                                             A09878
      GO TO 22                                                          A09890
C                                                                       A09898
C     FORMAT STATMENTS FOLLOW.                                          A09898
C                                                                       A09898
    1 FORMAT(I4)                                                        A09898
    2 FORMAT(I4,I5)                                                     A09920
    3 FORMAT(F10.5,I5)                                                  A09948
9FORMAT(I4,3X1HN,6X5HSUM X,2X8HSUM X SQ,10X11HX=RAW SCORE,13XI2)        A09976
   16 FORMAT(3X1HA,3X1HB,8X1HC,7X1HD,9X1HE,10X1HF,9X1HG/)               A10116
   17 FORMAT(I4,I5,F10.2,2F10.4,F10.5,F10.6)                            A10250
   18 FORMAT(2F10.2,F10.2,33XI2)                                        A10302
41FORMAT(I4,3X1HN,6X5HSUM X,2X8HSUM X SQ,10X12HX=SQRT(X+.5),12XI2)      A10346
42FORMAT(I4,3X1HN,6X5HSUM X,2X8HSUM X SQ,10X19HX=SQRT(X)+SQRT(X+1),5XI2) A10488
43FORMAT(I4,3X1HN,6X5HSUM X,2X8HSUM X SQ,10X8HX=LOG(X),16XI2)           A10644
44FORMAT(I4,3X1HN,6X5HSUM X,2X8HSUM X SQ,10X17HX=RECIPROCAL OF X,7XI2)  A10778
   45 FORMAT(2F10.2,F10.6,33XI2)                                        A10930
   25 FORMAT(28HPUSH START TO BEGIN NEW RUN.)                           A10974
   26 FORMAT(14HA=GROUP NUMBER)                                         A11054
   27 FORMAT(16HB=SUBJECT NUMBER)                                       A11106
   28 FORMAT(11HC=RAW SCORE)                                            A11162
   29 FORMAT(12HD=SQRT(X+.5))                                           A11208
   31 FORMAT(19HE=SQRT(X)+SQRT(X+1))                                    A11256
   32 FORMAT(8HF=LOG(X))                                                A11318
   33 FORMAT(17HG=RECIPROCAL OF X/)                                     A11358
  909 FORMAT(50HPROGRAMMED BY R.C. IRONS, SCHAVMED, PENSACOLA, FLA)     A11422
C     END OF PROGRAM.                                                   A11546
      END                                                               A11546
                                                                         00162
```

TRANSFORMATION PROGRAM SYMBOL TABLE

ADDRESSES	NAME	DIMENSION	STATE LOC	FORM LOC	FIX	FLOATING	A
19999	SIN						A
19989	SINF						A
19979	COS						A
19969	COSF						A
19959	ATAN						A
19949	ATANF						A
19939	EXP						A
19929	EXPF						A
19919	LOG						A
19909	LOGF						A
19899	SQRT						A
19889	SQRTF						A
19879	SIGN						A
19869	SIGNF						A
19859	ABS						A
19849	ABSF						A
19839	RDNM						A
19829	RDNMF						A
19819				0909 11422			A
19809	CON						A
19799						43429448 00	A
19789	SUMX						A
19779						00000000 99-	A
19769	SUMX2						A
19759	SMXA2						A
19749	SMXA						A
19739	SMXB						A
19729	SMXB2						A
19719	SMXC						A
19709	SMXC2						A
19699	SMXD						A
19689	SMXD2						A
19679			0022 07346				A
19669				0001 09898			A
19659	NOGRP						A
19649	KOUNT						A
19639					0001-		A
19629			0023 07394				A
19619				0002 09920			A
19609	NUMGP						A
19599	NOCAS						A
19589	LAST						A
19579					0000-		A
19569			0021 07454				A
19559				0003 09948			A
19549	XRAW						A
19539	KASNO						A
19529					000		A
19519					001		A
19509	XA2						A
19499						50000000 00	A
19489	XA						A
19479	XB						A
19469						10000000 01	A

TRANSFORMATION PROGRAM SYMBOL TABLE (Cont'd)

19459	XB2				A
19449		0004 08034			A
19439		0005 07954			A
19429	XC				A
19419	XC2				A
19409		0006 08106			A
19399	XD				A
19389	XD2				A
19379		0013 08690			A
19369		0007 08390			A
19359	NUMTR				A
19349			0009 09976		A
19339				0002-	A
19329			0041 10346		A
19319				0003-	A
19309			0042 10488		A
19299				0004-	A
19289			0043 10644		A
19279				0005-	A
19269			0044 10778		A
19259		0015 08890			A
19249		0014 08794			A
19239			0026 11054		A
19229			0027 11106		A
19219			0028 11162		A
19209			0029 11208		A
19199			0031 11256		A
19189			0032 11318		A
19179			0033 11358		A
19169			0016 10116		A
19159			0017 10250		A
19149		0019 09102			A
19139	CASES				A
19129			0018 10302		A
19119			0045 10930		A
19109		0024 09866			A
19099			0025 10974		A

00092

```
12
1    6
    24.    227
    41.    231
    37.    233
    4Ø.    2Ø4
    37.    2Ø6
    32.    236
2    6
    4Ø.    2Ø9
    36.    222
    29.    23Ø
    44.    237
    38.    239
    35.    24Ø
3    17
    35.    2Ø5
    38.    2Ø7
    39.    211
    49.    212
    5Ø.    213
    35.    215
    36.    216
    54.    217
    52.    218
    4Ø.    22Ø
    43.    221
    38.    224
    42.    225
    38.    232
    44.    234
    46.    238
    37.    251
4    1Ø
    3Ø.    2Ø3
    34.    2Ø8
    44.    21Ø
    36.    214
    43.    219
    48.    223
    25.    226
    42.    228
    44.    229
    32.    235
5    4
    29.    247
    31.    256
    36.    261
    28.    262
6    12
    31.    246
    33.    249
    46.    25Ø
    39.    253
    4Ø.    254
    33.    255
    37.    259
    44.    26Ø
    46.    263
    45.    265
    35.    266
    25.    243
7    5
    39.    242
    45.    244
    53.    245
    37.    291
    48.    248
8    5
    47.    241
    35.    252
    24.    257
    5Ø.    258
    42.    264
9    5
    33.    27Ø
    37.    274
    37.    275
    46.    288
    35.    29Ø
1Ø   11
    36.    267
    2Ø.    269
    37.    271
    38.    272
    32.    278
    44.    281
    18.    282
    33.    284
    46.    285
    35.    286
    35.    289
11   5
    41.    268
    38.    273
    38.    276
    36.    28Ø
    27.    283
12   3
    41.    277
    48.    279
    26.    287
```

TRANSFORMED SCORES TABLE

A=GROUP NUMBER
B=SUBJECT NUMBER
C=RAW SCORE
D=SQRT(X+.5)
E=SQRT(X)+SQRT(X+1)
F=LOG(X)
G=RECIPROCAL OF X

A	B	C	D	E	F	G
1	227	24.00	4.9497	9.8989	1.38021	.041666
1	231	41.00	6.4420	12.8838	1.61278	.024390
1	233	37.00	6.1237	12.2471	1.56820	.027027
1	204	40.00	6.3639	12.7276	1.60206	.025000
1	206	37.00	6.1237	12.2471	1.56820	.027027
1	236	32.00	5.7008	11.4014	1.50515	.031250
2	209	40.00	6.3639	12.7276	1.60206	.025000
2	222	36.00	6.0415	12.0827	1.55630	.027777
2	230	29.00	5.4313	10.8623	1.46239	.034482
2	237	44.00	6.6708	13.3414	1.64345	.022727
2	239	38.00	6.2048	12.4094	1.57978	.026315
2	240	35.00	5.9581	11.9160	1.54406	.028571
3	205	35.00	5.9581	11.9160	1.54406	.028571
3	207	38.00	6.2048	12.4094	1.57978	.026315
3	211	39.00	6.2849	12.5695	1.59106	.025641
3	212	49.00	7.0356	14.0710	1.69019	.020408
3	213	50.00	7.1063	14.2124	1.69897	.020000
3	215	35.00	5.9581	11.9160	1.54406	.028571
3	216	36.00	6.0415	12.0827	1.55630	.027777
3	217	54.00	7.3824	14.7646	1.73239	.018518
3	218	52.00	7.2456	14.4912	1.71600	.019230
3	220	40.00	6.3639	12.7276	1.60206	.025000
3	221	43.00	6.5954	13.1906	1.63346	.023255
3	224	38.00	6.2048	12.4094	1.57978	.026315
3	225	42.00	6.5192	13.0381	1.62324	.023809
3	232	38.00	6.2048	12.4094	1.57978	.026315
3	234	44.00	6.6708	13.3414	1.64345	.022727
3	238	46.00	6.8190	13.6379	1.66275	.021739
3	251	37.00	6.1237	12.2471	1.56820	.027027
4	203	30.00	5.5226	11.0449	1.47712	.033333
4	208	34.00	5.8736	11.7470	1.53147	.029411
4	210	44.00	6.6708	13.3414	1.64345	.022727
4	214	36.00	6.0415	12.0827	1.55630	.027777
4	219	43.00	6.5954	13.1906	1.63346	.023255
4	223	48.00	6.9641	13.9282	1.68124	.020833
4	226	25.00	5.0497	10.0990	1.39794	.040000
4	228	42.00	6.5192	13.0381	1.62324	.023809
4	229	44.00	6.6708	13.3414	1.64345	.022727
4	235	32.00	5.7008	11.4014	1.50515	.031250
5	247	29.00	5.4313	10.8623	1.46239	.034482
5	256	31.00	5.6124	11.2246	1.49136	.032258
5	261	36.00	6.0415	12.0827	1.55630	.027777
5	262	28.00	5.3385	10.6766	1.44715	.035714
6	246	31.00	5.6124	11.2246	1.49136	.032258
6	249	33.00	5.7879	11.5755	1.51851	.030303

6	250	46.00	6.8190	13.6379	1.66275	.021739
6	253	39.00	6.2849	12.5695	1.59106	.025641
6	254	40.00	6.3639	12.7276	1.60206	.025000
6	255	33.00	5.7879	11.5755	1.51851	.030303
6	259	37.00	6.1237	12.2471	1.56820	.027027
6	260	44.00	6.6708	13.3414	1.64345	.022727
6	263	46.00	6.8190	13.6379	1.66275	.021739
6	265	45.00	6.7453	13.4905	1.65321	.022222
6	266	35.00	5.9581	11.9160	1.54406	.028571
6	243	25.00	5.0497	10.0990	1.39794	.040000
7	242	39.00	6.2849	12.5695	1.59106	.025641
7	244	45.00	6.7453	13.4905	1.65321	.022222
7	245	53.00	7.3143	14.6285	1.72427	.018867
7	291	37.00	6.1237	12.2471	1.56820	.027027
7	248	48.00	6.9641	13.9282	1.68124	.020833
8	241	47.00	6.8920	13.7838	1.67209	.021276
8	252	35.00	5.9581	11.9160	1.54406	.028571
8	257	24.00	4.9497	9.8989	1.38021	.041666
8	258	50.00	7.1063	14.2124	1.69897	.020000
8	264	42.00	6.5192	13.0381	1.62324	.023809
9	270	33.00	5.7879	11.5755	1.51851	.030303
9	274	37.00	6.1237	12.2471	1.56820	.027027
9	275	37.00	6.1237	12.2471	1.56820	.027027
9	288	46.00	6.8190	13.6379	1.66275	.021739
9	290	35.00	5.9581	11.9160	1.54406	.028571
10	267	36.00	6.0415	12.0827	1.55630	.027777
10	269	20.00	4.5276	9.0547	1.30103	.050000
10	271	37.00	6.1237	12.2471	1.56820	.027027
10	272	38.00	6.2048	12.4094	1.57978	.026315
10	278	32.00	5.7008	11.4014	1.50515	.031250
10	281	44.00	6.6708	13.3414	1.64345	.022727
10	282	18.00	4.3011	8.6015	1.25527	.055555
10	284	33.00	5.7879	11.5755	1.51851	.030303
10	285	46.00	6.8190	13.6379	1.66275	.021739
10	286	35.00	5.9581	11.9160	1.54406	.028571
10	289	35.00	5.9581	11.9160	1.54406	.028571
11	268	41.00	6.4420	12.8838	1.61278	.024390
11	273	38.00	6.2048	12.4094	1.57978	.026315
11	276	38.00	6.2048	12.4094	1.57978	.026315
11	280	36.00	6.0415	12.0827	1.55630	.027777
11	283	27.00	5.2440	10.4876	1.43136	.037037
12	277	41.00	6.4420	12.8838	1.61278	.024390
12	279	48.00	6.9641	13.9282	1.68124	.020833
12	287	26.00	5.1478	10.2951	1.41497	.038461

INPUT DATA FOR BARTLETT PROGRAM FROM TRANSFORMATION PROGRAM

12	N	SUM X	SUM X SQ	X=RAW SCORE	1
	6.00	211.00	7619.00		1
	6.00	222.00	8342.00		1
	17.00	716.00	30754.00		1
	10.00	378.00	14790.00		1
	4.00	124.00	3882.00		1
	12.00	454.00	17672.00		1
	5.00	222.00	10028.00		1
	5.00	198.00	8274.00		1
	5.00	188.00	7168.00		1
	11.00	374.00	13448.00		1
	5.00	180.00	6594.00		1
	3.00	115.00	4661.00		1
12	N	SUM X	SUM X SQ	X=SQRT(X+.5)	2
	6.00	35.70	214.00		2
	6.00	36.67	225.00		2
	17.00	110.71	724.50		2
	10.00	61.60	383.00		2
	4.00	22.42	126.00		2
	12.00	74.02	460.00		2
	5.00	33.43	224.50		2
	5.00	31.42	200.50		2
	5.00	30.81	190.50		2
	11.00	64.09	379.50		2
	5.00	30.13	182.50		2
	3.00	18.55	116.50		2
12	N	SUM X	SUM X SQ	X=SQRT(X)+SQRT(X+1)	3
	6.00	71.40	855.95		3
	6.00	73.33	899.95		3
	17.00	221.43	2897.89		3
	10.00	123.21	1531.93		3
	4.00	44.84	503.96		3
	12.00	148.04	1839.91		3
	5.00	66.86	897.97		3
	5.00	62.84	801.96		3
	5.00	61.62	761.96		3
	11.00	128.18	1517.91		3
	5.00	60.27	729.96		3
	3.00	37.10	465.97		3
12	N	SUM X	SUM X SQ	X=LOG(X)	4
	6.00	9.23	14.25		4
	6.00	9.38	14.70		4
	17.00	27.54	44.69		4
	10.00	15.69	24.70		4
	4.00	5.95	8.87		4
	12.00	18.85	29.69		4
	5.00	8.21	13.52		4
	5.00	7.91	12.60		4
	5.00	7.86	12.37		4
	11.00	16.67	25.45		4
	5.00	7.76	12.06		4
	3.00	4.70	7.42		4

INPUT DATA FOR BARTLETT PROGRAM FROM TRANSFORMATION PROGRAM (Cont'd)

12	N	SUMX,	SUM X SQ	X=RECIPROCAL OF X	5
	6.00	.17	.005393		5
	6.00	.16	.004611		5
	17.00	.41	.010120		5
	10.00	.27	.007899		5
	4.00	.13	.004276		5
	12.00	.32	.009261		5
	5.00	.11	.002671		5
	5.00	.13	.003972		5
	5.00	.13	.003668		5
	11.00	.34	.012297		5
	5.00	.14	.004123		5
	3.00	.08	.002508		5

127

4.4.14

THE BARTLETT TEST FOR HOMOGENEITY OF VARIANCE PROGRAM

This program computes the Bartlett Chi-Square Test for homogeneity of variance, degree of freedom, and the level of significance. In addition to these measures the program provides a table containing many of the computational steps in performing the Bartlett Test and gives tables values of chi square for the appropriate degrees of freedom for the following levels of significance: 0.10, 0.05, 0.02, 0.01, and 0.001.

FORMAT FOR INPUT CARDS

Card	Columns	Information	Example
EDIT	1- 4	Number of groups	0012
	65	Transform Type	1,2,3,4, or 5
CASE	1-10	Number of cases	000000013.

Blank columns may be used in place of leading zeros in the data fields for the original data, and for each set of transformed data the input consists of an EDIT card and a CASE card for each group. These are followed by the cards containing the values for the sums of (X) and sums of (X) squared. The groups of data can follow one another directly.

OPERATING THE PROGRAM

1. Place the program in the card reader.
2. Place the five decks of data cards produced by the transformation program on top of the program deck.
3. Press the RESET button on the console, the LOAD button on the card reader, and the START button on the card punch.

128

4.4.15

4. When LOAD DATA is typed, the program will automatically read the data and process the problem.

5. When END OF JOB message and instructions for initiating a new run are typed, NON PROCESS RUNOUT for the CARD PUNCH must be pressed to obtain all the output cards.

6. The output cards may be listed directly with an 80-80 board on a 407 machine.

PROCESSING TIME

The approximate computer time necessary to accomplish the Bartlett Program is given by the following equation:

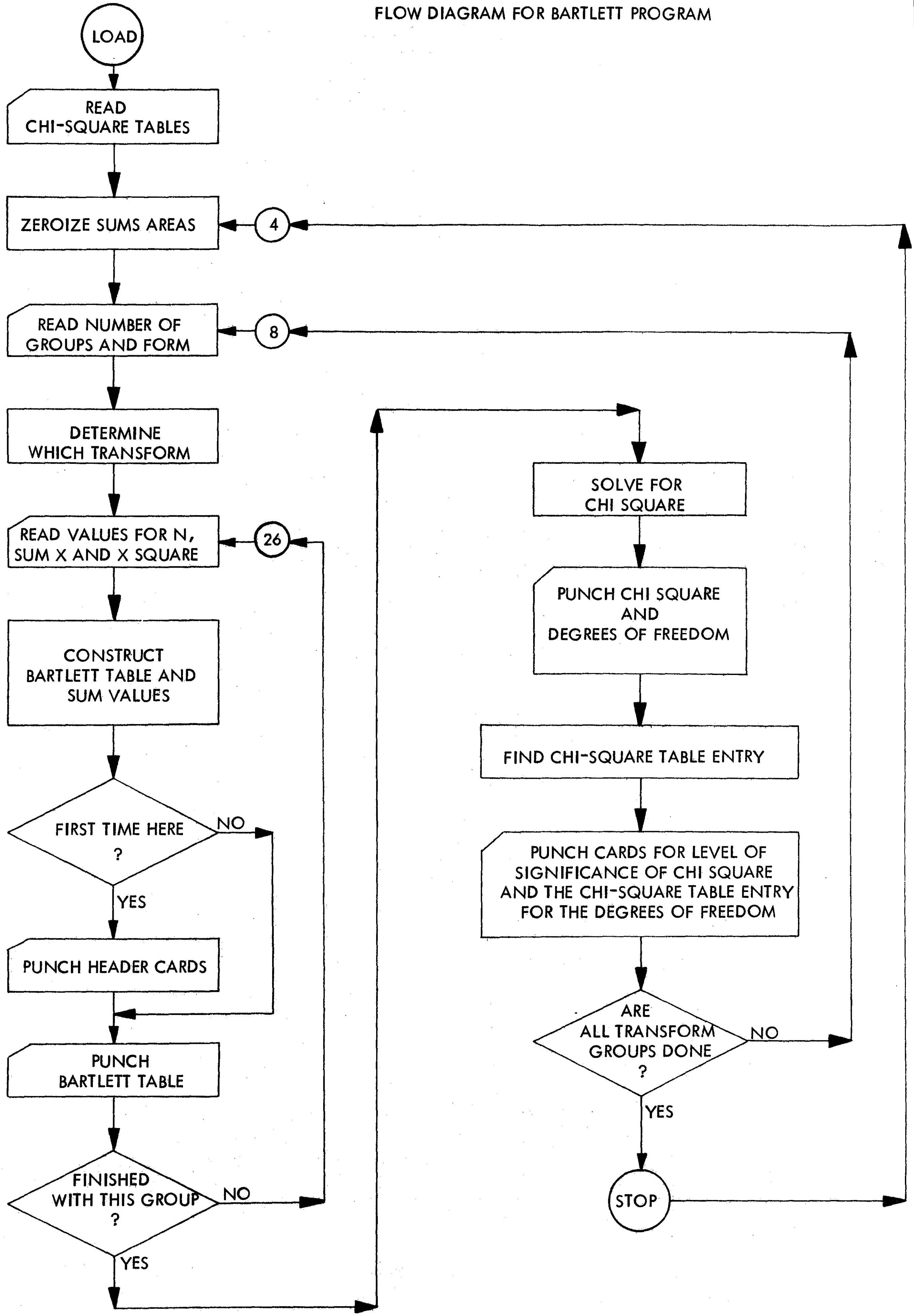
Execution time = 120 + 6K seconds

where K = number of data cards per transform group.

129

4.4.16

FLOW DIAGRAM FOR BARTLETT PROGRAM



```
C      DEFINITION OF SYMBOLS FOLLOW.                                    A07070
C                                                                       A07070
C                                                                       A07070
C      MATRX= STORAGE LEVELS FOR CHI SQUARE TABLE.                      A07070
C      SUMDF= SUMMATION OF D.F.                                         A07070
C      BIGN = SUMMATION OF N.                                           A07070
C      SMRCP= SUMMATION OF THE RECIPROCALS OF D.F.                      A07070
C      SUMS2= SUMMATION OF S SQUARE.                                    A07070
C      SMSD2= SUMMATION OF THE SUM OF X SQUARE.                         A07070
C      SMDFL= SUMMATION OF D.F. TIMES THE LOG OF S SQUARE.              A07070
C      NOGRP= NUMBER OF GROUPS.                                         A07070
C      NUMTR= TRANSFORMED DATA TYPE.                                    A07070
C      CHIDF= LEVEL OF SIGNIFICANCE.                                    A07070
C      KOUNT= A COUNTER FUNCTION.                                       A07070
C      LEVEL= LEVEL OF SIGNIFICANCE.                                    A07070
C      CASES= NUMBER OF CASES IN GROUP.                                 A07070
C      SUMX = SUM OF X.                                                 A07070
C      SUMX2= SUM OF X SQUARE.                                          A07070
C      DF   = N-1, WHERE N IS NUMBER OF SUBJECTS.                       A07070
C      RECDF= RECIPROCAL OF D.F.                                        A07070
C      S2   = S SQUARE.                                                 A07070
C      TLOG = LOG OF S SQUARE.                                          A07070
C      DFLOG= D.F. TIMES LOG OF S SQUARE.                               A07070
C      ONE  = FIRST EQUATION IN SOLVING FOR CHI SQUARE.                 A07070
C      TWO  = SECOND EQUATION IN SOLVING FOR CHI SQUARE.                A07070
C      THREE= THIRD EQUATION IN SOLVING FOR CHI SQUARE.                 A07070
C      CHI  = CHI SQUARE.                                               A07070
C      LAST =COUNTER TO FIND END OF ROUTINE.                            A07070
C                                                                       A07070
C      PROGRAM BEGINS.                                                  A07070
C                                                                       A07070
       DIMENSION T(5,30)                                                A07070
       MATRX=1                                                          A07070
C      CHI SQUARE TABLE IS READ IN AT THIS POINT.                       A07094
     1 READ 2,T(1,MATRX),T(2,MATRX),T(3,MATRX),T(4,MATRX),T(5,MATRX)    A07094
       IF(MATRX-30)3,4,4                                                A07406
     3 MATRX=MATRX+1                                                    A07474
       GO TO 1                                                          A07510
     4 CN=.43429448                                                     A07518
C      PROGRAM HEADER (IDENTIFICATION) CARDS TO BE PUNCH NOW.           A07542
       PUNCH 5                                                          A07542
       PUNCH 6                                                          A07554
       PUNCH 7                                                          A07566
C      ZEROING SUMMATION AREA NEXT TO BE DONE.                          A07578
       LAST=1                                                           A07578
     8 SUMDF=0.                                                         A07602
       BIGN=0.                                                          A07626
       SMRCP=0.                                                         A07650
       SUMS2=0.                                                         A07674
       SMSD2=0.                                                         A07698
       SMDFL=0.                                                         A07722
C      READ VALUES FOR NUMBER OF GROUPS AND TYPE TRANSFORM USED.        A07746
       READ 9,NOGRP,NUMTR                                               A07746
C      ROUTINE FOR IDENTIFICATION OF TRANSFORM TYPE.                    A07782
       IF(NUMTR-1)11,12,11                                              A07782
```

```
   12 PUNCH 18
      GO TO 13
   11 IF(NUMTR-2)14,15,14
   15 PUNCH 19
      GO TO 13
   14 IF(NUMTR-3)16,17,16
   17 PUNCH 21
      GO TO 13
   16 IF(NUMTR-4)22,23,22
   23 PUNCH 24
      GO TO 13
   22 PUNCH 25
   13 CHIDF=NOGRP-1
      KOUNT=1
      LEVEL=CHIDF
   26 CASES=0.
      SUMX=0.
      SUMX2=0.
C     READ VALUES FOR N, SUM OF X AND SUM OF X SQUARE.
      READ 27,CASES,SUMX,SUMX2
C     BEGIN COMPUTATIONS
      DF=CASES-1.
      RECDF=1./DF
      SUMD2=SUMX2-((SUMX*SUMX)/CASES)
      S2=SUMD2/DF
      TLOGS=(LOGF(S2))*CN
      DFLOG=DF*TLOGS
      BIGN=BIGN+CASES
      SUMDF=SUMDF+DF
      SMRCP=SMRCP+RECDF
      SMSD2=SMSD2+SUMD2
      SUMS2=SUMS2+S2
      SMDFL=SMDFL+DFLOG
C     IF FIRST TIME HERE, PUNCH TABLE HEADER CARDS, IF NOT PUNCH TABLE.
      IF(KOUNT-1)28,29,28
   29 PUNCH 31
      PUNCH 32
   28 PUNCH 33,KOUNT,CASES,DF,RECDF,SUMD2,S2,TLOGS,DFLOG
C     DETERMINE IF LAST OF GROUP.
      IF(NOGRP-KOUNT)34,34,35
   35 KOUNT=KOUNT+1
C     GO BACK AND READ A NEW DATA CARD FOR THIS GROUP.
      GO TO 26
C     MAJOR EQUATIONS FOR THE TEST.
   34 ONE=(LOGF(SMSD2/SUMDF))*CN
      TWO=((SUMDF*ONE)-SMDFL)*2.3026
      THREE=1.+(1./(3.*CHIDF))*(SMRCP-(1./SUMDF))
      CHI=TWO/THREE
C     PUNCH VALUES FOR THE SUMS,CHI SQUARE AND LEVEL OF SIGNIFICANCE.
      PUNCH 36,BIGN,SUMDF,SMRCP,SMSD2,SUMS2,SMDFL
      PUNCH 37,CHI,LEVEL
C     FOLLOWING PROCEDURE FOR DETERMING IF CHI SQUARE IS SIGNIFICANT.
      IF(T(1,LEVEL)-CHI)38,39,41
   41 PUNCH 42
      GO TO 43
```

```
   39 PUNCH 44                                                               A09698
      GO TO 43                                                               A09710
   38 IF(T(2,LEVEL)-CHI)45,46,47                                             A09718
   46 PUNCH 48                                                               A09834
      GO TO 43                                                               A09846
   47 PUNCH 49                                                               A09854
      GO TO 43                                                               A09866
   45 IF(T(3,LEVEL)-CHI)51,52,53                                             A09874
   53 PUNCH 54                                                               A09990
      GO TO 43                                                               A10002
   52 PUNCH 55                                                               A10010
      GO TO 43                                                               A10022
   51 IF(T(4,LEVEL)-CHI)56,57,58                                             A10030
   58 PUNCH 59                                                               A10146
      GO TO 43                                                               A10158
   57 PUNCH 61                                                               A10166
      GO TO 43                                                               A10178
   56 IF(T(5,LEVEL)-CHI)62,63,64                                             A10186
   64 PUNCH 65                                                               A10302
      GO TO 43                                                               A10314
   63 PUNCH 66                                                               A10322
      GO TO 43                                                               A10334
   62 PUNCH 67                                                               A10342
   43 PUNCH 68                                                               A10354
      PUNCH 69,T(1,LEVEL),T(2,LEVEL),T(3,LEVEL),T(4,LEVEL),T(5,LEVEL)        A10366
C     DETERMINE IF THE JOB IS DONE. IF FINISHED TYPE MESSAGE.                A10678
      IF(LAST-5)71,72,72                                                     A10678
   71 LAST=LAST+1                                                            A10746
      GO TO 8                                                                A10782
   72 PRINT 73                                                               A10790
      PAUSE                                                                  A10802
      GO TO 4                                                                A10814
C                                                                            A10822
C     THE FOLLOWING CARDS ARE FOR FORMAT OF THE PROGRAM ONLY                 A10822
C                                                                            A10822
9FORMAT(I4,59XI2)                                                            A10822
27FORMAT(F10.0,1XF10.0,1XF10.0)                                              A10856
68FORMAT(50HLEVEL OF SIGNIFICANCE    .100    .050    .020    .010,3X4H.001)  A10902
69FORMAT(22HCHI SQUARE TABLE ENTRY,5F7.3/)                                   A11048
31FORMAT(5H CASE,25X8HSUM X(I),5X4HS(I),4X8HLOG S(I),1X13H(N-1)LOG S(I))     A11146
32FORMAT(4H  NO,3X1HN,3X4HD.F.,3X6H1/D.F.,7X2HSQ,10X2HSQ,8X2HSQ,9X2HSQ/)     A11302
33FORMAT(I5,2F5.0,F10.5,F13.5,3F11.5)                                        A11476
36FORMAT(/5HSUMS ,2F5.0,F10.5,F13.5,F11.5,11XF11.5/)                         A11534
37FORMAT(15X11HCHI SQUARE=F10.3,5X20H DEGREES OF FREEDOM=I4/)                A11616
2FORMAT(5F7.3)                                                               A11738
5FORMAT(41HBARTLETT TEST FOR HOMOGENEITY OF VARIANCE)                        A11780
42FORMAT(20X23HNOT SIGNIFICANT AT .100/)                                     A11886
44FORMAT(20X19HSIGNIFICANT AT .100/)                                         A11968
48FORMAT(20X19HSIGNIFICANT AT .050/)                                         A12042
49FORMAT(20X33HSIGNIFICANT BETWEEN .100 AND .050/)                           A12116
54FORMAT(20X33HSIGNIFICANT BETWEEN .050 AND .020/)                           A12218
55FORMAT(20X19HSIGNIFICANT AT .020/)                                         A12320
59FORMAT(20X33HSIGNIFICANT BETWEEN .020 AND .010/)                           A12394
61FORMAT(20X19HSIGNIFICANT AT .010/)                                         A12496
65FORMAT(20X33HSIGNIFICANT BETWEEN .010 AND .001/)                           A12570
```

```
66FORMAT(20X19HSIGNIFICANT AT .001/)                                    A12672
67FORMAT(20X12HBEYOND   .001/)                                          A12746
6FORMAT(37HREFERANCE EDWARDS REV. ED., P,125-129/)                      A12806
7FORMAT(50HPROGRAMMED BY R.C. IRONS  SCHAVMED, PENSACOLA,FLA./)         A12910
18FORMAT(25HDATA FROM RAW SCORES USED/)                                 A13040
19FORMAT(40HDATA FROM SQRT(X+.5) TRANSFORMATION USED/)                  A13120
21FORMAT(47HDATA FROM SQRT(X)+SQRT(X+1) TRANSFORMATION USED/)           A13230
24FORMAT(36HDATA FROM LOG(X) TRANSFORMATION USED/)                      A13354
25FORMAT(30HDATA FROM RECIPROCAL OF X USED/)                            A13456
73FORMAT(41HJOB COMPLETED, PUSH START TO BEGIN AGAIN.//)                A13546
C                                                                       A13662
C       END OF PROGRAM.                                                 A13662
C                                                                       A13662
        END                                                             A13662
                                                                          00179
```

BARTLETT PROGRAM SYMBOL TABLE

ADDRESSES		NAME	DIMENSION	STATE	LOC	FORM	LOC	FIX	FLOATING	A
19999		SIN								A
19989		SINF								A
19979		COS								A
19969		COSF								A
19959		ATAN								A
19949		ATANF								A
19939		EXP								A
19929		EXPF								A
19919		LOG								A
19909		LOGF								A
19899		SQRT								A
19889		SQRTF								A
19879		SIGN								A
19869		SIGNF								A
19859		ABS								A
19849		ABSF								A
19839		RDNM								A
19829		RDNMF								A
19819	18329	T	0005 0030							A
18319		MATRX								A
18309								0001-		A
18299				0001	07094					A
18289						0002	11738			A
18279								0030-		A
18269								000		A
18259				0003	07474					A
18249				0004	07518					A
18239		CN								A
18229									43429448 00	A
18219						0005	11780			A
18209						0006	12806			A
18199						0007	12910			A
18189		LAST								A
18179				0008	07602					A
18169		SUMDF								A
18159									00000000 99-	A
18149		BIGN								A
18139		SMRCP								A
18129		SUMS2								A
18119		SMSD2								A
18109		SMDFL								A
18099						0009	10822			A
18089		NOGRP								A
18079		NUMTR								A
18069				0011	07870					A
18059				0012	07850					A
18049						0018	13040			A
18039				0013	08146					A
18029								0002-		A
18019				0014	07958					A
18009				0015	07938					A
17999						0019	13120			A
17989								0003-		A
17979				0016	08046					A

135 4.4.22

17969		0017 08026				A
17959			0021 13230			A
17949				0004-		A
17939		0022 08134				A
17929		0023 08114				A
17919			0024 13354			A
17909			0025 13456			A
17899	CHIDF					A
17889	KOUNT					A
17879	LEVEL					A
17869		0026 08254				A
17859	CASES					A
17849	SUMX					A
17839	SUMX2					A
17829			0027 10856			A
17819	DF					A
17809					10000000 01	A
17799	RECDF					A
17789	SUMD2					A
17779				002		A
17769	S2					A
17759	TLOGS					A
17749				001		A
17739	DFLOG					A
17729		0028 08934				A
17719		0029 08910				A
17709			0031 11146			A
17699			0032 11302			A
17689			0033 11476			A
17679		0034 09154				A
17669		0035 09110				A
17659	ONE					A
17649	TWO					A
17639					23026000 01	A
17629	THREE					A
17619					30000000 01	A
17609				005		A
17599	CHI					A
17589			0036 11534			A
17579			0037 11616			A
17569		0038 09718				A
17559		0039 09698				A
17549		0041 09678				A
17539			0042 11886			A
17529		0043 10354				A
17519			0044 11968			A
17509		0045 09874				A
17499		0046 09834				A
17489		0047 09854				A
17479			0048 12042			A
17469			0049 12116			A
17459		0051 10030				A
17449		0052 10010				A
17439		0053 09990				A
17429			0054 12218			A

BARTLETT PROGRAM SYMBOL TABLE (Cont'd)

17419		0055 12320		A	
17409	0056 10186			A	
17399	0057 10166			A	
17389	0058 10146			A	
17379		0059 12394		A	
17369		0061 12496		A	
17359	0062 10342			A	
17349	0063 10322			A	
17339	0064 10302			A	
17329		0065 12570		A	
17319		0066 12672		A	
17309		0067 12746		A	
17299		0068 10902		A	
17289		0069 11048		A	
17279			0005-	A	
17269	0071 10746			A	
17259	0072 10790			A	
17249		0073 13546		A	
					00128

CHI-SQUARE TABLE

2.706	3.841	5.412	6.635	10.83
4.605	5.991	7.824	9.210	13.82
6.251	7.815	9.837	11.341	16.27
7.779	9.488	11.668	13.277	18.46
9.236	11.070	13.388	15.086	20.52
10.645	12.592	15.033	16.812	22.46
12.017	14.067	16.622	18.475	24.32
13.362	15.507	18.168	20.090	26.12
14.684	16.919	19.679	21.666	27.88
15.987	18.307	21.161	23.209	29.59
17.275	19.675	22.618	24.725	31.26
18.549	21.026	24.054	26.217	32.91
19.812	22.362	25.472	27.688	34.53
21.064	23.685	26.873	29.141	36.12
22.307	24.996	28.259	30.578	37.70
23.542	26.296	29.633	32.000	39.25
24.769	27.587	30.995	33.409	40.79
25.989	28.869	32.346	34.805	42.31
27.204	30.144	33.687	36.191	43.82
28.412	31.410	35.020	37.566	45.32
29.615	32.671	36.343	38.932	46.80
30.813	33.924	37.659	40.289	48.27
32.007	35.172	38.968	41.638	49.73
33.196	36.415	40.270	42.980	51.18
34.382	37.652	41.566	44.314	52.62
35.563	38.885	42.856	45.642	54.05
36.741	40.113	44.140	46.963	55.48
37.916	41.337	45.419	48.278	56.89
39.087	42.557	46.693	49.588	58.30
40.256	43.773	47.962	50.892	59.70

138

4.4.25

BARTLETT PROGRAM OUTPUT

BARTLETT TEST FOR HOMOGENEITY OF VARIANCE
REFERENCE EDWARDS REV. ED., P.125-129

PROGRAMMED BY R.C. IRONS SCHAVMED PENSACOLA,FLA

DATA FROM RAW SCORES USED

CASE NO	N	D.F.	1/D.F.	SUM X(I) SQ	S(I) SQ	LOG S(I) SQ	(N-1)LOG S(I) SQ
1	6.	5.	.20000	198.83330	39.76666	1.59951	7.99759
2	6.	5.	.20000	128.00000	25.60000	1.40824	7.04120
3	17.	16.	.06250	597.76500	37.36031	1.57241	25.15856
4	10.	9.	.11111	501.60000	55.73333	1.74611	15.71503
5	4.	3.	.33333	38.00000	12.66666	1.10266	3.30798
6	12.	11.	.09090	495.66700	45.06063	1.65379	18.19177
7	5.	4.	.25000	171.20000	42.80000	1.63144	6.52577
8	5.	4.	.25000	433.20000	108.30000	2.03462	8.13851
9	5.	4.	.25000	99.20000	24.80000	1.39445	5.57780
10	11.	10.	.10000	732.00000	73.20000	1.86451	18.64511
11	5.	4.	.25000	114.00000	28.50000	1.45484	5.81937
12	3.	2.	.50000	252.66670	126.33335	2.10151	4.20303
SUMS	89.	77.	2.59785	3762.13200	620.12096		126.32178

CHI SQUARE= 7.958 DEGREES OF FREEDOM= 11

NOT SIGNIFICANT AT .100

LEVEL OF SIGNIFICANCE	.100	.050	.020	.010	.001
CHI SQUARE TABLE ENTRY	17.275	19.675	22.618	24.725	31.260

139

4.4.26

BARTLETT PROGRAM OUTPUT (Cont'd)

DATA FROM SQRT(X+.5) TRANSFORMATION USED

CASE NO	N	D.F.	1/D.F.	SUM X(I) SQ	S(I) SQ	LOG S(I) SQ	(N-1)LOG S(I) SQ
1	6.	5.	.20000	1.58500	.31700	-.49894	-2.49470
2	6.	5.	.20000	.88518	.17703	-.75193	-3.75969
3	17.	16.	.06250	3.51742	.21983	-.65789	-10.52633
4	10.	9.	.11111	3.54400	.39377	-.40474	-3.64273
5	4.	3.	.33333	.33590	.11196	-.95091	-2.85273
6	12.	11.	.09090	3.41997	.31090	-.50737	-5.58107
7	5.	4.	.25000	.98702	.24675	-.60773	-2.43093
8	5.	4.	.25000	3.05672	.76418	-.11680	-.46721
9	5.	4.	.25000	.64878	.16219	-.78996	-3.15985
10	11.	10.	.10000	6.08836	.60883	-.21549	-2.15499
11	5.	4.	.25000	.93662	.23415	-.63049	-2.52198
12	3.	2.	.50000	1.79917	.89958	-.04595	-.09191
SUMS	89.	77.	2.59785	26.80414	4.44623		-39.68417

CHI SQUARE= 9.386 DEGREES OF FREEDOM= 11

NOT SIGNIFICANT AT .100

LEVEL OF SIGNIFICANCE	.100	.050	.020	.010	.001
CHI SQUARE TABLE ENTRY	17.275	19.675	22.618	24.725	31.260

140

4.4.27

BARTLETT PROGRAM OUTPUT (Cont'd)

DATA FROM SQRT(X)+SQRT(X+1) TRANSFORMATION USED

CASE NO	N	D.F.	1/D.F.	SUM X(I) SQ	S(I) SQ	LOG S(I) SQ	(N-1)LOG S(I) SQ
1	6.	5.	.20000	6.29000	1.25800	.09968	.49840
2	6.	5.	.20000	3.73518	.74703	-.12665	-.63329
3	17.	16.	.06250	13.69920	.85620	-.06742	-1.07879
4	10.	9.	.11111	13.85960	1.53995	.18750	1.68757
5	4.	3.	.33333	1.30360	.43453	-.36197	-1.08593
6	12.	11.	.09090	13.58990	1.23544	.09182	1.01005
7	5.	4.	.25000	3.91808	.97952	-.00898	-.03594
8	5.	4.	.25000	12.18688	3.04672	.48383	1.93533
9	5.	4.	.25000	2.55512	.63878	-.19464	-.77859
10	11.	10.	.10000	24.26350	2.42635	.38495	3.84953
11	5.	4.	.25000	3.46542	.86635	-.06230	-.24921
12	3.	2.	.50000	7.16667	3.58333	.55428	1.10857
SUMS	89.	77.	2.59785	106.03315	17.61223		6.22769

CHI SQUARE= 9.548 DEGREES OF FREEDOM= 11

NOT SIGNIFICANT AT .100

LEVEL OF SIGNIFICANCE	.100	.050	.020	.010	.001
CHI SQUARE TABLE ENTRY	17.275	19.675	22.618	24.725	31.260

DATA FROM LOG(X) TRANSFORMATION USED

CASE NO	N	D.F.	1/D.F.	SUM X(I) SQ	S(I) SQ	LOG S(I) SQ	(N-1)LOG S(I) SQ
1	6.	5.	.20000	.05118	.01023	-1.98983	-9.94917
2	6.	5.	.20000	.03593	.00718	-2.14346	-10.71732
3	17.	16.	.06250	.07520	.00470	-2.32790	-37.24643
4	10.	9.	.11111	.08239	.00915	-2.03836	-18.34531
5	4.	3.	.33333	.01937	.00645	-2.18987	-6.56963
6	12.	11.	.09090	.07979	.00725	-2.13943	-23.53376
7	5.	4.	.25000	.03918	.00979	-2.00899	-8.03598
8	5.	4.	.25000	.08638	.02159	-1.66564	-6.66258
9	5.	4.	.25000	.01408	.00352	-2.45345	-9.81382
10	11.	10.	.10000	.18737	.01873	-1.72729	-17.27293
11	5.	4.	.25000	.01648	.00412	-2.38510	-9.54041
12	3.	2.	.50000	.05666	.02833	-1.54770	-3.09540
SUMS	89.	77.	2.59785	.74403	.13109		-160.78279

CHI SQUARE= 12.034 DEGREES OF FREEDOM= 11

NOT SIGNIFICANT AT .100

LEVEL OF SIGNIFICANCE	.100	.050	.020	.010	.001
CHI SQUARE TABLE ENTRY	17.275	19.675	22.618	24.725	31.260

BARTLETT PROGRAM OUTPUT (Cont'd)

DATA FROM RECIPROCAL OF X USED

CASE NO	N	D.F.	1/D.F.	SUM X(I) SQ	S(I) SQ	LOG S(I) SQ	(N-1)LOG S(I) SQ
1	6.	5.	.20000	.00057	.00011	-3.93829	-19.69148
2	6.	5.	.20000	.00034	.00006	-4.16199	-20.80995
3	17.	16.	.06250	.00023	.00001	-4.83907	-77.42515
4	10.	9.	.11111	.00060	.00006	-4.16962	-37.52662
5	4.	3.	.33333	.00005	.00001	-4.76955	-14.30865
6	12.	11.	.09090	.00072	.00006	-4.17946	-45.97406
7	5.	4.	.25000	.00025	.00006	-4.20238	-16.80954
8	5.	4.	.25000	.00059	.00014	-3.82973	-15.31895
9	5.	4.	.25000	.00028	.00007	-4.14266	-16.57067
10	11.	10.	.10000	.00178	.00017	-3.74765	-37.47654
11	5.	4.	.25000	.00020	.00005	-4.29456	-17.17825
12	3.	2.	.50000	.00037	.00018	-3.72738	-7.45476
SUMS	89.	77.	2.59785	.00603	.00104		-326.54467

CHI SQUARE= 22.221 DEGREES OF FREEDOM= 11

SIGNIFICANT BETWEEN .050 AND .020

LEVEL OF SIGNIFICANCE	.100	.050	.020	.010	.001
CHI SQUARE TABLE ENTRY	17.275	19.675	22.618	24.725	31.260

00

143

4.4.30

REFERENCE

1. Edwards, A.L., Experimental Design in Psychological Research. New York: Holt, Rinehart and Winston, 1960.

144

4.4.31

A GENERAL FUNCTION SUBPROGRAM

M. E. Munroe

University of New Hampshire

A General Function Subprogram

by M.F. Munroe, University of New Hampshire

This is a Fortran II function subprogram that accepts X from the main program and returns F(X). This single subprogram can be used to yield values of any desired function. The function to be computed is controlled by data read into the main program.

The code for describing a function by numerical data is as follows. A block of four integers will give instructions to combine to functions in a specified way and store the result. The first integer in each block of four specifies the combining operation to be performed. The next two specify the basic functions (or previously stored results) that are to be combined in this way. The fourth integer specifies a label for storage. The specific code chosen is this:

Operations	Library Functions	Identity Function	Constants	Storage
1. addition	11. log	[f(x) = x]	30	101 - 115
2. subtraction	12. sin	21		
3. multiplication	13. cos			
4. division	14. exp			
5. composition	15. sqrt			
[f(g(x))]	16. atan			
6. integer exponentiation	17. abs			

Every time a 30 appears there must be an auxiliary entry to indicate what constant. Thus, the data consists of a vector of 4k fixed point numbers plus another vector of floating point numbers - one for each 30 in the first vector.

Illustration: Suppose we want the subprogram to compute

$$\sqrt{\sin \pi x + 3x^5} \quad .$$

The vector of integers would be as follows, and we indicate at each storage stage what is being stored.

			πx				$\sin \pi x$				x^5
3	30	21	101	5	12	101	102	6	21	30	103

			$3x^5$				$\sin \pi x+3x^5$				$\sqrt{\sin \pi x + 3x^5}$
3	30	103	104	1	102	104	105	5	15	105	106

The vector of constants would be

3.1415926   5.   3.

Note that the exponent 5 is read as a floating point number. The subprogram will convert it to an integer before exponentiating.

A listing of the subprogram follows. The idea is to use these code numbers as branching controls by means of IF statements and computed GO TO statements.

```
C       GENERAL FUNCTION SUBPROGRAM
        FUNCTION F(X)
        DIMENSION I(200),C(100),FF(15)
        COMMON I,C,NI
C       INITIALIZE CONSTANT COUNTER
        KON=1
        DO 100 J=1,NI,4
        IJ=I(J)
C       PREPARE FPR FIRST PASS (THROUGH SECOND FUNCTION)
        L=2
        S=X
 3      M=J+L
        IM=I(M)
C       BRANCH FOR FUNCTION, CONSTANT, STORED RESULT
        IF(IM-30)1,41,2
C       BRANCH FOR LIBRARY, IDENTITY FUNCTION
 1      IF(IM-20)4,4,42
C       LIBRARY FUNCTION
 4      K=IM-10
C       BRANCH TO SPECIFIC LIBRARY FUNCTION
        GO TO (43,44,45,46,47,48,49),K
C       STORED RESULT
 2      K=IM-100
        R=FF(K)
        GO TO 65
C       CONSTANT
 41     R=C(KON)
C       RESET CONSTANT COUNTER
        KON=KON+1
        GO TO 65
C       IDENTITY FUNCTION
 42     R=S
        GO TO 65
C       SPECIFIC LIBRARY FUNCTIONS
 43     R=LOGF(S)
        GO TO 65
 44     R=SINF(S)
        GO TO 65
 45     R=COSF(S)
        GO TO 65
 46     R=EXPF(S)
        GO TO 65
 47     R=SQRTF(S)
        GO TO 65
 48     R=ATANF(S)
        GO TO 65
 49     R=ABSF(S)
```

148

4.53

```
C       BRANCH ACCORDING AS THIS IS FIRST OR SECOND PASS
 65     GO TO (66,67),L
C       STORE RESULT OF FIRST PASS AS Z
 67     Z=R
C       PREPARE FOR SECOND PASS (THROUGH FIRST FUNCTION)
        L=1
C       START SECOND PASS UNLESS OPERATION IS COMPOSITION
        IF(IJ-5)3,11,3
C       RESET S FOR COMPOSITION
 11     S=Z
C       START SECOND PASS
        GO TO 3
C       AFTER SECOND PASS BRANCH TO OPERATION CALLED FOR
 66     GO TO (31,32,33,34,35,36),IJ
C       PERFORM OPERATION AND RECORD RESULT AS F
 31     F=R+Z
        GO TO 70
 32     F=R-Z
        GO TO 70
 33     F=R*Z
        GO TO 70
 34     F=R/Z
        GO TO 70
 35     F=R
        GO TO 70
 36     IZ=Z
        F=R**IZ
C       NOTE STORAGE INSTRUCTIONS
 70     K=I(J+3)-100
C       STORE RESULT AS BOTH FF(K) AND AS F
C       IF DO LOOP CONTINUES IT WILL BE CALLED AS FF(K)
C       IF RETURN TO MAIN PROGRAM IT WILL BE CALLED AS F
 100    FF(K)=F
        RETURN
        END
```

FORTRAN PROGRAMS FOR COMPUTING ELLIPTIC INTEGRALS AND FUNCTIONS

Henry E. Fettis
James C. Caslin

May 1964

Applied Mathematics Research Laboratory
Aerospace Research Laboratories
Office of Aerospace Research
Wright-Patterson Air Force Base, Ohio

4.6.D

FORTRAN PROGRAMS FOR COMPUTING ELLIPTIC INTEGRALS AND FUNCTIONS

1. Scope of the Programs

The programs presented in this paper provide subroutines for direct computation of the three types of elliptic integrals in so-called "Legendre Normal Form". They are

a. Integrals of the first kind:

$$F(\theta,k)=\int_0^\theta \frac{d\phi}{\sqrt{1-k^2\sin^2\phi}} \quad ; \qquad 0\le k^2<1 \; ; \quad 0\le\theta\le\pi/2 .$$

b. Integrals of the second kind:

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

c. Integrals of the third kind:

$$\Pi(\theta,\alpha^2,k)=\int_0^\theta \frac{d\phi}{(1-\alpha^2\sin^2\phi)\sqrt{1-k^2\sin^2\phi}} \; , \; -\infty<\alpha^2<\infty .$$

The quantity k is called the modulus, θ the amplitude and α^2 the parameter.

Closely associated with the above are the Jacobian elliptic functions, which are defined as follows:

Let

$$F(\theta,k) = u \; .$$

Then θ, the "amplitude" of u, is written as

$$\theta = \text{am}\, u \; .$$

Similarly $\sin\theta$ and $\cos\theta$ or the "sine-amplitude" and "cosine amplitude" of u are written

$$\sin\theta = \text{sn}\,u$$

and

$$\cos\theta = \text{cn}\,u \; .$$

In addition, the quantity $\sqrt{1-k^2\sin^2\theta}$, known as the "delta-amplitude" is introduced and is denoted by dnu. Since $\frac{dF}{d\theta} = \frac{1}{\sqrt{1-k^2\sin^2\theta}} = 1/\text{dnu}$, it follows that $\frac{d}{du}[\text{am}(u)] = \text{dnu}$. Thus, three types of integrals in Legendre normal form may also be expressed as follows:

$$
\begin{aligned}
&F(\theta,k) = u;\\
3.1\qquad &E(\theta,k) = \int_0^u dn^2 v\, dv;\\
&\Pi(\theta,\alpha^2,k) = \int_0^u \frac{dv}{1-\alpha^2 sn^2 v}\ .
\end{aligned}
$$

2. <u>Special Values</u>

When k = 0 or 1, the three canonical forms are reducible to elementary functions. Specifically,

a. $F(\theta,0) = E(\theta,0) = \theta$

b. $\Pi(\theta,\alpha^2,0) = \frac{1}{\sqrt{1-\alpha^2}}\tan^{-1}\left[\sqrt{1-\alpha^2}\tan\theta\right], \quad \alpha^2<1;$

$= \frac{1}{2\sqrt{1-\alpha^2}}\ln\left|\frac{1+\sqrt{1-\alpha^2}\tan\theta}{1-\sqrt{1-\alpha^2}\tan\theta}\right|, \quad \alpha^2>1;$

$= \tan\theta$, if $\alpha^2 = 1$

c. $F(\theta,1) = \ln(\tan\theta + \sec\theta)$

d. $E(\theta,1) = \sin\theta$

e. $\Pi(\theta,\alpha^2,1) = \frac{1}{1-\alpha^2}\left\{\ln(\tan\theta+\sec\theta) - \frac{\alpha}{2}\ln\left|\frac{1+\alpha\sin\theta}{1-\alpha\sin\theta}\right|\right\}$

$\alpha^2>0;\ \alpha^2\neq 1;\ \alpha\sin\theta\neq 1;$

$= \frac{1}{1-\alpha^2}\left\{\ln(\tan\theta+\sec\theta) + |\alpha|\tan^{-1}(|\alpha|\sin\theta)\right\},$

$\alpha^2<0\ ;$

$= \frac{1}{2}\left\{\ln(\tan\theta+\sec\theta) + \tan\theta\sec\theta\right\},$

$\alpha^2 = 1\ .$

3. Mathematical Formulation

The programs presented here are based on a transformation, known as Gauss' transformation or sometimes Landen's second transformation. It permits each of the three normal integrals to be expressed in terms of elementary functions and similar ones with a lesser modulus. Since the same transformation may be applied to the new set, and the process continued until a modulus of sufficiently small magnitude is obtained, a point is ultimately reached where the integrals reduce to element functions as outlined in section 2. It is then a simple matter to go backwards and find the desired values of the original integrals. The transformation goes as follows:

a. A new modulus k_1 is computed from the formula

$$k_1 = \frac{1-k'}{1-k'}$$

where $k' = \sqrt{1-k^2}$ is known as the complementary modulus.

b. A new amplitude θ_1 is computed from

$$\sin\theta_1 = \frac{(1+k')\sin\theta}{1+\sqrt{1-k^2\sin^2\theta}}$$

c. A new parameter α_1 is computed from

$$\alpha_1^2 = \alpha^2\left(\frac{1+p}{1+k'}\right)^2$$

where $p = \sqrt{1-k^2/\alpha^2}$.

The original integrals, functions of k, θ, α^2, may now be expressed in terms of ones with arguments $k_1, \theta_1, \alpha_1^2$, as follows:

$$F(\theta,k) = (1+k_1)F(\theta_1,k_1)$$

$$E(\theta,k) = (1+k')E(\theta_1,k_1) - k'F(\theta_1,k_1) + \frac{k^2\sin\theta\cos\theta}{1+\sqrt{1-k^2\sin^2\theta}}$$

$$\Pi(\theta,\alpha^2,k) = \frac{1}{p}\left\{2(1+k_1)\Pi(\theta_1,\alpha_1^2,k_1) - \Pi(\theta,\alpha^2,0) - (1-p)F(\theta,k)\right\}.$$

d. Steps a,b,c are repeated with θ, k, α^2 replaced by $\theta_1, k_1, \alpha_1^2$ and this procedure is continued until at the n-th step, the modulus k_N is essentially zero. $F(\theta_N,k_N)$, $E(\theta_N,k_N)$ and $\Pi(\theta_N,\alpha_N,k_N)$ are then elementary functions as given by formulae 2-a, 2b.

e. From the values of $F(\theta_N,k_N)$, $E(\theta_N,k_N)$, $\Pi(\theta_N,\theta_N, k_N)$ just found, equation 3.1 is employed to find $F(\theta_{N-1},k_{N-1})$, $E(\theta_{N-1},k_{N-1})$, $\Pi(\theta_{N-1},k_{N-1})$ and this process is continued until finally the values of the functions of the original arguments are obtained.

4. Recurrence Formulae

As an alternative to the above described process, which requires that intermediate values of k, α^2, θ be saved, a general recurrence type formula for direct computation may be written. This is accomplished by defining sequences of numbers Q_n, R_n, S_n, T_n, X_n, Y_n, Z_n, according to the following equations:

$$Q_o = 0 \qquad X_o = 0$$

$$k_o = 1 \qquad Y_o = 1$$

$$S_o = 1 \qquad Z_o = 0$$

$$T_o = 0$$

$$Q_{n+1} = Q_n + \left\{\frac{R_n^2 \sin\theta_n \cos\theta_n}{1+\sqrt{1-R_n^2\sin^2\theta_n}}\right\} T_n$$

$$R_{n+1} = (1+k_{n+1})R_n$$

$$S_{n+1} = (1+k_{n+1})\left\{S_n + k'_n T_n\right\}$$

$$T_{n+1} = (1+k'_n)\, T_n;$$

$$X_{n+1} = (1+k_{n+1})[X_n + \frac{p_n - 1}{p_n} Y_n] ,$$

$$Y_{n+1} = \frac{2(1+k_{n+1})}{p_n} Y_n ,$$

$$Z_{n+1} = Z_n + \frac{G_n}{p_n} Y_n ,$$

where

$$k_{n+1} = \frac{1-k'_n}{1+k'_n} ,$$

$$\alpha^2_{n+1} = \alpha^2_n \left(\frac{1+p_n}{1+k'_n}\right)^2 ,$$

$$\sin\theta_{n+1} = \frac{(1+k'_n)\sin\theta_n}{1+\sqrt{1-k^2_n \sin^2\theta_n}} ,$$

and

$$k_o = k$$

$$\alpha^2_o = \alpha^2$$

$$\sin\theta_o = \sin\theta$$

$$p_n = \sqrt{1-k^2_n/\alpha^2_n}$$

$$G_n = \Pi(\theta_n, \alpha^2_n; 0) .$$

Then if N is a value of n such that $k_N \cong 0$,

$$F(\theta,k) = R_N\theta_N$$

$$E(\theta,k) = Q_N + [T_N - S_N]\,\theta_N$$

$$\Pi(\theta,\alpha^2,k) = X_N\theta_N + Y_N G_N - Z_N$$

5. <u>Restrictions on the Use of the Formulae</u>

The formulae for F and E can be used for all values of k and θ with the exception of the trivial case $k = 1$. On the other hand, those for Π can not be universally applied unless p_n is real for each n. Evidentally, the initial value

$$p_0 = \sqrt{1 - k^2/\alpha^2}$$

will be real provided that either $\alpha^2 < 0$ or $\alpha^2 > k^2$, and it is easily shown that if these conditions are satisfied by the original values of k and α^2, it will always be satisfied by the subsequent values of these quantities. This leaves one remaining case, namely

$$0 < \alpha^2 \leq k^2$$

where the transformation will not work directly. Fortunately, there exists the following relationship:

5.1
$$\Pi(\theta,\alpha^2,k^2) + \Pi(\theta,k^2/\alpha^2,k^2) = F(\theta,k) + \frac{1}{\sqrt{(1-\alpha^2)(k^2/\alpha^2-1)}}\ln\sqrt{\left|\frac{1+u}{1-u}\right|},$$

where

$$u = \sqrt{\frac{(1-\alpha^2)(k^2/\alpha^2-1)}{1-k^2\sin^2\theta}}\,\tan\theta .$$

Also since $k^2/\alpha^2 > 1$, $\Pi(\theta,k^2/\alpha^2,k^2)$ can be calculated by the previously described method, and $\Pi(\theta,\alpha^2,k^2)$ found from the above formula.

6. <u>Some Aids for Improving the Numerical Accuracy</u>

Numerous sources of inaccuracy may be inherent in the previously described formulae. These will be briefly discussed below.

a. Subtraction of large but nearly equal quantities.

This occurs both in the calculation of E and Π. In the formulae for E, T_n and S_n increase in magnitude but their difference remains finite; similarly, in the expression for Π, the quantity $Y_n G_n - Z_n$ approaches a finite limit although both $Y_n G_n$ and Z_n increase. The inaccuracy which might result due to loss of significant figures may be avoided by reformulating the expressions directly in terms of the differences.

b. <u>Inaccuracies in Computing θ_N from $\sin\theta_N$.</u>

Most computers find inverse trigonometric functions by means of an inverse tangent subroutine. This requires that both $\sin\theta_N$ and $\cos\theta_N$ be calculated. If $\sin\theta_N$ is close to zero or one, inaccurate values of $\cos\theta_N$ will result if the latter is computed from the relation:

$$\cos\theta_N = \sqrt{1-\sin^2\theta_N},$$

to avoid this difficulty, $\cos\theta_{n+1}$ should be computed directly from $\sin\theta_n$ and $\cos\theta_n$, since the initial of those quantities may be computed with reasonable precision, and the final results will accordingly be more accurate. A suitable formula for accomplishing this is the following:

$$\cos\theta_{n+1} = \frac{1+R_{n+1}\sin^2\theta_{n+1}}{\sqrt{1-R_{n+1}^2\sin^2\theta_{n+1}}}\cos\theta_n .$$

Unreasonably large numbers may also occur if the initial value of p is small, since this quantity appears in the denominator in Eq. 3.1. For this reason it may be preferable to avoid using the direct method when α^2 is near to k^2, and this may be accomplished by employing Eq. 3.1 only when $\alpha^2 < 0$ or $\alpha^2 > k$, and using the transformation to the parameter k^2/α^2 otherwise.

APPENDIX A

```
C THIS SUBRØUTINE CALCULATES THE  ELLIPTIC INTEGRALS,F(SSK,TH) AND E(SSK,TH)
C SSK=MØDULUS TH=AMPLITUDE E=ACCURACY
C IND=1 ØNLY F(SSK,TH)IS CØMPUTED , IND=2 F(SSK,TH) AND E(SSK,TH) ARE CØMPUTED
      SUBRØUTINE ELLIP (SSK,TH,IND,E,BK,EK)
      P1=1.5707963267948 75
      TH1=P1*TH/90.0
      IF(SSK-1.0)8,9,8
    9 IF(TH-90.0)12,13,12
   12 SN=SINF(TH1)
      A=(1.0+SN)/(1.0-SN)
      BK=0.5*LØGF(A)
      EK=SN
      GØ TØ 30
   13 GØ TØ (14,15),IND
   15 BK=9.9999999E+98
      EK=1.0
      GØ TØ 30
   14 PRINT 100,SSK
      BK=9.9999999E+98
      GØ TØ 30
    8 IF(TH-90.)5,6,5
    6 SN=1.0
      CN=0.0
      GØ TØ 7
    5 SN=SINF(TH1)
      CN=CØSF(TH1)
    7 SK=SSK
      R=1.0
      T=1.0
      Q=0.0
      S=1.0
      D=SQRTF(1.-SK*SK*SN*SN)
    3 SK2=SK*SK
      SKP=SQRTF(1.0-SK2)
      GØ TØ (11,10),IND
   10 Q=Q+(T*SN*CN*SK2/(D+1.0))
   11 SK=(1.-SKP)/(1.+SKP)
      X=(1.0+SKP)/(1.0+D)
      SN=X*SN
      D=SQRTF(1.-SK*SK*SN*SN)
      Z=(1.+SK*SN*SN)/D
      CN=Z*CN
      S=(S+S-SK2*T)/(1.0+SKP)
      T=(1.0+SKP)*T
      R=(1.+SK)*R
      IF(SK2-E)4,4,3
    4 IF(SN-CN)22,22,24
22    P=ATANF(SN/CN)
      GØ TØ 23
24    P=P1-ATANF(CN/SN)
   23 GØ TØ (25,26),IND
   26 EK=P*S+Q
   25 BK=P*R
   30 RETURN
100   FØRMAT(2HK( F4.1,16H) IS NØT DEFINED)
      END
```


4.6.9

```
C  FØRTRAN II PRØGRAM THAT CALCULATES ELLIPTIC INTEGRALS ØF THE
C  THIRD KIND.   INPUT:  SK=MØDULUS, Ø=AMPLITUDE IN RADIANS
C  AND   PARAMETER (AL)=ALPHA SQUARED.
```

C PI= $\displaystyle\int_0^{\phi} \frac{d\theta}{(1-\alpha^2 \sin^2\theta)\sqrt{1-k^2 \sin^2\theta}}$

```
1       ACCEPT 100,SK,Ø,AL,E
        P1=1.57079632679487 5
        ALN=AL
        SKN=SQRTF(SK)
        SN=SINF(Ø)
        CN=CØSF(Ø)
        R=1.0
        S=1.0
        T=0.0
        Q=0.0
        TS1=AL-SKN
        TS2=ALN-1.0
        TS3=SN-CN
        IF(AL)35,40,3
3       IF(TS1)4,4,35
4       ALN=SK/AL
        TS2=ALN-1.0
        T3=1.-ALN
        T4=1.-SK/ALN
        IF(TS2)43,45,44
43      U=SQRTF((T3*T4)/(1.-SK*SN*SN))
        GØ TØ 45
44      U=SQRTF(((ALN-1.)*(1.-(SK/ALN)))/(1.-SK*SN*SN))
45      IF(TS2)9,14,15
9       C=SQRTF(T3)
        IF(TS3)10,10,11
10      T1=ATANF(U*(SN/CN))
        T2=ATANF(C*(SN/CN))
13      CØRR=(T1-T2)/SQRTF(T3*T4)
        GØ TØ 5
11      T1=P1-ATANF(CN/(U*SN))
        T2=P1-ATANF(CN/(C*SN))
        GØ TØ 13
14      T1=SQRTF(1.-SK)
        T2=SQRTF(1.-SK*SN*SN)
        CØRR=(-SK*SN*CN)/((T1*T2)*(T1+T2))
        GØ TØ 5
15      T1=(U*SN+CN)/(SQRTF(ALN-1.0)*SN+CN)
        T2=(1.-SK*SN*SN)/(1.-(SK/ALN)*(SN*SN))
        CØRR=(.5*LØGF(T1*T1*T2))/(SQRTF((ALN-1.)*(1.-SK/ALN)))
5       P2=SQRTF(1.-(SKN*SKN)/ALN)
2       SK2=SKN*SKN
        P=P2
        SKP=SQRTF(1.-SK2)
        AL1=ALN
        IF(TS3)16,16,17
16      SAL2=SN/CN
        GØ TØ 18
```

```
17      SAL2=CN/SN
18      SN=((1.+SKP)*SN)/(1.+SQRTF(1.-SK2*SN*SN))
        SKN=(1.-SKP)/(1.+SKP)
        TD=SQRTF(1.-SKN*SKN*SN*SN)
        CN=((1.+SKN*SN*SN)/TD)*CN
        R=R*(1.+SKN)
        P2=SQRTF(1.-SKN*((1.-P )/(1.+P )))
        ALN=ALN*((1.+P )/(1.+SKP))**2
        IF(AL1-1.0)19,22,23
19      T1=SQRTF(1.-AL1)
        IF(TS3)20,20,21
20      HN=(2.0/T1)*(ATANF(T1*SAL2)-ATANF(SQRTF(1.-ALN)*(SN/CN)))
        GØ TØ 24
21      C=SQRTF(1.-ALN)
        HN=(2./T1)*((P1-ATANF(SAL2/T1))-(P1-ATANF(CN/(C*SN))))
        GØ TØ 24
22      T2=SKN*SKN
        T3=SQRTF(1.-T2)
        T4=SQRTF(1.-T2*SN*SN)
        HN=(-2.*T2*(1.+SKN)*SN*CN)/((T3*T4)*(T3+T4))
        GØ TØ 24
23      T1=(CN+SQRTF(ALN-1.)*SN)**2*(1.-((SKN*SKN)/ALN)*(SN*SN))
        T2=(CN*SQRTF(1.-SKN*SKN*SN*SN)+P2*SQRTF(ALN-1.)*SN)**2
        HN=LØGF(T2/T1)/SQRTF(AL1-1.)
24      T1=1.+SKN
        T=T+(HN/P )*S
        Q=T1*(Q+((P -1.)/P )*S)
        S=((2.*T1)/P )*S
        IF(SKN-E)25,25,2
25      IF(SN-CN)26,26,27
26      ØN=ATANF(SN/CN)
        GØ TØ 28
27      ØN=P1-ATANF(CN/SN)
28      PIBAR=ØN*Q-T
        IF(AL)31,40,29
29      IF(TS1)30,30,31
30      PI=ØN*R-PIBAR+CØRR
        GØ TØ 32
31      PI=PIBAR+(GØ/RHØ)
32      PRINT 101,PI
        GØ TØ 1
35      RHØ=SQRTF(1.-SK/AL)
        IF(AL-1.0)36,39,42
36      C=SQRTF(1.-AL)
        IF(SN-CN)37,37,38
37      GØ=ATANF(C*(SN/CN))/C
        GØ TØ 5
38      GØ=(P1-ATANF(CN/(C*SN)))/C
        GØ TØ 5
39      IF(CN-E)40,40,41
40      PRINT 103,ALN,Ø
        GØ TØ 1
41      GØ=SN/CN
        GØ TØ 5
42      C=SQRTF(AL-1.)
        T1=(CN+C*SN)/(CN-C*SN)
        T1=ABSF(T1)
        GØ=(.5*LØGF(T1))/C
        GØ TØ 5
100     FØRMAT(E17.10)
103     FØRMAT(30HPRØBLEM NØT DEFINED FØR AHPHA=F16.10 10HAND THETA=F16.9)
101     FØRMAT(3HPI= F15.10)
        END
```


4.6.11

THE USE OF DISCRETE CONVOLUTION IN THE ANALYSIS OF DIFFRACTION PATTERNS

J. S. LIEBMAN

P. M. JOHNSON

GENERAL ELECTRIC COMPANY

ADVANCED ELECTRONICS CENTER

ITHACA, NEW YORK

This paper is about a program for convolution of two complex discrete functions. We will discuss uses of complex convolution, describe the experimental problem which led us to develop the program, and then will present the technique itself.

The convolution process is used in the analysis of several engineering problems. In electrical signal processing by networks, the circuit output equals the complex convolution of the input time function with the circuit impulse response function. In schlieren optics, light passing through transparent media undergoes phase diffraction instead of amplitude diffraction, and may be analyzed by complex convolution.

Our experimental system, a schlieren system, is a special case of a coherent optical processing system, and it uses the phenomenon of phase diffraction for a rather unique purpose.

Figure 1 is a diagram of the experimental setup. If monochromatic light is passed through a grooved transparent film and is partially blocked in the diffraction plane, it will then project a pattern on the image screen. The image intensity varies (in a manner depending upon the type of operation at the diffraction plane) with the groove depth and slope. The variations of light intensity on the screen present the information that is stored on the film. If no blocking of the light takes place in the diffraction plane, then there will be no variation of light intensity on the image screen. We desire to find out what kind of blocking at the diffraction plane produces the best results.

It happens that Fourier Transform relations exist between the object, diffraction, and image planes. It also happens that the light amplitude at the object plane can be represented as a multiple product. If we use convolution theorem that the Fourier Transform of a product is equal to the convolution of the Fourier transforms of the individual functions, and if we know the individual transforms of the factors in the product, we can easily compute the diffraction pattern. Hence, a major part of our computer simulation of the optical display system was the development of an efficient technique for the convolution of two or more complex discrete functions.

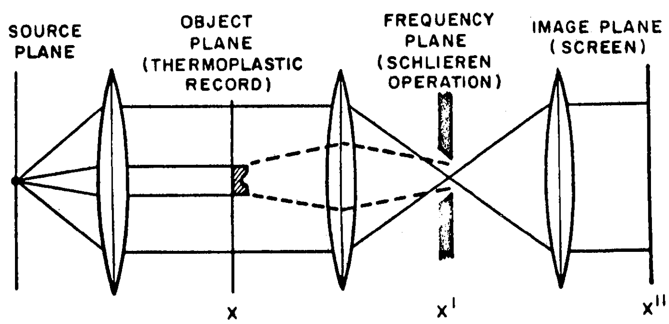


FIGURE 1

Because the functions we convolved have real and imaginary parts, the **following** series of equations develop the model used for our complex convolution.

If F_1 & F_2 are complex functions of x, the distance from the center of the diffraction plane.

$$F_1(x) = a(x)\,e^{-i\alpha(x)}$$

$$F_2(x) = b(x)\,e^{-i\beta(x)}$$

Let

$$F_{12}(x) = F_1(x) * F_2(x) = a(x)e^{-i\alpha(x)} * b(x)e^{-i\beta(x)}$$

Then

$$\text{Real}\,[F_{12}(x)] = \sum_{x=-\ell}^{x=+\ell} a_\ell\, b_{x-\ell} \cos(\alpha_\ell + \beta_{x-\ell})$$

$$\text{Imag.}\,[F_{12}(x)] = \sum_{x=-\ell}^{x=+\ell} a_\ell\, b_{x-\ell} \sin(\alpha_\ell + \beta_{x-\ell})$$

Since

$$\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$$

$$\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$$

Then

$$R[F_{12}(x)] = \sum_{x=-\ell}^{x=+\ell} a_\ell b_{x-\ell} \cos\alpha_\ell \cos\beta_{x-\ell} - \sum_{x=-\ell}^{x=+\ell} a_\ell b_{x-\ell} \sin\alpha_\ell \sin\beta_{x-\ell}$$

$$\mathcal{I}[F_{12}(x)] = \sum_{x=-\ell}^{x=+\ell} a_\ell b_{x-\ell} \sin\alpha_\ell \cos\beta_{x-\ell} + \sum_{x=-\ell}^{x=+\ell} a_\ell b_{x-\ell} \cos\alpha_\ell \sin\beta_{x-\ell}$$

Thus, complex convolution may be expressed in terms of four real convolutions. The following operations represent part of our simulation and, for various reasons, they were carried out by several computer programs instead of one. At the end of this paper is a Fortran program for the simple convolution performed in part b.

(a) Prepare four tables.

(1) x vs $a\cos\alpha$
(2) x vs $a\sin\alpha$
(3) x vs $b\cos\beta$
(4) x vs $b\sin\beta$

(b) Convolve

(1)*(3) $= R_1(x)$
(2)*(4) $= R_2(x)$
(1)*(4) $= I_1(x)$
(2)*(3) $= I_2(x)$

(c) Combine into real and imaginary parts.

$$R(x) = R_1(x) - R_2(x)$$
$$\mathcal{I}(x) = I_1(x) + I_2(x)$$

(d) The amplitude of the m^{th} point of the diffraction pattern is

$$A(m) = \sqrt{[R(m)]^2 + [\mathcal{I}(m)]^2}$$

(e) The phase of the m^{th} point of the diffraction pattern is

$$\alpha(m) = \tan^{-1}\left[\frac{\mathcal{I}(m)}{R(m)}\right]$$

In addition to possible uses in your own computing group, this technique offers interesting opportunities for university physics and electrical engineering labs. More and more universities and colleges are teaching their students how to program and are giving their students machine time. Complicated diffraction patterns could be predicted before being observed in the physics labs, as well as schlieren optics. Computer calculations could replace the electrical engineering students' frequent use of the graphical technique for complex convolution.

In closing, we present the Fortran program for the basic convolution of two discrete functions that is the fundamental unit of our complex convolution. It is offered as an example of a technique to be used when the variable (m in our case) can be restricted to integer values. Then each F(m) can be stored in location m for efficiency of storage.

```
C       CONVOLUTION OF TWO DISCRETE FUNCTIONS
C       J.S.LIEBMAN,GENERAL ELECTRIC,ITHACA,N.Y.
        DIMENSION M1(201),M2(201),A1(201),A2(201),A1A2(402)
1       READ 95
        PUNCH 95
        READ 90,N1
        DO 10 I=1,N1
        READ 90, M1(I),A1(I)
*10     M1(I)=M1(I)+100
        READ 90,N2
        DO 20 I=1,N2
        READ 90, M2(I),A2(I)
*20     M2(I)=M2(I)+100
        DO 30  K=1,402
30      A1A2(K)=0.
        DO 40 I=1,201
        DO 40 J=1,201
        K=M1(I)+M2(J)
40      A1A2(K)=A1A2(K)+A1(I)*A2(J)
        DO 60 K=1,402
        IF (A1A2(K)) 50,60,50
50      J=K-200
        PUNCH 90,J,A1A2(K)
60      CONTINUE
        PAUSE
        GO TO 1
90      FORMAT (14X,I6,14X,F6.4)
95      FORMAT (49H                                                 )
        END
```

* THESE STEPS WERE MERELY MANIPULATIONS TO AVOID NEGATIVE INDICES.

A COMPUTER APPROACH TO THE ANALYSIS OF GAMMA SPECTRA
BY THE METHOD OF CHARACTERISTIC ENERGY REGIONS

by

WENDELL E. CARR[1]

1/ Systems Analyst Programmer, Data Processing and Computations Section, Technical Operations Branch, Division of Radiological Health, Bureau of State Services, Public Health Service, U. S. Department of Health, Education, and Welfare, Radiological Health Laboratory, Rockville, Maryland.

4.8.D

A COMPUTER APPROACH TO THE ANALYSIS OF GAMMA SPECTRA BY THE METHOD OF CHARACTERISTIC ENERGY REGIONS

by

WENDELL E. CARR

The simultaneous equations method[1/] for gamma spectrum analysis utilizing characteristic energy regions has been programmed for the IBM 1620 MONITOR I System -- a combined operating and programming system. The method uses an IBM 1620 Computer (with a 40,000 digit memory, automatic divide, and indirect addressing), a 1622 Card Read-Punch Unit, a 1621 Paper Tape Reader, and two 1311 disk storage drives.

The method has been mainly applied to environmental samples, such as milk, food, air filters, etc. The gamma spectrum is obtained using a sodium-iodide crystal and a multi-channel gamma spectrometer. A spectrometer is calibrated so that counts from the sample are stored in channels on the basis of their energy level; counts of equivalent energy level are recorded in the same channel. The principles of multi-channel gamma spectroscopy have been fully described by many authors; one good summary is that of Crouthamel.[2/] The paper tape output from the spectrometer includes the channel numbers (energy levels) and the total counts (radioactivity) in each channel. The channels and counts comprise the gamma spectrum.

The essential problem in gamma spectrum analysis is to determine which counts are due to which nuclides and ultimately the content of each

nuclide present in one given sample. The counts in any one channel are usually due to several nuclides. For example, channels 32 - 40 (.01 MEV per channel) may be the characteristic region for I-131, but other nuclides also contribute to the counts in this region. How many counts in the I-131 region are due only to I-131? One method for solving this problem of spectral interference among the nuclides is the simultaneous equations method.

Description of Method

Standard solutions of known activity containing only one nuclide are prepared and counted for each of the nuclides of interest. In the present treatment it is assumed that the identity of the pertinent nuclides is known, and furthermore, that the gamma spectrum of each standard is obtained under the same conditions as the sample spectrum, i.e., same crystal, same geometry, etc.

The equations are set up by examining the standards. Since each standard contains only one nuclide, there is no interference problem (except background, which is subtracted out). For example, a Cs-137 standard has no counts from other nuclides in the Cs-137 region (channels 60 - 72), so that the counts there are all due to Cs-137. However, the Cs-137 causes counts in other channels, and the interference in other nuclide regions can be calculated as a fraction of the counts in channels 60 - 72. Suppose there are 100 counts in the Cs-137 region and 30 counts in the I-131 region; this means that in any sample, whatever the number of counts in the Cs-137 region that are due to Cs-137, .300 of that number will be in the I-131

region as a result of the Cs-137. By examining other standards, the fractional contribution to the I-131 region from each of the other nuclides can be determined.

In the system of equations there is one equation for each nuclide; this gives the counts in the nuclide region as a sum of the fractional contributions from the other nuclides. The I-131 equation might be:

$$I_R = 1.000\ I + .300\ Cs + - - -$$

where I_R is the counts in the I-131 region, I is the counts in the I-131 region due only to I-131, Cs is the counts in the Cs-137 region due only to Cs-137, - - -. For any sample, the counts in each region (e.g., I_R) can be found by simply adding the counts in each contributing channel. This leaves I, Cs, - - - as the unknowns; the solution to the system of equations will yield these and hence will be a solution to the gamma spectrum analysis problem.

For N nuclides, the system of N equations could be written in matrix notation[3/] as:

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,N} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,N} \\ \vdots & & & \vdots \\ A_{N,1} & A_{N,2} & \cdots & A_{N,N} \end{bmatrix} \cdot \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{bmatrix}$$

or $X = A\ Z$

In practice, counts per minute rather than counts are used so that

X_i is the gross* counts per minute in region i

Z_i is the true* counts per minute of nuclide i

A_{ij} is the fraction of Z_j which will be found in region i

The equations are usually solved explicitly for $Z_1, Z_2, ---, Z_N$ by multiplying both sides of the matrix equation by the inverse of A:

$$A^{-1}X = Z$$

<u>The Programs</u>

Three programs execute the method. The first program -- the <u>Set-Up Program</u> -- sets up the simultaneous equations describing the spectral interference among the nuclides (any number up to and including 18), solves the equations by the inverted matrix method, and calculates the factors needed for each nuclide to convert net counts per minute to picocuries per unit. Input to the <u>Set-Up Program</u> includes:

1. A standard spectrum with indicative information such as the date counted, counting duration, etc., for each nuclide.
2. The channels to be used for each nuclide region.
3. The volume or weight of the samples which will be analyzed with the inverted matrix. This will be used to calculate the conversion factors.

Output from the <u>Set-Up Program</u> includes:

1. The original matrix of the coefficients in the simultaneous equations.
2. The inverted matrix which is the solution to the simultaneous equations.
3. Conversion factors for each nuclide.

The computer group maintains a library of the output from the <u>Set-Up Program</u> for all the stations which it serves. The second program -- the

---

* "Gross" refers to the total counts per minute in a nuclide energy region (with the background subtracted out). "True" refers to that portion of of the gross counts per minute in the nuclide energy region which are due only to the particular nuclide.

Store Program -- writes this library on magnetic disk auxiliary storage: a disk pack on an IBM 1311 Disk Storage Drive. The Write Addresses Routine, part of the MONITOR Disk Utility Program, protects the library with read-only flags.

The third program -- the Analysis Program -- analyzes the gamma spectrum, obtained by counting an environmental sample, by using the library stored on a disk pack. The station number, matrix code, and analyzer number in the sample indicative determine a unique inverted matrix and set of conversion factors with which to analyze the sample. The matrix code specifies the particular combination of nuclides as well as the geometry. For example, for a specific geometry there could be one code for milk samples (4 nuclides) and another for food samples (8 nuclides or more). The output from this program is the concentration of each nuclide in the sample together with the associated error in this determination.

The concentrations are corrected for decay by a table-look-up method. The table of correction factors is included in the library.

An error is calculated for each concentration at the 95% confidence level based on the counting errors and the propagation of these errors due to subsequent calculations. Since the counts have a Poisson distribution which can be approximated by a normal distribution, there is a 0.95 probability that the observed value plus or minus two standard deviations will include the true value. Therefore, the counting errors are calculated as:

$$2S = 2\sqrt{\frac{N_{s+b}}{t_{s+b}} + \frac{N_b}{t_b}}$$, where

S is the standard deviation

N_{s+b} is the counting rate of the sample plus background

t_{s+b} is the time counted for the sample plus background

N_b is the counting rate of the background

t_b is the time counted for the background [4]

The propagation of the counting errors is calculated by applying the general propagation of error formula:

$$E_Z = \sqrt{\left(E_X \frac{\partial Z}{\partial X}\right)^2 + \left(E_Y \frac{\partial Z}{\partial Y}\right)^2 + - - - },$$

where $Z = f(X, Y, - - -)$; E_X is the error in X, E_Y is the error in Y, - - -; E_Z is the error in Z and is the same type of error as E_X, E_Y, - - -.[5] For example, if E_X, E_Y, - - - are 2S errors, then E_Z will also be a 2S error. The general form of the equation giving the true counts per minute of nuclide Z in terms of the gross counts per minute in regions X, Y, --- is

$$Z = a\,X + b\,Y + - - - ,$$

where a, b, - - - are considered to be constant coefficients. For this special case the general propagation of error formula reduces to

$$E_Z = \sqrt{\left(a\,E_X\right)^2 + \left(b\,E_Y\right)^2 + - - - },$$

where E_X, E_Y, - - - are the 2S counting errors. Multiplying Z by a factor k to convert from counts per minute to picocuries per unit multiplies the error by k also, so that the final error is

$$\begin{aligned} E_{kZ} &= k\,E_Z \\ &= k\sqrt{\left(a\,E_X\right)^2 + \left(b\,E_Y\right)^2 + - - - } \end{aligned}$$

Output from the Analysis Program is punched on cards and consists of the activity of each nuclide and the associated error in picocuries per unit of volume or weight. These cards are listed on the IBM 407.

Spectrum Verification

A special routine in both the Set-Up and Analysis Programs inspects the gamma spectrum and indicative for invalid characters in the paper tape by doing a simple multiplication with each digit after the tape has been read into memory. An invalid character will cause a CHECK STOP, and the operator must manually branch to a routine which will dump on the typewriter:

1. Indicative for the spectrum
2. Number of the digit which is invalid, e.g., 31st, 50th, etc.
3. The invalid character preceded by other digits.

Another routine in both programs affirms that the N^{th} channel in the spectrum is numbered correctly, since if a digit has been dropped or added, calculations will be meaningless.

Language

All these programs are written in SPS II-D, although the Set-Up Program uses a subprogram written in FORTRAN II-D to invert the matrix.

Application

The system of programs described here is being used by the Division of Radiological Health, United States Public Health Service, to analyze environmental samples. Samples are collected by State and local health department personnel under the nationwide networks of the Public Health Service and are shipped to Division of Radiological Health regional laboratories for counting. Gamma spectral data are obtained at the laboratories

and sent to the Data Processing and Computations Section for computer analysis. The resultant activity concentrations are used by other units in the PHS to assess levels of radiation in specific media and in the environment.

REFERENCES

1/ G. Richard Hagee, Gerald J. Karches, and Abraham S. Goldin, "Determination of Iodine-131, Cesium-137, and Barium-140 in Milk by Gamma Spectroscopy," Talanta, 5: 36-43 (1960)

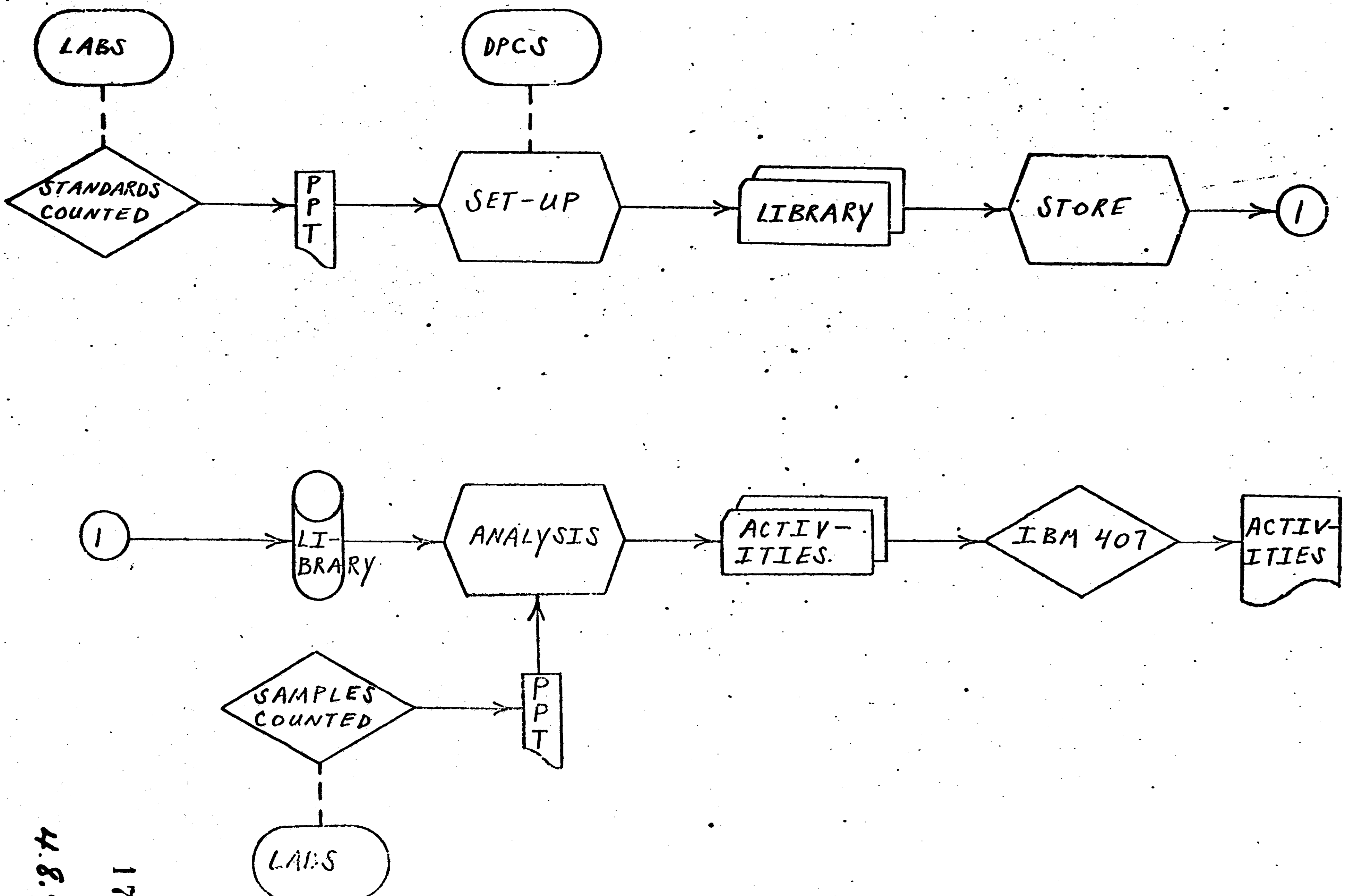
2/ C. E. Crouthamel, Applied Gamma Ray Spectrometry, Pergamon Press, New York, 1960

3/ Daniel T. Finkbeiner, II, Introduction to Matrices and Linear Transformations, W. H. Freeman and Company, San Francisco, London, 1960

4/ Alan A. Jarrett, "Statistical Methods Used in the Measurement of Radioactivity With Some Useful Graphs and Nomographs," AECU-262, June 17, 1946

5/ Lyman G. Parratt, Probability and Experimental Errors in Science, John Wiley and Sons, New York, 1961

SYSTEM AS USED BY PUBLIC HEALTH SERVICE



SUN OIL COMPANY
RESEARCH AND ENGINEERING
ENGINEERING DIVISION

"ZIP"

A MULTIPLE LINE KEYWORD IN CONTEXT PROGRAM

by

Thomas J. Scott

For presentation at the Spring, 1964 1620 Users Group Meeting

May 6 - 8
Washington, D. C.

A MULTIPLE LINE KEYWORD IN CONTEXT PROGRAM

The purpose of this program is to perform a keyword (or keyphrase) in context analysis on a multiple line (card) abstract, paragraph, or magazine article. The program will accept this keypunched information and:

1) Rearrange it in core according to an output format specified by the user.

2) Cross-reference it by keywords or phrases chosen by the user. These keywords need not appear in the abstract itself, but may be added by means of a keyword, or asterisk card, that immediately follows the abstract.

3) Produce the following output, once for each keyword selected.

   a) A keyword or phrase title card.
   b) The original abstract rearranged to an output format selected by the user.
   c) A reference card, containing information pertaining to the abstract's source, location, etc. This information will come from a similar input card.
   d) A blank card, to separate adjacent, cross-referenced abstracts.

Four identification fields appearing on the right hand side of every output card enable the user to perform an alphameric merge of the cards with the output from other runs.

A sample input-output is attached on pages 2 and 3.

Further information may be obtained by requesting a write-up of the program from the 1620 Users Group. The program number is 10.3.032.

Thomas J. Scott

TJS:hsp

- SAMPLE INPUT -

```
REVISEDZSPECIFICATIONS BEING PREPARED BY THEZAMERICAN CONCRETE INSTITUTE00924301
 WILL BRING THE DESIGN OFZTALL CONCRETE CHIMNEYSOINTO AGREEMENT WITH THE00924302
LATEST PRINCIPLES FORZSEISMIC AND WIND FORCESO,AND WILL INCLUDE GUIDANCE00924303
FOR PROTECTION AGAINSTZCORROSION .ZCIRCULAR STIFFENERSOSHOULD BE USED WH00924304
EN NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL ABOVE THE MAXIMUM T00924305
HAT WILL OCCUR.                                                         00924306
*VIBRATIONZSTACKSZDAMPING CHARACTERISTICS                               00924307
    CIVIL ENGINEERING, FEB. 1964, PP.32-35                            6400924308
```

- SORTED OUTPUT -

```
 AMERICAN CONCRETE INSTITUTE                         AMERICAN CONCRETE IN092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE         AMERICAN CONCRETE IN092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN    AMERICAN CONCRETE IN092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE    AMERICAN CONCRETE IN092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND   AMERICAN CONCRETE IN092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST         AMERICAN CONCRETE IN092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHENAMERICAN CONCRETE IN092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL    AMERICAN CONCRETE IN092430864
ABOVE THE MAXIMUM THAT WILL OCCUR                    AMERICAN CONCRETE IN092430964
      CIVIL ENGINEERING, FEB. 1964, PP. 32-35        AMERICAN CONCRETE IN092431064
                                                     AMERICAN CONCRETE IN092431164
 CIRCULAR STIFFENERS                                 CIRCULAR STIFFENERS 092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE         CIRCULAR STIFFENERS 092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN    CIRCULAR STIFFENERS 092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE    CIRCULAR STIFFENERS 092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND   CIRCULAR STIFFENERS 092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST         CIRCULAR STIFFENERS 092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHENCIRCULAR STIFFENERS 092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL    CIRCULAR STIFFENERS 092430864
ABOVE THE MAXIMUM THAT WILL OCCUR                    CIRCULAR STIFFENERS 092430964
      CIVIL ENGINEERING, FEB. 1964, PP. 32-35        CIRCULAR STIFFENERS 092431064
                                                     CIRCULAR STIFFENERS 092431164
 CORROSION                                           CORROSION           092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE         CORROSION           092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN    CORROSION           092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE    CORROSION           092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND   CORROSION           092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST         CORROSION           092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHENCORROSION           092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL    CORROSION           092430864
ABOVE THE MAXIMUM THAT WILL OCCUR                    CORROSION           092430964
      CIVIL ENGINEERING, FEB. 1964, PP. 32-35        CORROSION           092431064
                                                     CORROSION           092431164
 DAMPING CHARACTERISTICS                             DAMPING CHARACTERIST092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE         DAMPING CHARACTERIST092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN    DAMPING CHARACTERIST092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE    DAMPING CHARACTERIST092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND   DAMPING CHARACTERIST092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST         DAMPING CHARACTERIST092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHENDAMPING CHARACTERIST092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL    DAMPING CHARACTERIST092430864
ABOVE THE MAXIMUM THAT WILL OCCUR                    DAMPING CHARACTERIST092430964
      CIVIL ENGINEERING, FEB. 1964, PP. 32-35        DAMPING CHARACTERIST092431064
                                                     DAMPING CHARACTERIST092431164
```

SEISMIC AND WIND FORCES	SEISMIC AND WIND FOR	092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE	SEISMIC AND WIND FOR	092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN	SEISMIC AND WIND FOR	092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE	SEISMIC AND WIND FOR	092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND	SEISMIC AND WIND FOR	092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST	SEISMIC AND WIND FOR	092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHEN	SEISMIC AND WIND FOR	092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL	SEISMIC AND WIND FOR	092430864
ABOVE THE MAXIMUM THAT WILL OCCUR	SEISMIC AND WIND FOR	092430964
CIVIL ENGINEERING, FEB. 1964, PP. 32-35	SEISMIC AND WIND FOR	092431064
	SEISMIC AND WIND FOR	092431164
SPECIFICATIONS	SPECIFICATIONS	092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE	SPECIFICATIONS	092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN	SPECIFICATIONS	092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE	SPECIFICATIONS	092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND	SPECIFICATIONS	092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST	SPECIFICATIONS	092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHEN	SPECIFICATIONS	092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL	SPECIFICATIONS	092430864
ABOVE THE MAXIMUM THAT WILL OCCUR	SPECIFICATIONS	092430964
CIVIL ENGINEERING, FEB. 1964, PP. 32-35	SPECIFICATIONS	092431064
	SPECIFICATIONS	092431164
STACKS	STACKS	092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE	STACKS	092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN	STACKS	092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE	STACKS	092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND	STACKS	092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST	STACKS	092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHEN	STACKS	092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL	STACKS	092430864
ABOVE THE MAXIMUM THAT WILL OCCUR	STACKS	092430964
CIVIL ENGINEERING, FEB. 1964, PP. 32-35	STACKS	092431064
	STACKS	092431164
TALL CONCRETE CHIMNEYS	TALL CONCRETE CHIMNE	092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE	TALL CONCRETE CHIMNE	092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN	TALL CONCRETE CHIMNE	092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE	TALL CONCRETE CHIMNE	092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND	TALL CONCRETE CHIMNE	092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST	TALL CONCRETE CHIMNE	092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHEN	TALL CONCRETE CHIMNE	092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL	TALL CONCRETE CHIMNE	092430864
ABOVE THE MAXIMUM THAT WILL OCCUR	TALL CONCRETE CHIMNE	092430964
CIVIL ENGINEERING, FEB. 1964, PP. 32-35	TALL CONCRETE CHIMNE	092431064
	TALL CONCRETE CHIMNE	092431164
VIBRATION	VIBRATION	092430164
REVISED SPECIFICATIONS BEING PREPARED BY THE	VIBRATION	092430264
AMERICAN CONCRETE INSTITUTE WILL BRING THE DESIGN	VIBRATION	092430364
OF TALL CONCRETE CHIMNEYS INTO AGREEMENT WITH THE	VIBRATION	092430464
LATEST PRINCIPLES FOR SEISMIC AND WIND FORCES ,AND	VIBRATION	092430564
WILL INCLUDE GUIDANCE FOR PROTECTION AGAINST	VIBRATION	092430664
CORROSION . CIRCULAR STIFFENERS SHOULD BE USED WHEN	VIBRATION	092430764
NECESSARY TO ASSURE A RESONANT WIND VELOCITY WELL	VIBRATION	092430864
ABOVE THE MAXIMUM THAT WILL OCCUR	VIBRATION	092430964
CIVIL ENGINEERING, FEB. 1964, PP. 32-35	VIBRATION	092431064
	VIBRATION	092431164

1620 USERS GROUP

GENERAL DATA PROCESSING

"An Electric Load Statistics System for An Electric Utility"

Prepared by
Edward H. Parker
Engineering Computer Applications

Baltimore Gas and Electric Company
Baltimore, Maryland 21203

May 8, 1964

183 4.10.D

1620 USERS GROUP

An Electric Load Statistics System for An Electric Utility

The routine hourly records of generation, interchange with other utilities, supply and interchange with the Bethlehem Steel Company and supply to the Pennsylvania Railroad were mechanized through the use of the 1620 in 1961. The work included in the operation are arithmetical calculations necessary to provide the various combinations of basic data to obtain the several electrical system loads and their components. The data used are obtained from operators' log sheets and by telephone. The output is typewritten on 8-1/2 inch by 11 inch pre-printed forms which give hourly values and daily totals. Ultimately, monthly forms are produced which show the daily and monthly totals and also the daily one-hour peaks for selected items. The resulting information is used in the preparation of reports, studies and forecasts. The typewritten sheets are the counterpart of some of the records which have been kept for many years.

Twenty-four individual programs which may be used singly or in groups comprise a series of twelve operations from the time key-punchers provide the data packages each day until about two months later when the data are put into record files. Five types of operations are used in the process. They are checking, storing of data in core, production of off-line printing decks, production of data to be filed and combination of filed data with programmed identification for off-line printing.

The system is prepared for two different printers. One is the 870 system which has the appearance of a key-punch connected to a typewriter. The other is a 407 printer. The 870 is used when one day's work is processed. The 407 is used when a large quantity, which would overload the 870, is to be printed.

The various data are obtained from a variety of meters read by station or machine operators and by automatic recording equipment in many widely dispersed installations in the three states of the Company's interconnection area. The majority of the data are sent from the operating location into the Electric Engineers Department by regular Company mail. The data assembled onto a particular form for permanent records are not all interrelated. In carrying out a manual operation, data forms can be pieced together gradually throughout the day as the required data are available. In the meantime, the data which are complete can be disseminated. The programs written for the machine operation do not allow for variations of availability of data. Such flexibility was rejected after consideration of the various factors involved. All data are key-punched from the various log sheets which have been organized according to priority groups related to the sequence of need for the output forms to be produced. The total group of data cards required to produce selected sets of output forms have been labeled "packages".

Operation of the 1620 phase of the daily operation is scheduled to start at 9:15 a.m. A data package is first checked for omissions, then for accuracy by means of two editing programs. The first check is accomplished by means of a "hash total" of the identification numbers punched in the cards. Corrections of data are accomplished during the second edit phase. When an error is detected, the operation halts after an indication is typed on the console typewriter. Checking continues while errors are corrected by the key-punch operator. There is no need to restart the entire package during the checking phase. Part of this check requires that the midnight readings of the previous day - that is to say the low reading for the current day - be read into appropriate storage locations in order to check the twenty-four hour total. This reading, subtracted from its counterpart in the current day's data, should obviously be equal to the

total for the day obtained by adding the individual hourly values. When this detail is accepted by the program, the midnight-hour reading for the current day is stored in a special location so that all such readings may be punched to produce a new meter summary deck of cards for use as input data the following day.

There are five different processes by which the data are handled in the checking operation. The simplest is the acceptance of a twenty-four hour total. The next involves a meter reading with a twenty-four hour total as suggested before. Another is a series of hourly readings and a twenty-four hour total. Then naturally, hourly readings, twenty-four hour total and meter statements make a double check possible. Finally - one not quite so obvious - sets of meter statements and their twenty-four hour totals, without individual hourly values, are compared with the twenty-four hour sum of the combined hourly values of all the components. We call these editing routines. I have explained them because they are the basis of the adaptability of the system to electrical load measuring and recording methods.

When all data, approximately 400 key-punched cards, have been accepted by the program, a process requiring slightly over two minutes for correct data, all of the hourly data and separate twenty-four hour totals will have been stored in the core at specific addresses assigned to the particular item. The operator is then instructed to insert the production program.

These programs are so arranged that they select data from core according to specific requirements of a particular typewritten output sheet or form. These production programs are the sum total of all the manual steps previously carried out. They involve more logic than arithmetic. Since some data are positive and others negative and records are kept in terms of sales, purchases,

interchange, high and low transmission voltage, steam generation, losses, water power or cycles per second, many variations in the form make-up are possible. Network switching and equipment usage also insert variations in the programs. When the programs for a given package of data have been completed the machine is cleared and the next set of data can be edited. This may be the same work for the next day or, if available, the second package of the same day.

One might assume that the considerations mentioned in regard to the data and forms would be enough to make a system work. But, as you might have suspected, such is not the case. In this particular operation, it was necessary to allow for backward-turning meters, back feed through normally forward-turning meters and non-generation operation of generators or what is known as condenser operation. In a few cases, power might be sent through a second meter when a tie line was closed. Such values must be "marked" and properly handled. As you all know, any meter has a small number of dials so that when the reading reaches 999 it must "turn over". The next reading is zero not 1000. Normal and negative meter turnover are programmed in the edit phase.

The punched card output of the 1622 is of three types: complete printing decks for immediate use, records to be stored for use at the end of the month as input for the end of the month operation and "converted" data obtained from that assembled on the first cards for printing. Conversion involves selection and identification by means of a special code for system components which are part of the permanent record.

The output cards for preparing the printed sheets are saved for a period of two months. Some forms cannot be prepared as soon as the majority are prepared because data for these are from a different source. When the late data are in, combinations of more specific items can be prepared.

Operation is only approximately the same each day. The major difference occurring on Monday when three days are processed. Certain data are allowed to accumulate for several days at the source before delivery and are, therefore, processed when available. On the average, the system almost repeats itself every week and a half to two weeks with the exception of the month-end work.

The daily operation consists of two periods, one in the morning and one in the afternoon. The morning is the larger being from an hour and a half to sometimes three hours. The afternoon operation which usually is concerned only with the data which became available after the noon hour and operations requiring the entire data package for the day, can be accomplished in as little as ten minutes on the 1620. Up to an hour and a half is required when four or five days of irregular data become available because this must all be checked for errors. Naturally, a fair amount of card filing is an integral part of the whole operation. This averages out to roughly fifteen minutes a day.

Output consists of fourteen daily sheets, twenty-seven monthly sheets and approximately 250 cards of permanent record after unneeded data cards are sorted out. The operation is found to produce a higher quality, more legible record and makes it possible for the Daily Statistics Group to take on other work.

This operation is carried out as nearly as possible on a current basis. The fact that some data errors are cleared up by immediate telephone calls, while machine operation is held up, puts the daily statistics operation in a special category compared to usual practices. In regard to delay times, it is conceivable to start the operation only after all log sheet data are on hand. Intermittent operation could be scheduled, but it was decided that any arbitrary intentional delays on days when data are available could so seriously reduce the value of the information that every effort should be made to carryout as nearly as possible a

continuous process starting early every day.

The data structure is somewhat complicated but unavoidably so as was implied earlier. This recommends a higher degree of operator familiarity with the data and operation than a straightforward calculation would demand. The fact that the system accounts for 30% of the 1620 time on a monthly basis with a high operator activity level recommends a program of rotation of operators. Since the generation and distribution system is constantly being revised, the programs are brought up-to-date repeatedly.

(1311)-1B-16 SLIDE 1 B. G. Co. Eng Statistics

60 CYCLE GENERATION

RIVERSIDE

D.F. CODE | b5 | bbx | 3 | bbb | 742 | bb |

DATE: MONDAY APR 20 1964

	MVARHR.										MWHR.										
CODE →											3008846		3008852		3008858		3008864		3008870		3108841
TIME	NO. 1 GEN.		NO. 2 GEN.		NO. 3 GEN.		NO. 4 GEN.		NO. 5 GEN.		NO. 1 GEN.		NO. 2 GEN.		NO. 3 GEN.		NO. 4 GEN.		NO. 5 GEN.		TOTAL
	READING	DIFF.	READING	DIFF.	READING	DIFF.	READING	DIFF.	READING	DIFF.	READING	DIFF.	READING	DIFF.	READING	DIFF.	READING	DIFF.	READING	DIFF.	
CODE →							1103409		1103603								1103312		1103506		
2400	88390		82863		49840		76563		80286		66056		20718		05517		80021		00761		
0100																					
0200																					
0300																					
0400																					
0500							70	7									27	6			6
0600							87	17	97	11							52	25	78	17	42
0700			65	2	56	16	601	14	313	16			20	2	28	11	94	42	819	41	96
0800	94	4	77	12	68	12	19	18	30	17	61	5	45	25	53	25	163	69	888	69	193
0900	406	12	92	15	83	15	37	18	48	18	105	44	801	56	613	60	239	76	965	77	313
1000	20	14	911	19	901	18	56	19	68	29	167	57	858	57	676	63	316	77	043	78	332
1100	33	13	30	19	15	14	77	21	92	24	222	60	916	58	741	65	390	77	120	77	337
1200	88447	14	82950	20	49932	17	76697	20	80416	24	66281	59	20972	56	05803	62	80467	76	01197	77	330
12 Hr. Total		57		87		92		134		130		225		254		286		448		436	1649
CODE →							1203457		1203651								1203360		1203554		3208889
1300	59	12	66	16	51	19	716	19	43	27	341	60	21024	52	858	55	844	75	271	74	316
1400	73	14	83	17	67	16	35	19	70	27	400	59	080	56	923	65	619	75	347	76	331
1500	86	13	83002	19	84	17	55	20	97	27	459	59	134	54	985	62	694	75	421	74	324
1600	99	13	20	18	99	15	75	20	522	25	518	59	188	54	06047	62	767	73	495	74	322
1700	514	15	35	15	50013	14	94	19	41	19	578	60	245	57	112	65	845	78	573	78	338
1800	31	17	52	17	27	14	811	17	57	16	632	54	97	52	76	64	917	72	646	73	315
1900	45	14	66	14	40	13	29	18	73	16	690	58	355	58	238	62	994	77	723	77	332
2000	59	14	81	15	53	13	49	20	95	22	745	55	409	54	303	65	81070	76	800	77	327
2100	74	15	96	15	72	19	66	17	617	22	800	55	463	54	364	61	147	77	877	77	324
2200	85	11	114	18	92	20	85	19	37	20	827	27	508	45	411	47	224	77	955	78	274
2300		—	28	14	96	4	903	18	50	13		—	541	33	421	10	285	61	987	32	136
2400	88585	—	83138	10	50096	—	76906	3	80155	—	66827	—	21568	81	06421	—	81877		01987	—	[illegible]
12 Hr. Total		138		188		124		209		234		546		590		618		822		790	3366
24 Hr. Total		195		275		256		343		364		771		844		904		1270		1226	5015

CODE →	5018609	5018615	5018621	5018627	5-12294	5018585	5018591	5018597	5018603	5-12288
	AUXILIARY GENERATION					STATION SERVICE				
	NO. 1 UNIT	NO. 2 UNIT	NO. 3 UNIT	NO. 4 UNIT	NO. 5 UNIT	STA. SER.	P-2300-1	P-2300-2	P-2300-3	P-2300-4
PREV.	80533	66201	0995	7387	2320	4826	14923	34490	6506	0970
PRES.	80560	66243	1037	7368	2395	4827	14949	34490	6528	0979
DIFF.	27	42	42	81	75	1	26		22	9

4107

NOTE:
USE INDIVIDUAL PLANT
NO. CODE FOR X

OPERATOR 2400-0800 HRS. [illegible]

" 0800-1600 " [illegible]

" 1600-2400 " [illegible]

(1274-A)-1B-16 60 SLIDE #2

B. G. & E. Co.

LOA LOG

Engineering Statistics

BETH. STEEL CO. LOAD & INTERCHANGE DATA

D.F. CODE | bb | bbb | b | bbb | bbb | bb |

DATE: MONDAY APR 20 1964

MWHR.

TIME	60 ~ TIES TO STEEL	60 ~ TIES FROM STEEL	NET 60 ~ TIES TO STEEL	33973-A	NET 60 ~ TO STEEL	25 ~ TIE TO STEEL	25 ~ TIE FROM STEEL READING	25 ~ TIE FROM STEEL DIFF.	NET 25 ~ TIES TO STEEL	NET 25 & 60 ~ TIES TO STEEL	GENERATION 25 ~	GENERATION 60 ~	GENERATION TOTAL	PLANT LOAD	FREQ. CHANGER (25 ~ OUTPUT) READING	FREQ. CHANGER DIFF.	GAS ENGINES 25 ~	7 & 8 TURBINES 25 ~	FIRM SALES TO STEEL
CODE →	3109423	3109520		1112993		3109229		3109326			3108453	3109617			1101857				2106125
2400							76702								52997				
0100	124		124		124			1	-1	123	24	71	95	218	98	1		24	41
0200	131		131		131	1			1	132	24	77	101	233	99	1		24	56
0300	134		134		134	2			2	136	23	80	103	239	3000	1		23	62
0400	114		114		114	3			3	117	24	81	105	222	02	2		24	45
0500	100		100		100	2			2	102	25	97	122	224	05	3		25	47
0600	116		116		116	3			3	119	21	98	119	238	09	4		21	61
0700	105		105		105	3			3	108	21	108	129	237	12	3		21	60
0800	134		134		134	5			5	139	28	105	133	272	16	4		28	96
0900	105		105		105	8			8	113	21	105	126	239	24	8		21	62
1000	81		81		81	7			7	88	21	127	148	236	30	6		21	59
1100	68		68		68	3			3	71	30	140	170	241	37	7		30	66
1200	81		81	3	78 ~~81~~	8	76703		8	86 ~~89~~	22	133	155	~~244~~ 241	45	8		22	64
12Hr. Total	1293		1293		1290	45		1	44	~~1337~~	284	1222	1506	2843		48			
CODE →	3209471	3209568		1213041		3209277		3209374			3208501	3209665			1201905				2206173
1300	97		97		97	8			8	105	23	113	136	241	54	9		23	64
1400	107		107		107	9			9	116	22	124	146	262	62	8		22	85
1500	102		102		102	11			11	113	18	134	152	265	71	9		18	88
1600	153		153		153	8			8	161	24	66	90	251	78	7		24	74
1700	155		155		155	8			8	163	23	67	90	253	86	8		23	76
1800	134		134		134	10			10	144	22	79	101	245	95	9		22	68
1900	147		147		147	11			11	158	20	84	104	262	101	6		20	86
2000	123		123		123	11			11	134	20	82	102	236	09	8		20	59
2100	142		142		142	9			9	151	23	87	110	261	15	6		23	84
2200	132		132		132	10			10	142	24	112	136	278	22	7		24	101
2300	116		116		116	6			6	122	24	99	123	245	27	5		24	68
2400	109		109		109	6	76703	—	6	115	21	96	117	232	53131	4		21	55
12Hr. Total	1517		1517		1517	107		—	107	1624	264	1143	1407	3031		86		264	
24Hr. Total	2810		2810	3	~~2810~~ 2807	152		1	151	~~2961~~ 2958	548	2365	2913	~~5874~~ 5871		134		548	1625

[illegible] +1 at 2400

	PREV.	PRES.	DIFF.
Top Bus "IN" 3009234	44144	44144	—
Reactor Bus "IN" 3009240	50552	50704	152
Top Bus "OUT" 3009331	00094	00094	—
Reactor Bus "OUT" 3009337	76702	76703	1
25~ PD Veeder	699001	699153	152
#7 Turbine 3008458	03238	03416	178
#8 Turbine 3008464	17923	18293	370
#1 Unit 3009622	75078	75706	628
#2 Unit 3009628	94223	94813	590
#3 Unit 3009634	71013	71637	624
#4 Unit 3009640	27062	27585	523

	PREV.	PRES.	DIFF.
33973 "OUT" 3009428	93515	94019	504
33974 "OUT" 3009434	12024	12527	503
33981 "OUT" 3009440	12402	13317	915
33980 "OUT" 3009446	91527	92415	888
33973 "IN" 3009525	27529	27529	—
33974 "IN" 3009531	28724	28724	—
33981 "IN" 3009537	05297	05297	—
33980 "IN" 3009543	03438	03438	—
60~ PD Veeder	377278	380087	2810 → +1
33973-A	13981	13984	3
33973-A Veeder	0355808	0355812	4

OPERATOR

2400-0800 HRS. E V [illegible]

0800-1600 " [illegible]

1600-2400 " Witt

SLIDE #3

ITEM IDENTIFICATION a	b	ADDRESS	CONST.		DATE MO.	DAY	YR.	D. OF W.	24 HOUR TOTAL	UNIT CODE c	d	DEC. e	INIT. HR.	0100	0200	0300	0400	0500	0600	0700	0800	0900	1000	1100	1200
0	0	0 0 0 0 0	0 0 0 0	0 0 0	0 0	0 0	0 0	0	0 0 0 0 0 0	0	0	0	0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
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 78 79 80
1	1	1 1 1 1 1	1 1 1 1	1 1 1	1 1	1 1	1 1	1	1 1 1 1 1 1	1	1	1	1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
2	2	2 2 2 2 2	2 2 2 2	2 2 2	2 2	2 2	2 2	2	2 2 2 2 2 2	2	2	2	2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2
3	3	3 3 3 3 3	3 3 3 3	3 3 3	3 3	3 3	3 3	3	3 3 3 3 3 3	3	3	3	3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3
4	4	4 4 4 4 4	4 4 4 4	4 4 4	4 4	4 4	4 4	4	4 4 4 4 4 4	4	4	4	4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4
5	5	5 5 5 5 5	5 5 5 5	5 5 5	5 5	5 5	5 5	5	5 5 5 5 5 5	5	5	5	5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5
6	6	6 6 6 6 6	6 6 6 6	6 6 6	6 6	6 6	6 6	6	6 6 6 6 6 6	6	6	6	6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6
7	7	7 7 7 7 7	7 7 7 7	7 7 7	7 7	7 7	7 7	7	7 7 7 7 7 7	7	7	7	7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7
8	8	8 8 8 8 8	8 8 8 8	8 8 8	8 8	8 8	8 8	8	8 8 8 8 8 8	8	8	8	8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8
9	9	9 9 9 9 9	9 9 9 9	9 9 9	9 9	9 9	9 9	9	9 9 9 9 9 9	9	9	9	9 9	9 9 9 9	9 9 9 9	9 9 9 9	9 9 9 9	9 9 9 9	9 9 9 9	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 78 79 80

HOURLY DATA FOR HOUR ENDING (A.M. OR P.M.)

NOTES: (a) EDITING ROUTINE NO. (b) CARD CODE
(c) DURATION OF MEASUREMENT CODE (d) UNIT OF MEASUREMENT CODE
(e) NUMBER OF PLACES TO RIGHT OF DECIMAL POINT

IBM D31058

ENGINEERING STATISTICS
DAILY 2400 HOUR METER READINGS

SIGNAL	DATE MO.	DAY	YR.	D. OF W.		1 METER READING	1 ADDRESS	2 METER READING	2 ADDRESS	3 METER READING	3 ADDRESS	4 METER READING	4 ADDRESS	5 METER READING	5 ADDRESS	6 METER READING	6 ADDRESS
0	0 0	0 0	0 0	0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0
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 78 79 80
1	1 1	1 1	1 1	1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1
2	2 2	2 2	2 2	2	2 2 2 2 2 2	2 2 2 2 2 2	2 2 2 2 2	2 2 2 2 2 2	2 2 2 2 2	2 2 2 2 2 2	2 2 2 2 2	2 2 2 2 2 2	2 2 2 2 2	2 2 2 2 2 2	2 2 2 2 2	2 2 2 2 2 2	2 2 2 2 2
3	3 3	3 3	3 3	3	3 3 3 3 3 3	3 3 3 3 3 3	3 3 3 3 3	3 3 3 3 3 3	3 3 3 3 3	3 3 3 3 3 3	3 3 3 3 3	3 3 3 3 3 3	3 3 3 3 3	3 3 3 3 3 3	3 3 3 3 3	3 3 3 3 3 3	3 3 3 3 3
4	4 4	4 4	4 4	4	4 4 4 4 4 4	4 4 4 4 4 4	4 4 4 4 4	4 4 4 4 4 4	4 4 4 4 4	4 4 4 4 4 4	4 4 4 4 4	4 4 4 4 4 4	4 4 4 4 4	4 4 4 4 4 4	4 4 4 4 4	4 4 4 4 4 4	4 4 4 4 4
5	5 5	5 5	5 5	5	5 5 5 5 5 5	5 5 5 5 5 5	5 5 5 5 5	5 5 5 5 5 5	5 5 5 5 5	5 5 5 5 5 5	5 5 5 5 5	5 5 5 5 5 5	5 5 5 5 5	5 5 5 5 5 5	5 5 5 5 5	5 5 5 5 5 5	5 5 5 5 5
6	6 6	6 6	6 6	6	6 6 6 6 6 6	6 6 6 6 6 6	6 6 6 6 6	6 6 6 6 6 6	6 6 6 6 6	6 6 6 6 6 6	6 6 6 6 6	6 6 6 6 6 6	6 6 6 6 6	6 6 6 6 6 6	6 6 6 6 6	6 6 6 6 6 6	6 6 6 6 6
7	7 7	7 7	7 7	7	7 7 7 7 7 7	7 7 7 7 7 7	7 7 7 7 7	7 7 7 7 7 7	7 7 7 7 7	7 7 7 7 7 7	7 7 7 7 7	7 7 7 7 7 7	7 7 7 7 7	7 7 7 7 7 7	7 7 7 7 7	7 7 7 7 7 7	7 7 7 7 7
8	8 8	8 8	8 8	8	8 8 8 8 8 8	8 8 8 8 8 8	8 8 8 8 8	8 8 8 8 8 8	8 8 8 8 8	8 8 8 8 8 8	8 8 8 8 8	8 8 8 8 8 8	8 8 8 8 8	8 8 8 8 8 8	8 8 8 8 8	8 8 8 8 8 8	8 8 8 8 8
9	9 9	9 9	9 9	9	9 9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9	9 9 9 9 9 9	9 9 9 9 9	9 9 9 9 9 9	9 9 9 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 78 79 80

IBM D31059

POT	MISC	b	SUB	MASTER	SYS.
ITEM IDENTIFICATION					

INIT. HR	DAY OF YR	MO	DAY	YR
		DATE		

ENGINEERING STATISTICS
HOURLY DATA

ITEM IDENTIFICATION POT	MISC.	b	SUB NO.	MASTER NO.	SYS.	DATE MO	DAY	YR.	D. OF W.	24 HOUR TOTAL	UNIT CODE c	d	DEC. e	INIT. HR.	0100	0200	0300	0400	0500	0600	0700	0800	0900	1000	1100	1200
0 0	0 0 0	0	0 0 0	0 0 0	0 0	0 0	0 0	0 0	0	0 0 0 0 0 0	0	0	0	0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
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 78 79 80
1 1	1 1 1	1	1 1 1	1 1 1	1 1	1 1	1 1	1 1	1	1 1 1 1 1 1	1	1	1	1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
2 2	2 2 2	2	2 2 2	2 2 2	2 2	2 2	2 2	2 2	2	2 2 2 2 2 2	2	2	2	2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2	2 2 2 2
3 3	3 3 3	3	3 3 3	3 3 3	3 3	3 3	3 3	3 3	3	3 3 3 3 3 3	3	3	3	3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3
4 4	4 4 4	4	4 4 4	4 4 4	4 4	4 4	4 4	4 4	4	4 4 4 4 4 4	4	4	4	4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4	4 4 4 4
5 5	5 5 5	5	5 5 5	5 5 5	5 5	5 5	5 5	5 5	5	5 5 5 5 5 5	5	5	5	5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5	5 5 5 5
6 6	6 6 6	6	6 6 6	6 6 6	6 6	6 6	6 6	6 6	6	6 6 6 6 6 6	6	6	6	6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6	6 6 6 6
7 7	7 7 7	7	7 7 7	7 7 7	7 7	7 7	7 7	7 7	7	7 7 7 7 7 7	7	7	7	7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7	7 7 7 7
8 8	8 8 8	8	8 8 8	8 8 8	8 8	8 8	8 8	8 8	8	8 8 8 8 8 8	8	8	8	8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8	8 8 8 8
9 9	9 9 9	9	9 9 9	9 9 9	9 9	9 9	9 9	9 9	9	9 9 9 9 9 9	9	9	9	9 9	9 9 9 9	9 9 9 9	9 9 9 9	9 9 9 9	9 9 9 9	9 9 9 9	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 78 79 80

HOURLY DATA FOR HOUR ENDING (A.M. OR P.M.)

NOTES: (a) Voltage code (b) Miscellaneous equipment code.
(c) Duration of measurement code. (d) Unit of measurement code.
(e) Number of places to right of decimal point.

IBM D50303

BALTIMORE GAS AND ELECTRIC CO

192

4.10.9

SLIDE #4

COLUMN HEADINGS

1	BALTIMORE - COMP. 8
2	MD. P.R.R. INCL. BFCL - COMP. 9
3	STEEL PLANT - COMP. 10
4	BALTIMORE GROUP - COMP. 11
5	WASHINGTON - COMP. 12
6	BALTO. WASH. GROUP - COMP. 13
7	BALTO. PLUS WASH.
8	BALTO. AND STEEL PLANT
9	BALTO. AND STEEL SALES
10	BALTO. AND STEEL SALES AND MD. P.R.R.

COLUMN HEADINGS

11-13	25 CYCLE
11	BALTIMORE
12	BALTIMORE PLUS STEEL SALES
13	BALTIMORE GROUP
14-15	60 CYCLE
14	BALTIMORE
15	BALTIMORE PLUS STEEL SALES

HR.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
01	511	15	218	744	488	1232	999	729	634	649	18	18	42	493	616
02	484	21	233	738	452	1190	936	717	616	637	17	18	43	467	598
03	473	18	239	730	435	1165	908	712	609	627	17	19	43	456	590
04	467	19	222	708	434	1142	901	689	584	603	18	21	47	449	563
05	476	19	224	719	433	1152	909	700	578	597	17	19	47	459	559
06	502	20	238	760	454	1214	956	740	621	641	18	21	46	484	600
07	573	21	237	831	547	1378	1120	810	681	702	18	21	45	555	660
08	757	30	272	1059	731	1790	1488	1029	896	926	20	25	57	737	871
09	911	17	239	1167	894	2061	1805	1150	1024	1041	25	33	62	886	991
10	989	14	236	1239	992	2231	1981	1225	1077	1091	27	34	61	962	1043
11	1029	11	241	1281	1032	2313	2061	1270	1100	1111	26	29	66	1003	1071
12	1031	12	241	1284	1042	2326	2073	1272	1117	1129	27	35	65	1004	1082
13	1012	12	241	1265	1019	2284	2031	1253	1117	1129	23	31	63	989	1086
14	1039	12	262	1313	1034	2347	2073	1301	1155	1167	28	37	67	1011	1118
15	1022	21	265	1308	1029	2337	2051	1287	1135	1156	28	39	66	994	1096
16	1007	23	251	1281	1030	2311	2037	1258	1168	1191	23	31	62	984	1137
17	995	24	253	1272	1003	2275	1998	1248	1158	1182	24	32	63	971	1126
18	971	28	245	1244	982	2226	1953	1216	1115	1143	24	34	65	947	1081
19	999	24	262	1285	987	2272	1986	1261	1157	1181	22	33	59	977	1124
20	1020	24	236	1280	1002	2282	2022	1256	1154	1178	22	33	61	998	1121
21	1006	11	261	1278	968	2246	1974	1267	1157	1168	21	30	59	985	1127
22	928	14	278	1220	881	2101	1809	1206	1070	1084	23	33	64	905	1037
23	815	19	245	1079	749	1828	1564	1060	937	956	20	26	55	795	911
24	706	20	232	958	613	1571	1319	938	821	841	21	27	52	685	794
TOTAL	19723	449	5871	26043	19231	45274	38954	25594	22681	23130	527	679	1360	19196	22002

4.10.2

193

1.1.1A SYSTEM LOAD COMPONENTS MWHR - MAIN UNIT BASIS MON APR 20 1964

SLIDE #5

(M-1679) B.G. & E. CO. ENGINEERING STATISTICS

COLUMN HEADINGS

1 - 6 SAFE HARBOR NET SEND OUT
1 230 KV
2 138 KV
3 69 KV
4 TOTAL - SEE NOTE
5 2/3
6 1/3

COLUMN HEADINGS

7 MANOR-BRUNNER IS. TO BRUNNER IS.
8 NET TO BALTO.-WASH. GROUP FROM OTHERS
9 NET INTERCHANGE TO B.-W. GRP.
10 NET INTERCHANGE TO BALTO. GRP.
11 NET INTER. FROM WASH. TO BALTO. GRP.
12 S.H. 230 KV NET TO BALTO.-WASH. GRP.

HR.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
01	102	52	20	174	116	58	21	222	106	200	94	81			
02	102	51	18	171	114	57	10	249	135	214	79	92			
03	108	51	16	175	117	58	-11	233	116	205	89	119			
04	110	46	17	173	115	58	-14	228	113	191	78	124			
05	109	52	14	175	117	58	-5	190	73	149	76	114			
06	109	51	20	180	120	60	35	166	46	97	51	74			
07	104	52	23	179	119	60	80	51	-68	-21	47	24			
08	94	58	36	188	125	63	133	-41	-166	-90	76	-39			
09	104	59	48	211	141	70	179	-173	-314	-227	87	-75			
10	107	61	52	220	147	73	190	-213	-360	-217	143	-83			
11	110	62	57	229	153	76	192	-209	-362	-224	138	-82			
12	110	64	57	231	154	77	179	-164	-318	-183	135	-69			
13	112	64	52	228	152	76	158	-53	-205	-126	79	-46			
14	109	64	55	228	152	76	165	-102	-254	-158	96	-56			
15	111	64	53	228	152	76	159	-79	-231	-144	87	-48			
16	110	64	52	226	151	75	144	-24	-175	-106	69	-34			
17	107	61	54	222	148	74	159	-73	-221	-141	80	-52			
18	104	65	49	218	145	73	153	-62	-207	-156	51	-49			
19	107	61	50	218	145	73	147	-28	-173	-131	42	-40			
20	107	66	50	223	149	74	150	-45	-194	-163	31	-43			
21	110	63	48	221	147	74	147	-33	-180	-151	29	-37			
22	109	64	43	216	144	72	132	0	-144	-125	19	-23			
23	106	58	35	199	133	66	103	98	-35	-10	25	3			
24	108	57	28	195	129	66	79	121	-8	64	72	29			
TOTAL	2569	1410	947	4928	3285	1643	2685	259	-3026	-1253	1773	-116			

INCL.RAMP ENERGY OF 0 MWHR TO..1558....589...1120...1773....660
AND P.P.& L STA.SER.OF 2 MWHR FROM.1299...3615...2373......0....776

1.1.25 S.H. ENTITLEMENT & INTERCHANGE DATA MWHR MON APR 2 1964

COLUMN HEADINGS

1 NET BEN. 138KV FROM WASH.
2 NET BEN. 115KV FROM WASH.
3 NET FROM WASH.
4 NET 230KV FROM BALTO. TO S.H.& BURT.
5 NET TO PA. FROM B.W. GROUP
6 NET PA. INTER. FROM B.W. GP. BILL
7 NET DELV. FROM BALTO. GROUP

COLUMN HEADINGS* IND. MINUS

8 NET INTER. FROM BALTO. GP. BILL.
9 BALTO. AND MD. P.R.R.
10 BALTO.,ST.SALES,MD.P.R.R. WASH.
11 BALTO.,WASH.GEN.& 23 S.H.NET S.O.
12 NET APS 230 KV FROM WASH.
13 NET VEPCO 230 KV FROM WASH.

DATE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	9 27	3 53	5 191	9 13	1 9	1 6			20 796	20 1642	19 1348	1 21			
2	6 26	2 36	2 147	13 85	9 86				19 1047	20 2152	11 2030	13 85			
3	7 25	5 29	5 142	9 91	9 64				19 1070	19 2224	11 2056	9 91			
4	2 25	6 34	5 126	22 93	1 5	1 2			19 1083	19 2235	19 2074	22 93			
5		5 17	5 106	18 100	-	1 50			20 1056	20 2153	11 2095	18 100			
6		3 15	5 102	9 114	9 65	1 127		1 18	20 1029	20 2104	11 2140	9 114			
7		3 14	3 100	18 52	18 4	1 125		1 32	20 879	20 1827	19 1702	18 52			
8		3 31	3 85	10 17					20 770	20 1644	20 1425	10 17			
9		3 40	3 101	13 58					20 1043	11 2159	11 2077	13 58			
10		5 37	4 104	18 94	18 8				20 1047	20 2151	12 2071	18 94			
11	15 9	5 47	2 125	9 82					20 1049	20 2177	11 2040	9 82			
12	13 7	1 46	1 128	14 42					20 1090	11 2170	14 2076	14 42			
13	17 12	5 58	5 177	8 73					20 1030	20 2134	20 2082	8 73			
14		4 65	2 203	11 66		1 48			20 883	19 1837	19 1639	11 66			
15		3 51	2 139	9 8					20 769	20 1640	20 1560	9 8			
16	16 11	1 63	1 165	15 73					20 1048	20 2166	12 1987	15 73			
17	16 9	5 59	5 182	14 53					20 1032	12 2136	14 1942	14 53			
18	12 10	1 43	3 164	9 63					20 1083	20 2222	11 2066	9 63			
19	14 9	3 49	4 174	12 57					20 1068	20 2174	20 2047	12 57			
20	16 8	4 49	5 178	21 68					20 1046	20 2148	11 2045	21 68			
21	23 7	3 53	5 173	23 93		1 54			20 946	20 1941	20 1733	23 93			
22	4 9	5 32	6 149	13 82		1 102			20 797	20 1679	20 1508	13 82			
23		3 44	3 159	11 97		1 120			20 1063	20 2193	11 2088	11 97			
24		5 61	5 201	10 44					20 1038	20 2167	11 1959	10 44			
25		3 79	3 239	9 57					20 1012	20 2143	20 1992	9 57			
26		6 25	6 97	18 77	12 21				20 1053	20 2187	12 1977	18 77			
27	14 3	5 54	5 195	11 52					20 932	12 1997	11 1750	11 52			
28		5 36	1 105	20 12					20 873	20 1834	20 1496	20 12			
29		3 29	3 69						21 729	21 1524	21 1164				
30	9 7	3 39	5 101	22 53					20 1048	12 2155	20 2036	22 53			
31	9 8	21 61	1 127	22 52					20 1052	14 2175	20 2106	22 52			
TOTAL	1 27	25 79	25 239	6 114	2 86	6 127	1	7 32	12 1090	4 2235	6 2140				

1620 USERS GROUP

GENERAL DATA PROCESSING

A WORK PERFORMANCE MONITORING SYSTEM DESIGNED FOR AN ELECTRIC UTILITY METER AND INSTALLATION DEPARTMENT

by

D. D. Williams

Engineering Computer Applications Group

Baltimore Gas and Electric Company

Baltimore, Maryland 21203

196

4.11.0

A WORK PERFORMANCE MONITORING SYSTEM DESIGNED FOR AN ELECTRIC UTILITY METER AND INSTALLATION DEPARTMENT

by

D. D. Williams
Baltimore Gas and Electric Company

The Baltimore Gas and Electric Company has for a number of years conducted a Work Measurement Program in a number of its operating departments, notably the Gas Service and Meter and Installation Departments. This Program was started and has spread as a means of providing Management not with a whip with which to increase work output, but rather with information that would suggest improvements in budgeting and scheduling results and in work practices and planning. In other words, its purpose was to provide Management with a device with which it could exert better control over operations.

The Program fell a little short of this objective because it left the analysis of the mass of information which it produced up to the Department Heads. To eliminate this shortcoming, the Installation Division of the Meter and Installation Department requested that a monthly analysis be made of this information on the Company's type 1620 computer. The Installation Division has the function of installing and maintaining metering equipment and service drops on customers' premises. The attached monthly operations report was developed as an answer to this request. It analyzes between 30,000 and 40,000 time cards per month for eight work districts and for the Division as a whole. The

Meter and Installation Department uses contractors to supplement its own forces, and for the purposes of this report, all contractors regardless of the districts in which they work are treated as comprising District 9. The copy of the report attached hereto contains the sheets for only one of these districts where a sheet type is produced for the districts as well as for the Division as a whole.

I will not attempt to discuss here the contents of all of the tabulations contained in the report. However, from a general data processing point-of-view, a few salient features are noteworthy. As a control device, the report would be virtually useless without the bases for comparison provided by the columns headed, "Last Month", "Same Month Last Year" and "12-Month Average". The "One-Month Average" shown in this particular issue of the report will grow to be a "12-Month Average" in January, 1965, and will continue thereafter to be a "12-Month Average". To provide these figures, the system turns out summary cards each month which become input to the system the following month. In addition to the time cards and the summary cards, there are approximately a thousand master function cards which are input to the system and which provide the bases for calculating performance indexes and for distributing work and travel time as indicated by the various tabulations. By definition, a function is a prescribed process such as installing a "service drop" or removing a meter, and a job is a unit of work which covers a single completion of a function. As indicated above, the work of the Installation Division has been divided into approximately a thousand functions which are broadly classified as productive or non-productive.

The system was designed to be as flexible as possible with respect to such items as the addition and subtraction of tabulations to and from the report, the addition, changing and subtraction of lines within any tabulation and the changing of the master function list. The system consists of several programs, the first of which compiles and summarizes data from the time cards after they have been sorted by function and district. The sorting takes between eight and ten hours to complete on an IBM Corporation type 082 sorter. The next two programs contain the headings and interpretatively programmed steps for compiling all of the tabulations in the report, except the one contained in sheet type 9. This sheet is produced by a separate program. It takes from one and a half to two hours to turn out the complete report on a type 1620 Model 1 computer connected with a type 1622 card read-punch unit. A complete report consists of approximately 60 pages, all of which are turned out by the system ready for printing on blank paper on the type 870 printer. It takes about six hours to print a complete report.

As experience is gained in the interpretation and use of the information contained in this report, it is expected that the system will be extended to cover the operations of other divisions and departments in the Company. It is hoped that its flexibility will make it easily adaptable to their needs.

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

DIVISION SUMMARY

DISTRIBUTION OF WORK TIME BY FUNCTIONAL CATEGORY (1)

CATEGORY	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
PRODUCTIVE WORK				
JOBS RECEIVED				
TOTAL	3148			3148
RATE (NO/MAN-DAY)	1.2			1.2
JOBS COMPLETED				
TOTAL	4002			4002
COMPANY	3273			3273
CONTRACTOR	729			729
RATE (NO/MAN-DAY)	1.5			1.5
TOTAL MEASURED	3475			3475
ACTUAL JOB TIME (HRS/MAN-DAY)	2.85			2.85
EXPECTED JOB TIME (HRS/MAN-DAY)	2.13			2.13
JOB PERFORMANCE INDEX	1.34			1.34
TOTAL UNMEASURED	527			527
ACTUAL JOB TIME (HRS/MAN-DAY)	.32			.32
SUPER., INSPECT., CLER.,(HRS/MAN-DAY)	2.49			2.49
NON-PRODUCTIVE WORK				
JOB-RELATED				
HEADQUARTERS (HRS/MAN-DAY)	.49			.49
FIELD (HRS/MAN-DAY)	.27			.27
WORK PERFORMANCE INDEX (MEASD FUNC)	1.22			1.22
TRAVEL (HRS/MAN-DAY) (2)	1.09			1.09
TRAVEL PERFORMANCE INDEX	1.09			1.09
OTHER (HRS/MAN-DAY) (3)	.48			.48

DISTRIBUTION OF WORK TIME BY TIME CATEGORY (1) (HRS/MAN-DAY)

CATEGORY	CURRENT MONTH	LAST MONTH	SAME MO. LAST YR.	1-MO. AVG.
STRAIGHT TIME	7.88			7.88
OVERTIME	.12			.12

NOTES. (1) TIMES GIVEN IN TERMS OF EQUIVALENT MAN-DAYS.
(2) EXCLUDING TRAVEL FOR SUPERVISION, INSPECTION, CLERICAL.
(3) MEETINGS, TRAINING, AND ABSENCES COUNTED AS WORK TIME.

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

ANNAPOLIS SUMMARY

DISTRIBUTION OF WORK TIME BY FUNCTIONAL CATEGORY (1)

CATEGORY	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
PRODUCTIVE WORK				
JOBS RECEIVED				
TOTAL	209			209
RATE (NO/MAN-DAY	.8			.8
JOBS COMPLETED				
TOTAL	478			478
COMPANY	478			478
CONTRACTOR				
RATE (NO/MAN-DAY)	1.8			1.8
TOTAL MEASURED	410			410
ACTUAL JOB TIME (HRS/MAN-DAY)	2.98			2.98
EXPECTED JOB TIME (HRS/MAN-DAY)	2.33			2.33
JOB PERFORMANCE INDEX	1.28			1.28
TOTAL UNMEASURED	68			68
ACTUAL JOB TIME (HRS/MAN-DAY)	.25			.25
SUPER., INSPECT., CLER.,(HRS/MAN-DAY)	2.10			2.10
NON-PRODUCTIVE WORK				
JOB-RELATED				
HEADQUARTERS (HRS/MAN-DAY)	.50			.50
FIELD (HRS/MAN-DAY)	.21			.21
WORK PERFORMANCE INDEX (MEASD FUNC)	1.07			1.07
TRAVEL (HRS/MAN-DAY) (2)	1.54			1.54
TRAVEL PERFORMANCE INDEX	1.10			1.10
OTHER (HRS/MAN-DAY) (3)	.42			.42

DISTRIBUTION OF WORK TIME BY TIME CATEGORY (1)
(HRS/MAN-DAY)

CATEGORY	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
STRAIGHT TIME	7.94			7.94
OVERTIME	.06			.06

NOTES. (1) TIMES GIVEN IN TERMS OF EQUIVALENT MAN-DAYS.
(2) EXCLUDING TRAVEL FOR SUPERVISION, INSPECTION, CLERICAL.
(3) MEETINGS, TRAINING, AND ABSENCES COUNTED AS WORK TIME.

4.11.5

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

JOB PERFORMANCE INDICES

(MEASURED JOBS)

	CURRENT MONTH	LAST MONTH	SAME MO. LAST YR.	1-MONTH AVG.
DIVISION	1.34			1.34
MONUMENT STREET	1.58			1.58
ANNAPOLIS	1.28			1.28
ESSEX	1.16			1.16
GLEN BURNIE	1.20			1.20
HOWARD	1.12			1.12
COCKEYSVILLE	1.43			1.43
BEL AIR	.97			.97
WESTMINSTER	1.08			1.08
CONTRACTORS	1.57			1.57

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

TRAVEL INDICES (MEASURED TRAVEL)

PERSONNEL PERFORMANCE

	CURRENT MONTH		LAST MONTH		SAME MO. LAST YR.		1-MONTH AVG.	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
DIVISION	1.09	42.84					1.09	42.84
MONUMENT STREET	1.04	34.51					1.04	34.51
ANNAPOLIS	1.10	51.86					1.10	51.86
ESSEX	1.63	45.48					1.63	45.48
GLEN BURNIE	1.68	50.64					1.68	50.64
HOWARD	1.05	38.92					1.05	38.92
COCKEYSVILLE	.89	44.30					.89	44.30
BEL AIR	.78	51.92					.78	51.92
WESTMINSTER	1.00	64.61					1.00	64.61
CONTRACTORS	1.03	34.54					1.03	34.54

NOTES. (1) ACTUAL TIME/NORMAL TIME.
(2) TOTAL TRAVEL MINUTES/COMPLETED PRODUCTIVE JOBS.

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

TRAVEL INDICES (MEASURED TRAVEL)

DISPATCH PERFORMANCE

	CURRENT MONTH		LAST MONTH		SAME MO. LAST YR.		1-MONTH AVG.	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
DIVISION	5.6	10.8					5.6	10.8
MONUMENT STREET	3.1	5.9					3.1	5.9
ANNAPOLIS	9.6	17.8					9.6	17.8
ESSEX	4.9	8.6					4.9	8.6
GLEN BURNIE	6.4	11.4					6.4	11.4
HOWARD	5.7	12.1					5.7	12.1
COCKEYSVILLE	6.8	14.2					6.8	14.2
BEL AIR	8.5	16.4					8.5	16.4
WESTMINSTER	10.1	21.0					10.1	21.0
CONTRACTORS	3.2	6.2					3.2	6.2

NOTES. (1) VEHICLE-MILES/FIELD JOB.
(2) MAN-MILES/FIELD JOB.

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

DIVISION DETAIL

DISTRIBUTION OF WORK TIME BY PERSONNEL CLASSIFICATION
(EQUIVALENT MAN-DAYS)

CLASSIFICATION	CURRENT MONTH (1)	CURRENT MONTH (2)	LAST MONTH (1)	LAST MONTH (2)	SAME MO. LAST YR. (1)	SAME MO. LAST YR. (2)	1-MONTH AVG. (1)	1-MONTH AVG. (2)
M.+I.	1609	652					1609	652
TRANSFERRED								
BORROWED	32	22					32	22
CONTRACTOR	196	88					196	88
SUB-TOTAL	1838	761					1838	761
TOTAL	2599						2599	

NOTES. (1)PRODUCTIVE WORK (2)NON-PRODUCTIVE WORK

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

ANNAPOLIS DETAIL

DISTRIBUTION OF WORK TIME BY PERSONNEL CLASSIFICATION
(EQUIVALENT MAN-DAYS)

CLASSIFICATION	CURRENT MONTH (1)	CURRENT MONTH (2)	LAST MONTH (1)	LAST MONTH (2)	SAME MO. LAST YR. (1)	SAME MO. LAST YR. (2)	1-MONTH AVG. (1)	1-MONTH AVG. (2)
M.+I.	170	86					170	86
TRANSFERRED								
BORROWED	2	1					2	1
CONTRACTOR								
SUB-TOTAL	173	87					173	87
TOTAL	260						260	

NOTES. (1)PRODUCTIVE WORK (2)NON-PRODUCTIVE WORK

4.11.10

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

DIVISION SUMMARY

DISTRIBUTION OF PRODUCTIVE TIME BY FUNCTIONAL CLASSIFICATION
(HRS/MAN-DAY)

CLASSIFICATION	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
SUPERVISION	.62			.62
INSPECTION	1.13			1.13
CLERICAL	.74			.74
INSTALLATION				
SELF-CONTAINED	2.51			2.51
TRANSFORMER	.18			.18
MISCELLANEOUS	.48			.48
TOTAL	3.17			3.17
STORM TROUBLE				
TOTAL	5.66			5.66

DISTRIBUTION OF JOB-RELATED NON-PRODUCTIVE TIME BY FUNCTIONAL CLASS.
(HRS/MAN-DAY)

CLASSIFICATION	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
HEADQUARTERS				
PREPARG & COMPLTG DAYS WORK	.30			.30
MISCELLANEOUS	.18			.18
TOTAL	.48			.48
FIELD				
CGI	.06			6
TURN-BACKS & TURN-DOWNS	.08			.08
MISCELLANIOUS	.13			.13
TOTAL	.27			.27
TOTAL	.75			.75

DISTRIBUTION OF OTHER NON-PRODUCTIVE TIME BY FUNCTIONAL CLASS.
(HRS/MAN-DAY)

CLASSIFICATION	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
MEETINGS	.05			.05
TRAINING				
ON-THE-JOB	.10			.10
OFF-THE-JOB	.30			.30
ABSENCES COUNTED AS WORK TIME	.03			.03
TOTAL	.48			.48

 4.11.11

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

ANNAPOLIS SUMMARY

DISTRIBUTION OF PRODUCTIVE TIME BY FUNCTIONAL CLASSIFICATION (HRS/MAN-DAY)

CLASSIFICATION	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
SUPERVISION	.70			.70
INSPECTION	.68			.68
CLERICAL	.73			.73
INSTALLATION				
SELF-CONTAINED	2.41			2.41
TRANSFORMER	.21			.21
MISCELLANEOUS	.60			.60
TOTAL	3.22			3.22
STORM TROUBLE				
TOTAL	5.32			5.32

DISTRIBUTION OF JOB-RELATED NON-PRODUCTIVE TIME BY FUNCTIONAL CLASS. (HRS/MAN-DAY)

CLASSIFICATION	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
HEADQUARTERS				
PREPARG + COMPLTG DAYS WORK	.30			.30
MISCELLANEOUS	.19			.19
TOTAL	.49			.49
FIELD				
CGI	.02			.02
TURN-BACKS + TURN-DOWNS	.11			.11
MISCELLANEOUS	.08			.08
TOTAL	.21			.21
TOTAL	.70			.70

DISTRIBUTION OF OTHER NON-PRODUCTIVE TIME BY FUNCTIONAL CLASS. (HRS/MAN-DAY)

CLASSIFICATION	CURRENT MONTH	LAST MONTH	SAME MO LAST YR	1-MO. AVG.
MEETINGS	.03			.03
TRAINING				
ON-THE-JOB				
OFF-THE-JOB	.35			.35
ABSENCES COUNTED AS WORK TIME	.03			.03
TOTAL	.42			.42

4. 11. 12

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

DISTRIBUTION OF JOB-RELATED TRAVEL DISTANCE BY WORK UNIT (1)
(MILES AND PER CENT)

WORK UNIT	CURRENT MONTH (2)	(3)	LAST MONTH (2)	(3)	SAME MO. LAST YR. (2)	(3)	1-MONTH AVG. (2)	(3)
DIVISION	27317	-		-		-	27317	-
DISTRICT								
MONUMENT ST.	6039	22.1					6039	22.1
ANNAPOLIS	5512	20.2					5512	20.2
ESSEX	2499	9.1					2499	9.1
GLEN BURNIE	2342	8.6					2342	8.6
HOWARD	2119	7.8					2119	7.8
COCKEYSVILLE	4036	14.8					4036	14.8
BEL AIR	2348	8.6					2348	8.6
WESTMINSTER	2422	8.9					2422	8.9
CONTRACTORS	2837	10.4					2837	10.4

DISTRIBUTION OF JOB-RELATED TRAVEL TIME BY WORK UNIT (1)
(VEHICLE HOURS AND PER CENT)

WORK UNIT	CURRENT MONTH (2)	(3)	LAST MONTH (2)	(3)	SAME MO. LAST YR. (2)	(3)	1-MONTH AVG. (2)	(3)
DIVISION	1396.0	-		-		-	1396.0	-
DISTRICT								
MONUMENT ST.	478.0	34.2					478.0	34.2
ANNAPOLIS	208.2	14.9					208.2	14.9
ESSEX	173.1	12.4					173.1	12.4
GLEN BURNIE	118.3	8.5					118.3	8.5
HOWARD	92.0	6.6					92.0	6.6
COCKEYSVILLE	152.1	10.9					152.1	10.9
BEL AIR	73.1	5.2					73.1	5.2
WESTMINSTER	101.4	7.3					101.4	7.3
CONTRACTORS	213.9	15.3					213.9	15.3

NOTES. (1) EXCLUDING TRAVEL FOR SUPERVISION, INSPECTION, + CLERICAL.
(2) TOTAL DISTANCE OR TIME.
(3) PER CENT OF DIVISION TOTAL.

 4.11.13

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

DISTRIBUTION OFJOB RELATED CREW MEMBER TRAVEL TIME BY WORK UNIT
(MAN HOURS AND PER CENT)

WORK UNIT	CURRENT MONTH (1)	CURRENT MONTH (2)	LAST MONTH (1)	LAST MONTH (2)	SAME MO. LAST YR. (1)	SAME MO. LAST YR. (2)	1-MONTH AVG. (1)	1-MONTH AVG. (2)
DIVISION	1440.6	-		-		-	1440.6	-
DISTRICT								
MONUMENT ST.	442.3	30.7					442.3	30.7
ANNAPOLIS	192.0	13.3					192.0	13.3
ESSEX	133.9	9.3					133.9	9.3
GLEN BURNIE	132.4	9.2					132.4	9.2
HOWARD	102.0	7.1					102.0	7.1
COCKEYSVILLE	227.4	15.8					227.4	15.8
BEL AIR	92.2	6.4					92.2	6.4
WESTMINSTER	118.3	8.2					118.3	8.2
CONTRACTORS	205.8	14.3					205.8	14.3

NOTES. (1) TOTAL MAN-HOURS MINUS TOTAL VEHICLE-HOURS.
(2) PER CENT OF DIVISION DIFFERENCE.

DISTRIBUTION OF JOB RELATED EXCESS CREW MEMBER TRAVEL TIME BY WORK UNIT
(MAN HOURS AND PER CENT)

WORK UNIT	CURRENT MONTH (1)	CURRENT MONTH (2)	LAST MONTH (1)	LAST MONTH (2)	SAME MO. LAST YR. (1)	SAME MO. LAST YR. (2)	1-MONTH AVG. (1)	1-MONTH AVG. (2)
DIVISION	238.0	-		-		-	238.0	-
DISTRICT								
MONUMENT ST.	97.5	41.0					97.5	41.0
ANNAPOLIS	25.1	10.5					25.1	10.5
ESSEX	26.2	11.0					26.2	11.0
GLEN BURNIE	12.7	5.3					12.7	5.3
HOWARD	14.5	6.1					14.5	6.1
COCKEYSVILLE	36.2	15.2					36.2	15.2
BEL AIR	8.0	3.4					8.0	3.4
WESTMINSTER	18.1	7.6					18.1	7.6
CONTRACTORS	41.9	17.6					41.9	17.6

NOTES. (1) TOTAL TIME DUE TO CREW SIZE LARGER THAN NECESSARY.
(2) PER CENT OF DIVISION EXCESS.

4.11.14

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

DIVISION SUMMARY

DISTRIBUTION OF WORK TIME (*)(EXCLUDING DEFINED TRAVEL TIME) BY FUNCTION

		CURRENT MONTH					PERFORMANCE INDEX		
LINE NO.	FUNC. NO.	TOTAL TIME (*) (MHRS)	NO. JOBS	AVERAGE TIME TRAVEL (MHRS)	AVERAGE TIME FUNC. (MHRS)	PERF. INDEX	LAST MO.	SAME MO. LAST YR.	12-MO. AVG.
1	901	2,926.4			.01				
2	902	1,928.1	8		241.01				
3	900	1,619.0							
4	771	1,401.7	517	.75	2.71	1.33			1.33
5	781	1,369.0	381	.83	3.59	1.05			1.05
6	477	779.6	24		32.48				
7	481	456.5	901		.51	1.20			1.20
8	751	358.2	119	.57	3.01	2.23			2.23
9	904	336.9	451	.07	.75	1.38			1.38
10	741	334.6	140	.37	2.39	1.89			1.89
11	941	333.6	150	.60	2.22	1.18			1.18
12	482	329.6	876	.84	.38	1.04			1.04
13	497	269.8							
14	977	233.7	257	.30	.91				
15	976	201.5	57	.35	3.54				
16	905	200.0	85	.44	2.35	1.42			1.42
17	791	199.6	166	.62	1.20	1.16			1.16
18	915	195.3	48	.82	4.07	.79			.79
19	493	192.4	250	.72	.77	1.23			1.23
20	975	182.9	55	.70	3.33				
21	914	179.0	31	.44	5.77	2.17			2.17
22	499	153.1	370		.41				
23	721	147.7	102	.60	1.45	2.23			2.23
24	490	146.4	238	.61	.62	1.50			1.50
25	495	140.0	93	.20	1.51				
26	487	134.7	27		4.99				
27	937	133.3	54	.94	2.47	1.04			1.04
28	484	130.3	30		4.34				
29	761	118.1	142	.17	.83	1.08			1.08
30	845	117.6	7	1.97	16.80	1.40			1.40
31	731	104.4	101	.60	1.03	2.22			2.22
32	801	104.0	31	.63	3.35	1.75			1.75
33	951	100.6	48	.29	2.10	3.92			3.92
34	711	97.1	55	.63	1.77	3.14			3.14
35	947	89.7	60	.55	1.50	.96			.96
36	913	88.4	22	.62	4.02	2.51			2.51

37	940	84.3	65	.26	1.30	1.80	1.80
38	978	78.3	13		6.02		
39	843	77.9	7	1.04	11.13	1.12	1.12
40	496	77.7	71	.29	1.09		
41	476	76.3	17		4.49		
42	483	76.0	38	.02	2.00		
43	919	74.8	62	.33	1.21		
44	933	70.9	5	.67	14.18	7.13	7.13
45	946	64.2	34	.60	1.89	1.28	1.28
46	841	54.0	5	1.40	10.80	1.00	1.00
47	938	53.9	51	.43	1.06	.96	.96
48	475	50.8	8		6.35		
49	972	50.6	86	.01	.59	1.61	1.61
50	910	50.3	15	.58	3.35	1.05	1.05
51	942	50.1	46	.29	1.09	1.37	1.37
52	480	50.0	35		1.43		
53	939	49.4	13	.42	3.80	2.31	2.31
54	945	48.4	14	1.07	3.46	3.12	3.12
55	944	47.9	52	.68	.92	1.26	1.26
56	743	45.9	11	.59	4.17	3.50	3.50
57	909	45.9	11	.93	4.17	1.34	1.34
58	846	43.9	2	2.02	21.95	1.48	1.48
59	957	43.9	5	.29	8.78	1.53	1.53
60	811	39.2	11	.67	3.56	1.16	1.16
61	921	38.8	54	.56	.72		
62	745	36.6	7	.66	5.23	3.59	3.59
63	773	34.9	9	.62	3.88	1.91	1.91
64	917	31.2	36	.77	.87	1.58	1.58
65	479	31.0	17		1.82		
66	489	30.0	88	.17	.34		
67	474	29.2	17		1.72		
68	853	23.3	2	.73	11.65	.75	.75
69	488	21.9	47	.79	.47		
70	494	21.9	30	.53	.73	1.10	1.10
71	407	19.0					
72	821	19.0	18	.83	1.06	1.14	1.14
73	020	18.3	35	.03	.52	1.20	1.20
74	908	17.9	10	.44	1.79	.84	.84
75	922	16.3	1	.33	16.30	1.63	1.63
76	713	15.5	2	.58	7.75	11.65	11.65
77	492	14.8	28	.68	.53	1.36	1.36
78	907	14.3	7	.40	2.04	.63	.63
79	918	14.1	11	.66	1.28	.94	.94
80	844	13.1					
81	935	13.0	1	1.17	13.00	1.30	1.30
82	765	12.6	7	.17	1.80	1.67	1.67
83	755	12.4	2	.37	6.20	2.57	2.57
84	983	11.7	13	.16	.90	.83	.83
85	953	9.5	9	.09	1.06	1.06	1.06
86	911	9.4	3	.88	3.13	1.04	1.04
87	865	9.2	2		4.60	1.62	1.62
88	984	9.1	15		.61	1.82	1.82
89	803	9.0	3	.22	3.00	1.47	1.47
90	863	8.3	2	1.37	4.15	1.56	1.56
91	030	8.2	10	.34	.82	1.36	1.36

92	793	7.1	5	.27	1.42	.92	.92
93	498	7.0	9		.78		
94	950	6.6	3	.46	2.20	2.27	2.27
95	763	6.5	1	.67	6.50	8.62	8.62
96	943	6.3	5	.42	1.26	.99	.99
97	080	6.2	16	.01	.39	.96	.96
98	486	6.0	7		.86	3.03	3.03
99	200	5.9	16	.31	.37	.82	.82
100	491	5.8	10	.62	.58	1.48	1.48
101	851	5.6	2	.45	2.80	.20	.20
102	963	4.8	1	.60	4.80	.36	.36
103	916	4.6	3	.30	1.53	.99	.99
104	783	4.5	1		4.50	1.29	1.29
105	435	4.3	10	.27	.43	1.13	1.13
106	752	4.0	1		4.00	2.29	2.29
107	126	3.8	4		.95	1.15	1.15
108	725	3.6	2	.89	1.80	2.00	2.00
109	090	3.3	6	.22	.55	1.37	1.37
110	485	2.8	1		2.80		
111	733	2.5	2	1.05	1.25	2.86	2.86
112	602	2.4	10	.21	.24		
113	969	2.4	2	.67	1.20	2.92	2.92
114	715	2.3	1	.50	2.30	2.55	2.55
115	813	2.3	1		2.30	.82	.82
116	971	2.3	4	.41	.58	1.50	1.50
117	601	2.2	1	.73	2.20		
118	120	2.0	2		1.00	1.44	1.44
119	979	2.0	1		2.00		
120	434	1.9	4		.48	.91	.91
121	923	1.9	1	.42	1.90	.16	.16
122	449	1.8	7	.26	.26	.95	.95
123	723	1.7	1		1.70	2.22	2.22
124	958	1.7	1		1.70	1.32	1.32
125	603	1.6	3	.14	.53		
126	110	1.5	2		.75	1.09	1.09
127	964	1.5	1		1.50	.38	.38
128	432	1.4	4	.24	.35	1.60	1.60
129	861	1.4	1		1.40	.53	.53
130	439	1.2	3	.90	.40	.68	.68
131	408	1.0					
132	422	1.0	4	.33	.25	.56	.56
133	013	.8	1	.33	.80	1.02	1.02
134	116	.8	1		.80	.94	.94
135	418	.8	6	.45	.13		
136	437	.8	3		.27		
137	023	.7	1		.70	.91	.91
138	117	.7	1		.70	.76	.76
139	428	.6	3	.42	.20		
140	431	.6	2	.21	.30	1.31	1.31
141	562	.6	1	.08	.60	.67	.67
142	440	.5	1	.42	.50	1.00	1.00
143	609	.5	1	.23	.50		
144	442	.4	2	.16	.20	.69	.69
145	559	.2	1	.17	.20		
TOTAL		17,957.6	7,242				

4.11.17

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

ANNAPOLIS SUMMARY

DISTRIBUTION OF WORK TIME (*)(EXCLUDING DEFINED TRAVEL TIME) BY FUNCTION

		CURRENT MONTH					PERFORMANCE INDEX		
LINE NO.	FUNC. NO.	TOTAL TIME (*) (MHRS)	NO. JOBS	AVERAGE TRAVEL (MHRS)	TIME FUNC. (MHRS)	PERF. INDEX	LAST MO.	SAME MO. LAST YR.	12-MO. AVG.
1	771	199.8	71	.95	2.81	1.36			1.36
2	902	189.1	4		47.27				
3	900	181.3							
4	901	175.8							
5	781	124.5	34	.90	3.66	1.18			1.18
6	477	90.8							
7	481	58.6	105		.55	1.56			1.56
8	941	50.2	19	.92	2.64	1.37			1.37
9	915	40.8	1	.91	3.70	.74			.74
10	845	34.9	3	1.36	11.63	.83			.83
11	947	30.6	18	.72	1.70	1.13			1.13
12	493	29.6	39	.80	.75	1.23			1.23
13	751	24.3	7	.90	3.47	2.67			2.67
14	937	21.6	8	1.05	2.70	1.16			1.16
15	741	20.4	9	.47	2.26	2.09			2.09
16	933	20.2	1	1.40	20.20	10.10			10.10
17	482	19.1	107	.85	.17	.54			.54
18	407	19.0							
19	484	18.8	3		6.26				
20	020	18.3	35	.03	.52	1.20			1.20
21	791	17.6	12	.51	1.46	1.57			1.57
22	499	16.0	27		.59				
23	976	15.4	22	.03	.70				
24	801	12.6	2	1.08	6.30	3.49			3.49
25	977	11.5	6	.66	1.91				
26	487	11.3	2		5.65				
27	944	10.6	7	.67	1.51	1.52			1.52
28	945	10.5	5	.89	2.10	1.58			1.58
29	913	10.1	3	.43	3.36	2.03			2.03
30	946	9.6	5	.94	1.92	1.28			1.28
31	984	9.1	15		.60	1.82			1.82
32	910	8.3	2	.87	4.15	1.45			1.45
33	030	8.2	10	.34	.82	1.36			1.36
34	721	7.8	4	.50	1.95	3.16			3.16
35	496	7.2	18	.20	.40				
36	921	6.6	9	.72	.73				

37	480	6.4	6		1.06		
38	495	6.4	6	.42	1.06		
39	917	6.4	15	.32	.42	.64	.64
40	080	6.2	16	.01	.38	.96	.96
41	811	6.1	2	.68	3.05	1.19	1.19
42	200	5.9	16	.31	.36	.82	.82
43	476	5.8	2		2.90		
44	821	5.6	4	.81	1.40	1.55	1.55
45	975	5.0	4	1.18	1.25		
46	435	4.3	10	.27	.43	1.13	1.13
47	483	4.2	2	.13	2.10		
48	490	4.2	13	.78	.32	1.04	1.04
49	773	4.0	1	1.00	4.00	1.71	1.71
50	126	3.8	4		.95	1.15	1.15
51	474	3.8	1		3.80		
52	951	3.8	1	.77	3.80	5.63	5.63
53	904	3.4	2	.14	1.70	2.86	2.86
54	090	3.3	6	.22	.55	1.37	1.37
55	939	3.1	2	.63	1.55	1.31	1.31
56	475	3.0					
57	731	2.9	2	1.85	1.45	3.48	3.48
58	479	2.5	1		2.50		
59	489	2.4	23		.10		
60	602	2.4	10	.21	.24		
61	942	2.1	2	.33	1.05	1.26	1.26
62	120	2.0	2		1.00	1.44	1.44
63	434	1.9	4		.47	.91	.91
64	983	1.9	2	.98	.95	.85	.85
65	449	1.8	7	.26	.25	.95	.95
66	711	1.8	1	1.65	1.80	3.00	3.00
67	603	1.6	3	.14	.53		
68	918	1.6	2	.80	.80	.67	.67
69	110	1.5	2		.75	1.09	1.09
70	488	1.5	7	.76	.21		
71	432	1.4	4	.24	.35	1.60	1.60
72	938	1.3	4	.11	.32	.31	.31
73	940	1.3	1		1.30	1.11	1.11
74	408	1.0					
75	422	1.0	4	.33	.25	.56	.56
76	013	.8	1	.33	.80	1.02	1.02
77	116	.8	1		.80	.94	.94
78	418	.8	6	.45	.13		
79	437	.8	3		.26		
80	439	.8	2	.26	.40	.64	.64
81	023	.7	1		.70	.91	.91
82	117	.7	1		.70	.76	.76
83	492	.7	3	.48	.23	.90	.90
84	428	.6	3	.42	.20		
85	431	.6	2	.21	.30	1.31	1.31
86	562	.6	1	.08	.60	.67	.67
87	440	.5	1	.42	.50	1.00	1.00
88	486	.5	1		.50	3.00	3.00
89	609	.5	1	.23	.50		
90	761	.5	1		.50	.75	.75
91	442	.4	2	.16	.20	.69	.69
92	559	.2	1	.17	.20		
TOTAL		1,679.2	848				

4.11.19.

METER AND INSTALLATION DEPARTMENT

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

CONTRACTOR DETAIL

DISTRIBUTION OF WORK TIME (INCLUDING DEFINED TRAVEL TIME)

BY BUDGET CATEGORY

(MAN- HOURS)

CATEGORY	SPONSORED BY M.&I.	SPONSORED BY E.D.D.	SPONSORED BY OTHERS	TOTAL
OPERATION	445.2		4.6	449.8
MAINTENANCE	91.5	27.5		119.0
CONSTRUCTION	1139.0	482.2		1621.2
MISCELLANEOUS	81.9			81.9
SUB-TOTAL	1757.5	509.7	4.6	2271.8
TOTAL		2271.8		
DEFINED TRAVEL TIME		419.7		
DEFFERENCE		1852.1		

METER AND INSTALLATION DEPARTMENT

INSTALLATION DIVISION

MONTHLY OPERATIONS REPORT

SERVICES INSTALLED AND CHANGED IN DWELLING UNITS

	UNDER 100 AMPERES	100 AMPERES & OVER	TOTAL
RESIDENCES			
NEW			
INSTALLED	26	452	478
OLD			
INSTALLED	26	26	52
CHANGED	161	311	472
APARTMENTS			
NEW			
INSTALLED	220	229	449
OLD			
INSTALLED	48	17	65
CHANGED	52	44	96
TOTAL			
NUMBER	533	1079	1612
PER CENT	33.1	66.9	100.0

4.11.21

A MULTIPLE-ITERATION PROCEDURE FOR ENGINEERING DESIGN

Meriwether L. Baxter, Jr.

Chief, Mathematical Research

Gleason Works

Rochester, N.Y.

Prepared for presentation at the 1620 Users Group

Eastern Region Meeting

Washington, D. C.

May 6-8, 1964

A MULTIPLE-ITERATION PROCEDURE FOR ENGINEERING DESIGN

Introduction

The process of design is essentially one of synthesis; but in most real engineering design problems direct synthesis is out of the question due to the complex interrelation of variables. Methods of analysis, on the other hand, have vastly improved, and there is hardly any structure or mechanism that cannot be analyzed (with the aid of computers) to determine quantitatively any property we may wish to study.

Fortunately, we can convert analysis into synthesis by means of iteration, and obtain thereby a powerful design tool. The purpose of this paper is to describe one such procedure that has been found effective at the Gleason Works.

Statement of the Problem

The elements of the problem are the following:

1) A set of n variable design parameters (a, b, c ---);
2) A calcuation procedure of any complexity based on the parameters;
3) A set of n results, functionally dependent on the n parameters, whose desired values are known.

The problem is to establish a procedure for determining the values of the n parameters that will yield the desired n results.

Obviously, some restrictions are necessary for such a procedure to work, and some discussion of them will come later in this paper. However at this stage it is sufficient to require only that

1) a solution to the problem probably exists (a matter of engineering judgment or experience)
2) a set of reasonable starting values for the n parameters can be chosen.

Outline of the Method

It is not particularly difficult to handle multiple iteration when the relating functions are simple and short enough to make differentiation easy. In many engineering problems, however, this is not the case and we need a method which does not require any analysis of the functions involved, but merely the ability to calculate them rapidly.

Let us say that if we are given values of three parameters a, b, c, we can calculate three quantities, x, y, z which depend on a, b, c. It is not, however, possible to calculate a, b, c from given values of x, y, z. Nevertheless we must find somehow what values of a, b, c to choose to make x = y = z = 0, and we know as a practical fact that such a solution exists.

If we make a small change in a and determine its effect on x, y and z, we have obtained approximations of three partial derivatives

$$\frac{\partial x}{\partial a}, \frac{\partial y}{\partial a}, \frac{\partial z}{\partial a}.$$

Similarly by stepping b and c small amounts we can complete the matrix of 9 partial derivatives, or rates, and with these we can calculate changes in a, b, c merely by solving three simultaneous linear equations. Repeated application of this procedure will produce convergence, if

a) there is a unique answer

b) the starting values of the parameters are reasonable

c) high accuracy is maintained in computation, particularly in the solution of the linear equations.

Figure 1 is a computation flow-chart for one way of handling such an iterative procedure; not all necessary details are included, but only enough to indicate the plan.

We begin by initializing the starting values of a, b, c and specifying the small steps Δa, Δb, Δc to be used to get rates in the first pass. We also set switch 1 for the first pass.

The computational block calculates x, y and z, and the test block determines whether the errors are sufficiently small. If so, we exit from the iteration program. Usually the test will fail and control will pass to switch 1.

22

Branch (1) of switch 1 resets itself to branch (2), proceeds to set switch 2, step a, store the last answers, and return to computation. On subsequent passes a new set of rates is figured following each change in a, b, or c.

When three sets of rates have been determined, new values of Δa, Δb, Δc are calculated which are first approximations to a solution. If we had reason to believe the functions were linear it would be desirable to substitute all of these changes at once; however it is generally more practical to apply the changes one at a time as was done with the original steps, obtaining new rates at each stage in preparation for subsequent passes. As a result, we only expect that the test may be passed every third time through the loop.

This procedure, for one, two, three or more variables, has been of tremendous value in our engineering computation; however it must of course be applied intelligently. One particularly troublesome condition is unsuspected discontinuity in the functional relationships; if such discontinuities exist they may be located in advance, and the program modified to prevent stepping across this region.

Multiple solutions have not been troublesome since their existence can usually be predicted; sufficiently good starting values will force the solution we are after.

When the iteration does not converge, and the program has been carefully written to avoid round-off errors, we usually find that there is a physical meaning corresponding to the non-convergence. For example, in certain gear analysis programs, failure to converge indicates either

a) line contact instead of point contact (infinite number of solutions), or

b) contact at or near the cusp of a generated surface (dual solutions very close together).

<u>Use as a Subprogram</u> (FORTRAN)

Iteration of the type described has sufficiently varied usefulness in our work to justify preparing standard subprograms for various numbers of variables. The subprogram approach is also desirable for us since the iteration seldom constitutes the whole problem, but is more likely to represent merely a detail in the main-line computation.

It is desirable to leave as many controls as possible in the hands of the programmer; we therefore set up the subprogram to include only the necessary mechanics of iteration, leaving to the main-line program such items as

a) initializing the variable parameters

b) testing the results

c) taking action on failure of iteration.

We also permit the programmer to specify in his CALL statement the values of the initial steps of the parameters.

It is recommended that the iteration loop be made within the range of a DO statement, to provide easy initializing of the iteration and an automatic count for stopping iteration after a preset number of tries.

Figure 2 shows a program employing the iteration subprogram INTP3A. This program was designed as a short test of the subroutine and includes its own data. Naturally a real program would be much more complex.

Starting values of the parameters A, B, C are set at +25°, -25°, and +25°, and the functional relations are

$$X = \sin A + \sin B = 0 \qquad (A + B = 0)$$

$$Y = \sin (A + C) - \sin 50° = 0 \qquad (A + C = 50°)$$

$$Z = \sin (B + C) + \sin 10° = 0 \qquad (B + C = -10°)$$

It will be seen that the correct final values of A, B, C fulfilling these relations are 30°, -30°, 20°.

Notice that the main calculation and the test lie within the range of the DO loop. The limit of 100 for the index I permits about 33 tries before exiting; this is probably more than will generally be needed.

Failure of the test sends control to the iteration subroutine, which changes the value of either A, B, or C, as indicated by the dashed line in the Figure; control is then returned to the CONTINUE statement so as to restart calculation at the beginning of the DO loop. Passing of the test transfers out of the loop.

Two refinements that we generally employ are not shown in this example. One of these is to follow the CONTINUE statement with a PRINT statement indicating that convergence has failed within the specified count, followed by other appropriate action. The second is a sense-switch-controlled PRINT statement, just before the test, causing print-out of A, B, C, X, Y, Z, I each time through the loop for tracing purposes when convergence is in question.

Structure of the Subprogram

Since the logic of the subprogram is essentially represented by the right-hand part of Figure 1, it is not necessary to show a flow chart. Figure 3 shows the FORTRAN listing as now being used. Switch 1 of Figure 1 is represented by the DO index I, and switch 2 by the index K.

One addition has been made to cover the possibility that sometime during the iteration the calculated increment of one of the parameters may be zero, resulting in division by zero when new rates are being computed.

In reviewing this program at this time I believe I see several ways in which it might be improved. One of these is to use appropriate subscripting (if you have a compiler that can handle subscripting efficiently); another possibility is to call a separate subroutine to solve the set of simultaneous linear equations.

Extension of this method to handle other numbers of variables should present no difficulty; one can also consider the possibility of writing a general procedure for handling n variables.

Comments and Conclusions

Since we have established this method without proof of existence or convergence, it is essential that it be applied with care and common sense. There is no magic here that will produce answers where none exist.

Within these limitations there are many engineering design problems where this multiple iteration procedure is extremely valuable.

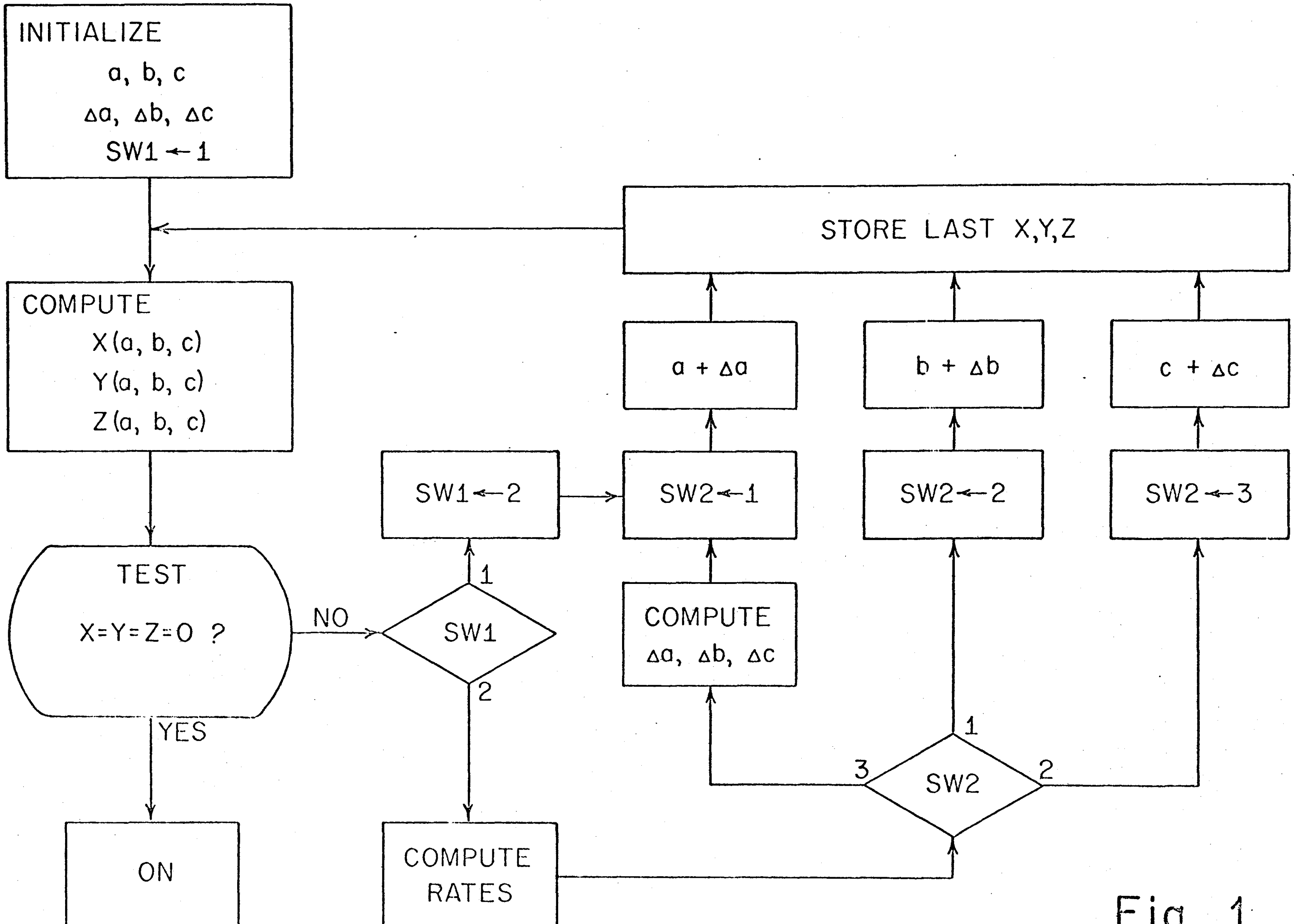

INITIALIZE
a, b, c
Δa, Δb, Δc
SW1 ← 1
STORE LAST X,Y,Z
COMPUTE
X(a, b, c)
Y(a, b, c)
Z(a, b, c)
a + Δa
b + Δb
c + Δc
SW1 ← 2
SW2 ← 1
SW2 ← 2
SW2 ← 3
TEST
X=Y=Z=O ?
NO
SW1
1
2
COMPUTE
Δa, Δb, Δc
YES
SW2
1
2
3
ON
COMPUTE
RATES

Fig. 1

```
C     INTP3A TEST
      ADEG=25.
      A=ADEG/57.2958
      C=A
      B=-A
      DO 1 I=1,100,1
      X=SINF(A)+SINF(B)
      Y=SINF(A+C)-0.76604
      Z=SINF(B+C)+0.17365
      IF(ABSF(X)+ABSF(Y)+ABSF(Z)-.00002)2,2,3
    3 CALL INTP3A(A,B,C,X,Y,Z,I,.01,.01,.01)
    1 CONTINUE
    2 ADEG=A*57.2958
      BDEG=B*57.2958
      CDEG=C*57.2958
      PRINT 4,ADEG,BDEG,CDEG,I
    4 FORMAT(4X,3(F7.4,2X),I3)
      STOP 11111
      END
```

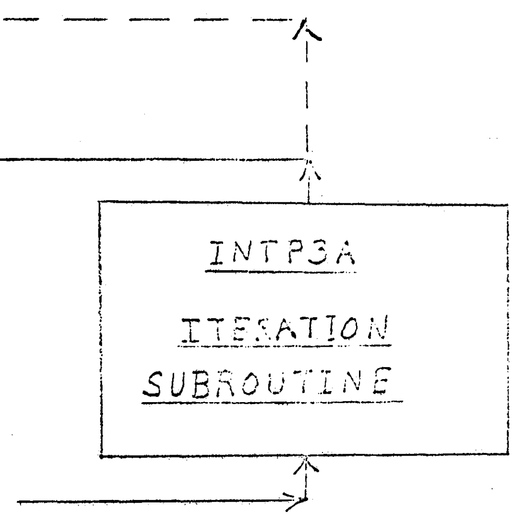


Fig. 2

4.12.70

```
      SUBROUTINE INTP3A(A,B,C,X,Y,Z,I,ASTEP,BSTEP,CSTEP)
      IF(I-1)1,1,2
    1 DA=ASTEP
      DB=BSTEP
      DC=CSTEP
    3 K=1
      A=A+DA
    4 XO=X
      YO=Y
      ZO=Z
      RETURN
    2 GO TO(5,8,11),K
    5 IF(DA)6,7,6
    6 DXDA=(X-XO)/DA
      DYDA=(Y-YO)/DA
      DZDA=(Z-ZO)/DA
    7 K=2
      B=B+DB
      GO TO 4
    8 IF(DB)9,10,9
    9 DXDB=(X-XO)/DB
      DYDB=(Y-YO)/DB
      DZDB=(Z-ZO)/DB
   10 K=3
      C=C+DC
      GO TO 4
   11 IF(DC)12,13,12
   12 DXDC=(X-XO)/DC
      DYDC=(Y-YO)/DC
      DZDC=(Z-ZO)/DC
   13 D1=DYDB*DZDC-DYDC*DZDB
      D2=DXDB*DZDC-DXDC*DZDB
      D3=DXDB*DYDC-DXDC*DYDB
      DEN=DXDA*D1-DYDA*D2+DZDA*D3
      DA=(Y*D2-X*D1-Z*D3)/DEN
      DB=(DXDA*(Z*DYDC-Y*DZDC)-DYDA*(Z*DXDC-X*DZDC)
     1   +DZDA*(Y*DXDC-X*DYDC))/DEN
      DC=(DXDA*(Y*DZDB-Z*DYDB)-DYDA*(X*DZDB-Z*DXDB)
     1   +DZDA*(X*DYDB-Y*DXDB))/DEN
      GO TO 3
      END
```

Fig.3

4.12.11

"A Distribution Feeder and Substation Load Forecasting System"

by

J. C. Hubbard

Baltimore Gas and Electric Company

Prepared for the Eastern Region 1620 Users Group

Spring Meeting

Washington, D. C., May 6-8, 1964

A Distribution Feeder and Substation Load Forecasting System

Introduction

In the Baltimore Gas and Electric Company, peak electric loads are forecast for the succeeding five years. A separate five-year forecast is made for summer and winter, each.

For purposes of forecasting, the Baltimore System consists of a group of master substations whose peak loads are related to the System peak load by a factor which includes losses and diversity. The typical master supplies several customer and distribution substations over subtransmission circuits; masters may or may not supply local distribution load. A substation consists of one or more transformers, each of which supplies one or more feeders. Where a substation includes two or more transformers, there is generally more than one supply circuit.

Metering

Substations and master substations have integrating watt and var meters, integrating watt and graphic var, or type RI (watt and volt-ampere) meters. All integrations are for a clock-hour interval. A number of magnetic tape recorders, three to a set, recently have been installed - to record watts, leading vars and lagging vars. For this type installation, a program has been developed that selects and tabulates the daily clock-hour mva peak and its components and the daily mw peak and its corresponding mvar.

Feeders at attended substations have indicating ammeters, one to a phase, whose indications are recorded hourly. Many feeders at unattended stations have peak demand ammeters, one to a phase; the peak indicators are read at intervals, generally of two weeks, and then reset. A few feeders have graphic averaged three-phase ampere hourly indications. Other feeders are metered in pairs, and some are without metering; in these special cases, spot checks are used to get an estimate of their loads.

Indicating and graphic voltmeters are located on various buses around the system.

Purpose of programs

Forecasts are required for all feeders, all distribution substations, some customer stations, and all master substations. The Baltimore System forecast is made separately. A check is made on the diversity between the Baltimore System and master substation peaks to determine if it is satisfactory.

There follows an outline of a set of fortran programs developed in connection with the feeder and station forecasts.

Program 1: Peak load data

The first program is concerned with the interpretation of actual load data from the most recent season. Its objects are to calculate the seasonal mva peak and its mw and mvar components for each feeder and substation, and to calculate the diversity factor between each station and its associated feeders.

Inputs for each item include actual peak load data, the mvar of capacitors in service at the time of the peak, the previous year's corresponding seasonal mw peak component, and the mw evaluation of load transfers since the previous year.

In this and other programs, capacitors connected on distribution feeders and on substation buses are treated as mvar sources and their outputs are added algebraically to feeder or station mvar to determine load mvar. Calculations concerned with power factors, growth rates, and diversity factors are always related to load mvar.

This program is constructed to calculate any of the missing items of mw, mvar and mva when supplied with one of the following three sets of data: mw and mvar; mva and mw; amperes, volts and power factor. In the case where a station's input data consists of mva and mw, a test load power factor is included in order to provide a way for the machine to test whether the mvar which it calculates is leading or lagging. This power factor is based on a separate history of information relative to the station's load power factors at the time of its summer and winter peaks.

Outputs include mva, mw, mvar and load mvar, power factor, and a year-to-year percentage load growth based on a constant load area. Mw and mvar diversity factors between the station and the feeders which it supplies are produced for each substation. Test power factors are replaced in output by the actual computed power factors.

Program 2: Weather adjustments

A part of the variation in actual peak loads from year to year is caused by differences in weather. Weather effects tend to obscure basic patterns of load change. Much of the variation can be eliminated by the reduction of actual peaks to peaks under normal peak load temperature conditions. The adjusted figures reveal more clearly the year-to-year growth which is occurring. In our system, weather adjustments are determined by means of a linear regression program.

The program is applied to substations only. It is based on information for a twenty-week period in summer, a ten-week period in winter. Stations are divided into two groups, those that normally peak in daytime and those that normally peak in nighttime. Inputs consist of two sets of weekly maximum temperature indices, one each for day and night, and weekly maximum mw loads for each station. Output for each station includes the regression constants of its load on the temperature index, the average mw peak and average temperature index, the normal temperature index and the corresponding mw load computed by the program. The ratio of the maximum weekly peak load to the load at the

normal temperature index is also produced; this becomes a gauge of the normality of the season's weather, which is either extreme, normal or mild, according to whether the ratio is high, normal or low. Produced, too, is the computed load at a very high temperature index; this makes available a distant point for assisting in locating a line for visualization. The slope of the line indicates whether and how much the station is temperature-sensitive.

Program 3: Feeder and substation forecasting

The major program has inputs including actual peak load data, reference mw and load mvar to be used as the current year's base load, rating of capacitors assumed to be in service during the period forecast, growth rates, mw and mvar diversity factors, and various adjustments.

Items to be forecast are divided into several classifications: feeders, single-feeder substations, multi-feeder substations, customer substations, loads to be included in diversity calculations but not forecast per se, and master substations. Each classification has its own path through the program. Within limits, the inputs are intermingled. The identification fields are used for discrimination between classes.

The reference mw and load mvar are selected by the forecaster after due attention is paid the actual peak, the estimated peak under normal weather conditions, previous forecasts, and whatever additional information is deemed relevant. These references become the effective current year's peak load on which the forecast is based. Growth rates are entered for the first and fifth years of the forecast. The machine will apply constant or smoothly-varying rates to the consecutive annual peaks, beginning with the references, according to whether the specified growth rates are the same or different. A zero growth rate is acceptable.

Separate mw and mvar diversity factors are provided for because, among other reasons, losses sometimes are involved in the calculations. Diversity factors are entered only for multi-feeder substations and master substations.

Adjustments to the reference-rate process are necessary when, in the judgment of the forecaster, any change is likely to occur within a load area which will affect the continuity of normal growth. Such changes take many forms. For example, an abnormally large number of houses of the same type as now exist within an area may be scheduled for completion within the year. The forecaster, therefore, may decide that it is necessary to add an abnormal increment of load for next season, yet allow succeeding years to change at the area's normal rate. Or perhaps a commercial area is being developed which is likely to add large loads for the next two years but little additional load afterwards. The forecaster may decide in this case that he will compute the entire five-year forecast at the normal growth rate without including the new load and make an incremental adjustment to each of the five years in order to account for the new load. To allow for many possibilities, the program permits two types of adjustments, one for loads which will be included in, the other for loads which will be excluded from the application of growth rates. Either or both types can be included in any portion or all years of the forecast. Therefore a wide latitude is available to the forecaster.

For each item a complete set of input data can be entered on from two to four cards, according to whether and what type of adjustments are applied. The first card for each item includes a field for the variable indicating how many cards are to be used. Input data are assembled by masters; within the master, by stations. Items are processed one at a time. As each card is read, checks are made on the correctness of its order and on whether it is the final data card for the item. All identification fields on each card are checked, to prevent the mixing of data.

Each item proceeds by its appropriate path to its forecast peak loads. On the way, all distribution substations and customer stations set a program switch to positive values, and cumulate their peak load components. The switch is set to zero before the next item's data are read. Individual feeders of a multi-feeder station cumulate their mw peak loads; the switch is left to have the value "zero".

In the case of a multi-feeder station, the forecast is computed first from reference and growth rate. Later the program switch initiates the computation of certain of its forecast peak loads by means of the summation of its feeder peaks and its diversity factor. The results are compared. Where the difference is outside of a preset allowable percentage difference, the fact is so noted in output; the note serves as an indication to the forecaster that further study of the station and its feeders is called for.

For a master substation, the forecast is computed by use of its diversity factors and the summation of the peaks of the stations which it supplies. The program switch is set negative.

The initialization procedure preceding each group of input cards is determined by the value (positive, zero, negative) of the program switch.

Output for each item includes the actual load and the forecast load in mva and components for all five years. The mvar of capacitors considered to be in service during the period of the forecast appears also. The mvar component of the forecast is that which the feeder or station must carry. This output, by means of an SPS format modification program, becomes the five-year forecast for feeders and substations.

Program 4: The revised forecast

In practice, the initial forecasts are made strictly on the basis of change within a given feeder or station load area. They are released for study in connection with system planning and development. The studies result in a schedule of load transfers and capacitor installations prior to each season.

The planning schedule is inserted into a modified forecasting program. This has all of the inputs of the earlier program; provision is made also for including the capacitor changes and the mw and mvar components of load transfers for each of the first three forecast years.

The output of this program is identical in format with that of the initial forecast. The new figures reveal the effects of the planning schedule, and assist in planning for the intermediate-term. Full annotation describes how the revised forecasts differ from the original.

J. C. Hubbard
Baltimore Gas and Electric Company
May 5, 1964

LIST OF FIGURES

Program 1

Program 2

Program 3

Program 4

FEEDER AND TRANSFORMER CALCS.
CRW + JCH, ENGINEERING STAT.
5-31-62, (MOD. 6-14-62), DWG. REV.

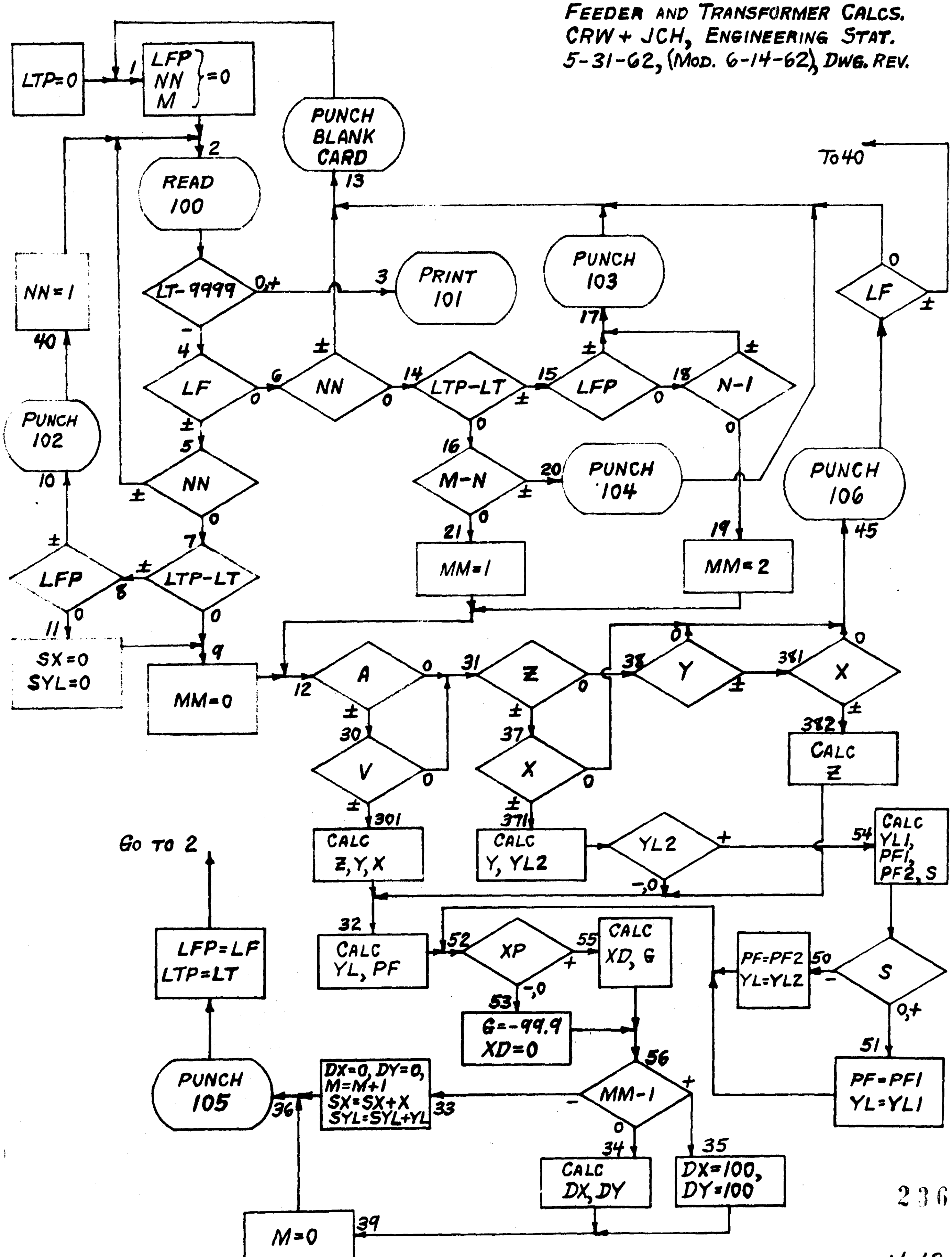


FIG. 1

Feeder and Transformer Calculations

Input - Output Lists

	Item	Input	Output
LF	Feeder	✓	✓
LT	Substation	✓	✓
N	No. of feeders	✓	✓
LY	Year, season	✓	✓
XA	mwhr adjustment	✓	
XP	mwhr peak, prev. year	✓	
YC	mvar caps.	✓	
V	voltage, kv	✓	
A	current, amperes	✓	
Z	mvahr	✓	✓
X	mwhr	✓	✓
Y	mvarhr	✓	✓
PF	power factor	✓	✓
YL	mvarhr, load		✓
XD	adjusted mwhr		✓
G	growth rate		✓
DX	mwhr diversity factor		✓
DY	mvarhr diversity factor		✓

FIG. 2

FEEDER AND TRANSFORMER DATA SHEET

#219

(6)

CEDAR PARK

FEEDER 1 5	TRANSF. 9 10	N 11	YEAR-SEAS 14 15	ADJUSTMENT MWHR 20 21	PREV. YR. PEAK MWHR 25 27	CAPS. AT PEAK TIME MVAR 30 34 35	VOLTAGE KV 40	CURRENT AMPERES 45	EQUIPMENT PEAK MVAHR 50 52	ACTIVE COMPONENT MWHR 55 59 60	REACT. COMPONENT MVARHR 65 66	LOAD POWER FACTOR PERCENT 70 71 72
	52.0	1	63.1	0	18.0	15.0				22.0	-1.5	
	56.9	1	63.1	0	15.9	12.0			17.8	17.8		.83
	60.0	1	63.1	0	4.5	0				2.4	1.8	
	61.0	1	63.1	0	12.4	6.0				12.8	2.2	
4715	76.0		63.1	0	10.1	6.9	25.3	431				.86
4716	76.0		63.1	0	7.1	6.9	25.3	336				.86
	76.0	2	63.1	0	16.8	13.8				18.8	-2.7	
4798	76.6		63.1	.8	4.7	3.0	25.3	150				.86
4799	76.6		63.1	0	10.4	3.0	25.3	439				.86
	76.6	2	63.1	.8	14.7	6.0				13.9	2.8	
4791	79.0		63.1	3.0	7.1	0	25.3	210				.83
4792	79.0		63.1	0	7.1	7.5	25.3	352				.83
	79.0	2	63.1	3.0	13.8	7.5				12.5	.8	
	81.5	1	63.1	0	7.0	6.0				10.0	-1.8	
4771	81.6		63.1	7.5	18.3	6.0	25.3	515				.84
4772	81.6		63.1	11.8	15.6	12.0	25.3	538				.84
4773	81.6		63.1	0	0	7.5	25.3	700				.84
	81.6	3	63.1	0	33.8	25.5				42.0	1.5	
4718	88.6		63.1	0	9.3	3.0	25.3	194				.81
4719	88.6		63.1	0	7.3	6.0	25.3	245				.81
	88.6	2	63.1	0	16.3	7.0				10.7	-1.4	

(3)

FIG. 3

Fdr No.	Sta No.	No. of Fdrs	Yr., Seas	Fdr. or Sta. Peak Mva	Mw	Mvar	Load PF	Mvar	Adj. Mw	% Growth	D. F. Mw	Mvar
0	8	1	632	1.15	1.15	-.05	87.	.64	1.15	1.7	100.	100.
0	136	1	632	1.19	1.19	-.15	93.	.45	1.19	4.3	100.	100.
0	56	1	632	1.44	1.44	-.09	87.	.81	1.69	3.0	100.	100.
0	1076	1	632	.89	.81	.37	90.	.37	.81	2.5	100.	100.
0	2076	1	632	.81	.75	-.31	94.	.26	.75	-3.8	100.	100.
4780	196	0	632	1.69	1.58	.61	83.	1.06	1.58	8.2	.	.
4783	196	0	632	1.03	.97	-.34	87.	.55	.97	7.4	.	.
4784	196	0	632	.69	.69	.01	83.	.46	1.30	14.5	.	.
4786	196	0	632	1.46	1.46	-.06	83.	.98	1.46	3.2	.	.
0	196	4	632	4.40	4.40	-.10	83.	2.95	7.99	6.5	107.	104.
4796	232	0	632	.34	.32	.14	91.	.14	.32	14.2	.	.
4797	232	0	632	.86	.59	-.63	90.	.27	.59	1.7	.	.
0	232	2	632	1.00	.88	-.49	90.	.41	.88	4.7	103.	100.
0	300	1	632	.88	.85	.26	86.	.49	.85	3.6	100.	100.
0	302	1	632	1.34	1.06	-.82	84.	.68	1.06	-6.1	100.	100.
0	1304	1	632	1.35	1.34	.23	89.	.68	1.34	13.5	100.	100.
0	2304	1	632	4.34	4.30	-.60	94.	1.50	4.30	16.2	100.	100.

FIG. 4

Program #334

Variation of peak load with weather

Eng Stat. 9-3-63 J.C.H.

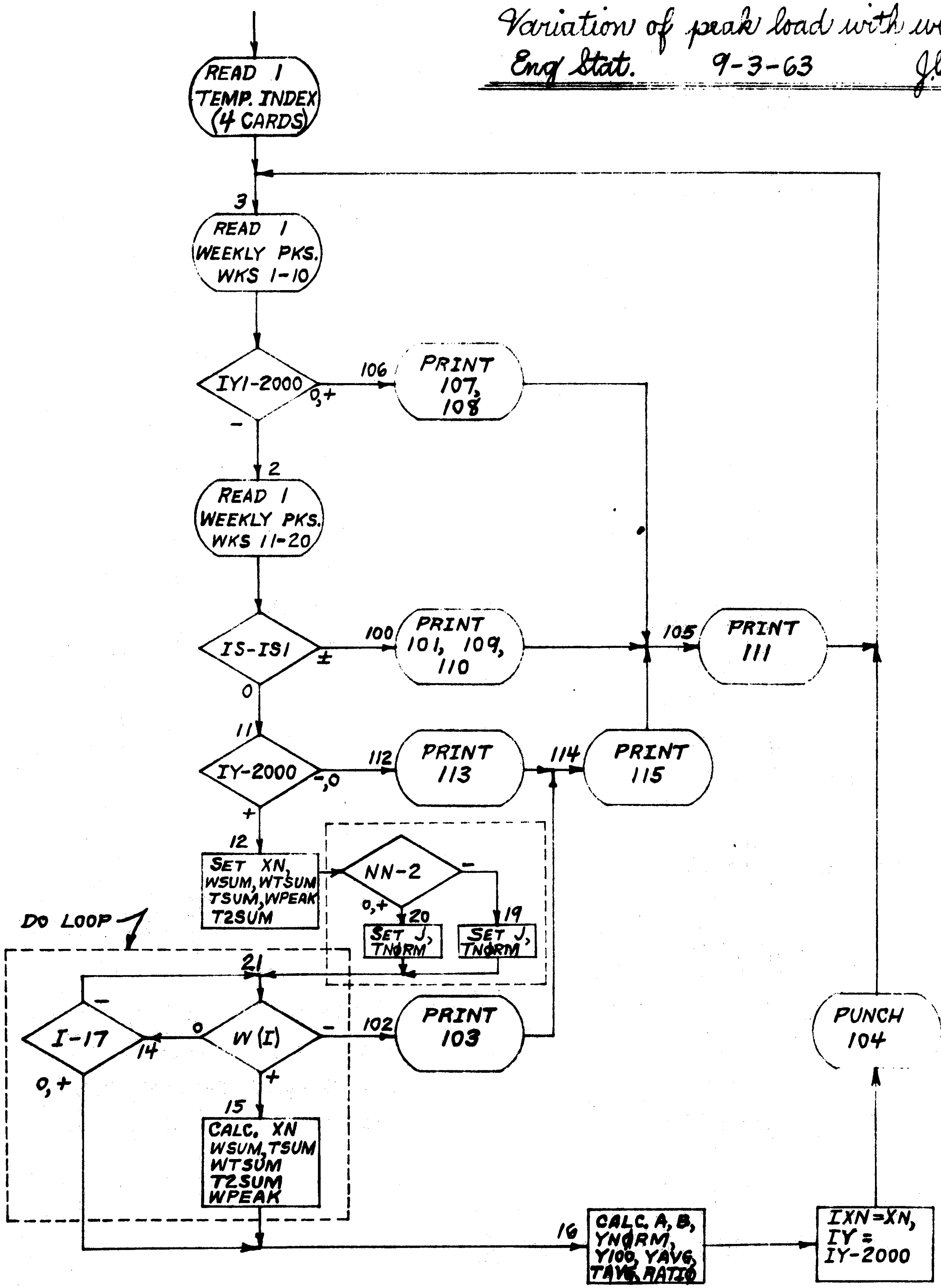


FIG. 5

240

4.13.10

Summer

For temp. index cards, IS = 0

Cols 5-8, designated "Yr., season" (IY):

col	5	6	7	8
	1	6	1	1

$\underbrace{6\ 1}_{\text{Year}}$

Col 5: 1 for weeks 1-10
2 " " 11-20

Col 8: Season: 1 for summer
2 " winter

Before punching, let IY = IY - 2000

NN = 1 for night peak sta., 2 for day peak sta.

T(I): Pk. temp. indexes, weeks 1-20 [I = 1 to 20, night ; 21-40 day]

W(I): " mw loads, " 1-20

IXN = XN = No. of weeks in calcs.

WSUM = Sum of mw loads

TSUM = " " temp. indexes

$WTSUM = \sum_{n=1}^{20} W_n T_n$

$T2SUM = \sum_{n=1}^{20} T_n T_n$

WPEAK = Peak mw load

Eq: $Y = A + BT$

A, B: constants of eq.

YAVG = WSUM/XN

TNØRM = temp index normal for period = 37 for night station, 47 for day peak sta.

YNØRM = mw load at TNØRM

RATIØ = peak mw load/YNØRM

Y100 = mw load at T=100

FIG. 6

#341

VARIATION OF PEAK LOAD WITH WEATHER

~~TRANS~~ SUB ~~AND~~ NO. ~~D SUB~~	YR. AND SEA.	TI-LOAD REGRESSION EQUATION CONSTANTS A	B	TEMPERATURE INDEX AVG.	NORM.	LOAD AT AVG. TI	NORM TI	100 TI	PK. TO ~~NORM~~ NORM. ~~RATIO~~ RATIO	NO. WKS ~~OCC~~
268	632	.172039E+03	.295987E-00	54.6	40.	188.2	183.8	201.6	1.082	5
296	632	.100171E+03	.249677E-00	45.8	40.	111.6	110.1	125.1	1.116	8
1389	632	.341257E+03	.136376E+01	45.2	40.	402.9	395.8	477.6	1.162	10
2628	632	.191679E+03	.272586E-00	45.2	40.	204.0	202.5	218.9	1.076	10
676	632	.237973E+03	.383324E-00	45.2	40.	255.3	253.3	276.3	1.065	10
759	632	.287007E+03	.610460E-00	45.2	40.	314.6	311.4	348.0	1.098	10
3874	632	.225404E+02	.433526E-01	45.2	40.	24.5	24.2	26.8	1.071	10

FIG. 7

2/2

4.13.12

Feeder and station peak load forecasting

	Inputs	Outputs
LF	Feeder identification	LF
LT	Transf. "	LT
LM	Master "	
LY	Year and season	
XR	Reference mw	
YR	" mvar	
CE	Caps in during forecast period	CE
ZA	Actual mva peak	ZA
XA	" mw comp. of ZA	XA
YA	" mvar " " ZA	YA
CAP	Caps in during actual peak	CAP
NM	Month } of actual peak	NM
ND	Day } of actual peak	ND
NH	Hour } of actual peak	NH
NN	No. of cards required	$(ZE)_K$, K=1 to 5 Mva peak forecast } 1st to 5th years
JJ	Card no.	$(XE)_K$, K=1 to 5 Mw comp of $(ZE)_K$ } 1st to 5th years
G(1)	Growth rate, 1st yr. forecast	$(YE)_K$, K=1 to 5 Mvar " " $(ZE)_K$ } 1st to 5th years
G(5)	" " , 5th " "	
DX(1)	Mw diversity factor, 1st yr.	
DX(5)	" " " , 5th yr.	

DY(1) Mvar " " , 1st yr.

DY(5) " " " , 5th yr.

X_K, K=1 to 5 Mw adjs included in growth rate calcs, 1st to 5th years

X_K, K=6 to 10 Mvar " " " " " " , "

V_K, K=1 to 5 Mw " excluded from " " " , "

V_K, K=6 to 10 Mvar " " " " " " , "

C_K, K=1 to 3 Caps added } Planning schedule, 1st to 3rd years

P_K, K=1 to 3 Mw comp. of load transfers } Planning schedule, 1st to 3rd years

Q_K, K=1 to 3 Mvar " " " " } Planning schedule, 1st to 3rd years

2 13 J.e.H. 4.13.13

FIG. 8

(M-1791)-2B-20

PEAK LOAD FORECAST DATA

Engineering Statistics

GUNPOWDER (8)

63 YEAR   S SEASON

WHITE MARSH

F	T	M	Y	S	R-MW	R-MVAR	CE	A-MVA	A-MW	A-MVAR	A-CAP	M	D	H	N	
4971	883	394	631		144	140	90	153	144	140	90				4	1

F	T	M	Y	S	G1	G5	DM1	DM5	DR1	DR5	
4971	883	394	631		00	00					2

F	T	M	Y	S	M1	R1	M2	R2	M3	R3	M4	R4	M5	R5		
4971	883	394	631													3
4971	883	394	631		15	15	20	20	25	25	30	30	35	35	100+1100 for incr. in 564	4

WINTERS RUN FEEDERS

F	T	M	Y	S	R-MW	R-MVAR	CE	A-MVA	A-MW	A-MVAR	A-CAP	M	D	H	N	
4615	910		631		94	65	60	94	94	00	60				2	1

F	T	M	Y	S	G1	G5	DM1	DM5	DR1	DR5	
4615	910		631		1000	1000					2

F	T	M	Y	S	M1	R1	M2	R2	M3	R3	M4	R4	M5	R5	
															3
															4

F	T	M	Y	S	R-MW	R-MVAR	CE	A-MVA	A-MW	A-MVAR	A-CAP	M	D	H	N	
4616	910		631		88	61	90	93	88	27	30				2	1

F	T	M	Y	S	G1	G5	DM1	DM5	DR1	DR5	
4616	910		631		1000	1000					2

F	T	M	Y	S	M1	R1	M2	R2	M3	R3	M4	R4	M5	R5	
															3
															4

F	T	M	Y	S	R-MW	R-MVAR	CE	A-MVA	A-MW	A-MVAR	A-CAP	M	D	H	N	
4617	910		631		119	90	60	122	119	26	60				3	1

F	T	M	Y	S	G1	G5	DM1	DM5	DR1	DR5	
4617	910		631		1000	1000					2

F	T	M	Y	S	M1	R1	M2	R2	M3	R3	M4	R4	M5	R5		
4617	910		631		-35	-25									to Churchv. 7052 on 10/8/63	3
																4

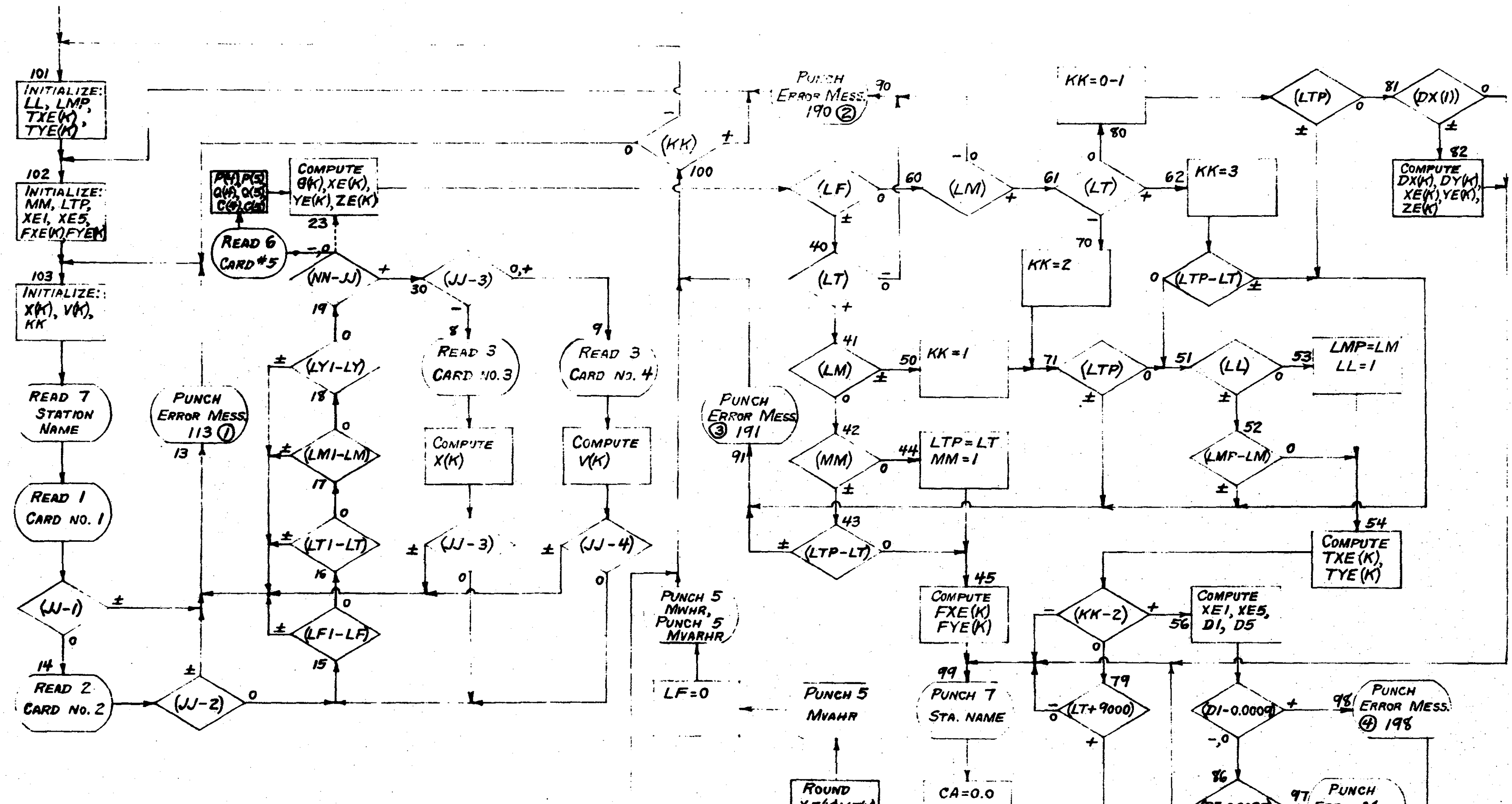
WINTERS RUN SUBSTATION

F	T	M	Y	S	R-MW	R-MVAR	CE	A-MVA	A-MW	A-MVAR	A-CAP	M	D	H	N	
	910	394	631		281	210	210	280	277	42	150	7	19	14	3	1

F	T	M	Y	S	G1	G5	DM1	DM5	DR1	DR5	
	910	394	631		1000	1000	10700	10700	10300	10300	2

F	T	M	Y	S	M1	R1	M2	R2	M3	R3	M4	R4	M5	R5		
	910	394	631		-35	-25									to Churchv. 13kv on 10/8/63	3
																4

FIG. 9



PEAK LOAD FORECASTING
ENG. STAT. 11-62

Modified to include planned transfers + cap. installations, 3/13/63

FIG. 10

ACTUAL AND FORECAST ONE-HOUR PEAK LOADS
ADJUSTED FOR PLANNED FEEDER AND CAPACITOR CHANGES
60 CYCLE FEEDERS AND SUBSTATIONS
SUMMER, 1963 TO 1968
MVAHR, MWHR & MVARHR

STATION, FEEDER NUMBER	CAPS MVARHR	ACTUAL 1963 TIME	ACTUAL 1963 LOAD	FORECAST 1964	FORECAST 1965	FORECAST 1966	FORECAST 1967	FORECAST 1968
SION HILL								
4939		7/18	1.26	1.06	1.17	1.29	1.43	1.59
	.48	22	1.22	1.06	1.17	1.29	1.42	1.56
	.48		.31	-.03	.03	.12	.20	.30
VAN BIBBER FEEDERS								
7053			3.17	4.09	4.53	5.01	5.54	6.17
	1.50		3.17	4.05	4.52	5.01	5.54	6.16
	2.40		-.10	-.55	-.35	-.12	.12	.39
7054			1.94	2.54	2.71	2.91	3.15	3.42
	1.20		1.90	2.29	2.52	2.77	3.05	3.35
	2.10		-.40	-1.10	-1.00	-.89	-.77	-.64
VAN BIBBER SUBSTATION								
		7/18	4.74	6.19	6.75	7.38	8.09	8.92
	2.70	22	4.70	5.93	6.59	7.29	8.05	8.91
	4.50		-.60	-1.77	-1.48	-1.15	-.80	-.41
WEST ABERDEEN								
4738		7/18	2.22	2.45	2.72	3.01	3.35	3.73
	1.53	22	2.22	2.45	2.71	3.00	3.31	3.66
	1.53		-.15	-.00	.16	.33	.53	.74
HARFORD MASTER								
		7/19	40.00	42.96	46.50	50.42	54.71	59.44
		15	37.20	41.66	45.10	48.42	52.02	55.94
	19.64		14.70	10.50	11.35	14.03	16.93	20.07

ARSENAL 4913 - 600 KVAR CAPACITORS ADDED IN 1964.
BELCAMP INCLUDES A LOAD INCREASE IN 1964.
CHESAPEAKE CARDENS 4918 - 600 KVAR CAPACITORS ADDED IN 1964.
CHURCHVILLE 7051 - 300+J140 FROM SION HILL 4939 IN 1964.
CHURCHVILLE 7052 IS A NEW ITEM AND WAS PLACED IN SERVICE AFTER THE SUMMER PEAK OCCURRED.ITS LOAD CONSISTS OF ALL CHURCHVILLE 4KV AND PORTIONS OF FOREST HILL 4912 AND WINTERS RUN 4617.FORECAST INCLUDES NEW LOAD FOR HARFORD JUNIOR COLLEGE IN 1964, 1800 KVAR CAPACITORS ADDED IN 1964.
FOREST HILL WAS RELIEVED BY CHURCHVILLE 7052 AFTER THE SUMMER PEAK OCCURRED 300 KVAR CAPACITORS ADDED IN 1965.
HAMILTON COURT 4927 - 300 KVAR CAPACITORS ADDED IN 1964 AND 300 KVAR ADDED IN 1965.
HAVRE DE GRACE 4619 - INCLUDES LOAD INCREASE FOR HARFORD MEMORIAL HOSPITAL IN 1964, 300 KVAR CAPACITORS TRANSFERRED TO 4625 IN 1964 AND 750 KVAR ADDED IN 1965.
HAVRE DE GRACE 4625 - EXCLUDES LOAD INCREASE FOR HARFORD MEMORIAL HOSPITAL IN 1964, 300 KVAR CAPACITORS TRANSFERRED FROM 4619 IN 1964 AND 750 KVAR ADDED IN 1965.
SION HILL 4939 - 300 KVAR CAPACITORS ADDED AND 300+J140 TRANSFERRED TO

FIG. 11

PROCESSING O-D SURVEY DATA

ON A SMALL COMPUTER

A. D. Stasi and M. B. Lipetz

Edwards and Kelcey, Inc.

247

4.14.C

PROCESSING O-D SURVEY DATA ON
A SMALL COMPUTER

By
A. D. Stasi and M. B. Lipetz

The title of this paper is not intended to lead anyone to believe that Edwards and Kelcey has found a way to process a complex comprehensive transportation study on its bare minimum Model I, paper tape, 20K IBM 1620. Although there are programs available to do this on the 1620, they all require 60K and are limited in the number of traffic zones they can handle. Rather I would like to describe to you some of the things we have done in the way of preparing the data for the larger computers such as the 7074 and 7094. Most of us do not have ready nor easy access to these larger computers which are used with the Bureau of Public Road's battery of Traffic Forecasting and Assignment Programs for comprehensive traffic studies. Consequently, when we do get on them, we must use them most efficiently. Being certain that our data is as correct as it can possibly be and in the proper form for processing is, therefore, of prime importance. Editing for correctness and re-formatting of the original data is well within the capabilities of the small computer, i.e. a 20-60K IBM 1620. We have used all configurations of 1620's by the way our

own Model I - 20K with paper tape I/O; a Model I - 20K with a slow card reader I/O and a souped-up Model I 60K with paper tape, card reader and 6 magnetic tape drives. This last configuration was available at Rutgers University until about a month ago -- we were sorry to see it go -- it cost us only $50/hr!!

To get down to specifics, we have written editing programs, volume factoring programs, zone conversion programs and re-formatting programs.

The remainder of this paper will be devoted to an explanation of the purpose and some of the mechanics of these programs.

I. Editing Program

O-D surveys are generally based on a sampling of the travelling public which is later expanded to represent the entire area being studied. It is, therefore, essential that the maximum number of survey trip records be maintained in the file. Towards this end, it is absolutely necessary that all trip records be edited for validity. The specific requirements for the editing program are dictated by the particular trip record being edited (Roadside Interview, Home, Truck, Taxi, Tourist, Non-Tourist, etc.).

Our program simply checks various items of data, both single characters and fields, on each card read in. Several methods are used

to accomplish the checking. The simplest is a direct comparison between the item of data and its parameters for each item. However, utilization of this method will slow the program down considerably. A faster method involves the use of binary decision tables. Here the item of data itself is used to compute a location within a corresponding table. The value of the digit (zero or one) found at this location indicates whether the data is acceptable or not.

Another check incorporated into the program is the logic or cross data check. This is done to insure that several interrelated items do not contradict each other and wreak havoc in subsequent programs. For example, a trip entering the survey area at its western boundary with an indicated travel direction of West, contains an obvious inconsistency. A zone of origin that does not lie within the county of origin also indicates an inconsistency. The only option available to the programmer, and hence to the program, is to designate both items of data as errors and leave the decision as to which is right and which is wrong to the traffic engineer.

Error indication posed a problem. How could the program clearly indicate which items of data are wrong and yet not destroy the original

value of the data? The answer was to punch two error cards. The first card is simply a duplicate of the input card. The second card is also a duplicate, however, it contains asterisks in all the incorrect data fields. Printing the error cards on a 407 produces a listing from which corrections may easily be made. The two types of error cards are tied together by a sequence number assigned by the program to each input card. This number permits each output record to be uniquely identifiable in all future processing. It also provides us with a relatively simple way to insert corrected data cards back into their original position in the file.

To aid in the determination of the necessity for error correction, the program categorizes and summarizes the types of errors. Should the decision be made to correct the errors, these summaries indicate which types of e rrors may economically be corrected by the program and which can only be done manually. The program also keeps a count of the total number of cards read in and the total in error.

Ideally, the program should completely separate the acceptable data from unacceptable data by punching or writing two files. This eliminates the problem of separating the data at some future time by makeshift means, facilitates the application of corrections by a program and maintains the order of the original input file. On computers

essentially equipped with only one output device, the 1620 typewriter being much too slow to be seriously considered for high-volume applications, the production of two output files becomes a problem. An easy solution is an identifying punch in an available card column and subsequent separation on a sorter. The best solution is to utilize a 1620 with both punched card and magnetic tape input/output. The acceptable data file could then be written on the tape and the error file punched on cards. This has the added advantage of placing the acceptable data on the proper medium for input to the larger computer at relatively little extra cost.

A program for editing 70,000 trip records has been satisfactorily run on a 20K 1620 with card reader, and one for editing 500,000 trip records on the souped-up 1620 previously described.

As is evident from this description of our Editing Program, quite a bit was packed into it. There is need for more information, however, such as what are the numbers of correct and error records at each interview station. This information is valuable and often will aid in pinning down the location of large blocks of errors.

A station volume accumulation program was, therefore, written to summarize, by interview station, the output of the editing program. So used, it produces an up-to-the-run tabulation of the number of acceptable trips through each interview station. By using the error cards as input, a tabulation of unacceptable trips can also be produced. These two figures for a particular station provide an indication of how many, if any, unacceptable trip records must be corrected to increase the sample to a satisfactory level.

Part one of the program reads and accumulates the number of trips for each station. The address of the proper accumulator is computed directly from the station number, thereby eliminating the need to search a table for each record. The second part tests each accumulator for zero and, if not zero, punches a card containing the corresponding station number and the number of trips through that station. After all the non-zero volume stations have been punched, the program types the total number of input records and the number of input records which did not have a logical station number.

After both the editing and summary runs are complete, a decision must be made to either discard or attempt to correct the unacceptable trip records. It has been our experience that these records usually comprise a significant portion, between 5 and 10 per cent, of the total number of trip records. The usual decision, therefore, is to correct all the records that are correctable by a program and discard the remainder. Many errors seem to occur in groups. One interviewer at a particular station, for example, consistently entered the wrong hour period on his interview forms. All the trips recorded by this man at that station on that day contained the wrong time. This can be corrected by scanning the other records for that station and determining during what part of that day the station was being operated. Errors such as this lend themselves well to correction by a program. These corrections normally increase the acceptable portion of the interview sample to approximately 98 per cent.

The correction or FUDGE program, as it has come to be known, is a conglomerate. It contains the appropriate routines from the editing program that originally discovered the particular errors

to be corrected. In a sense, the errors are re-discovered by the program. This time, however, instead of simply indicating the errors the program applies corrections and punches the corrected card. A card is punched for each error card read whether corrections are made or not.

After the unacceptable trip file has been essentially corrected, it is run back through the editing program. This is necessary to provide a final check and to weed out the relatively small group of records that contain items of data which were not corrected and are, therefore, still unacceptable. These records now become the final unacceptable trip file. The acceptable trip records are combined and tabulated by the station volume accumulation program. A tabulation is also made of the unacceptable records. Processing of these records is now complete. They are deleted from the survey data at this point.

As previously mentioned, there is presently available from the BPR a package of programs related to O-D surveys. These programs are all written for a definite purpose and have definite requirements relative to their use. This is not to say, however, that we cannot fool the computer into thinking it is doing one thing

while we are actually making it do something else. For example, a program written to prepare tables of trips taking place between various zones based on various trip purposes has been used to prepare tables of trips between specific zones based on the interview station rather than the trip purpose. This was accomplished by placing a "dummy code" in a specific field and defining this field to the program as the purpose field. Not really difficult and not really fancy, but a simple 1620 program did the job for us and a simple 1620 placed the factor. A not so simple 7094 produced the tables which would have had to be produced manually otherwise.

Another problem frequently encountered in O-D surveys has to do with the numbering system used to designate the survey zone. These survey zones frequently require in excess of five digits to uniquely identify a trip end. These survey zones are later combined and must be reduced to a 5-digit maximum code in order to be acceptable to the BPR programs. Again this is a job for a small 1620 -- we have done this for 70,000 records on a 20K card 1620. The 500,000 record job will be done on a 7074 although we might have tried it on the souped-up 1620 if changing zones were the only changes required and it had not been moved out.

These examples show how we have made use of our computer for preparing data from an O-D survey. There are many more uses which we will undoubtedly uncover in the future. We feel our 1620 has a definite place in conjunction with the larger computers and intend to make maximum use of it.

4.... /0

THE PLOT ROUTINE AS AN

EXAMPLE OF METHODS OF ADDING

SUBROUTINGS TO THE NCE LOAD AND GO FORTRAN

Hubbard Seward

Newark College of Engineering

25

4.15.

The Plot Routine as an Example of Methods of Adding Subroutines to the NCE Load and Go Fortran

by

Hubbard Seward

Newark College of Engineering

Presented at the May meeting of the IBM 1620 Users' Group in Washington, D.C.

NCE Load and Go Fortran was written to provide a 20K Load and Go system with greater arithmetic abilities than the IBM Gotran system. In order to conserve space it was necessary to omit or simplify some features of Fortran I. Whenever possible features were provided to replace those which had to be eliminated. These substitutions make it necessary for the programmer to compile in the Fortran source program some things which are handled automatically in other systems, however in many cases additional usefulness was created. The original description of the system states that subroutines cannot be added, however this is not exactly true. Because of space limitations, no facilities for adding subroutines have been included, but any user may add such subroutines as will increase the processors utility. Of course he must do so with the realization that, at least for those restricted by a 20K machine, space is at a premium.

Recompilation will be necessary to add subroutines, but an unlimited number of functions and up to four new nonarithmetic pseudo op codes can be added. Functions can be added by merely including their names and the associated addresses in the function table symbolic location FTBL. The table format is an eight digit constant representing the alphameric function name followed by a five digit constant for the address of the execution routine. The address must have a flag over the low order digit. Note that functions can be made removable by locating them after the I/O area and properly relocating the record mark placed by the initialization routine and also changing the address used to initialize several fields. (Chi, output, wk, and associated constants should remain in their present relationship but may be moved to any location. between 10K and 20K. See for example the routine at location 15824 on the listing. For maximum space the EXP and Lôg routines can be eliminated but when this is done the instruction at 07380 should be made to branch to an error message, since the ** operation requires these subroutines.

To add new features to the system, additional nonarithmetic type pseudo instructions can be added. Because of the way this part of the interpreter is set up there is a limit of four more which can be added. The length of the nonarithmetic pseudo instructions is variable. The op code must have two digits and the instruction must be terminated by a record mark. All operands are two, three, or four digits long and are of four types:

(1) Variables and constants use the four-digit symbol table pseudo address which must be decoded by SUBER; (2) Statement numbers are represented by their three-digit symbol table address; (3) USE and do end pseudo ops which may have to refer to statements which have no statement number use the four low order digits of the location of the pseudo ops for that statement; (4) IF (sense switch) pseudo ops contain a two-digit indicator as the first operand. Currently the maximum size of pseudo instructions is limited to 26 digits by the size of the area WK.

The plot pseudo instruction is complied by recognizing the word PLOT on the source statement. The control is passed to the I/O compiler routine which assembles the operands in groups of five. If more operands are present, a new pseudo op is produced.

The execution routine increments the card number and then enters a loop which puts a character in the output area for each variable. When there are no variables left in the list, the card is punched and control returns to the interpreter to execute the next pseudo instruction. If more than five variables are needed on a plot or if axes are desired this routine can be easily modified to provide these features. More variables could be accommodated by changing the plot routine to examine the next pseudo op before returning control, and returning control only if it were not another plot. Note that the card numbers in columns 79 and 80 also provide a means of sorting out plot cards from simultaneously punched data output.

NON - ARITHMETIC PSEUDO INSTRUCTION TABLE

FORTRAN Equivalent	Pseudo op code	Description	Length	Operands
END	+ 00	Branch to compiler	5	X
*FXFLT	- 00	Change mode of FAC	3	X
*RSGN	+ 01	FAC = - FAC	3	X
*DO END	- 01	Increment Index; check for Index Limit	7	C
STOP	+ 02	Branch to Intrepreter Initialization	3	X
*DO START	- 02	Set Index Equal to Initial Value	19	A,A,A,A
PAUSE	+ 03	Halt	3	X
GO TO	- 03		6	B
READ	+ 04			
ACCEPT	- 04		3-23	X or From
PRINT	+ 05			1 to 5 TYPE A
PUNCH	- 05			
*IF( )	+ 06	Tests FAC For -,0,+	12	B,B,B
IF(sense SW)	- 06	Tests Indicator	11	D,B,B
USE		Stores LOC in Table	6	B
RETURN	- 07	Returns to LOC in Table	3	X

* These routines are not exact equivalents of FORTRAN Statements.

Operand types

A Pseudo Address - The three center digits of a symbol table address the first and last being always 1 and 9 respectively plus a fourth digit to indicate subscript.

B Same as A without subscript digit.

Refers to statement number location in symbol table.

C 4 low order digits of location of DO start pseudo op.

D 2 digits for machine indicator number

X No operands

Note On I/O instructions flagged subscript digit indicates fixed point variable.

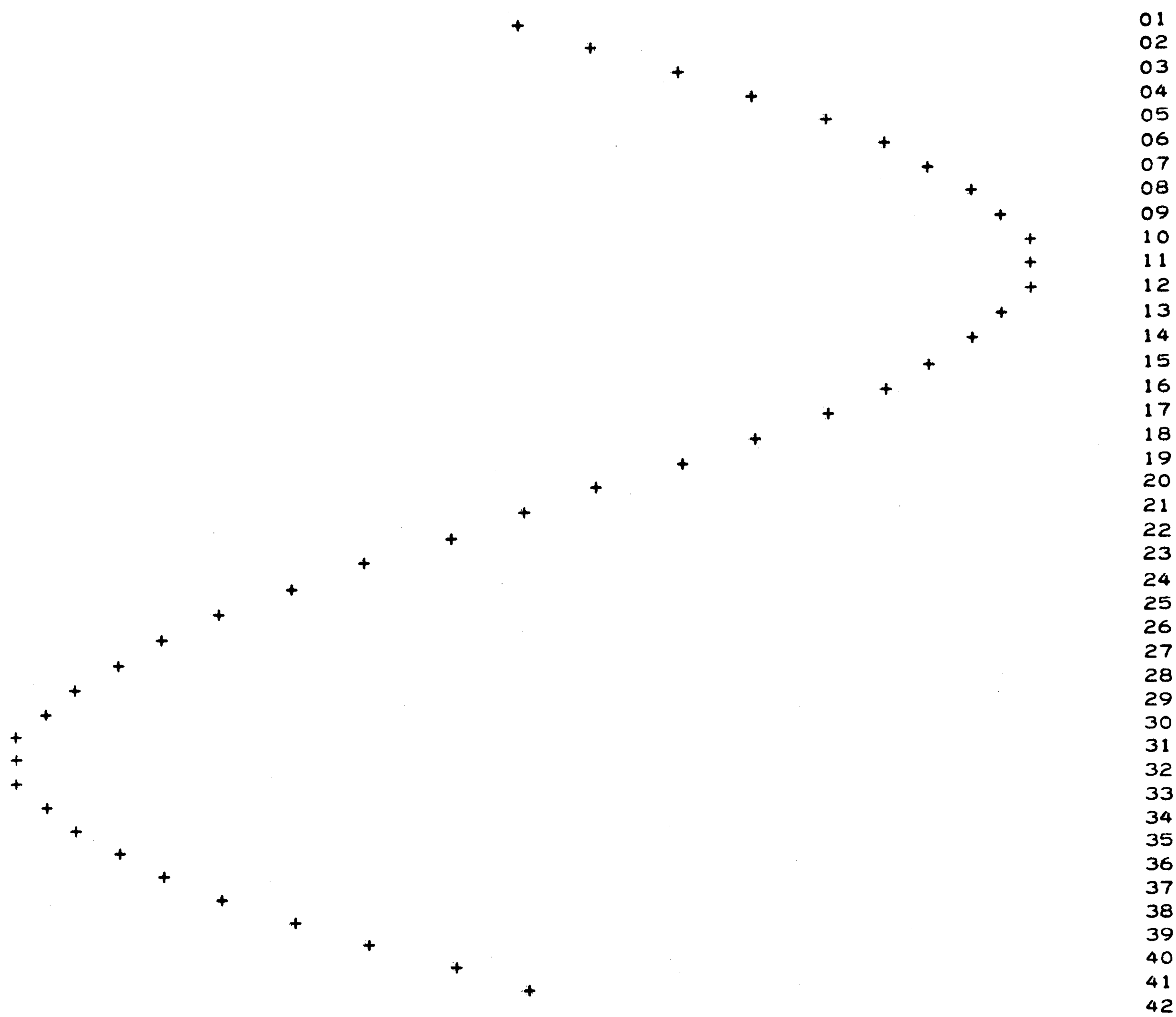
```
******       PLOT ROUTINE FOR LOAD AND GO FORTRAN
***          COMPILATION ROUTINE AND CONSTANTS
***          PLACE AFTER CARD NO. 24
       DSA   INPLOT

***          PLACE AFTER CARD  155
       C     PLT,CHI+6
       BE    COPLOT

***          PLACE AFTER CARD  178
       DAC   4,PLOT
PLT    DS

***          PLACE AFTER CARD 526
COPLOT TFM   WK+1,80,9R
       B     COREAD+12
       DORG  *-3

***          IF PLOT ARRAY  STATEMENTS ARE DESIRED, REPLACE CARD 392 WITH
       BNE   ER11-120


***          EXECUTION ROUTINE
***          IF THE PLOT ROUTINE IS DESIRED TO BE REMOVABLE, ( SW 1 OPTION )
***          PLACE THIS AFTER CARD 1930,  IF NOT PLACE AFTER CARD 1810
******         NOTE THAT NUMBERS MUST BE FIXED POINT AND IN THE RANGE 0-77    0007
       AM    LOC-1,4,10
       TR    WK,WK+4                               20214 31 14625 14629 12-0031
       AM    P4-1,2,10                             20226 11 20321 000-2 12-0032
       MM    FAC,2,10                              20238 13 00079 000-2 12-0033
       BN    ER66                                  20250 47 20382 01300 12-0034
       AM    99,OUT                                20262 11 00099 J4459 12-0035
       CM    99,OUT+154                            20274 14 00099 J4613 12-0036
       BP    ER66                                  20286 46 20382 01100 12-0037
       C      P4+11,-99                            20298 24 20333 0009R 12-0038
       TF    -99,CHAR,7                            20310 26 0009R K0415 12-0039
P4     BE    *+24,,10                              20322 46 20346 012-0 12-0040
       TFM   -99,21,10                             20334 16 0009R 000K1 12-0041
INPLOT AM    LOC-1,3,10                            20000 11 09139 000-3 12-0009
       TFM   P4-1,CHAR-2                           20012 16 20321 K0413 12-0010
       BT    SWEEP,OUTX                            20024 27 08482 08735 12-0011
       AM    CDNO,1,10                             20036 11 20145 000-1 12-0012
       BD    *+48,CDNO                             20048 43 20096 20145 12-0013
       AM    CDNO-1,9,10                           20060 11 20144 000-9 12-0014
       BD    *+24,CDNO-2                           20072 43 20096 20143 12-0015
       TFM   CDNO-2,7170,8                         20084 16 20143 0P170 12-0016
       TF    OUT+158,CDNO                          20096 26 14617 20145 12-0017
P1     BNR   P3,WK+1
******              EXIT HERE IF FINISHED                                0019
       WACD  OUT                                   20120 39 14459 00400 12-0020
       B     LOC

CDNO   DC    4,7070,*+2                            20142 P0 70    20145 04-0022
       DORG  *+2                                   20145                 0023
P3     BT    SUBER,WK+4
       TF    FAC,-SUB1-23                          20158 26 00099 1135J 12-0025
       BNR   *+20,FAC                              20170 45 20190 00079 12-0026
       B     ER71                                  20182 49 09060 00000 12-0027
       DORG  *-3                                   20190                 0028
       BNF   ER74,FAC-3                            20190 44 20370 00076 12-0029
```


4.15.6

```
       B     P1                                          20358 49 20108 00000 12-0043

******       ERRORS  74 FLOATING POINT VARIABLE.     66  OUTSIDE RANGE        0044
ER74   BTM   ERROR.77400.5R11                        20370 17 027-4 7740R 12-0045
ER66   BTM   ERROR.67600.7R                          20382 17 02704 0760Z 12-0046
       TFM   OUT+154.4559.8                          20394 16 14613 0M559 12-0047
       B     P1                                      20406 49 20108 00000 12-0048
       DORG  *-3                                     20414                   0049
******       LIST OF CHARACTERS FOR PLOTTING                                  0050
CHAR   DAC   1.+                                     20414 J0        20415 02-0051
       DAC   1.*                                     20416 J4        20417 02-0052
       DAC   1.X                                     20418 07        20419 02-0053
       DAC   1.0                                     20420 P0        20421 02-0054
       DAC   1..                                     20422 -3        20423 02-0055

***          IN ORDER TO MAKE THE PLOT REMOVABLE ADD THIS AFTER CARD 1947
       SF    PLT

***          IF MORE THAN FIVE VARIABLES ARE TO BE PLOTTED. REPLACE
***          THE   B    LOC   IN THE ABOVE ROUTINE WITH THE FOLLOWING
***          AND MAKK SUITABLE EXPANSION TO THE LIST OF CHARACTERS.
       AM    LOC-1.2.10
       CM    1-LOC.4.10
       BNE   EXIT
       AM    LOC-1.1.10
       B     P1
EXIT   SM    LOC-1.2.10
       B     LOC
```

THE CARD NUMBERS IN THIS LISTING REFER TO THE CARDS IN THE LISTING OF THE UNCOMPRESSED DECK AS DISTRIBUTED BY THE PROGRAM LIBRARY. ALL MACHINE LANGUAGE INSTRUCTIONS AND ADDRESSES ON THIS LISTING SHOULD BE IGNORED.

```
A=0.0
PI=3.1415926536/20.
DO 8 I=1,41
DIMENSION ISIN(41)
ISIN(I)=SIN(A)*35.+35.5
8 A=A+PI
PLOT, ISIN(I),I=1,41
END
```



4.15.8

AUTOMATIC SCHEDULING AND REGISTRATION

IN A SMALL COLLEGE

Arthur F. Jackson

Director of Guidance and Computer Center

Greensboro, North Carolina

AUTOMATIC SCHEDULING AND REGISTRATION IN A SMALL COLLEGE

by

Arthur F. Jackson
Director of Guidance and Computer Center

The Agricultural and Technical College of North Carolina is a small land grant college located in Greensboro, North Carolina with an enrollment of approximately 3,000 students. It has five undergraduate schools: Agriculture, Education and General Studies, Engineering, Nursing, and Technical Institute. Each student registers in one of sixty-five different major fields.

The college has been using unit record equipment in the registration process for several years beginning with a Remington Rand installation of Punch and Sorter and developing into a complete Remington installation. In 1962, the college changed to an IBM Series 50 including the Sorter, Interpreter, 402 Accounting Machine. With this equipment, registration proceeded as follows.

The Deans of the various schools requested that a specific number of class cards be prepared for a certain quarter. This information including class code number, class description, meeting place, time, and credit hours, was punched into a Master Course Card. These cards were then duplicated on the reproducer in the numbers indicated by the Deans and then sent to the Registrar. At pre-registration time, the students completed a course schedule on a four-part form at a pre-registration station. These forms were completed and sent

to the Registrar's office where clerical help manually matched each course on the schedule form with a pre-punched class card. Each student's class cards were assembled in an envelope and returned to the Data Processing Center where student's name and number were gang punched into his class cards. From these cards, class rosters and individual student schedule forms were printed on the 402 Accounting Machine. In cases where cards or courses had all been distributed before a student had received all his class cards, his schedule was sent back to his advisor for revision.

In June, 1963, the college accepted delivery of an IBM 1620 Computer, a card system with 20 K Memory positions and without automatic divide. It was used primarily for teaching purposes except for occasional use in the computation of statistics and compilation of certain registration data. During the 1964 Spring Quarter, the 1620 played a prominent part in the registration. This was possible because of the development of three programs: STUDENT, a program devised by IBM, and two post processor programs designed by the author.

STUDENT, an abbreviation of "Scheduling Technique Using Defined Increments of Time", was designed for use in scheduling students into a section where the number of sections does not exceed 1000. It is designed to store a Master Schedule (not to exceed 1000 sections), accept student requests (not to exceed 8 per student), and to place students in sections. Students are allowed to designate specific sections desired and also to request free time. Students who request

specific sections are scheduled until requested sections fill; students who do not specify sections are scheduled as long as any section is open. When a student is not scheduled, a report is written on the typewriter indicating the reason for failure to schedule.

This program is controlled by course code number, section number, and load limit. This means that each student request is checked to determine whether or not the number of spaces provided for that section has been reached. On the basis of these two checks, the student is scheduled or not scheduled. The card format for the student request allows for twenty-eight columns of information about the student while the rest of the request card is allotted to the eight courses.

In order to test the feasibility of STUDENT for use in our system, the decision was made to use the class cards and Master cards from the 1963 Fall Quarter registration in a trial run. A computer program was, therefore, written to condense the several individual class cards for each student into the single request card format required by the program STUDENT. The Master Cards were also transformed in the STUDENT format. The process was then reversed, the Masters were loaded into memory, and the students' requests fed into the program. Every possible conflict situation was simulated in the test. The test resulted in the "go ahead" signal for the Spring Quarter.

Once this decision was made, there remained two problems. The punched output from STUDENT could not be used effectively as the class

card which is used as basis for class rosters, admission to class, and finally as a grade report. The second problem was that of determining the spaces remaining in sections which were still open as soon as the scheduling had been completed. This information would eventually be obtained from temporary class rosters, but administrative officials felt the need to have this information earlier than possible by this method. This information was vital to the opening of new sections or the deletions of courses in which too few persons were registered. Two programs, both storage and search programs, were written to achieve these ends.

The program to punch class cards was designed to use as input the condensed output from STUDENT. This was a single card with name, I.D. number, and as many as eight course code numbers and sections. The program first stored up to 225 Master Course Cards in memory. It was limited to 225 because of the alphabetic content of our Master cards. The program then attempted to match each course on the student request card with one of the masters in storage. When there was a match, a duplicate of the Master Card was moved to output, the student's name and number were added from the request card, and the card was punched.

The search was sequential, from the first master stored to the last. There was, therefore, no need to sort the master course cards before storing in memory. If no match occurred during a search, the computer simply moved to the next course on the card or the next card to begin the search over again.

The inability to store all of the Master Course Cards necessitated four passes with the request cards. The Masters, however, were loaded by school groups which made possible the printing of class rosters for one school while the computer was punching course cards for another school. This process was not timed with stop watches, but it was possible to prepare all of the class cards within one working day. In addition, the preparation was much more complete than ever before.

The card count program was similar in nature to the program for punching class cards. The course code number and section of each Master Card was stored in memory together with a two position counter. Each unit was thirteen storage positions as follows: a four digit course code number, a blank; a two digit section, a blank; a two digit number representing the number of spaces allotted to that section, a blank; and finally, a two digit counter. Storage was searched sequentially, and when a match occurred, the number "one" was added to the count position of the section matched. At the end of the run, the typewriter printed out the contents of memory where the course codes had been stored. This was a report of the number of spaces allotted to each section and the number used at this point.

Pre-registration was conducted as previously with the exception that each person listed, in addition to the courses, the course code number and section for each course on a special card. These cards were sent to the Data Processing Center where, as soon as received, they were punched. No sorting was required, but the cards

were sorted by school for ease of handling of individuals who could not be scheduled. These were then run into the 1620 with the program switches set for no punched output. This afforded an opportunity for checking such things as missing master cards, wrong sections, and the like. Errors in course code numbers were expected since this was the first time the student body had been required to use them. All errors thus discovered were corrected and the request cards were then run again to discover schedule conflicts. Most of these resulted from the choice of favorite hours for classes. Where possible, these were resolved by reproducing the request cards, omitting the section, or by punching in sections which could not create conflicts. No punched output had been obtained on either run.

With conflicts reduced to a minimum, the request cards were once more fed into the 1620, but this time the program switches were set to obtain punched output in condensed form. Those individuals (their request cards) who were not scheduled on this run were sent to the department in which they were enrolled for a resolution of this conflict.

Now, the program for punching class cards was used. Class cards were made for each class section for all of those who had been scheduled successfully and a count made of the number of seats taken in every section. Class rosters were then printed for every class and section on stock paper. Schedules were printed on the 402 Accounting Machine for the individual students who had been scheduled.

In the interim between the final run on the computer and

registration, the various administrative officials used the data provided by the program described to open new sections, expand limits in sections, et cetera. This made possible the scheduling of most of those individuals who had not been scheduled prior to this time.

At registration time, the student picked up his schedule and other things such as meal books, et cetera. The time required for this was about fifteen minutes as opposed to several hours under the old system. Lines moved very rapidly so no long lines were formed.

Problems in registration were reduced to a minimum but the problems in changing courses was not solved. In our system, a student must re-take a course he has failed in the next quarter it is offered. This created most of the requests for changes.

The information gained in this effort suggests pre-registration in May for the next Fall Quarter, registration of the entering freshmen with test scores taken into account, and eventually pre-registration for an entire year together with automatic scheduling.

4.16.7

AFIT

SPS WITH SIMULATED TNF, TNS AND MF

INSTRUCTIONS

David Olson, Computing Center

Newark College of Engineering

AFIT SPS with Simulated TNF, TNS and MF Instructions

David Olson
Computing Center
Newark College of Engineering

Introduction

AFIT SPS by Mr. Pratt has just recently become widely available through the Program Library.

The system has many advantages and improvements over SPS II.

However the system requires a machine with indirect addressing, auto-divide, 40K and the TNF, TNS and MF instructions.

Description of Simulation Method

We have removed the requirement of the TNF, TNS and MF instructions.

TNS

The TNS instruction is used only twice in the processor, therefore TD instructions were used to simulate this instruction.

TNF

The TNF instruction is used several times therefore it was felt that a subroutine would be the most efficient method of simulating it. This was done as follows:

Replace the 73 with a 41, then follow the TNF with BTM TNF, *-1.

The TNF subroutine looks like this:

```
            DS      5
TNF         SF      TNF-1
            TF      A999,CON1
            TF      304,1-TNF
            A       A999,CON2
            TF      310,RSTOR
            AM      TNF-1,5
            TF      1-TNF,A999
            BB
CON1        DC      10,6070707070
CON2        DC      10,0001020304
A999        DS      10
RSTOR       DC      12,801234567891
```

This subroutine is essentially the one described by Mr. Pratt in the May 1963 issue of the <u>1620 User Group Newsletter</u>.

<u>MF</u>

The MF instruction is also used quite frequently, therefore it was also simulated by a subroutine.

This was accomplished as follows:

Replace the 71 with a 41 then follow the MF with a BTM MF,1-*

The MF subroutine is as shown below:

```
QADD        DS      5
MF          TF      PADD,QADD
            AM      PADD,5,10
            CF      -PADD
PADD        DS      5,*
            BNF     *+36,-QADD
            CF      -QADD
            SF      -PADD
            BB
```

Of course there was some additional rewriting of the processor to allow use of the BTM instruction, since the routines in the processor which used TNF or MF were quite often entered by another BT or BTM instruction.

Mr. Pratt is to be congratulated on his excellent contribution to the growing list of Symbolic Programming Systems. We have found no significant errors in his system.

In our opinion, it is the best 40K SPS available.

NCE HIGH SPEED SPS ASSEMBLER

Kurt Germann and George Rumrill

Newark College of Engineering

4.18.0

NCE HIGH SPEED SPS ASSEMBLER

by

Kurt Germann and George Rumrill

Newark College of Engineering

Presented at the May meeting of the IBM 1620 Users' Group in Washington, D.C.

This assembler is an attempt to provide Users' Group members with an assembler that has the language capabilities of AFIT SPS, and a symbol table size of the order of LAMP (File 1.00.001) in a 20K system. Although some compromises had to be made, the resulting system, provides space for 450 labels, and has a language capability similiar to IBM 1620/1710 SPS.

The need for an assembler with a large label capacity in 20K developed when the Computer Center Staff of Newark College of Engineering was writing NCE Load and Go Fortran (File #2.0.029). During compilation of the Load and Go System, the IBM SPS Symbol table overflowed, so another version of SPS was looked for in the 1620 Library. It was found that OSAP (File #1.01.013) best suited the need, but to be useful to us, several modifications and corrections had to be made. These initial modifications were the work of Joseph Petersack, an undergraduate student at NCE. After the 20K Load and Go processor was completed, the authors decided to follow through on these initial modifications, and to rewrite completely the OSAP processor.

The NCE assembler is approximately two to three times as fast as IBM 1620/1710 SPS, exclusive of input-output. This means that on a Model I 1620 with a 500/250 cpm reader-punch, the assembler will run at reader speed on Pass 1, and at punch speed while punching an uncompressed object deck on Pass 2.

The assembler requires card I/O and indirect addressing. Automatic divide is required only if division is performed in address adjustment in the source statements.

The assembler has been written so that the input format can be of <u>two</u> types. It can be standard SPS, or a slightly modified version. The modified format is easier to keypunch, and also easier to read on a 80 - 80 printout. The format is as follows:

Columns	1 - 6	Label
	7	Blank
	8 - 11	Op Code
	12	Blank
	13 - 50	Operands and Remarks
	51 - 80	Arbitrary

This format does not allow page and line counts; if they are desired the standard SPS format can be used by setting a console switch.

The output format of an uncondensed object deck is in most cases a single card per statement, and can be printed on a standard 80 - 80 board. The uncondensed format is:

Columns 1 - 50 Original Source Statement

52 - 80 Machine Language and Sequence Numbers

This format enables using the object deck as a source deck, making the original source obsolete. Besides being more convenient, this format produces object decks which are about one half as large as those of SPS II. In addition, a compressor is built in, which produces compressed decks smaller than those of SPS II. Also provided, at the operator's option, is a symbol table punch-out, five symbols to a card, in alphabetic order, at the end of pass 1.

The number of labels that can be stored by a 20K machine varies. If, upon initialization, the compressor, its loader, and the uncompressed loader are eliminated, approximately 450 six-digit labels can be stored. If the compressor and loaders are not eliminated, about 350 labels can be stored. If a larger model 1620 is used the assembler expands automatically, increasing the symbol table accordingly.

Depending on the size of the machine, the size of the program, and the number of labels, it is possible to store the source statements of a program in core during pass 1.

If the program fits in core, it can be run in a single pass, if it does not fit, however, the source program must be reentered for pass 2 as usual. In a 20K machine, again depending on the program, 60 - 100 source statements can be stored and processed. A 60K machine on the other hand can store 1000 - 1300 source statements. Besides reducing card handling, this will also, of course, save machine time.

In addition to the above, a different error message format has been incorporated. Each error message consists of a typeout of the actual source statement in error, the type of error, followed by the sequence number of the card in the uncompressed or compressed object deck that contains the error. To aid in trouble shooting, column 51 of the uncompressed object card in error contains the letter E.

As far as the language is concerned, all SPS II op-codes for the 1620 card system are valid, both mnemonic and numeric, with the exception of HEAD, TRA, TCD, and SEND. In addition, many of the op-codes used in AFIT SPS have been incorporated, as well as several interesting programming procedures devised by Richard L. Pratt. A variable and fixed length subroutine package similar to that of AFIT SPS is also included.

Blanks in Source Statements

Blanks are ignored in the operand field of all source statements, including the flag operand of instructions.

In a DAC statement, the rightmost character of the constant does not have to be followed immediately by a comma. If no address is specified, only one comma is needed between the length and the constant itself.

Examples:

A	DS	, 1001 *10-9	
B	DS	, 20002	
	TF	A , B , 1 2 3 4 5	$2\bar{6}$ $\bar{1}0\bar{0}\bar{0}1$ 20002
	TR	A, B , ,	31 10001 20002
	DAC	5,	$\bar{0}0$ 00000 000
	DAC	2, A @ , ,	$\bar{4}1$ 0*

* The spaces between the second and third commas will not cause the constant to be loaded at zero

	DC	5,@ , ,	$\bar{0}000$ *

DAC Statements

The maximum length can not be greater than 35 because of the restriction on the length of the source statement (columns 1 to 50).

DSAC Statement

The Define Special Alphameric Constant is the same as the DAC except no flag is placed over the high order digit. This allows long alphameric constants to be constructed by one DAC followed by DSAC's and this long constant can be moved by a transmit field instruction.

DC and DSC Statements

A) The processor can supply any number of lending zeros, up to the maximum constant length of 99999

Example:

X23 DC 5600,@

This will cause 5599 zeros followed by a record mark to be loaded into care, the label will refer to the location of the record mark and the high order zero will be flagged.

B) The @ sign which indicates a record mark may appear anywhere in a constant.

Example:

DSC 6,-@123@@

Will cause $\overline{\ddagger}12\overline{3}\ddagger\ddagger$ to be stored.

C) Any character of the constant may be flagged by punching a minus sign over that character in the source statement.

Example:

DC 7, M4*P8

Will cause $\overline{0}0\overline{4}\overline{4}\overline{\ddagger}78$ to be stored

DC	1, @	will	produce	$\ddagger$
DC	1,-@	will	"	$\overline{\ddagger}$
DC	1,*	"	"	$\overline{\ddagger}$
DSC	1,-@	"	"	$\overline{\ddagger}$

D) Constants may be preceeded by a plus sign if desired.

285

4.186

DFC Statements

Floating point constants, as required by the subroutines, may be specified as follows:

first operand - mantissa length, which may be symbolic, (must be less than 50)
second " - mantissa (trailing zeros automatically supplied.)
Third " - exponent (may be symbolic)
fourth " - address (if desired)

The label, if present, will refer to the units position of the exponent.

Example:

DFC 10 ,-2,1

will produce $\bar{2}00000000\bar{0}$ $\bar{0}1$

Operands

The previous IBM restriction on - 0 is removed and this will now be assembled correctly.

Example:

TDM A , - 0 15 10001 0000$\bar{0}$

Division is now allowed in address arithmetic (if the auto divide feature is available). Division can be used in conjunction with subtraction and multiplication to produce truncation.

Example:

AM B/A * 1000, B/A +213 11 02000 $\bar{0}$0215

Record Marks in the Q11 position of instructions are frequently required. This usually has required the two instructions:

```
TFM     A    ,, 10
DC      1,@,*
```

However on this system an R in the flag operand will cause a record mark to be placed in the Q11 position of the instruction:

Example:

TFM	A ,, 10R	results in	16	10001	000$\bar{0}$‡
TFM	A ,, R	" "	16	10001	$\bar{0}$000‡
TDM	A ,, 11R	" "	15	10001	0000$\bar{‡}$

DSA Statements

The number of operands in a DSA or DSSA is no longer limited to ten, but may be any number of address operands up to column 50.

DSSA Statements

This is the same as DSA, except that no flags are set over the high order digit of the DSSA's, and the label refers to the high order digit of the first DSSA.

TABLE	DSSA	A,B,	10001	20002	00000

DNB Statements

The length in the DNB and DSNB statements must be no greater than 99999.

DSNB Statements

Define Special Numeric Blank acts the same as DNB except that the label, if present, is equated to the high order position of the numeric blanks.

INC Statement

Input Conversion Subroutine allows input data in the free format style of AFIT Fortran to be converted to internal floating point form:

287

4.18.

first operand — address where exponent of number is to be stored.

second operand — alphameric record address of data to be converted.

Data in the input area may be separated by blanks or commas and although the input area must contain an alphameric record mark, no flags are required. Library No. 1.6.053.

OUTC Statements

Output Conversion Subroutine allows internal floating point numbers to be converted to external Fortran style constants under format control.

first operand	alphameric record address where external form is to be stored
second operand	address of exponent of floating point number to be converted.
third operand	five digits specifying the format:

In this five digit constant, the first digit represents the type of format desired:

0	for E
1	for F
2	for G

The next two digits are the length of the field, and the last two the number of spaces after the decimal point.

E 14.8	01408
F 7.2	10702
G 16.10	21610

The E and F conversion work exactly the same as 1620 Fortran. The G w.d conversion gives F type output if as many as d significant figures can fitted in the space provided. Otherwise, a format of E w.d will be used. Library File # 1.6.053

D O O D L E

Do-It-Yourself Problem Solver

M. V. Farina

General Electric Company

289

4.19.0

DOODLE

Do-It-Yourself Problem Solver

M.V. Farina G.E. Company

An engineer busily at work in the solution of a mathematical problem reminds me of a trombone player. He can manipulate a slide rule with all the skill of a musician. Unfortunately, slide rules are slow and limited in accuracy.

Suggest to him the possibility of programming his problem for solution on a computer, however, and chances are he'll give you a hundred reasons why he can't. He doesn't know anything about programming; Fortran has too many rules to remember; his problem is too small for a computer, and he can go on and on with excuses.

An accountant industriously engaged at making debits equal credits reminds me of a pianist. He can finger the keyboard of a desk calculator easily as well as Van Clyburn.

But the typical accountant, like the engineer, believes that he himself cannot program his problem for a computer.

Perhaps it does take some skill to program an engineering or business problem. But have you noticed -- it takes no skill to doodle? We all do it -- while on the telephone, during a meeting, while waiting for the doctor.

If programming could be made as simple as doodling, then maybe engineers and accountants would program. In an effort to simplify programming we devised a system for those people to use and called it DOODLE.

My discussion of DOODLE is in three parts: First, what DOODLE is; second, how we are inducing engineers and accountants to use it; and third, some ideas of what the second generation of DOODLE might produce.

DOODLE is a system which enables the user to write out a problem in a very simple way. There are few rules to know and a knowledge of programming is not required. We'll show you some examples in a few moments.

DOODLE is an interpretive scheme. It works directly with the user's program as he has devised it. The system interprets each instruction as it encounters the instruction and executes it. This means that no assembly or compilation process need be performed before the problem is given to the computer. Computer time is thus conserved. Very often it takes longer to compile or assemble a program than it does to run it. This is particularly expensive on one-shot jobs. DOODLE bypasses this expense.

Mechanically, DOODLE is a Fortran II program written for an IBM 1620 Computer with indirect addressing, auto-divide, floating-point hardware and card I-O. Our Fortran compiler in Johnson City has been changed to origin at 11K instead of the standard 12K, as specified in the Fortran II Operator's Guide, Page 25. (40 K machine)

I can best continue showing you what DOODLE is by telling you how we are inducing our engineers and others to use DOODLE. The words "tricking them" seem to be applicable here since the write-up of DOODLE does not mention the words "program" or "programming." The write-up pre-supposes no programming knowledge and speaks only in terms of: "you have a problem to solve -- a computer can solve it for you--here's how you do it."

The DOODLER publication exists as a 2-volume work. The first booklet is the "Primer" and the second booklet is called "For The Advanced Doodler." I'd like to show you a few of the pages from the Primer.

(SLIDE 1 - PAGE 1 OF DOODLE WRITE-UP)

The key word here is "EASY." On this first page we are hoping to spark the reader's interest.

The second page tells the reader that DOODLE is a method of telling the computer what his problem is; that he writes out the problem on a simple form; that cards are punched from the form and delivered to the computer; and that he gets answers in return.

(SLIDE 2 - PAGE 3 OF DOODLE WRITE-UP)

This slide shows how the user tells the computer what numbers his problem deals with. Note the illustration. We feel the illustrations, of which there are many, are very effective.

A card will later be punched for every line shown on this coding sheet. Notice that item numbers have already been filled in.

(SLIDE 3 - PAGE 5 OF DOODLE WRITE-UP)

We have told the reader how he feeds numbers into the computer. He will now be told how to tell the computer what to do with those numbers.

(SLIDE 4 - PAGE 6 OF DOODLE WRITE-UP)

This slide shows how the "ops" are used.

Page 7 of the DOODLE write-up reads as follows:

"You told the computer to divide item 101 by item 102. The result became item 105. Then you told the computer to multiply item 103 by item 104. The result became item 106. Finally you told the computer to add items 105 and 106. The result, item 107, is the answer to your problem."

Finally the reader is instructed to put the op "END" at the end of his solution procedure, and he is shown how the computed coding form will look at that time.

(SLIDE 5 - PAGE 8 OF DOODLE WRITE-UP)

At item 107, note that a "1" has been placed under the column headed "P." This causes the computer to print out the result of the execution of the instruction at item 107. This, of course, is the answer to the example problem.

(SLIDE 6 - PAGE 15 OF DOODLE WRITE-UP)

Skipping ahead in the DOODLE write-up, page 15 shows a list of the ops that are available to the DOODLER.

At the bottom of the slide, notice that the publication "For The Advanced Doodler" is mentioned for the first time. It is not necessary for a DOODLER to ever read the advanced volume. It includes such exotic features as looping, address modification and use of subroutines. We recognize that these features are such that only professional programmers would be interested in them or DOODLER enthusiasts.

The DOODLE write-up continues in this light vein. The reader is let step by step to the point where he learns how to get repetitive solutions from problems, and how to make use of smybolic nomenclature.

He is told how to do this in a chapter called "That Mysterious Column C." Here is part of that chapter:

(SLIDE 7 - PAGE 17 OF DOODLE WRITE-UP)

In the last part of the booklet, we give an example of how the quadratic formula could be written up.

(SLIDE 8 - PAGE 21 OF DOODLE (MODIFIED)

Notice that the values for A, B, and C are given to the computer at item numbers 101, 102, and 103.

The constants used by the problem, a zero, a four and a two are given next at items 104, 105, and 106.

The actual programming of the problem start at item 107. We need a minus B, so we subtract B from zero and store it in MB, which stands for "minus B."

294

4.19.5

We need a B square, so we multiply B times B and put the result in B2. The names where results are stored can be made up by the DOODLER as he goes along. In the example, we have used names that remind us of what has been calculated. Notice that when A is multiplied times C, we have chosen as the result name "AC," and when AC is multiplied times four, we have used the result name "4AC."

There are two answers to this problem, and they are shown at items 116 and 117. These are the positive and negative roots respectively. Notice the "1" under the column headed "P." This causes the computer to print out the results at items 116 and 117.

When the computer comes to item 118, and spots the op "END," it goes back to the card reader looking for more cards. If it finds more cards, it will load what it finds on top of the program being run starting at the beginning of the program and continuing to load until it comes to another end card.

In this example, new values of A, B, and C will be loaded on top of the old values of A, B, and C. The program will then be executed with the new values, new answers will be printed out, and the computer will again look for new input in the card reader.

It is possible for DOODLE to set up branch conditions, to make tests and to branch about within the program depending upon the value of test conditions. It is even possible to do address modification and branching to subroutines. We don't recommend that the DOODLE user get involved with these more sophisticated features of DOODLE. It is possible, for example, to get into endless loops.

Besides, a person who was this sophisticated would probably be better off learning and using Fortran.

It is easy to find mistakes when using DOODLE. If sense switch one is turned on before the program is run, the result of every instruction is printed out.

Before closing I'll touch upon some ideas which have come to mind about how DOODLE can be used in some novel ways.

Suppose an engineer has a problem he wants to solve. Rather than pick up a coding sheet, writing down the instructions and then having them key punched, he can simply reach into his desk, pull out some pre-punched cards, put them together, and take them to the computer direct, thus by-passing the key-punching step.

Of course the situation isn't really as simple as this. The user is required to maintain in his desk several hundred cards made up of about 100 standard instructions. The next slide shows what some these standard instructions look like:

(SLIDE 9 - SPECIAL)

The secret in using this idea is that every instruction revolves around ARG1 and ARG2. If one wants to add two plus two, for example, the "desk programmer" would have to put a two into ARG1, then a two into ARG2 and finally cause the addition of ARG1 and ARG2. The result would go to ARG1 and the programmer would then usually have to put ARG1 somewhere in order to use its results later.

This sounds very wasteful and slow. It is! We're not suggesting this idea has any practicality at all. Or does it?

Suppose every standard statement had a unique 2-digit number associated with it. Then, the programmer could call for a standard statement by number rather than reaching for it in his desk drawer. Suppose he punched up all the standard statements he required for a particular problem on cards starting in column 1 and continuing to column 80. He could get forty 2-digit numbers per card, which could represent a fairly elaborate sequence of events. For example, suppose the programmer wants to compute the following:

$$\mathbf{X} = \mathrm{SIN}\ (\theta^2 - 1)$$

He reads in his data so that **X** is represented by "standard **A**" and θ by "standard B." The solution of this equation could then be as follows:

(SLIDE 10 - SPECIAL)

We have arbitrarily assigned 2-digit numbers to each of the standard instructions shown here. Note that the value for 1 is standard. So would other numerical values selected such that the programmer could build a number as he needed it. For example, the number 100 could be built by multiplying standard 10 by standard 10.

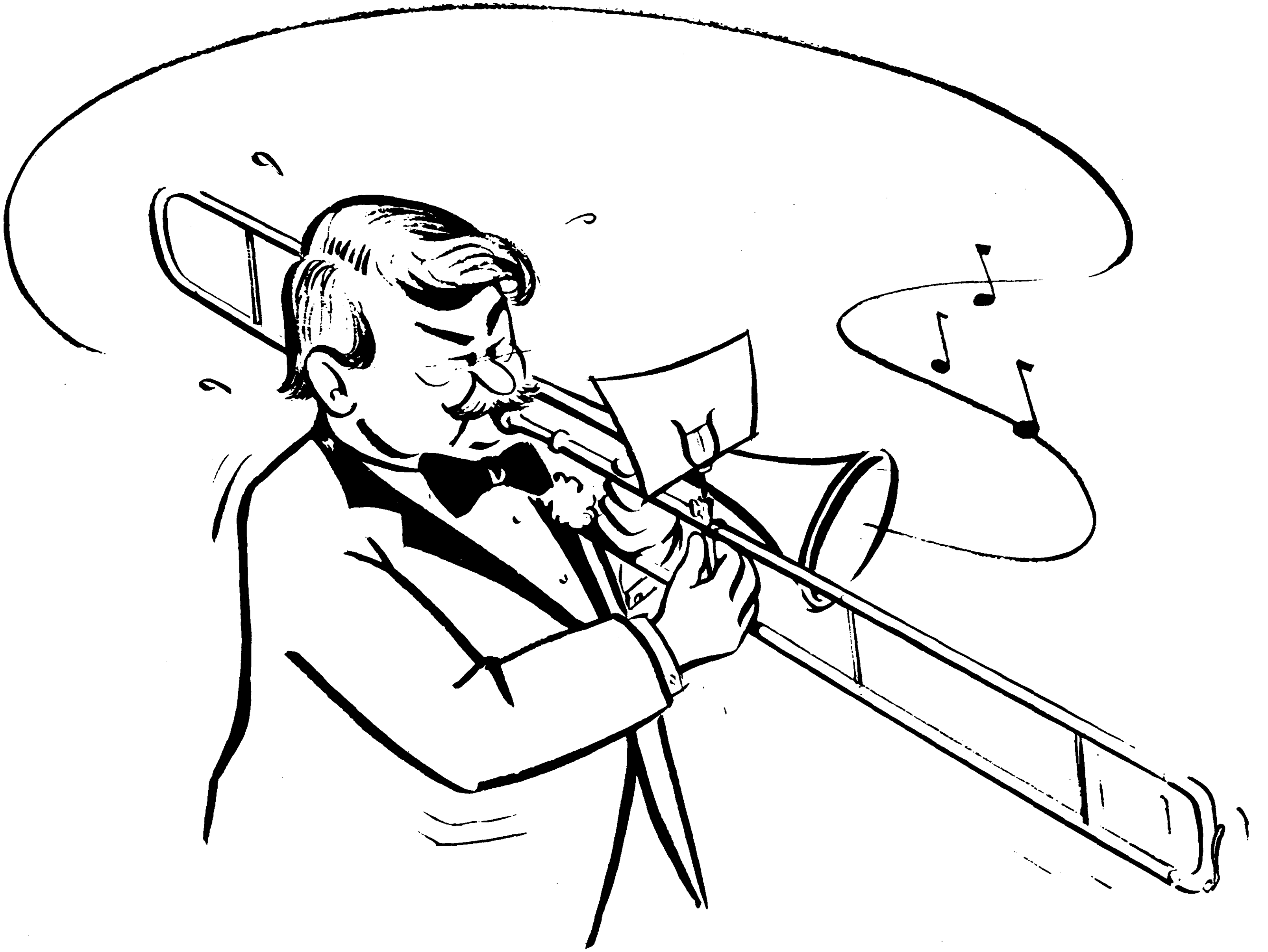
We need not stop here. All sorts of possibilities exist. Suppose the function $\mathbf{X} = \mathrm{SIN}\ (\theta^2 - 1)$ is to be used often in a program. The sequence of operations which compute this function (010214. . . . . 37) could be summarized by another 2-digit number (say 99) which will cause the proper sequence to be set up. In this way the programmer can generate his own pseudo macro instructions.

But now I feel I'm treading in the area of Fortran programming or of computer design and will close. I'd like to leave you with this thought: If you do see an engineer playing tunes on a slide rule or see an accountant's fingers pecking swiftly at the keys of a desk calculator, please hand him a pencil and ask him to start doodling.

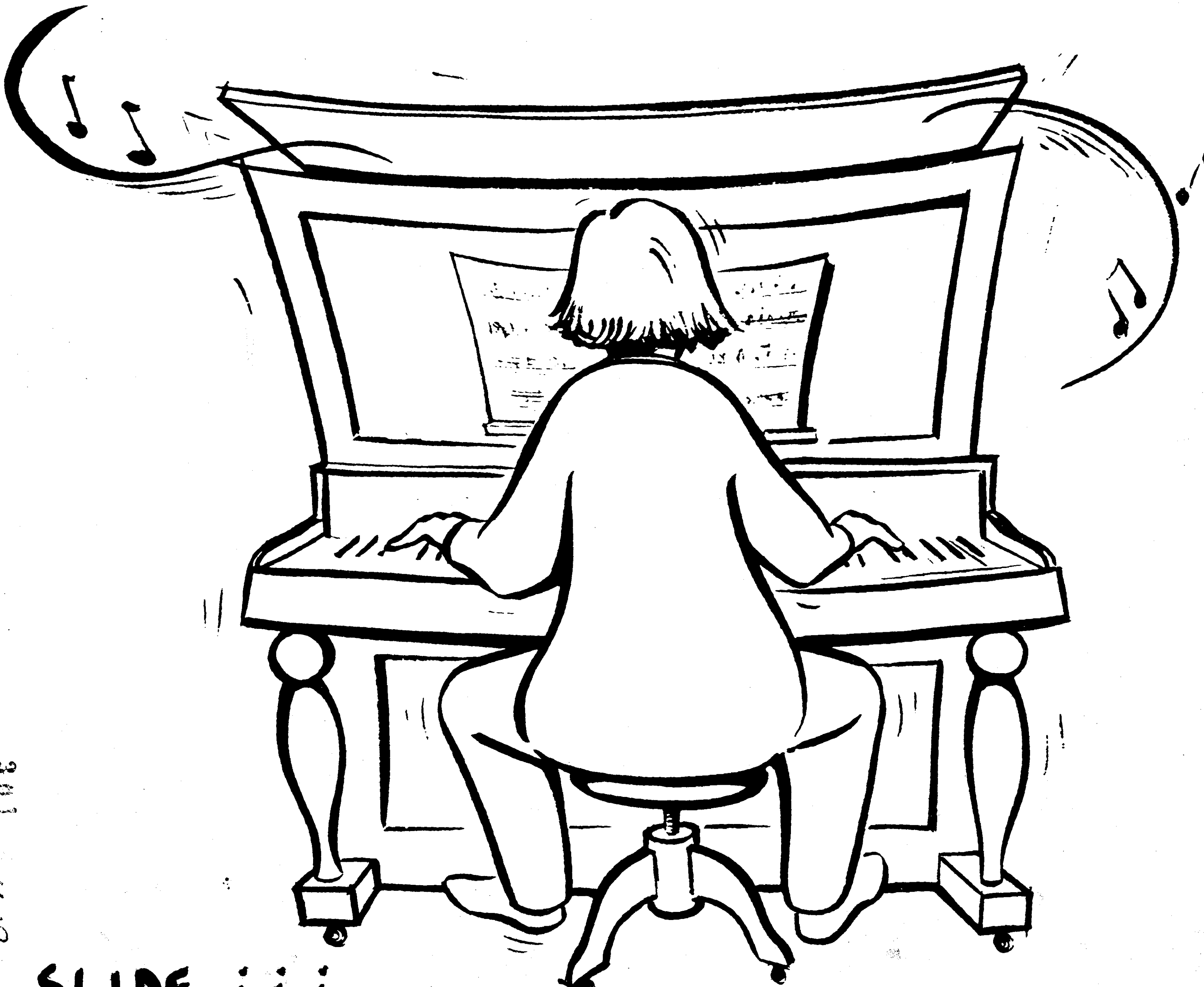
Doodle

299

4.19.10



SLIDE ii



SLIDE iii

301 4.19.1

DO YOU HAVE PROBLEMS?

do you do REPETITIVE WORK

on a desk calculator?

do your problems involve NUMBERS

and

WHAT HAS TO BE DONE with those numbers?

why not have a COMPUTER

solve your problems for you?

its EASY

SLIDE 1

first, tell the computer what <u>NUMBERS</u> you are going to use.

suppose they are:

191.25
1654.0
2.0
.3

---

on the form shown below, you'll see a column headed **OP** and a column headed NUMBERS

write LOD and 191. 25 on the first line under these columns.

do the same for the other numbers.

here's how the form will look after you have done this:

ITEM #	OP	A	B	C	R	P	NUMBERS	COMMENTS
101	LOD						191.25	
102	LOD						1654.	
103	LOD						2.	
104	LOD						.3	
105								
106								
107								
108								
109								

SLIDE 2

now tell the computer **WHAT TO DO**

to do this, you'll need some OP's.

you've already used one,

it was LOD.

some other OP's look like this:

ADD

SUB

MPY

DIV

you can guess what these mean.

SLIDE 3

suppose your problem is this:

(1) 191.25 ÷ 1654

(2) 2 x .3

(3) add the two results together.

the OP's you'll need are:

DIV

MPY

ADD

here's how you would write out the problem

ITEM #	OP	A	B	C	SR	P	NUMBERS	COMMENTS
101	LOD						191.25	
102	LOD						1654.	
103	LOD						2.	
104	LOD						.3	
105	DIV	101	102					
106	MPY	103	104					
107	ADD	105	106					
108								
109								
110								

SLIDE 4

you'll have to tell the computer to

print out item 107.

place a "1" in the column headed P

place END as the last OP

here is how the form looks when you have written out your problem completely:

ITEM #	OP	A	B	C	SR	P	NUMBERS	COMMENTS
101	LOD						191.25	
102	LOD						1654.	
103	LOD						2.	
104	LOD						.3	
105	DIV	101	102					
106	MPY	103	104					
107	ADD	105	106			1		
108	END							
109								

SLIDE 5

LIST OF OP's

OP	WHAT IT DOES
LOD	load a number into the specified item
ADD	item A + item B
SUB	item A - item B
MPY	item A x item B
DIV	item A ÷ item B
SIN	sine of item A (radians)
COS	cosine of item A (radians)
EXP	item A raised to the item B power
QRT	square root of item A
ATN	arctangent of item A (radians)
ABS	absolute value of item A
LOG	natural log of item A
END	last card of problem solution or of new set of numbers
*MUV	move item A
*RET	last OP in a subroutine
*BCH	branch to item A, B, or C
*SUM	sum from item A to item B

*For these OP's see chapter called "For the Advanced Doodler."

SLIDE 6

suppose you need the value of " π " in your problem.

you can tell the computer what the VALUE of " π " is

and

at the same time, write down a NAME for the π item.

you will then be able to refer to that item by the

NAME you have given it, as well as its NUMBER.

here's how to tell the computer to calculate the

area of a circle whose radius is 8.1623

the equation is $A = \pi r^2$

ITEM #	OP	A	B	C	SR	P	NUMBERS	COMMENTS
101	LOD			PI			3.14159	
102	LOD			R			8.1623	
103	MPY	R	R	RSQ				
104	MPY	RSQ	PI			1		
105	END							
106								

SLIDE 7

ITEM #	OP	A	B	C	SR	P		NUMBERS	COMMENTS
101	LOD			A				3.	
102	LOD			B				6.	
103	LOD			C				1.	
104	LOD			ZERO					
105	LOD			FOUR				4.	
106	LOD			TWO				2.	
107	SUB	ZERO	B	MB					
108	MPY	B	B	B2					
100	MPY	A	C	AC			D		
110	MPY	AC	FOUR	4AC			E		
111	SUB	B2	4AC	RAD			C		
112	QRT	RAD		ROOT			I		
113	MPY	A	TWO	DEN			M		
114	ADD	MB	ROOT	PNUM			A		
115	SUB	MB	ROOT	MNUM			L		
116	DIV	PNUM	DEN	X1		1			
117	DIV	NNUM	DEN	X2		1	R		
118	END						E		
119									

ITEM #	OP	A	3	C	SR	P	NUMBERS	COMMENTS
101	LOD			A			2.	
102	LOD			B			5.	
103	LOD			C			3.	
104	END							
105	END							
106								
107								

4.19.2(

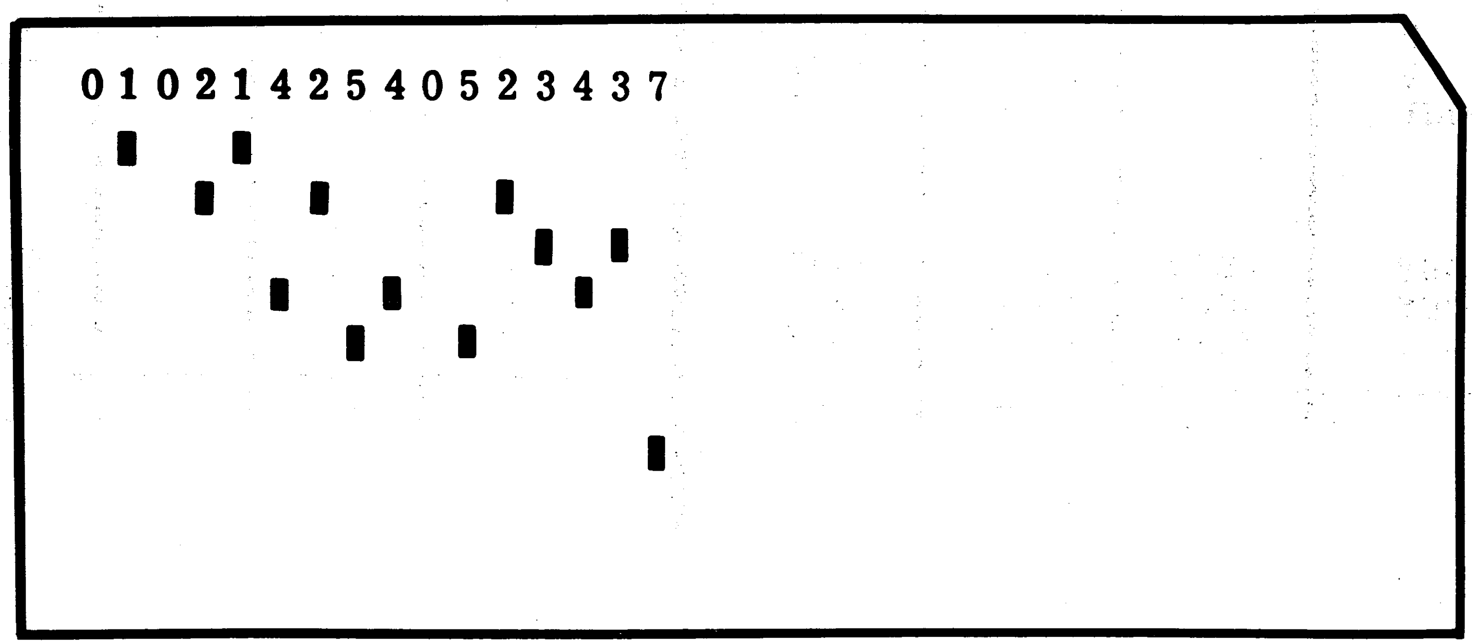
SLIDE 8

OP	A	B	C
LOD			A
LOD			B
LOD			C
.			
.			
.			
.			
MUV	A		ARG1
MUV	B		ARG1
MUV	C		ARG1
.			
.			
.			

OP	A	B	C
ADD	ARG1	ARG2	ARG1
SUB	ARG1	ARG2	ARG1
MPY	ARG1	ARG2	ARG1
.			
.			
.			
MUV	ARG1		ARG2
MUV	ARG1		A
MUV	ARG1		B
.			
.			
.			
MUV	ARG2		ARG1

SLIDE 9

OP NO.	OP	A	B	C	
01	LOD			A	
02	LOD			B	
14	LOD			ONE	
25	MUV	B		ARG1	
25	MUV	B		ARG2	
40	MPY	ARG1	ARG2	ARG1	
52	MUV	ONE		ARG2	
34	SUB	ARG1	ARG2	ARG1	
37	SIN	ARG1			1



SLIDE 10

PDQ FORTRAN
For Paper Tape 1620

Edited by

J. W. Trantum
P. G. Boekhoff -3193

MRD DIVISION
General American Transportation Corporation
7501 North Natchez Avenue
Niles, Illinois

Phone: 312 NI7-7000 Ext. 233

April 1964

We wish to acknowledge the generously given help of Frank H. Maskiell, and all associated with the development of P.D.Q. Fortran.

312

4.20.D

TABLE OF CONTENTS

312a

4.20.

I 1620 USERS GROUP LIBRARY PROGRAM ABSTRACT

Title: PDQ Fortran for Paper Tape

Author; Organization: J. W. Trantum, P. G. Boekhoff MRD-GATC

Date: April 20, 1964 Users Group Membership Code: 3193

Direct Inquiries to: J. W. Trantum, P. G. Boekhoff MRD/GATC
7501 N. Natchez Avenue, Niles, Ill.
Phone: NI 7-7000, Ext. 233

Description/Purpose: PDQ Paper tape is a modification of PDQ card system which generates shorter, faster object programs than any other system available for machines without Floating Point Hardware.

Specifications:

A. Storage: 1. Processor requires 17250 allowing 275 symbols

2. Class A subroutines require 6400

B. Equipment Required by System: Paper tape, Auto Divide, 20K. Program will adjust for additional memory.

Additional Remarks: Any program written in FO-004 or U.T.O. may be compiled in PDQ. Additional facilities in PDQ include: COMMON statements, batch compilation, continuation of I/Q and FORMAT statements, expanded format (A and D specifications) and multiple specifications (nFw.d), "Free Format" input, improved listing of source program, PROCEDURE statements, TRACE statements, MOVE statements, IF statement with continue feature, additional relocatable subroutines, Fixed Point ADD and SUBTRACT in line - No Fixed Constants in the Symbol Table.

313

4.20.1

II Language Specifications

Any proper statement in the FO-004 or UTO languages is allowed. List of Permissible Statements.

A. Arithmetic Statements:

A = B op C etc.

B. Control Statements:

1. GO TO n
2. GO TO (n_1, n_2, ----- n_m), i
3. IF (a) n_1, n_2, n_3
4. IF (SENSE SWITCH i) n_1, n_2

   (any of the n_i in 3 and 4 above may be replaced by the letter c)
5. DO n i = m_1, m_2 or

   DO n i = m_1, m_2, m_3
6. BEGIN PROCEDURE n
7. RETURN n
8. END PROCEDURE n
9. EXECUTE PROCEDURE n
10. CONTINUE
11. CONTROL m
12. PAUSE
13. STOP m
14. END
15. BEGIN TRACE
16. END TRACE
17. MOVE n, A, B or

    MOVE n, A, B, m

314

4.20.2

C. Input/Output Statements:

1. READ n, list

2. PUNCH n, list

3. ACCEPT n, list

4. PRINT n, list

5. TYPE n, list

D. Specification Statements:

1. COMMON

2. DIMENSION

3. FORMAT

Additional Specifications

A. PROCEDURES

A procedure is defined as a group of FORTRAN statements which are to be executed as a unit more than once in any larger program. Such a group of statements is preceded by a statement

BEGIN PROCEDURE n

where n is a procedure number. (Note - This number must not appear anywhere in the program as a statement number, and cannot be used in more than one procedure.)

The procedure statements must be followed by the statement

END PROCEDURE n

where n is the number of the procedure.

The procedure n may be executed at any time by the statement

EXECUTE PROCEDURE n

which transfers control to the statement immediately following the BEGIN PROCEDURE n statement and obeys the statements of procedure n until

315

4.20.3

a) the END PROCEDURE n statement is executed, which transfers control to the statement immediately following the EXECUTE PROCEDURE n statement that entered the procedure.

b) a RETURN n statement is executed. This returns control to the statement immediately following the EXECUTE PROCEDURE n statement that entered the procedure. A RETURN statement is valid only within the procedure from which it returns control.

Any statement may be included within a procedure, including EXECUTE PROCEDURE statements referring to other procedures, except:

a) BEGIN PROCEDURE, RETURN, or END PROCEDURE statements referring to other procedures.

b) EXECUTE PROCEDURE referring to the procedure within which it is placed.

B. CONTROL m

This statement controls the typewriter carriage

m	=	101	spaces typewriter
m	=	102	returns typewriter carriage
m	=	108	tabulates

C. PAUSE

The program execution is halted by a 48 op code in the main line program. Depression of the "START" key will cause the computer to continue to the next instruction.

D. STOP m

The program execution is halted after the carriage is returned and STOP m is typed. If m is omitted, STOP 0000 will be typed. Depressing "START" will cause the computer to continue to the next program instruction. m, if present, must be an unsigned constant, $0 \leq m \leq 9999$.

E. END

The program execution is halted after the carriage is returned and END is typed. Depressing "START" will cause the carriage to return and END to be typed.

F. TRACE

The results of all arithmetic statements may be traced without the generation of any additional in-line instructions. Replacement type statements (A = B, C = D(K), I = J) will not be traced since no arithmetic operation is required and, hence, FAC is not used. If the replacement statement includes a subscripted variable on each side of the equality, the statement may be traced. The tracing of arithmetic statements at object time is by the use of SWITCH 4.

> BEGIN TRACE will turn on the trace feature. This will replace the normal FMFAC instruction (26 ADDR, FAC) with a combination "FMFAC-TRACE" instruction (17 FMTR, ADDR). Each arithmetic statement thereafter will permit tracing. Tracing will be terminated when an END TRACE statement is encountered. Succeeding statements will not be traced until another BEGIN TRACE is read.

G. PRINT

PRINT n, list will cause the variables in the list to be printed according to the format specification n after a carriage return has been executed. This conforms with the PRINT statement of FO-004. No indexing is permissible within the list.

H. TYPE

TYPE n, list performs the same function as PRINT n, list with the exception that no carriage return will be executed prior to the output of data. This will permit the printing of columns of data utilizing the typewriter tab settings and the CONTROL 108 statement.

I. INPUT/OUTPUT LISTS

If the length of the list in an input/output statement requires more characters than are available before column 78, the list may be continued in the next statement. Terminate the list preceding a continuation with a comma. Continuation statements must have a digit (a continuation number of character) preceding the remainder of the list.

J. COMMON

The COMMON statement is followed by a list of variables whose object time addresses will begin in the first symbol location after the function symbols. This will enable the assignment of addresses to symbols common in several programs. Dimensioned variables which are listed in COMMON statements will have the necessary space reserved and must not be again listed in a DIMENSION statement. CAUTION: The COMMON statement(s) must be the first compiled statement in the program.

EXAMPLE: COMMON A,B, C,D(2,4),E

317

4.20.5

K. MOVE n, A, B or
MOVE n, A, B, m

The purpose of the MOVE statement is to transmit a specified set of fields to specified locations. m (if present) and n must be unsigned constants. A and/or B may be subscripted; the contents or the locations to be transmitted may be fixed, floating, or not. If m or n are specified to be more than 2 digits, they will be treated modulo 100.

If m is omitted or specified to be 1, n fields will be transmitted from A and (n-1) successively lower locations to B and (n-1) successively lower locations. If m is specified to be >1, n fields will be moved from every mth location beginning with A to every mth location beginning with B.

The MOVE statement generates 3 instructions in the object program (36 core positions).

Example:

```
MOVE 3, x(1), y(2), 4    (36 positions)
```

is the real way to say

```
DO 100 I = 1,9,4

100  y(I+1) = x(I)   (120 positions in line + 20 in
                       symbol table for I and 100)
```

Note: If n = 0, one field will be moved. If m = 0, it will be moved yet again.

L. IF(a) n_1, n_2, n_3

IF (SENSE SWITCH i) n_1, n_2

IF statements may have either statement numbers or the letter C (continue) in any of the positions in the transfer list. Statement numbers will be handled as they are in other systems. The letter C refers to the statement immediately following the IF statement. The statement following this IF need not be numbered unless referred to by some statement other than the IF. No transfer instruction or entry in the symbol table is generated by the letter C.

For example, the statements

```
IF (SENSE SWITCH 2) 6,5

5  A = B
```

may be written

318

4.20.6

```
IF (SENSE SWITCH 2) 6, C

A = B
```

M. <u>FORMAT</u>

FORMAT statements which are over 78 characters may be continued in the following statement. The statement to be continued must terminate with a comma following some specification. Continuation statements must have a continuation character preceding the remainder of the format specifications. Not more than 3 continuations are allowed.

Example:

```
100 FORMAT (I5, F8.4 ,
            1E14.8,
            2F8.4)
```

Permissible format specifications include I type (Iw), F type (Fw.d), E type (Ew.d), D type (Dw), A type (Aw), X type (wX), and H type (wH). All of the numerical field specifications and the A specification may have fields repeated by preceding the specification by the number of required repetitions such as (3F10.2) which is equivalent of (F10.2, F10.2, F10.2).

<u>I TYPE</u>: The I specification defines a field of w characters from which or into which data will be transferred. The data must be in fixed-point notation (maximum of 4 digits), however, the field width, w, is not so limited. On output, the sign of the variable will precede the digits, hence, a minimum field width of 5 is required to insure complete output of a fixed-point number. Should the field width be less than one plus the digits to be output, the low order digit or digits will be lost and the output data will be in error.

Example: Fixed point number minus 144 is output under an I3 specification. The result will be -14 which, of course, is incorrect.

<u>F TYPE</u>: The F specification defines a field width of w characters from which or into which floating-point data will be transferred. If a decimal point is not included in input data, the last d characters of the field will be recognized as decimal digits. On output, if the numerical value of the variable requires a field width greater than w to permit all non-decimal digits to be output, the specification will be automatically changed to an E type specification so that the output data will be complete.

Example: <u>+1728</u>. is the output under the specification (F6.0) of a variable. Had the specification been (F5.0), the output would have been +.17280000E+04.

319

4.20.7

(Note - the plus signs normally do not actually appear in output - see Modifications to the System, #3).

Variables in E type notation may be input under the F type specification.

E TYPE: The E specification permits the input or output of a floating point variable in the exponential form; that is, + .XXXE+XX. On output, 6 characters of the field are required for the sign, the decimal point, and the exponential notation.

Variables in F notation may be input under the E specification.

D TYPE: The D specifications permit input or output of data floating point variables in "excess 50" form. Only two forms of the D specification are acceptable: D10 for floating-point numbers which will permit transfer of a floating-point value e.g., 12, as $\overline{5}21200000\underline{0}$, and D4 which will permit transfer of a fixed-point number such as -123 as $\overline{0}12\overline{3}$.

(Note - Antique machines may require modification B/M 2115322 to print the Alpha character 50)

A TYPE: The A specification permits the input or output of up to five alphameric characters. The alphameric characters are converted to their two-digit form and stored in the symbol table as a 10-digit floating point variable. The width of the A field must not exceed 5. The number of alphameric characters input or output will be determined by the field width designation. The A format input will select the next available non-numeric character and the specified number of following characters

X TYPE (BLANK FIELD SPECIFICATION): Blank characters may be provided in an output record by means of specification wX where w is the width of the field to contain blanks. When the X specification is used with an input record, w characters are ignored.

H TYPE: The specification wH permits the w alphameric characters immediately following H in the specification to be input or output. CAUTION: the characters of an H specification may not be split on continuations. Should more characters than are available in a given statement be required in an H specification, the H specification on the first statement must include the count of the characters on the first statement only, and a new H specification must be stated in the format continuation.

Free-Form Input

The input accepted by this system has been relaxed so that variables need not correspond precisely with the field widths defined by the input FORMAT specification. The only requirements for the entry of data are that a space or blank column exists between data and that the data agree in mode (fixed or floating) with the FORMAT specification.

It is required that the letter E follow immediately the last digit of the mantissa when a variable is in the E form and no blanks are permitted until after the last digit of the exponent or the entry will be erroneous. Hence, the only acceptable E notations MM...ME+X, MM--ME+XX, MM--MEXX, MM--MEX, MM--ME-XX, MM--ME-X where M represents mantissa digits and X represents exponent digits.

Relocatable Subroutines

1. SIN and COS - evaluates sine(x) or cosine(x)

   Method - Hastings approximation with improved reduction to first quadrant.

   Length - 694 digits

   Speed - 128 msec.

   Accuracy - Error not more than one in last digit of mantissa for all angles, except sine of angles near $\pi/2$ and cosine of angles near zero, when error occasionally reaches two in last digit.

   Note: If all significance is lost, typewriter will indicate error condition, and machine will halt. Processing cannot continue.

2. EXP - evaluates e^x

   Method - Addition loop successively multiplies 1.0 by 2, 1.1, 1.01, 1.001, and 1.0001, reducing argument by the logs of those factors until argument mantissa is less than 0.0001; approximation $\exp(x) = 1 + x$ is then used.

   Length - 536 digits

   Speed - 76 msec.

   Accuracy - Error not more than one in last digit of mantissa for positive arguments, not more than two for negative arguments.

3. LOG - evaluates $\log_e (x)$

   Method - Addition loop successively multiplies argument by 2, 1.1, 1.01, 1.001, and 1.0001, accumulating negative logs of those factors until argument mantissa is greater than 0.9999; approximation $\log x = x - 1$ is then used.

321

4.20.9

Length - 678 digits

Speed - 79 msec.

Accuracy - Error not more than one in last digit of mantissa for arguments greater than 2.0 or less than 0.9; for arguments closer to one, loss of accuracy is inevitable, but routine is always more precise than is (x - 1).

4. SQRT - evaluates $\sqrt{x}$

Method - Odd-integer method generating result by adding complements.

Length - 378 digits

Speed - 116 msec.

Accuracy - Perfect, rounded to an eight-digit mantissa.

5. ABS - evaluates $|x|$

Method - load dividend to FAC.

Length - 14 digits

Speed - 1 msec.

Accuracy - Perfect

6. BAR - drops decimal digits of a variable (bracket function)

Method - obvious

Length - 104 digits

Speed - 4.2 msec.

Accuracy - Perfect

7. ATAN - evaluates arctan (x). Result is in the range

$$-\frac{\pi}{2} \leq \arctan(x) \leq \frac{\pi}{2}$$

Method - Table look-up and reduction of argument to less than 0.1; three-term Chebyshev approximation is then used. Result is complemented(if necessary) in fixed-point, avoiding loss of last digit in FSB.

322

4.20.10

Length - 818 digits

Speed - 126 msec. for argument less than 1
178 msec. for argument 1 or over

Accuracy - Error not more than one in last digit of mantissa.

8. SIGN - evaluates signum (x) $= \begin{Bmatrix} -1.0, & x < 0 \\ 0.0, & x = 0 \\ 1.0, & x > 0 \end{Bmatrix}$

Method - obvious

Length - 52 digits

Speed - 2.5 msec.

Accuracy - Perfect

9. ONE - evaluates the Heaviside unit step function 1(x) $= \begin{Bmatrix} 0.0, & x < 0 \\ 0.5, & x = 0 \\ 1.0, & x > 0 \end{Bmatrix}$

Method - obvious

Length - 76 digits

Speed - 2.2 msec.

Accuracy - Perfect.

Error Messages During Processing

PDQ FORTRAN has a limited ability to detect errors in the grammar and syntax of FORTRAN. This capability is not adequate and is included mainly to ensure that the processor does not destroy itself.

The PDQ FORTRAN error messages are:

Error No.	Condition
1	Incorrectly formed statement
2	Subscripted variable for which no DIMENSION statement has previously appeared in the program, dimensioned variable used without subscripts, variable in DIMENSION statement has already appeared in the source program.

323

4.20.11

Error No.	Condition
3	Floating point number not in allowable range of values, or fixed point number contains more than four digits.
4	Symbol table full.
5	Mixed mode expression.
6	Variable name in an expression contains more than five characters.
7	Switch number has been omitted in an IF (SENSE SWITCH n) statement.
8	A comma follows the statement number in a DO statement.
9	A DIMENSION statement ends with a comma, more than two dimensions have been specified in a DIMENSION statement.
10	Unnumbered FORMAT statement.
11	Incorrect representation in a FORMAT statement in one of the following ways:
	a. Special character (= @ - * + . ,) in numerical field specification.
	b. Alphabetic character other than D, E, F, I, or A in a numerical field specification.
	c. Decimal point missing in an E or F-type numerical field specification.
	d. The number of positions to the right of the decimal point has not been given in an E or F-type numerical field specification.
	e. A record mark appears in a numerical field specification or an alphanumeric field.
	f. The first character following the word FORMAT is not a (.
	g. The last character is not, or) .
12	The total record width specified in a FORMAT statement is greater than 99 characters.
13	A FORMAT statement number has been omitted in an input/output statement.

324

4.20.12

III Operating Instructions

1. Clear Core

2. Load PDQ Processor 36 00000 003

3. Switch Settings

Parity and I/O to Stop, OFLOW to program

Program Switch	On	Off
1	List ONLY object time addresses of source statements	List source statements and symbol table
2	Do not list names and locations of relocatable subroutines (functions)	List complete symbol table
3	Do not compile subroutines on object tape	Compile subroutines on object tape
4	Source program is on tape	Enter source program on typewriter

Note: If switch 1 is on at the end of the compilation, the symbol table will NOT be typed, but the highest address below the object-program-symbol-table will be typed.

Error Procedure

If a source statement error is encountered the processor will type an error message and halt. Pressing START will transfer control to the typewriter so that the statement may be correctly typed in. Sense Switch 4 serves its usual role with regard to typing errors and must be turned off before release-start is pressed. In some circumstances a source program error may result in a MAR CHECK. In this event one may enter a branch to the error routine (4913184) and proceed as above.

Debug and Protect Feature

The symbol-table locations for which entries are not defined at compile time will be initialized to flag-record marks in low address (high-order address) of each symbol location at the time subroutines are loaded. This will protect the subroutines and the remainder of the symbol table from the disastrous effects of referring to undefined symbols (fixed variables excepted). Further, the record mark will generate a MAR CHECK if an undefined symbol is encountered by the subroutines.

325

4.20.13

Object Program Errors

Error conditions detected by the subroutines generate the following error messages and results:

Condition	Message	Contents of FAC	Operation
Overflow in fixed or floating arithmetic	OFLO	NINES	continued
Underflow in fixed or floating arithmetic	UNFLO	ZERO	continued
Overflow in EXP	EX OFLO	NINES	continued
Underflow in EXP	EX UFLO	ZERO	continued
Negative argument in SQRT	I AIN'T REAL	$\sqrt{\lvert \text{argument} \rvert}$	continued
Argument out of range in SIN-COS	Function Undefined - Beyond Allowable Range (FUBAR)	argument	terminated
Zero argument in LOG	LOG 00	MINUS NINES	continued
Negative argument in LOG	I CAN'T TAKE YOUR LOG	$\log \lvert \text{argument} \rvert$	continued
Input out of range	(none)	NINES	continued

Fixed Point Arithmetic

The operators +, - and * in the fixed point mode give results mod 10000. Hence, the addition of 9998 and 4 gives 2 with no error detection due to the overflow. Should division by 0 be attempted, an error overflow will be indicated.

326

4.20.14

IV Physical Description

Processor tape consists of the following records

1. Loader and identifier (numeric)

2. Tables and program (numeric) dump of locations 00100 through 17850.

3. Subroutine names (alpha)

Subroutine tape consists of the following records (all numeric)

1. Loader and identifier

2. Tables

3. Relocator (dump or locations 00500 through 02553)

4.a Relocatable subroutines (many card images)

4.b End-of-relocatables record

•5. Class A subroutines (dump or locations 00100 through 06367)

6. Loader

7. Symbol-table initializer

8. I/O buffer initializer

9. Type message, halt, and branch to 06400

A duplicator program is provided with the system. It will duplicate both the processor and subroutines.

Adding Relocatable Subroutines

The name of the subroutine must be added to the last record of the processor. Its order number in that record is the subroutine # + 1. Subroutines must be written and processed in an SPS which produces card-image output e.g., 1620-1710 SPS. Subroutines must begin with DORG 5000. Flags on the zero and one positions of each instruction will cause the P and Q addresses respectively to be relocated. No other flags may be used in addresses to be relocated except for positions 6 and 11. Flags in those locations will generate proper indirect addresses.

327

4.20.15

To assemble a relocatable subroutine, delete all records from the SPS output except those containing information generated by program instructions; i.e., delete the loaders and tables. Then add header and trailer records as follows:

Header Record

XX (Subroutine No.)   62 blanks 0≠

(Subroutine No. must be less than 23)

Trailer Records

1. Memory-capacity digit records

   64 blanks $\overline{x}$xxxx≠ (location of any digit which must be adjusted for memory size.

2. Subroutine length record

   $\overline{x}$xxxx (length of subroutine - must be even) 57 blanks $\overline{0}$000≠

<u>Modifications of the System</u>

The following modifications to the system may be incorporated at the user's discretion.

1. <u>Compilation Errors</u>

If the user prefers not to correct source program errors at compile time he may (1) ignore the erroneous statement by changing the digit in location 13241 of the processor to 9, or (2) terminate compilation by changing the contents of locations 13241-6 to 909860.

2. <u>Trace Format</u>

To eliminate printing of the 4-digit address of the traced variable, change subroutine locations 02694-5, 02736-7 to 41.

To modify the Format for Fixed and Floating variable change the fields at 2327 and 2359 respectively to the desired specifications.

3. <u>Plus Signs in Output</u>

Plus signs are normally omitted in both tape and typewriter output. To obtain plus signs in output change the digit in location 03196 of the subroutines to $\overline{1}$ for floating output, change 04694 to $\overline{1}$ for fixed output.

328

4.20.16

Object Time Error Modification

Any error condition message may be deleted by changing the carriage return and output instruction op-codes to 41. The system now has a NOP instruction following each of the instructions which write error messages. These may be changed either to Halt instructions (48) or to branch to END (4902622) to terminate the program.

329

4.20.17

V Error in the System

There is one known error in PDQ FORTRAN inherited from FORTRAN with FORMAT. The statements

```
A = - B ** C or

A = - B ** I
```

will compile as if they were

```
A = (-B) ** C or

A = (-B) ** I
```

To obtain the desired result, it is necessary to write

```
A = - (B * * C) or

A = - (B * * I)
```

3 3 0

4.20.18

SEX FORTRAN

For Paper-Tape 1620

Edited by

P. G. Boekhoff - 3193

MRD DIVISION
General American Transportation Corporation
7501 North Natchez Avenue
Niles, Illinois

Phone: 312 NI 7-7000 Ext. 233

May 1964

The author wishes to acknowledge the assistance of John W. Trantum, Frank H. Maskiell, John W. Holmes, Messrs. Forbisher and Laroche, and all others whose work has made possible the development of this system.

331

4.21.D

I THE CONCEPT OF SEX

SEX FORTRAN is a system based on, and largely identical to, PDQ FORTRAN for Paper Tape. To the best of the author's knowledge, no one presently disputes the claim that PDQ is the most efficient and powerful FORTRAN in existence for a poor man's 1620 (20K, auto divide); this is a sufficient (though not entirely the actual) reason for taking PDQ as a starting point.

One reason for the writing of SEX FORTRAN is that any FORTRAN must be less than maximally efficient, and that many programmers are sophisticated enough to recognize and circumvent (e.g., by judicious patching) the inherent inefficiencies. In most systems, this requires the programmer to operate outside the language. SEX is designed to give this type of programmer a head start toward program efficiency by providing several new, frightfully nonstandard statements in the language, and by streamlining the treatment of parts of the standard FORTRAN language.

A second reason for SEX is that, in several instances at the author's installation, programs have appeared which have overlapped by several thousand locations - too much to permit a simple tightening of the source program, but small enough to suggest that there should be a reasonable way to squeeze them into core.

SEX provides an answer in two ways. First, for the programmer who does not wish to stray from conventional FORTRAN, any program in Olde FORTRAN, FO-004, UTO, or PDQ may simply be recompiled in SEX from the original source.* This will save from 3500 to perhaps 8000 locations on a large program, depending upon the system against which a comparison may be made. For example, any program requiring 25,000 locations in UTO, or about 23,500 in PDQ, will almost certainly fit a 20K machine in SEX. The SEX object program begins in location 03100, and both the object program and the symbol table are generally shorter (never longer) than those generated by any of the above systems.

Second, for the benefit of the programmer who has outgrown the idea that the 1620 is a mysterious black box, the more powerful language of SEX will permit corners to be cut in the source program.

The other side of the question of efficiency is time. Exclusive of I/O, running time in SEX is slightly but not significantly faster than PDQ. The I/O subroutines process faster in SEX; actual time spent on the I/O operations depends in other systems on the format, which is rigidly fixed in SEX. Thus no direct comparison is possible. The author claims, however, that SEX in general does yield a faster as well as a shorter object program than any other system for this simple-minded machine configuration.

---

* A few statements are not accepted by the SEX processor: SAY and OUT

II THE TECHNIQUE OF SEX

The best way to permit a significantly longer object program is, of course, to remove unnecessary instructions from the Class A subroutines. Since the author did not wish to cripple any of the PDQ arithmetic subroutines, the input-output routines (some 3500 locations) were discarded and replaced by a set of rather austere I/O subroutines requiring only 350 locations. This is the greatest space-saving innovation in SEX, and it is from the two new statements corresponding to these routines (SELECT and EXTERNAL) that the system derives its name.

Accompanying the adoption of this new set of I/O routines there are of course restrictions on data format. All numeric data must appear in excess-fifty or in 10-digit fixed-point representation (DIO format), one value per record. Alphameric data, in A5 format, may also be input or output if desired. Since all format (except typewriter control) is strictly defined,there are no FORMAT statements in the system (they will be accepted but ignored by the processor).

The arithmetic subroutines for fixed-point addition and subtraction have been eliminated. The add and subtract instructions now appear in line. This results in a saving of 30 locations in the Class A subroutines, at the expense of one instruction (12 locations) in the relocatable LOG subroutine.

Further, smaller savings of space were introduced by a slight change in the STOP routine and by the author's willingness to accept a somewhat questionable format for TRACE output (these are described in Section IV below).

The symbol table has been shortened in SEX by the elimination of fixed-point constants (these now appear in line) and by ignoring FORMAT statement numbers. The adoption of the PDQ "continue" feature for IF statements makes possible the elimination of many statement numbers from the source program (and hence from the symbol table).

The object program itself is for the most part the same as that generated by PDQ (i.e., shorter than in any other system), with two exceptions: FORMAT statements are ignored, and fixed-point arithmetic statements of the form $I = I \pm J$ (where J is any fixed-point expression) are compiled correctly by SEX, which saves one or two in-line instructions depending on the form of J.

In many programs, logic is determined by defining variables which function only as binary switches to be tested by the program. To eliminate the obvious waste of reserving 10 digits in the symbol table for each such creature, SEX provides a bank of ten single-digit SENSE LIGHTS, which are described in Section V.

For the sophisticated programmer, it is possible in SEX to save further space in cases which are beyond the limited psychic capabilities of the SEX processor. Provision is made for the programmer to patch his program at compile time. The HANG PATCH statement is discussed in Section V.

III THE LANGUAGE OF SEX

The following is a list of permissible SEX statements. For the most part, these are the usual statements found in any 1620 FORTRAN, from FO-004 on up. Those which were introduced (or altered in meaning and/or treatment) by PDQ have been marked with a single asterisk. Statements marked with a double asterisk are peculiar to SEX.

A. Arithmetic and Data Transmission

A = B op C etc.

* MOVE n, A, B or

* MOVE n, A, B, m

B. Input/Output and Format

** SELECT, wxyz

** EXTERNAL, List

ACCEPT n, List

PRINT n,List

PUNCH n, List

READ n, List

*TYPE n, List

FORMAT ( )

CONTROL n

C. Program Control and Logic

CONTINUE

GO TO n

GO TO (n_1, ..., n_j), i

DO n i = m_1, m_2 or

DO n i = m_1, m_2, m_3

334

4.21.3

** LIGHT i ON
OFF

** LIGHTS ON
OFF

* IF (Expression) n_1, n_2, n_3

* IF (SENSE SWITCH i) n_1, n_2

** IF (SENSE LIGHT i) n_1, n_2

PAUSE

STOP or

STOP n

EXECUTE PROCEDURE n

BEGIN PROCEDURE n

RETURN n

END PROCEDURE n

END

D. Declarative and Miscellaneous

** HANG PATCH

DIMENSION

* COMMON

* BEGIN TRACE

* END TRACE

All statements are interpreted and compiled as they are in PDQ FORTRAN except as herein noted.

IV NEW TREATMENT OF STATEMENTS

Most statements are handled by SEX as they are by PDQ. The exceptions:

A. ACCEPT, PRINT, PUNCH, READ, TYPE - each of these I/O statements is compiled in the "standard" manner, subject to the format restrictions of SEX; i.e., each generates a SELECT instruction (q.v.) defining the format that I have decreed to be the most "natural", followed by an EXTERNAL instruction for each list variable. AN I/O statement with no list (such as PRINT 1) will generate (a) a SELECT instruction in the object program, and (b) a compilation error. The error routine allows the operator to leave the SELECT instruction in the object program or to overlay it with the next statement.

   I/O continuation statements (as in PDQ) are accepted by the system.

B. FORMAT - the SEX processor will accept but ignore FORMAT statements and continuations (their statement numbers will not appear in the symbol table). All I/O defined by the statements listed in paragraph A must be numeric, in DIO format. Alphameric I/O (in A5 format) is also allowed but must be specified by a SELECT statement.

C. STOP or STOP n - The only innovation here is the abbreviation of the printed message to S NNNN.

D. TRACE - The format for trace output has been modified to AAAAAVVVVVVVVVV for floating and AAAAAGGGGGVVVV for fixed variables, where

   A is the address of the variable,

   V is the value of the variable, and

   G is whatever garbage may be in the accumulator.

V TREATMENT OF NEW STATEMENTS

The following is a description of the new statements mentioned in Section I.

A. SELECT, wxyz

This is the statement which defines format and controls all input and output. The arguments are as follows:

$$w = \begin{cases} 0 & \text{for numeric input} \\ 1 & \text{for alpha input} \\ 2 & \text{for numeric output} \\ 3 & \text{for alpha output} \end{cases}$$

$$x = \begin{cases} 1 & \text{for typewriter} \\ 2 & \text{for paper-tape punch} \\ 3 & \text{for paper-type reader} \end{cases}$$

$$y = \begin{cases} 0 & \text{for no control before first list item} \\ 1 & \text{for space before first list item} \\ 2 & \text{for carriage return before first list item} \\ 3 & \text{for tab before first list item} \end{cases}$$

z = same as y; applies to all items in list except the first.

The polypotent SELECT statement generates one instruction in line which sets up the I/O subroutines to handle the list variables in succeeding EXTERNAL statements. The arguments corresponding to "standard" I/O statements are:

ACCEPT - 0123

PRINT - 2123

PUNCH - 2200

READ - 0300

TYPE - 2133

B. EXTERNAL, List

This statement generates, for each list variable, an instruction to input or output the variable as specified by the last SELECT statement encountered. As an example of the use of SELECT and EXTERNAL statements, consider the following two programs:

```
C         PROGRAM I (PDQ)
          READ 1506*, A, B, C
          PRINT 1506, A, B, C
1506      FORMAT (D10, 5X, D10, 5X, D10)
          END

C         PROGRAM II (SEX)
          SELECT, 0300
          EXTERNAL, A, B, C
          SELECT, 2123
          EXTERNAL, A, B, C
          END
```

The operation of these programs is identical (in fact, the SEX object programs generated would be identical). In Program II, the first SELECT statement tells the I/O subroutines that data are to be read numerically (w = 0) from paper tape (x = 3), with no typewriter carriage control (y = z = 0). The first EXTERNAL statement causes the values of 3 variables to be read and stored. The second SELECT tells the subroutines to write numeric (w = 2) on the typewriter (x = 1) with a carriage return preceding the first item on each EXTERNAL list (y = 2) and a tab preceding each succeeding item (z = 3). The final EXTERNAL statement causes the values of the list variables to be read from storage and output as specified.

C. LIGHT i ON (or OFF)

The function of this statement is obvious except for the definition of the symbol i, which can be either a single digit (0 through 9) or the name

* cf. Forbisher-Laroche

of a variable presumably (though not necessarily) defined elsewhere. If a variable, the light affected will be that corresponding to the low-order digit of the value of the variable at the time this statement is executed. If the variable is fixed-point, this will be the units digit. Use of a floating - point variable in this statement is permissible, sometimes useful, and usually dangerous. The sign of the variables will have no effect unless the program is being run with IA turned on, in which case the author disclaims responsibility.

Example: The sequence

```
      Q = 0.0
      LIGHT Q ON
      DO 1 IDIOT = 6903, 6907, 2
1     LIGHT IDIOT ON
      LIGHT 9 ON
```

will cause lights 0, 3, 5, 7, and 9 to be turned on, in that order.

Loading the subroutines to start the program will reset all lights OFF. If any light is turned on, it may be turned off again only by means of an OFF statement; interrogating a light does NOT turn it off.

D. LIGHTS ON (or OFF)

This statement simultaneously sets (or resets) all 10 lights.

E. IF (SENSE LIGHT i) n_1, n_2

As in the ON and OFF statements, i may be the name of a variable; as in other IF statements, one (or both) of the statement numbers may be omitted. If the light referenced is on, the program branches to n_1; if off, to n_2.

If both statement numbers are omitted AND i is the name of a variable, the object program may not run properly.

The continuation feature for all IF statements has been modified from PDQ; the letter C for the continuation may be replaced by anything (or nothing) neither beginning with a numeric digit nor including a comma. This should be of great value to those of us who delight in writing fudge recipes and/or bad poetry in our programs.

339

F. HANG PATCH

This statement allows a machine-language patch to be entered at compilation. When a HANG PATCH statement is encountered, the SEX processor transfers control to the typewriter to accept the patch, the idea being that the map listing being produced will make the necessary addresses available.

Two of the more immediately obvious uses of this statement are for alpha output (such as column headings) and for direct branches within the object program to avoid using symbol-table space for statement numbers.

A few words of caution are in order:

(1) normally, no record mark should appear in the patch. In this case, the last digit entered at the typewriter will be the last digit inserted into the object program;

(2) if a record mark with no flag appears in the patch, it will go into the object program as the last digit of the patch, and anything following the record mark will be ignored;

(3) if a <u>flagged</u> record mark appears in the patch, the preceding digit will be assumed to be the last digit of the patch;

(4) the processor will accept unflinchingly a patch with an odd number of digits. Care is advised.

340

VI DESCRIPTION AND OPERATION OF THE SYSTEM

The processor occupies locations 00100-17480, permitting 252 entries in the symbol table. The Class A subroutines end in location 03099. Relocatable subroutines are stolen directly from PDQ (q.v.).

Errors detected at compile time are for the most part the same as those of PDQ, UTO, etc. The routine for handling them is a bit different. When an error is detected by the processor, a message is typed and the machine halts. If Switch 2 is turned on and the START key depressed, the carriage will return and a corrected statement may be entered at the typewriter. If Switch 2 is off when START is depressed, the erroneous statement will be ignored and the next statement will be read in.

This error routine and the existence of the HANG PATCH statement make it advisable for the programmer to be available at compile time. Simple patches are provided with the system to permit changing these procedures for closed-shop operation.

Except as noted here, operation of SEX is the same as with PDQ. Details and instructions are provided with the system.

THE ART OF DEBUGGING

E. J. Orth, Jr.

Southern Services, Inc.

Birmingham, Alabama

3 i -

4, 22, 0

The Art of Debugging

E. J. Orth, Jr.
Southern Services, Inc.
600 North 18th Street
Birmingham, Alabama 35202

This talk is based on experiences in writing and debugging large FORTRAN II programs at the Computer Center at Southern Services, Inc., in Birmingham, Alabama. As such, it is the sum total of the experiences of several people. Although specific reference will be to FORTRAN II oriented debugging, the techniques may be applied to any system. No claims are made for new ideas. Rather it is hoped that this representation of old ideas in what is hopefully a new framework will be useful to all concerned.

FORTRAN II provides two types of trace: An equals trace, and a logical trace of sorts. The equals trace prints the result of each arithmetic statement; the logical trace prints the results of computations specified within an IF statement. The normal output mode is the typewriter; however, an object time modification described in The 1620 Users Group Newsletter, Vol. 1, No. 2, allows conversion to punch mode. The normal FORTRAN trace will be called the micro trace in the rest of this discussion.

A long program involving many computations would require reams of micro trace to reach a bug occuring late in the program. The following techniques may be used to simplify the problem:

1. The subprogram concept
2. The PAUSE N statement
3. A special debugging subprogram.

The Subprogram Concept

Subprograms can materially cut the volume of a one for one trace. After a subprogram is checked out, it may be compiled in the no trace mode, thus eliminating all trace output from that subprogram at mainline debug time.

Subprograms need not be restricted to repetitive routines. Sections of the mainline logic may be checked out and CALLed as a subprogram.

The PAUSE N Statement

Use of this statement at strategic points in the program, each with a unique N, will allow the programmer to step through his program observing N in OR2. When a trouble spot is reached, trace may be initiated. Thus the PAUSE N statement allows an awkward type of macro trace.

A Special Debugging Subprogram

A more useful type of macro trace may be provided by a single argument subprogram whose function is to type and punch "STNO I" (where I is the argument) and pause. Type/punch of "STNO I" would indicate that the program is about to execute statement number I. Such a subprogram may be CALLed at the beginning of each main logical block. The argument would be the statement number of that logical block. Let us call this subprogram STNO.

The utility of STNO may be expanded by addition of a dump routine such as No. 1620-01.6.093. This dump routine is in the form of a library function, and is relocated from the subroutine deck by J = DUMP(J). J, the argument, is an integer of four digits, and specifies the section of core to be dumped. For instance if J is 1214, all core from 12000 thru 13999 will be dumped in 5/75 format. With the proper argument, the library function DUMP may be used to take snapshots of selected portion of core such as the symbol table or the common storage area, on up to all of core.

As a final refinement, both the STNO subprogram and the use of the DUMP routine within that program may be placed under console switch control. Thus, the entire subprogram may be bypassed, or only the DUMP routine bypassed. See flow chart in appendix B. The STNO debugging tool may now be used as follows:

1. A macro trace of the logic through the punch/type "STNO I"
2. Selected core dumps
3. Micro traces as desired, selected through CS4 at the proper "STNO I" type.

If punched micro trace is used, the punched "STNO I" will allow the programmer to readily find his place. Punched micro trace is most definitely recommended to save computer time.

If an on-line printer is available, read print for punch in the above.

The usefulness of selective core dumps cannot be overemphasized. A prime example concerns matric manipulation. A snapshot of the work area before and after completion of a matric operation may be compared to the micro trace of a FORTRAN statement. Indeed, a subprogram such as STNO incorporating some sort of dump routine is absolutely necessary in programs involving matrix algebra. STNO and DUMP are to matrix algebra as the FORTRAN micro trace is to arithmetic statements.

Another Use of DUMP

The following sequence may be placed at the very beginning of a program during debug time to allow variable length core snapshots at will.

```
1 FORMAT (I4)
  IF (SENSE SWITCH 4) 2,3
2 ACCEPT1,I
  I=DUMP(I)
  PAUSE 11111
3 CONTINUE
```

Flow Charting and Coding

Certain simple documentation practises will speed debugging. For instance, each page of the flow chart should be numbered. It is most useful if each numbered statement concerned with the logic on a given page use the flow chart page number as a hundreds digit.

Each logic block on the flow chart should be cross-indexed to the FORTRAN source program by statement number. That is, the number of the first statement in a block should appear above that block on the flow chart.

The obvious utility of comment cards need not be belabored here. 344

The FORTRAN statement STOP N is most useful as an error trap. Each N should be unique, and a table of N, statement number, and type of error should be prepared. STOP N should be used to trap unexpected, illogical conditions. For instance in testing a variable in an IF statement: If the variable may be positive or zero but never negative, use STOP N on negative. Programmed switches and the computed GO TO are very convenient. When testing a set of switches, STOP N may be used to trap theoretically impossible conditions. The only cost is the statement number, and the STOP statement. The STOP N statements may be removed when the program is debugged and no errors will occur.

Conclusion

Some sort of disciplined debugging procedure must be adopted in order to assure optimum utilization of programmers, operators, and equipment. If we are to grow in our profession, we must adopt and submit to this discipline. The days of bit chasing and typewriter core dumps are fast disappearing. We must be ready to program and debug without touching or seeing a computer. A painful thought? Yes. But have you heard of System 360?

APPENDIX A

When a card punched as below indicated is LOADED after a FORTRAN II object deck is loaded, all micro trace will be in the punched mode. The output may be listed 80-80 on the 407, with a card taped over printwheels N thru 80 to block printing of garbage. N=ff+3 or kk+1, whichever is greater. ff is floating precision; kk is fixed precision.

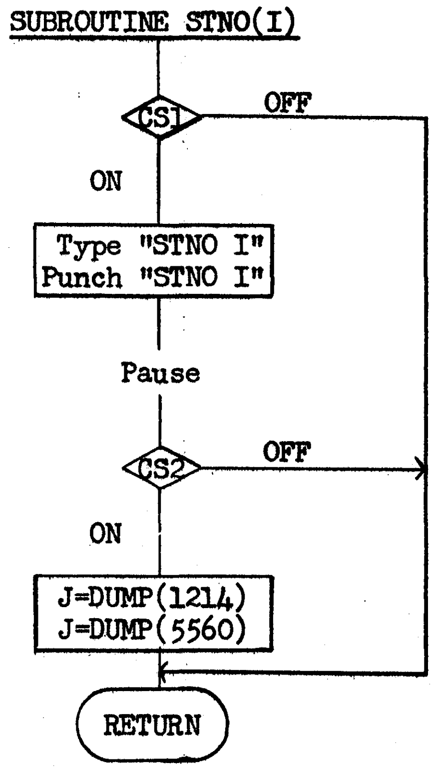
Procedure: Reset, LOAD this card. Typewriter will print "PCH TRACE MODE". Then branch to beginning of object program.

16	06701	000$\bar{4}$1	Replace RCTY with NOP
15	06733	00004	Replace WNTY with WNCD
15	06753	00004	Replace WNTY with WNCD
39	00051	00100	Print "PCH TRACE MODE"
48	57434	80063	
59	41434	50054	
56	44450	≠	

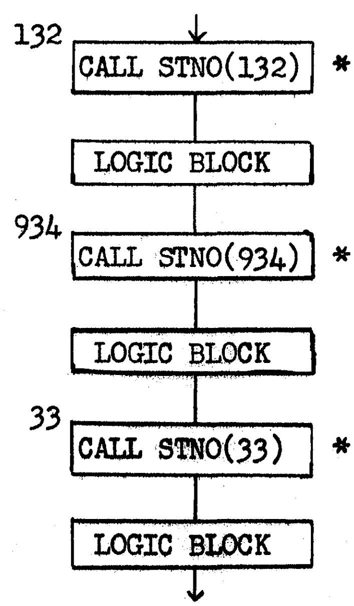
In most cases it is desirable to permanently modify the processor to punch trace.

APPENDIX B

One possible configuration of the STNO subprogram is shown. In this version, CS1 controls type/punch of "STNO I" and CS 2 controls DUMP.



EXAMPLE:



* These may be removed after debug time.

S A M P

Search and Memory Print

Jack M. Wolfe

Brooklyn College

Brooklyn, New York

347

4.23.0

(S A M P) SEARCH AND MEMORY PRINT

Jack M. Wolfe   Brooklyn College, Brooklyn 10, N.Y.   (User #1026)

DESCRIPTION

This utility program is available from the 1620 General Program Library under # 1.6.100. It can be used as an aid in debugging machine language programs in which the successive instruction locations are stepped up uniformly be 12 and in which the instructions are not interspersed with record areas. If these conditions do not prevail for the entire program, however, this utility program can be applied separately to each individual portion of the program for which these conditions are satisfied.

The memory print part of this program will write out on the typewriter any selected portion of the memory in either instruction format or data format. In instruction format the typeout contains the location, the OP code, the P-address and the Q-address, appropriately spaced for ease of reading. Flags and record marks are typed out. Each instruction is typed on a new line.

In data format each line contains the location and the contents of the ten memory positions starting at that address. Because the error in the program will often become apparent before the completion of the memory print, the typeout, whether in instruction or data format, can be terminated orderly by program switch control at any time in the processing.

The search part of this program may be particularly useful when the program being run stops on an invalid operation code due to a branch to a location that is actually not the beginning of an instruction. By displaying IR1 and subtracting 12 from that address, we know the address of the instruction with the invalid OP code. The SEARCH can then be used to search the instructions part of the program being debugged for all branching instructions whose P-part is the address of the instruction with the invalid OP code. The SEARCH will type out the locations and the instructions of all conditional and unconditional branches to that address. It takes indirect addressing into account to one level when an indirect address appears as the P-part of a branching instruction.

As a by-product of the search, the program types out all immediate instructions found without a flag in the first four positions of the Q-part of the instruction, except, of course, for the OP code 15. The locations of the typed instructions are also typed out.

(S A M P) continued

Sometimes it will be found that the search is completed without finding any branching instruction that could have caused the program to branch to the address being sought. This may be due to the fact that a branch might have been made to a location that is 12 positions lower than the address where the program hung up. If the 12 digits beginning at that address constituted an executable instruction, the computer would not have hung up until the following instruction was reached. Thus it is desirable to perform a memory print starting several instruction lengths before the address where the program hung up. This will reveal whether a branching might have taken place to an address preceding the address of the invalid OP code. With this information, the SEARCH can be planned more effectively. Quite often, moreover, the memory print itself will reveal the actual error and the SEARCH will then be made only when it is apparent that the program could have reached the invalid OP code only by branching to it or to an instruction preceding it.

STORAGE

This utility program itself occupies memory positions 37400-39999.

The following changes in the source program, which is written in SPS, will enable the program to be relocated for a 20K memory.

Source Card	Changes
01010	DORG 17400 (or any other address at which 2600 consecutive positions are available)
04080	Change column 16 from 3 to 1
04085	"
04088	"
04100	"
04110	"
04118	"
04130	"
04150	"
04160	"

For modification for 20K, positions 19995-19999 must be available.

EQUIPMENT

Indirect addressing.
Without automatic divide.

If automatic divide hardware is present, change the second operand of source card 02270 from $\bar{1}8$ to $\bar{2}0$. If the present object deck is available, change columns 35-36 from $\bar{1}8$ to $\bar{2}0$ on card 15. 349

4 R 2

OPERATING INSTRUCTIONS

For the most part the operating instructions are conveyed by typewritten messages. The SAMP object deck may be loaded initially and branched to when needed. If, however, an error in the program to be debugged destroys the utility program in memory, it can be reloaded readily; it contains 70 cards.

SOURCE PROGRAM AND WRITE-UP

The source program, write-up, and illustrative output appear in the material available from the 1620 General Program Library as 1.6.100.

350

SYMBOL TABLE PUNCH PROGRAMS

FOR

I.B.M., U.T.O. AND P.D.Q. FORTRAN SYSTEMS

By

Richard C. Irons

U.S. NAVAL SCHOOL OF AVIATION MEDICINE
U.S. NAVAL AVIATION MEDICAL CENTER
PENSACOLA, FLORIDA

USERS CODE: 1159

351

4.24.0

TABLE OF CONTENTS

352

4.24.01

INTRODUCTION

Fortran (FORmula TRANslation) is a popular computer language because it may be used without knowing the internal operations of the computer. A major drawback of the Fortran language, however, is the difficulty in finding program errors when they occur.

In processing a Fortran language program a table containing the storage addresses for calculated or constant values is produced. This table is known as a symbol table. The table can be a good aid in locating errors in a program because the values in question can be looked at and inferences made as to the causes of the difficulties.

Since the time required to type out the entire symbol table is relatively long, it is desirable to have the symbol table punched onto cards and then listed, perhaps, on an I.B.M. 407 machine. This makes the time required to obtain the table negligible, and the operation, therefore, more desirable.

Programs for punching the table have been mentioned in the 1620 USERS GROUP NEWSLETTER but these programs do not include all of the possible information that could be punched. In addition to the information punched out by these programs it is possible to provide the program execution phase addresses for any numbered fortran statement, or format type statement. With this additional information, statements can be referenced without necessarily referring to a program listing for the information.

To accomplish this, two programs have been developed, one of which is compatible with the I.B.M., U.T.O. Fortran systems, and the other with the P.D.Q.

Fortran system. These programs will provide the symbol table on cards, at the computer's card punch speed, in a format that is easy to read and that contains the maximum amount of information available.

DESCRIPTION OF PROGRAMS

I.B.M., U.T.O. SYMBOL TABLE PUNCH

The symbol table, described above, is generated during the processing of the Fortran source program and is not complete until the entire source program has been processed. This makes it necessary to load the symbol table punch program after the processing of the source program has been completed. In general it is advisable to use the Fortran Subroutine Deck during the compilation phase rather than during the object deck execution phase. This is mandatory for the U.T.O. version, optional for the I.B.M. versions. The U.T.O. processor will automatically clear the symbol table storage area if the subroutines are not used during the compilation of the source program.

P.D.Q. SYMBOL TABLE PUNCH

This program, like the program for the U.T.O. version, requires that the subroutines be used at the source deck compilation phase, since the symbol table storage area is also cleared if the subroutines are not used during this phase.

The original P.D.Q. system was developed with a symbol table punch routine incorporated within the CLT2 processor. The routine that was used will punch the table in a numeric form, which, besides being very difficult to read, destroys the area for the program discussed here. The routine used by the P.D.Q. processor is governed by switch two on the 1620 console. If switch two is off then the source program listing and

the symbol table will be punched on cards. It is desirable to have the program listing on cards because of the time saved, but it is also desirable to eliminate the symbol table punch routine used in the original P.D.Q. system. If the "branch control switch two" instruction in the symbol table routine is changed to an "unconditional branch" then it will be possible to retain the program listing on cards and eliminate the symbol table punch routine.

The above change would be necessary in the P.D.Q.—CLT2 processor only. In the P.D.Q. documentation from the 1620 USERS GROUP LIBRARY, page 193, line number 7552 is the instruction which reads "BC2 EXIT." This instruction should be changed to read "B EXIT."

OPERATING THE PROGRAMS

1. Compile source program.
2. Place symbol table punch program in card reader and press LOAD button on the reader.
3. Press START on the card punch.
4. Press START on the 1620 console when machine returns to a manual mode.
5. When punching is completed, remove cards from the card punch and list on an I.B.M. 407 machine with an 80-80 board.

PROCESSING TIME

The approximate computer time necessary to accomplish the symbol table punch is given by the following equation:

Execution time = (A x B) + 20 seconds

where A = number of symbols

and B = card punch speed of the computer

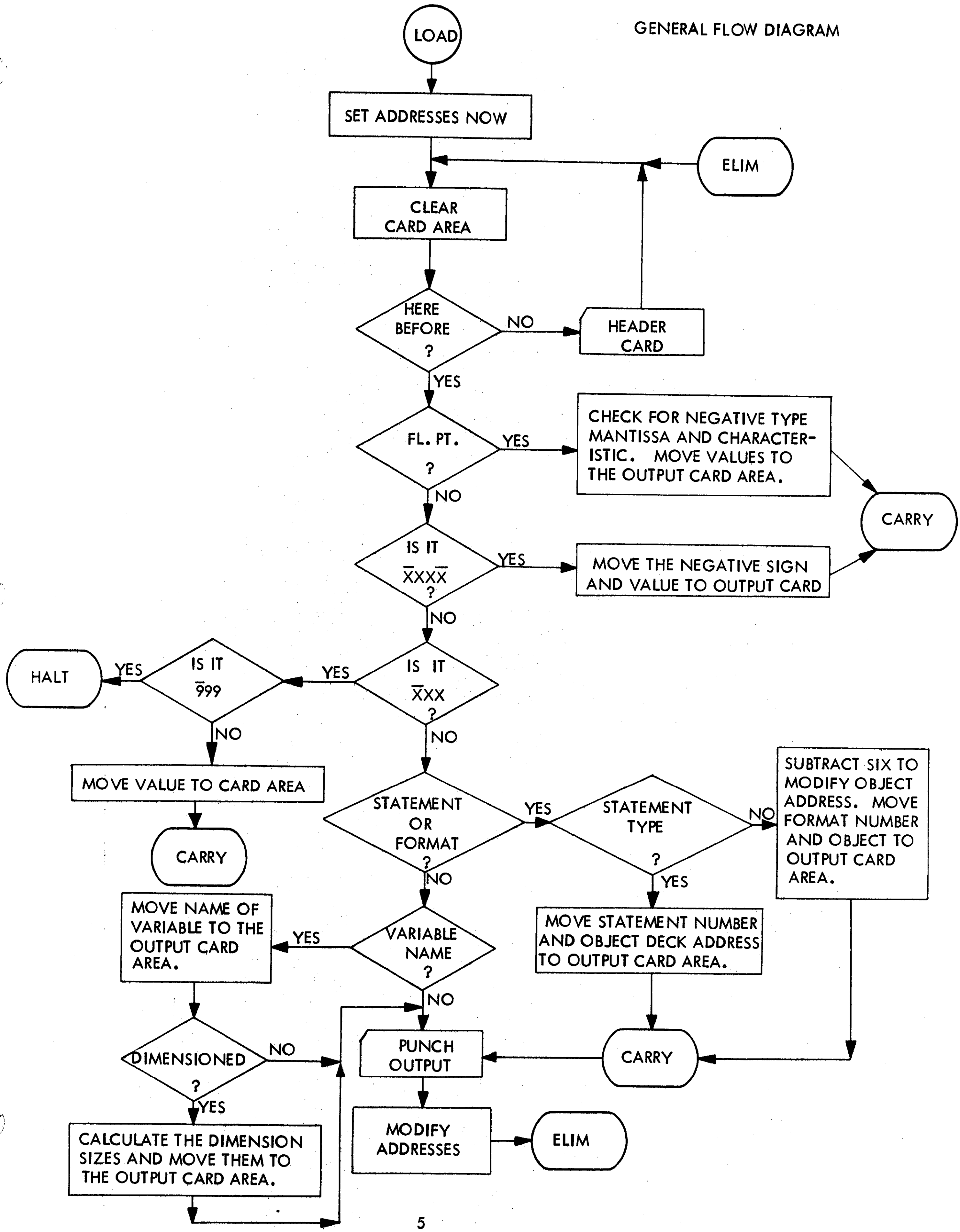
ADAPTATION TO OTHER THAN 20 K MACHINES

It is necessary to change the instruction labeled "DOIT" in both programs for machines having memory packages larger than 29K.

The instruction reads: "DOIT TFM INFAD, 19999." For a 40K machine the Q portion must be changed to read "39999", for 60K it would be "59999", et cetera.

356

4.24.4


GENERAL FLOW DIAGRAM
LOAD
SET ADDRESSES NOW
ELIM
CLEAR CARD AREA
HERE BEFORE ?
NO
HEADER CARD
YES
FL. PT. ?
YES
CHECK FOR NEGATIVE TYPE MANTISSA AND CHARACTERISTIC. MOVE VALUES TO THE OUTPUT CARD AREA.
CARRY
NO
IS IT $\overline{X}XX\overline{X}$?
YES
MOVE THE NEGATIVE SIGN AND VALUE TO OUTPUT CARD
NO
HALT
YES
IS IT $\overline{9}99$
YES
IS IT $\overline{X}XX$?
NO
NO
MOVE VALUE TO CARD AREA
CARRY
STATEMENT OR FORMAT ?
YES
STATEMENT TYPE ?
NO
SUBTRACT SIX TO MODIFY OBJECT ADDRESS. MOVE FORMAT NUMBER AND OBJECT TO OUTPUT CARD AREA.
NO
YES
MOVE NAME OF VARIABLE TO THE OUTPUT CARD AREA.
YES
VARIABLE NAME ?
MOVE STATEMENT NUMBER AND OBJECT DECK ADDRESS TO OUTPUT CARD AREA.
NO
DIMENSIONED ?
NO
PUNCH OUTPUT
CARRY
YES
CALCULATE THE DIMENSION SIZES AND MOVE THEM TO THE OUTPUT CARD AREA.
MODIFY ADDRESSES
ELIM

I.B.M., U.T.O. SYMBOL TABLE PUNCH PROGRAM LISTING

```
                            *
                            *      ** SYMBOL TABLE PUNCH PROGRAM FOR THE   **
                            *      ** I.B.M. AND U.T.O. FORTRAN PROCESSING**
                            *      ** SYSTEMS.                              **
                            *
                            *      ** DEVELOPED BY R.C. IRONS, U.S. NAVAL **
                            *      ** SCHOOL OF AVIATION MEDICINE, N.A.S. **
                            *      ** PENSACOLA, FLORIDA.                  **
                            *
                            *
                            *      ** TO CHANGE THIS PROGRAM FOR LARGER   **
                            *      ** SIZE MEMORYS IT IS ONLY NECESSARY   **
                            *      ** TO CHANGE THE FIRST DIGIT OF THE    **
                            *      ** Q FIELD OF INSTRUCTION LABLED DOIT **
                            *
                            *      ** DORG LOCATION MAY BE CHANGED TO SUIT**
                            *      ** THE USER.                             **
                            *
6000                               DORG6000
6000   16 07174 19999       DOIT   TFM INFAD,19999,,19999 MEANS 20K MEMORY
6012   26 07179 07174              TF  OBJAD,INFAD
6024   26 07184 07174              TF  MODAD,INFAD
6036   15 07186 00000       ELIM   TDM CHI-1,0,27,ZEROIZE OUTPUT AREA
6048   14 06042 07345              CM  *-6,CHI+158
6060   46 06096 01200              BE  *+36
6072   11 06042 00001              AM  *-30,1,10
6084   49 06036 00000              B   *-48
6096   16 06042 07186              TFM *-54,CHI-1
6108   44 06200 06043              BNF EMOD,ELIM+7,,BRANCH FOR NO HEADER CARD
6120   26 07217 07081              TF  CHI+30,ZA1+28
6132   26 07251 07111              TF  CHI+64,ZA2+28
6144   26 07281 07139              TF  CHI+94,ZA3+26
6156   26 07317 07169              TF  CHI+130,ZA4+28
6168   39 07187 00400              WACDCHI
6180   33 06043 00000              CF  ELIM+7
6192   49 06036 00000              B   ELIM
6200                               DORG*-3
6200   73 07195 07179       EMOD   TNF CHI+8,OBJAD,,MOVE SYMBOL TBL. ADDRESS
6212   12 07184 00001              SM  MODAD,1,10
6224   44 06352 07184              BNF NOTFL,MODAD,11,BRANCH NOT FLOATING PT.
6236   44 06272 07174              BNF *+36,INFAD,11,BRANCH POSITIVE EXPONENT
6248   33 07174 00000              CF  INFAD,,6
6260   16 07321 00020              TFM CHI+134,20,10,MOVE NEGATIVE SIGN
6272   73 07319 07174              TNF CHI+132,INFAD,11,MOVE EXPONENTAL
6284   12 07184 00001              SM  MODAD,1,10
6296   44 06332 07184              BNF *+36,MODAD,11,BRANCH POSITIVE MANTISSA
6308   33 07184 00000              CF  MODAD,,6
6320   16 07315 00020              TFM CHI+128,20,10,MOVE NEGATIVE SIGN
6332   73 07313 07184              TNF CHI+126,MODAD,11,MOVE MANTISSA
6344   49 06636 00000              B   CARRY
6352                               DORG*-3
6352   44 06408 07174       NOTFL  BNF NOTCN,INFAD,11,BRANCH NOT 4 DIGIT FX PT
6364   16 07295 00020              TFM CHI+108,20,10,MOVE NEGATIVE SIGN
6376   33 07174 00000              CF  INFAD,,6
```

358

I.B.M., U.T.O. SYMBOL TABLE PUNCH PROGRAM LISTING (CONT'D)

```
6388   73 07293 07174                 TNF  CHI+106,INFAD,11,MOVE FIXED PT. NO.
6400   49 06636 00000                 B    CARRY
6408                                  DORG*-3
6408   12 07184 00001         NOTCN   SM   MODAD,1,10
6420   44 06476 07184                 BNF  NOTAC,MODAD,11,NOT FIXED PT. TYPE
6432   14 07174 00999                 CM   INFAD,999,69,END OF SYMBOL TABLE
6444   46 07032 01200                 BE   EXIT,,,END OF SYMBOL TABLE ROUTINE
6456   73 07293 07174                 TNF  CHI+106,INFAD,11,MOVE 3 DIGIT CONSTANT
6468   49 06636 00000                 B    CARRY
6476                                  DORG*-3
6476   12 07184 00001         NOTAC   SM   MODAD,1,10
                              *
                              *       ** BRANCH IF NOT FLOATING PT., FIXED  **
                              *       ** PT., STATMENT OR FORMAT NUMBERS.   **
                              *
6488   44 06624 07184                 BNF  CLF,MODAD,11
6500   12 07184 00001                 SM   MODAD,1,10
6512   43 06568 07184                 BD   MAT,MODAD,11,BRANCH IF FORMAT NUMBER
6524   12 07184 00001                 SM   MODAD,1,10
6536   73 07249 07174                 TNF  CHI+62,INFAD,11,MOVE STATEMENT NUMBER
6548   73 07261 07184                 TNF  CHI+74,MODAD,11,MOVE OBJECT ADDRESS
6560   49 06636 00000                 B    CARRY
6568                                  DORG*-3
6568   12 07184 00001         MAT     SM   MODAD,1,10
6580   12 07184 00006                 SM   MODAD,6,610,MODIFY OBJECT ADDRESS
6592   73 07283 07184                 TNF  CHI+96,MODAD,11,MOVE OBJECT ADDRESS
6604   73 07271 07174                 TNF  CHI+84,INFAD,11,MOVE FORMAT NUMBER
6616   49 06636 00000                 B    CARRY
6624                                  DORG*-3
6624   33 06007 00000         CLF     CF   DOIT+7
6636   26 07184 07174         CARRY   TF   MODAD,INFAD
6648   44 06716 06007                 BNF  DIMEN,DOIT+7,,BRANCH FOR VARIABLE NAME
6660   39 07187 00400         RESAY   WACDCHI
6672   12 07174 00010                 SM   INFAD,10,10
6684   12 07179 00010                 SM   OBJAD,10,10
6696   26 07184 07174                 TF   MODAD,INFAD
6708   49 06036 00000                 B    ELIM
6716                                  DORG*-3
6716   26 07219 07174         DIMEN   TF   CHI+32,INFAD,11,MOVE NAME OF VARIABLE
6728   32 06007 00000                 SF   DOIT+7
6740   15 07184 00005                 TDM  MODAD,5
6752   12 07184 00010                 SM   MODAD,10,10
6764   44 06660 07184                 BNF  RESAY,MODAD,11,BRANCH NO DIMENSION
6776   11 07184 00001                 AM   MODAD,1,10
6788   44 06808 07184                 BNF  DIE,MODAD,11,BRANCH IF DIMENSIONED
6800   49 06660 00000                 B    RESAY
6808                                  DORG*-3
                              *
                              *       ** DIMENSION SIZES ARE FOUND BY SUB-   **
                              *       ** TRACTING ONE DIMENSION SIZE WHICH **
                              *       ** IS KNOWN FROM THE DIFFERENCE IN     **
                              *       ** SYMBOL TABLE ADDRESSES PLUS TEN.    **
                              *       ** THE NUMBER OF SUBTRACTIONS REQUIR-  **
                              *       ** ED TO BRING THE RESULT TO EQUAL     **
                              *       ** ZERO IS THE SIZE OF THE SECOND DI-  **
```

I.B.M., U.T.O. SYMBOL TABLE PUNCH PROGRAM LISTING (CONT'D)

```
                              *        ** MENSION.                                        **
                              *
                              *
6808    41                    DIE      NOP ,,,STORAGE FOR SYMBOL TABLE ADDRESSES
6820    15 07184 00009                 TDM MODAD,9
6832    26 06819 07184                 TF  DIE+11,MODAD,11,LOWER ADDRESS
6844    26 06814 07179                 TF  DIE+6,OBJAD,,HIGH ADDRESS
6856    73 07207 06819                 TNF CHI+20,DIE+11,,MOVE LOWER ADDRESS TO CD
6868    22 06814 06819                 S   DIE+6,DIE+11,,FIND DIFFERENCE
6880    11 06814 00010                 AM  DIE+6,10,10,UP DIFFERENCE BY TEN
6892    12 07184 00005                 SM  MODAD,5,10
6904    26 06830 07184                 TF  DIE+22,MODAD,11,BRING IN ONE DIMENSION
6916    73 07229 07184                 TNF CHI+42,MODAD,11,MOVE DIMENSION TO CARD
                              *
                              *        ** NEXT NOP USED AS COUNTER FOR SUBTRACTIONS
                              *
6928    41  -     -                    NOP ,,38
6940    11 06939 00001                 AM  *-1,1,10,COUNTER OF SUBTRACTIONS
6952    22 06813 06830                 S   DIE+5,DIE+22
6964    47 06940 01200                 BNZ *-24
6976    73 07239 06939                 TNF CHI+52,*-37,,MOVE SECOND DIMENSION
6988    26 06939 06934                 TF  *-49,*-54,,ZEROIZE COUNTER FIELD
7000    26 07179 06819                 TF  OBJAD,DIE+11,,RESET SYMBOL TABLE ADDR.
7012    12 07174 00010                 SM  INFAD,10,10
7024    49 06660 00000                 B   RESAY
7032                                   DORG*-3
                              *
                              *         ** THE SYMBOL TABLE PUNCH ROUTINE IS    **
                              *         ** COMPLETED AT THIS POINT.             **
                              *
7032    48                    EXIT     H
7044    49 07032 00000                 B   EXIT
7052                                   DORG*-3
                              *
                              *         ** DEFINITIONS FOLLOW NEXT.             **
                              *
7053       00015 00000        ZA1      DAC 15,ADDRESSES  NAME,
7083       00015 00000        ZA2      DAC 15,DIMENSION STATE,
7113       00014 00000        ZA3      DAC 14,LOC  FORM  LOC,
7141       00015 00000        ZA4      DAC 15,FIX    FLOATING,
7174       00005 00000        INFAD    DC  5,0
7179       00005 00000        OBJAD    DC  5,0
7184       00005 00000        MODAD    DC  5,0
7187       00080 00000        CHI      DAS 80,
6000                                   DENDDOIT
```

0324

SAMPLE PROBLEM # 1 LISTING

```
C                                                                                A07070
C     PROGRAM TO SOLVE SIMULTANEOUS LINEAR EQUATIONS BY INVERSION AND           A07070
C     MULTIPLICATION OF MATRICES.                                               A07070
C                                                                                A07070
C     IRONS, R.C., NAVSCHAVMED, PENSACOLA FEBRUARY 1964                         A07070
C                                                                                A07070
      DIMENSION W(28),RCP(28),R(28,29)                                          A07070
C                                                                                A07070
C     READ IN K, WHICH EQUALS THE NUMBER OF UNKNOWNS                            A07070
C                                                                                A07070
   24 READ 93,K                                                                 A07070
C                                                                                A07094
C     SET THE COUNTER N EQUAL TO K+1                                            A07094
C                                                                                A07094
      N=K+1                                                                     A07094
C                                                                                A07130
C     READ IN CONSTANT VALUES COLUMN BY COLUMN UNTIL MATRIX IS FULL.            A07130
C     THIS MEANS READ ALL X VALUES, ALL Y VALUES AND SO FORTH.                  A07130
C                                                                                A07130
      DO 26 M=1,K                                                               A07130
      DO 25 J=1,K                                                               A07142
   25 READ 94,R(J,M)                                                            A07154
   26 CONTINUE                                                                  A07274
C                                                                                A07310
C     DO LOOP NUMBER 18 CONTROLS THE INVERSION OF THE MATRIX. L GIVES           A07310
C     THE ROW WHICH IS OF MAIN INTEREST AT ANY PARTCULAR TIME.                  A07310
C                                                                                A07310
      DO 18 L=1,K                                                               A07310
C                                                                                A07322
C     DO LOOP NUMBER 3 SETS THE LAST COLUMN IN ALL ROWS EQUAL TO ZERO .         A07322
C                                                                                A07322
      DO 3 J=1,K                                                                A07322
    3 R(J,N)=0.                                                                 A07334
C                                                                                A07454
C     THE STATEMENT  R(L,N)=1. PUTS A VALUE OF 1 IN THE LAST COLUMN             A07454
C     OF THE ROW WHICH IS EQUAL TO L.                                           A07454
C                                                                                A07454
      R(L,N)=1.                                                                 A07454
C                                                                                A07538
C     DO LOOP NUMBER 43 CAUSES ALL VALUES FROM COLUMN ONE OF ALL ROWS           A07538
C     TO BE TRANSFERED TO A STORAGE AREA CALLED W.                              A07538
C                                                                                A07538
      DO 43 J=1,K                                                               A07538
   43 W(J)=R(J,1)                                                               A07550
C                                                                                A07658
C     DO LOOP NUMBER 2 CHECKS FOR VALUES OF ZERO IN COLUMN ONE OF ALL           A07658
C     ROWS, FINDS THEIR RECIPROCALS AND STORES THESE IN RCP.                    A07658
C                                                                                A07658
      DO 2 J=1,K                                                                A07658
      IF(R(J,1))1,61,1                                                          A07670
   61 RCP(J)=0.                                                                 A07750
      GO TO 2                                                                   A07798
    1 RCP(J)=1./R(J,1)                                                          A07806
    2 CONTINUE                                                                  A07890
C                                                                                A07926
```

```
      DO LOOP NUMBERS 5 AND 4 CAUSE ALL COLUMNS OF EACH ROW TO BE        A07926
      MULTIPLIED BY THEIR RESPECTIVE RCP VALUES.  ROWS WITH A RCP        A07926
      VALUE OF ZERO WILL NOT BE MULTIPIED.                               A07926
                                                                         A07926
      DO 5 J=1,K                                                         A07926
      DO 4 M=1,N                                                         A07938
      IF(RCP(J))6,4,6                                                    A07950
    6 R(J,M)=R(J,M)*RCP(J)                                               A08030
    4 CONTINUE                                                           A08210
    5 CONTINUE                                                           A08246
                                                                         A08282
      DO LOOP NUMBERS 15 AND 14 SUBTRACT THE ROW WHICH IS EQUAL TO L     A08282
      FROM ALL OTHER ROWS IN THE MATRIX.                                 A08282
                                                                         A08282
      DO 15 M=1,N                                                        A08282
      DO 14 J=1,K                                                        A08294
      IF(L-J)13,14,13                                                    A08306
   13 R(J,M)=R(J,M)-R(L,M)                                               A08374
   14 CONTINUE                                                           A08590
   15 CONTINUE                                                           A08626
                                                                         A08662
      DO LOOP NUMBERS 12 AND 7 CAUSES ALL COLUMNS OF EACH ROW TO BE      A08662
      MULTIPLIED BY THE VALUE WHICH WERE ORIGINALLY IN COLUMN ONE.       A08662
      THESE VALUES WERE STORED IN W. VALUES OF ZERO IN ANY W WILL NOT    A08662
      BE USED.                                                           A08662
                                                                         A08662
      DO 12 J=1,K                                                        A08662
      DO 7 M=1,N                                                         A08674
      IF(L-J)8,7,8                                                       A08686
    8 IF(W(J))11,7,11                                                    A08754
   11 R(J,M)=R(J,M)*W(J)                                                 A08834
    7 CONTINUE                                                           A09014
   12 CONTINUE                                                           A09050
                                                                         A09086
      DO LOOP NUMBERS 17 AND 16 CAUSE COLUMN ONE IN EACH ROW TO BE       A09086
      REPLACED BY THE VALUES IN COLUMN TWO.  COLUMN TWO BY COLUMN        A09086
      THREE AND SO FORTH.                                                A09086
                                                                         A09086
      DO 17 J=1,K                                                        A09086
      DO 16 M=1,K                                                        A09098
   16 R(J,M)=R(J,M+1)                                                    A09110
   17 CONTINUE                                                           A09290
                                                                         A09326
      STATEMENT NUMBER 18 IS THE LAST STATEMENT OF THE MATRIX INVERSION  A09326
      ROUTINE. IF L EQUALS K THEN INVERSION IS COMPLETE.                 A09326
                                                                         A09326
   18 CONTINUE                                                           A09326
                                                                         A09362
      DO LOOP NUMBER 19 READS CONSTANTS FOR EACH EQUATION INTO THE LAST  A09362
      COLUMN OF THE MATRIX.                                              A09362
                                                                         A09362
      DO 19 J=1,K                                                        A09362
   19 READ 94,R(J,N)                                                     A09374
                                                                         A09494
      DO LOOP NUMBER 20 WILL ZEROIZE RCP STORAGE AREA FOR STORING SUMS   A09494
```

```
C      IN MATRIX MULTIPLICATION.                                        A09494
C                                                                       A09494
       DO 20 J=1,K                                                      A09494
   20  RCP(J)=0 .                                                       A09506
C                                                                       A09590
C      DO LOOP NUMBERS 22 AND 21 CAUSE MATRIX MULTIPICATION AND SUMMING A09590
C      INTO RCP FOR EACH ROW TO BE PERFORMED.                           A09590
C                                                                       A09590
       DO 22 J=1,K                                                      A09590
       DO 21 M=1,K                                                      A09602
   21  RCP(J)=(R(J,M)*R(M,N))+RCP(J)                                    A09614
   22  CONTINUE                                                         A09866
C                                                                       A09902
C      HEADER CARD IS NOW PUNCHED FOR OUTPUT .                          A09902
C                                                                       A09902
       PUNCH 91                                                         A09902
C                                                                       A09914
C      DO LOOP NUMBER 23 CAUSES THE UNKNOWN NUMBER AND THE VALUE OF THE A09914
C      UNKNOWN TO BE PUNCHED.                                           A09914
C                                                                       A09914
       DO 23 J=1,K                                                      A09914
       PUNCH 95,J,RCP(J)                                                A09926
   23  CONTINUE                                                         A09986
C                                                                       A10022
C      NEXT STATEMENT TYPES PROBLEM COMPLETED STATEMENT.                A10022
C                                                                       A10022
       TYPE 92                                                          A10022
C                                                                       A10034
C      PROGRAM STOPS FOR OPERATOR AT THIS POINT .                       A10034
C                                                                       A10034
       PAUSE                                                            A10034
C                                                                       A10046
C      IF THERE IS ANOTHER PROBLEM THEN IT IS DESIRED TO READ IN A NEW  A10046
C      VALUE OF UNKNOWNS AND A NEW MATRIX.                              A10046
C                                                                       A10046
       GO TO 24                                                         A10046
C                                                                       A10054
C      HEADER CARD FORMAT IS NEXT                                       A10054
C                                                                       A10054
   91  FORMAT(30HUNKNOWN NUMBER   UNKNOWN VALUE/)                       A10054
C                                                                       A10144
C      PROBLEM COMPLETED MESSAGE FORMAT IS NEXT .                       A10144
C                                                                       A10144
   92  FORMAT(42HPROBLEM COMPLETED. PUSH START FOR NEW RUN./)           A10144
C                                                                       A10258
C      UNKNOWN VALUE FORMAT IS NEXT .                                   A10258
C                                                                       A10258
   93  FORMAT(I2)                                                       A10258
C                                                                       A10280
C      INPUT FOR VALUES ON UNKNOWNS FORMAT IS NEXT.                     A10280
C                                                                       A10280
   94  FORMAT(F10.8)                                                    A10280
C                                                                       A10302
C      FORMAT FOR OUTPUT VALUES IS NEXT.                                A10302
C                                                                       A10302
```

SAMPLE PROBLEM # 1 LISTING (CONT'D)

```
95 FORMAT(5XI2,10XF11 .5)                                              A10302
   END                                                                  A10344
                                                                         00167
```


4.24.12

SYMBOL TABLE LISTING OF PROBLEM # 1

ADDRESSES		NAME	DIMENSION		STATE	LOC	FORM	LOC	FIX	FLOATING	A
19999		SIN									A
19989		SINF									A
19979		COS									A
19969		COSF									A
19959		ATAN									A
19949		ATANF									A
19939		EXP									A
19929		EXPF									A
19919		LOG									A
19909		LOGF									A
19899		SQRT									A
19889		SQRTF									A
19879		SIGN									A
19869		SIGNF									A
19859		ABS									A
19849		ABSF									A
19839		RDNM									A
19829		RDNMF									A
19819	19549	W	0028	0001							A
19539	19269	RCP	0028	0001							A
19259	11149	R	0028	0029							A
11139					0024	07070					A
11129							0093	10258			A
11119		K									A
11109		N									A
11099									0001-		A
11089									000		A
11079					0026	07274					A
11069		M									A
11059					0025	07154					A
11049		J									A
11039							0094	10280			A
11029					0018	09326					A
11019		L									A
11009					0003	07334					A
10999										00000000 99-	A
10989										10000000 01	A
10979					0043	07550					A
10969					0002	07890					A
10959					0001	07806					A
10949					0061	07750					A
10939					0005	08246					A
10929					0004	08210					A
10919					0006	08030					A
10909					0015	08626					A
10899					0014	08590					A
10889					0013	08374					A
10879					0012	09050					A
10869					0007	09014					A
10859					0008	08754					A
10849					0011	08834					A
10839					0017	09290					A
10829					0016	09110					A
10819					0019	09374					A

SYMBOL TABLE LISTING OF PROBLEM # I (CONT'D)

9	0020 09506			A
9	0022 09866			A
9	0021 09614			A
9			001	A
9		0091 10054		A
9	0023 09986			A
9		0095 10302		A
9		0092 10144		A
				00063

366

4.24.14

P.D.Q. SYMBOL TABLE PUNCH PROGRAM LISTING

```
                                  *
                                  *        ** PROGRAM TO DUMP P.D.Q. FORTRAN TYPE **
                                  *        ** SYMBOL TABLE ON CARDS.              **
                                  *        *                                      **
                                  *        *                                      **
                                  *        **DORG CAN BE CHANGED AS DESIRED.      **
                                  *                                               **
6000                                       DORG6000
                                  *
                                  *        ** THE Q ADDRESS OF THE INSTRUCTION    **
                                  *        ** LABLED DOIT IS THE ONLY INSTRUCTION **
                                  *        ** REQUIRING A CHANGE FOR MACHINES OF  **
                                  *        ** GREATER THAN 20K MEMORY.  A THREE   **
                                  *        ** IN Q7 WOULD ADAPT TO A 40K SIZE, A  **
                                  *        ** FIVE FOR 60K AND SO ON .            **
                                  *
6000   16 07202 19999             DOIT     TFM INFAD,19999,,20K MEMORY MACHINE
6012   26 07207 07202                      TF  OBJAD,INFAD
6024   26 07212 07202                      TF  MODAD,INFAD
6036   15 07214 00000             ELIM     TDM CHI-1,0,27,ZEROIZE OUTPUT AREA
6048   14 06042 07373                      CM  *-6,CHI+158
6060   46 06096 01200                      BE  *+36
6072   11 06042 00001                      AM  *-30,1,10
6084   49 06036 00000                      B   ELIM
6096   16 06042 07214                      TFM ELIM+6,CHI-1
6108   44 06200 06043                      BNF EMOD,ELIM+7,,FLAG FOR HEADER CARD
6120   26 07245 07109                      TF  CHI+30,ZA1+28
6132   26 07279 07139                      TF  CHI+64,ZA2+28
6144   26 07309 07167                      TF  CHI+94,ZA3+26
6156   26 07345 07197                      TF  CHI+130,ZA4+28
6168   39 07215 00400                      WACDCHI,,,WRITE HEADER CARD
6180   33 06043 00000                      CF  ELIM+7
6192   49 06036 00000                      B   ELIM
6200                                       DORG*-3
6200   73 07223 07207             EMOD     TNF CHI+8,OBJAD,,SYMBOL TABLE ADDRESS
6212   44 06364 07202                      BNF NOTFL-12,INFAD,11,BRANCH NOT FL.PT.
6224   12 07212 00003                      SM  MODAD,3,10
6236   44 06272 07212                      BNF *+36,MODAD,11,BRANCH FLOATING POINT
6248   11 07212 00002                      AM  MODAD,2,10
6260   49 06376 00000                      B   NOTFL
6272   33 07202 00000                      CF  INFAD,,6,BEGIN FL. PT. TRANSFER
6284   12 07202 00007                      SM  INFAD,7,10
6296   32 07202 00000                      SF  INFAD,,6
6308   12 07202 00001                      SM  INFAD,1,10
6320   73 07331 07202                      TNF CHI+116,INFAD,11,MOVE CHARACTERISTIC
6332   11 07202 00008                      AM  INFAD,8,10
6344   73 07349 07202                      TNF CHI+134,INFAD,11,MOVE MANTISSA
6356   49 06660 00000                      B   CARRY
6364                                       DORG*-3
6364   12 07212 00001                      SM  MODAD,1,10
6376   44 06432 07202             NOTFL    BNF NOTCN,INFAD,11,NOT FOUR DIGIT FX. PT.
6388   16 07323 00020                      TFM CHI+108,20,10
6400   33 07202 00000                      CF  INFAD,,6
6412   73 07321 07202                      TNF CHI+106,INFAD,11,MOVE FIXED PT. NO.
```

```
6424   49 06660 00000          B    CARRY
6432                           DORG*-3
6432   12 07212 00001   NOTCN  SM   MODAD,1,10
6444   44 06500 07212          BNF  NOTAC,MODAD,11,NOT FIXED POINT TYPE.
6456   14 07202 00999          CM   INFAD,999,69,END OF SYMBOL TABLE.
6468   46 07056 01200          BE   EXIT
6480   73 07321 07202          TNF  CHI+106,INFAD,11,MOVE 3 DIGIT FX. PT.
6492   49 06660 00000          B    CARRY
6500                           DORG*-3
6500   12 07212 00001   NOTAC  SM   MODAD,1,10
6512   44 06648 07212          BNF  CLF,MODAD,11,VARIABLE NAME OR DIMENSION
6524   12 07212 00001          SM   MODAD,1,10
6536   43 06592 07212          BD   MAT,MODAD,11,BRACH IF FORMAT NO.
6548   12 07212 00001          SM   MODAD,1,10
6560   73 07277 07202          TNF  CHI+62,INFAD,11,MOVE STATEMENT NO.
6572   73 07289 07212          TNF  CHI+74,MODAD,11,MOVE OBJECT ADDRESS
6584   49 06660 00000          B    CARRY
6592                           DORG*-3
6592   12 07212 00001   MAT    SM   MODAD,1,10
6604   12 07212 00006          SM   MODAD,6,610
6616   73 07311 07212          TNF  CHI+96,MODAD,11,MOVE OBJECT ADDRESS
6628   73 07299 07202          TNF  CHI+84,INFAD,11,MOVE FORMAT NO.
6640   49 06660 00000          B    CARRY
6648                           DORG*-3
6648   33 06007 00000   CLF    CF   DOIT+7
6660   26 07212 07202   CARRY  TF   MODAD,INFAD
6672   44 06740 06007          BNF  DIMEN,DOIT+7,,VARIABLE NAME - DIMENSIO
6684   39 07215 00400   RESAY  WACDCHI,,,PUNCH OUTPUT CARD
6696   12 07202 00010          SM   INFAD,10,10
6708   12 07207 00010          SM   OBJAD,10,10
6720   26 07212 07202          TF   MODAD,INFAD
6732   49 06036 00000          B    ELIM
6740                           DORG*-3
6740   26 07247 07202   DIMEN  TF   CHI+32,INFAD,11,MOVE VARIABLE NAME.
6752   32 06007 00000          SF   DOIT+7
6764   15 07212 00005          TDM  MODAD,5
6776   12 07212 00010          SM   MODAD,10,10
6788   44 06684 07212          BNF  RESAY,MODAD,11,BRANCH NOT DIMENSIONED
6800   11 07212 00001          AM   MODAD,1,10
6812   44 06832 07212          BNF  DIE,MODAD,11,BRANCH IF DIMENSIONED
6824   49 06684 00000          B    RESAY
6832                           DORG*-3
                        *
                        *      ** THE DIMENSION SIZES ARE FOUND BY     **
                        *      ** FINDING THE DIFFERENCES IN THE       **
                        *      ** SYMBOL TABLE ADDRESSES PLUS TEN.     **
                        *      ** ONE DIMENSION SIZE IS ALREADY KNOWN  **
                        *      ** AND IS SUBTRACTED FROM THE ADDRESS   **
                        *      ** DIFFERENCES UNTIL THE RESULT IS      **
                        *      ** FOUND TO BE EQUAL TO ZERO.  THE      **
                        *      ** NUMBER OF SUBTRACTIONS NECESSARY TO  **
                        *      ** DO THIS IS THE SIZE OF THE SECOND    **
                        *      ** DIMENSION.                           **
                        *
6832   41               DIE    NOP  ,,,WORK AREA FOR DIMENSION SIZE
```

P.D.Q. SYMBOL TABLE PUNCH PROGRAM LISTING (CONT'D)

```
6844   15 07212 00009         TDM  MODAD,9
6856   26 06843 07212         TF   DIE+11,MODAD,11,LOWER SYMBOL TBL. ADDR.
6868   26 06838 07207         TF   DIE+6,OBJAD,,UPPER SYMBOL TBL. ADDRESS
6880   73 07235 06843         TNF  CHI+20,DIE+11,,MOVE LOWER ADDRESS
6892   22 06838 06843         S    DIE+6,DIE+11,,FIND DIFFERENCE
6904   11 06838 00010         AM   DIE+6,10,10
6916   12 07212 00065         SM   MODAD,5,10
6928   26 06854 07212         TF   DIE+22,MODAD,11,BRING IN ONE DIMENSION
6940   73 07257 07212         TNF  CHI+42,MODAD,11,MOVE ONE DIMENSION
6952   41  -     -            NOP  ,,38,AREA FOR SECOND DIMENSION
6964   11 06963 00001         AM   *-1,1,10
6976   22 06837 06854         S    DIE+5,DIE+22,,DIMEN. FROM TBL. DIFF.
6988   47 06964 01200         BNZ  *-24
7000   73 07267 06963         TNF  CHI+52,*-37,,MOVE SECOND DIMENSION
7012   26 06963 06958         TF   *-49,*-54,,ZEROIZE SECOND DIMENSION
7024   26 07207 06843         TF   OBJAD,DIE+11,,NEW SYMBOL TABLE ADDRESS
7036   12 07202 00010         SM   INFAD,10,10
7048   49 06684 00000         B    RESAY
7056                          DORG*-3
                        *
                        *     ** END OF SYMBOL TABLE ROUTINE          **
                        *
7056   48               EXIT  H
7068   49 07056 00000         B    *-12
                        *
                        *     ** AREA DEFINITIONS FOLLOW BELOW        **
                        *
7081      00015 00000   ZA1   DAC 15,ADDRESSES  NAME,
7111      00015 00000   ZA2   DAC 15,DIMENSION STATE,
7141      00014 00000   ZA3   DAC 14,LOC  FORM  LOC,
7169      00015 00000   ZA4   DAC 15,FIX     FLOATING,
7202      00005 00000   INFAD DC   5,0
7207      00005 00000   OBJAD DC   5,0
7212      00005 00000   MODAD DC   5,0
7215      00080 00000   CHI   DAS 80,
6000                          DENDDOIT
```

SAMPLE PROBLEM # 2 LISTING

```
06600  C       WHERRY TRANSFORM EQUATION FOR EQUATING SUBGROUPSZ
06600       DIMENSION CASES(9),SUMX(9),SUMX2(9),SUMY(9),SUMY2(9),SUMXY(9)Z
06600       DIMENSION EXX(9),EYY(9),EXY(9),BETA(9),ALPHA(9),R(9),XMEAN(9)
06600       DIMENSION YMEAN(9),STDVX(9),SXTR(9),SX2TR(9),SXYTR(9)Z
06600       DIMENSION BETA3(9),ALPH3(9),FMNX(9),ESDX(9),STDVY(9),ID(9)
06600  CZ
06600       BIGN=0.
06612       SSX=0.
06624       SSX2=0.
06636       SSY=0.
06648       SSY2=0.
06660       SSXY=0.
06672       X2TR=0.
06684       XYTR=0.
06696       XTR=0.
06708       PUNCH 202Z
06720       READ 201,KZ
06744       DO 3 I=1,KZ
06756       READ 103,ID(I),CASES(I),SUMX(I),SUMX2(I),SUMY(I),SUMY2(I),SUMXY(I)
07020     3 CONTINUEZ
07056      IF(SENSE SWITCH 2)4,5Z
07076     4 DO 6 I=1,KZ
07088       TYPE 103,ID(I),CASES(I),SUMX(I),SUMX2(I),SUMY(I),SUMY2(I),SUMXY(I)
07352     6 CONTINUEZ
07388     5 CONTINUEZ
07388       DO 1 I=1,KZ
07400       EXX(I)=(CASES(I)*SUMX2(I))-(SUMX(I)*SUMX(I))Z
07616       EYY(I)=(CASES(I)*SUMY2(I))-(SUMY(I)*SUMY(I))Z
07832       EXY(I)=(CASES(I)*SUMXY(I))-(SUMX(I)*SUMY(I))Z
08048       BETA(I)=EXY(I)/EXX(I)Z
08156       ALPHA(I)=(SUMY(I)-(BETA(I)*SUMX(I)))/CASES(I)Z
08348       R(I)=EXY(I)/SQRTF(EXX(I)*EYY(I))Z
08504       XMEAN(I)=SUMX(I)/CASES(I)Z
08612       YMEAN(I)=SUMY(I)/CASES(I)Z
08720       STDVX(I)=SQRTF(EXX(I)/(CASES(I)*CASES(I)))Z
08876       STDVY(I)=SQRTF(EYY(I)/(CASES(I)*CASES(I)))Z
09032       BIGN=BIGN+CASES(I)Z
09092       SSX=SSX+SUMX(I)Z
09152       SSX2=SSX2+SUMX2(I)Z
09212       SSY=SSY+SUMY(I)Z
09272       SSY2=SSY2+SUMY2(I)Z
09332       SSXY=SSXY+SUMXY(I)Z
09392       SXTR(I)=(BETA(I)*SUMX(I))+(CASES(I)*ALPHA(I))Z
09596       SX2TR(I)=((BETA(I)*BETA(I)*EXX(I))+(SUMY(I)*SUMY(I)))/CASES(I)Z
09872       SXYTR(I)=((BETA(I)*EXY(I))+(SUMY(I)*SUMY(I)))/CASES(I)Z
J0112       X2TR=X2TR+SX2TR(I)Z
J0172       XYTR=XYTR+SXYTR(I)Z
J0232       XTR=XTR+SXTR(I)Z
J0292     1 CONTINUEZ
J0328       EXYTR=(BIGN*XYTR)-(SSY*XTR)Z
J0424       EXXTR=(BIGN*X2TR)-(XTR*XTR)Z
J0520       EYYT=(BIGN*SSY2)-(SSY*SSY)Z
J0616       EXXT=(BIGN*SSX2)-(SSX*SSX)Z
J0712       EXYT=(BIGN*SSXY)-(SSX*SSY)Z
J0808       RT=EXYT/SQRTF(EXXT*EYYT)Z
```

```
J0868        RTR=EXYTR/SQRTF(EXXTR*EYYT)Z
J0928        SDTX=SQRTF(EXXT/(BIGN*BIGN))Z
J0988        SDTRX=SQRTF(EXXTR/(BIGN*BIGN))Z
J1048        TMNX=SSX/BIGNZ
J1084        TRMNX=XTR/BIGNZ
J1120        BETA2=SDTX/SDTRXZ
J1156        ALPH2=TMNX-(BETA2*TRMNX)Z
J1216        DO 2 I=1,KZ
J1228        BETA3(I)=BETA2*BETA(I)Z
J1312        ALPH3(I)=(BETA2*ALPHA(I))+ALPH2Z
J1408        EMNX(I)=(BETA3(I)*XMEAN(I))+ALPH3(I)Z
J1552        ESDX(I)=BETA3(I)*STDVX(I)Z
J1660      2 CONTINUEZ
J1696        PUNCH 101Z
J1708        PUNCH 102Z
J1720        DO 31 I=1,KZ
J1732        PUNCH103,ID(I),CASES(I),SUMX(I),SUMX2(I),SUMY(I),SUMY2(I),SUMXY(I)
J1996     31 CONTINUEZ
J2032        PUNCH 104,BIGN,SSX,SSX2,SSY,SSY2,SSXYZ
J2116        PUNCH 105Z
J2128        PUNCH 106Z
J2140        DO 32 I=1,KZ
J2152        PUNCH107,ID(I),CASES(I),XMEAN(I),YMEAN(I),STDVX(I),STDVY(I),R(I)Z
J2416     32 CONTINUEZ
J2452        PUNCH 108Z
J2464        PUNCH 109Z
J2476        PUNCH 111Z
J2488        DO 33 I=1,KZ
J2500        PUNCH 112,ID(I),EXX(I),EYY(I),EXY(I),ALPHA(I),BETA(I)
J2728     33 CONTINUEZ
J2764        PUNCH 113Z
J2776        PUNCH 102Z
J2788        DO 34 I=1,KZ
J2800        PUNCH103,ID(I),CASES(I),SXTR(I),SX2TR(I),SUMY(I),SUMY2(I),SXYTR(I)
J3064     34 CONTINUEZ
J3100        PUNCH 104,BIGN,XTR,X2TR,SSY,SSY2,XYTRZ
J3184        PUNCH 114Z
J3196        PUNCH 115Z
J3208        DO 35 I=1,KZ
J3220        PUNCH116,ID(I),XMEAN(I),STDVX(I),EMNX(I),ESDX(I),ALPH3(I),BETA3(I)
J3484     35 CONTINUEZ
J3520        PUNCH 117,TMNX,SDTX,TRMNX,SDTRX,ALPH2,BETA2Z
J3604        PUNCH 118,RT,RTRZ
J3640    101 FORMAT(2X18HORIGINAL DATA SUMS/)Z
J3712    102 FORMAT(7HGRP.   N,6X4HSUMX,9X5HSUMX2,6X4HSUMY,6X5HSUMY2,9X5HSUMXY/)
J3872    103 FORMAT(I3,F6.0,F11.2,F14.2,F9.0,F12.0,F14.2)Z
J3924    104 FORMAT(/3HSUM,F6.0,F11.2,F14.2,F9.0,F12.0,F14.2//)Z
J4000    105 FORMAT(15X24HORIGINAL DATA STATISTICS/)Z
J4084    106 FORMAT(43HGRP.  N    MEAN X   MEAN Y   ST.DEV.X ST.DEV.Y,7H  R(XY)/)Z
J4220    107 FORMAT(I3,F6.0,3F8.3,F9.3,F9.4)Z
J4272    108 FORMAT(//)Z
J4300    109 FORMAT(20X24HORIGINAL DATA STATISTICS/)Z
J4384    111 FORMAT(4HGRP.,7X3HLXX,12X3HLYY,11X3HLXY,5X5HALPHA,6X4HBETA/)Z
J4528    112 FORMAT(I3,3F14.0,2F10.3)Z
J4576    113 FORMAT(//15X33HTRANSFORM TO SLOPE=1, INTERCEPT=0/)Z
```

SAMPLE PROBLEM # 2 LISTING (CONT'D)

```
J4688      114 FORMAT(8X8HORIGINAL,7X11HTRANSFORMED/)Z
J4776      115 FORMAT(43HGRP.MEAN X ST.DEV.X MEAN X ST.DEV.X  ALPHA,4X4HBETA/)Z
J4914      116 FORMAT(I3,4F8.3,2F9.4)Z
J4966      117 FORMAT(/3HSUM,4F8.3,2F9.4//)Z
J5042      118 FORMAT(15HORIGINAL R(XY)=F7.4,3X18HTRANSFORMED R(XY)=F7.4)Z
J5156      201 FORMAT(I2)Z
J5178      202 FORMAT(44HWHERRY TRANSFORM EQ. FOR EQUATING SUBGROUPS.//)Z
J5300          ENDZ
                                                                      00118
```

SYMBOL TABLE LISTING FOR PROBLEM # 2

ADDRESSES		NAME	DIMENSION	STATE LOC	FORM LOC	FIX	FLOATING	A
19999		SIN						A
19989		SINF						A
19979		COS						A
19969		COSF						A
19959		EXP						A
19949		EXPF						A
19939		LOG						A
19929		LOGF						A
19919		SQRT						A
19909		SQRTF						A
19899		ABS						A
19889		ABSF						A
19879		DRH						A
19869		DRHF						A
19859		ATAN						A
19849		ATANF						A
19839	19759	CASES	0009 0001					A
19749	19669	SUMX	0009 0001					A
19659	19579	SUMX2	0009 0001					A
19569	19489	SUMY	0009 0001					A
19479	19399	SUMY2	0009 0001					A
19389	19309	SUMXY	0009 0001					A
19299	19219	EXX	0009 0001					A
19209	19129	EYY	0009 0001					A
19119	19039	EXY	0009 0001					A
19029	18949	BETA	0009 0001					A
18939	18859	ALPHA	0009 0001					A
18849	18769	R	0009 0001					A
18759	18679	XMEAN	0009 0001					A
18669	18589	YMEAN	0009 0001					A
18579	18499	STDVX	0009 0001					A
18489	18409	SXTR	0009 0001					A
18399	18319	SX2TR	0009 0001					A
18309	18229	SXYTR	0009 0001					A
18219	18139	BETA3	0009 0001					A
18129	18049	ALPH3	0009 0001					A
18039	17959	EMNX	0009 0001					A
17949	17869	ESDX	0009 0001					A
17859	17779	STDVY	0009 0001					A
17769	17689	ID	0009 0001					A
17679		BIGN						A
17669							00 00000000	A
17659		SSX						A
17649		SSX2						A
17639		SSY						A
17629		SSY2						A
17619		SSXY						A
17609		X2TR						A
17599		XYTR						A
17589		XTR						A
17579					0202 15178			A
17569					0201 15156			A
17559		K						A
17549				0003 07020				A

17539	I				A
17529			0103 13872		A
17519		0004 07076			A
17509		0005 07388			A
17499		0006 07352			A
17489		0001 10292			A
17479				001	A
17469				003	A
17459				000	A
17449				002	A
17439				004	A
17429	EXYTR				A
17419	EXXTR				A
17409	EYYT				A
17399	EXXT				A
17389	EXYT				A
17379	RT				A
17369	RTR				A
17359	SDTX				A
17349	SDTRX				A
17339	TMNX				A
17329	TRMNX				A
17319	BETA2				A
17309	ALPH2				A
17299		0002 11660			A
17289			0101 13640		A
17279			0102 13712		A
17269		0031 11996			A
17259			0104 13924		A
17249			0105 14000		A
17239			0106 14084		A
17229		0032 12416			A
17219			0107 14220		A
17209			0108 14272		A
17199			0109 14300		A
17189			0111 14384		A
17179		0033 12728			A
17169			0112 14528		A
17159			0113 14576		A
17149		0034 13064			A
17139			0114 14688		A
17129			0115 14776		A
17119		0035 13484			A
17109			0116 14914		A
17099			0117 14966		A
17089			0118 15042		A

00101

374 4.24.22

DEPARTMENT OF CIVIL ENGINEERING

Civil Engineering Systems Laboratory

MONITOR SUPERVISOR FOR 1620-1311
OR
SUPERMONITOR

E. E. Newman

Research Engineer

Paper Presented to the 1620 Users Group

EASTERN REGION MEETING

May 6-8, 1963

37

4.250

MONITOR SUPERVISOR FOR 1620-1311 OR SUPERMONITOR

The Civil Engineering Systems Laboratory at M.I.T. consists in part of an IBM 1620 60K computer with:

a. Floating-Point Hardware
b. Automatic Divide
c. 500/min Card Reader
d. 250/min Card Punch
e. Indirect Addressing
f. The 3 additional instructions
   Move Flag
   Transfer Numerical Strip
   Transfer Numerical Fill
g. A Model 565 Cal Comp Plotter
h. 2 - 1311 Disk Drives
i. 1 - IBM 1050 Remote Console
used to time share a 7094.

These facilities are never used for production at least not in the context of the general meaning of the word.

Rather, they are used to implement computer-aided teaching to show students how engineers analyse and solve complex problems. Two suspended TV screens enable, even those in the back row, to see everything that takes place at either of the 2 Consoles, the plotter or the typewriter. Research has become such an integral and essential part of the advancement of the engineering educator and of the graduate education that between classes the room is used as a Research Laboratory. Thus the computer is used to aid in the study of the fundamental principles of engineering disciplines and the methodology of civil engineering practice and as an aid to research so that nearly all

the available time is used for the compilation of programs rather than production.

The Civil Engineering Department has offered to all students for a number of years now a course entitled "Computer Approaches to Engineering Problems." The number of students electing to take this course has been steadily increasing. This year the number of students selecting this course reached such proportions that, because of the machine time available, only 1/2 of those enrolling were accommodated. The interest in this course is mainly attributable to the turn-around time which is measured in minutes rather than days and because of the hands-on-experience. The room when used as a Laboratory is operated on an open-shop basis. Time is available to students 7 days a week from 1 P.M. to midnight on a 1st come, 1st served basis. The time from 8 A.M. to 1 P.M. is reserved for both academic and sponsored research. Students and research personnel who require large blocks of time are permitted to sign up in advance for the hours from midnight to 8 A.M.

The course is set up so that the 1st two lectures and a lab session provide students with sufficient information to write simple FORTRAN Programs. Thus, after the first week of the course students begin to utilize the computer.

The 1st lab session consists of an introduction to the 1620, its peripheral equipment and the operating procedure to be used for the compilation and execution of programs. The lectures include a brief explanation of computers in general and some of the basic FORTRAN.

To meet these requirements and to accommodate the ever increasing number of students simply meant that the throughput be maximized.

take maximum advantage of both the hard- and software. To implement this a supervisor for the Monitor 1 System was written, debugged, tested and put into operation.

The Supervisor as written performs two tasks in addition to maximizing the throughput.

1. Maintains a log of all users.
2. Supervises the input.

A flow diagram which is self-explanatory is shown in Fig. 1.

A listing of the program is shown in Fig. 1A.

The supervisor at the present time includes the following systems:

1. LOG
2. EDIT
3. SSGO
4. SSGO/EDIT
5. SSGO/EDIT/GO
6. MONITOR 1
7. SUBSET
8. LOAD
9. COGO
10. CI/COGO
11. PLOT

All these systems cards are available at the Console as are end of job cards, cold start cards and clock cards. See Fig. 2.

The order of cards for any job is as follows:

1. User ID Card
2. System Card
3. Program Deck

FLOW DIAGRAM FOR SUPERVISOR OF MONITOR

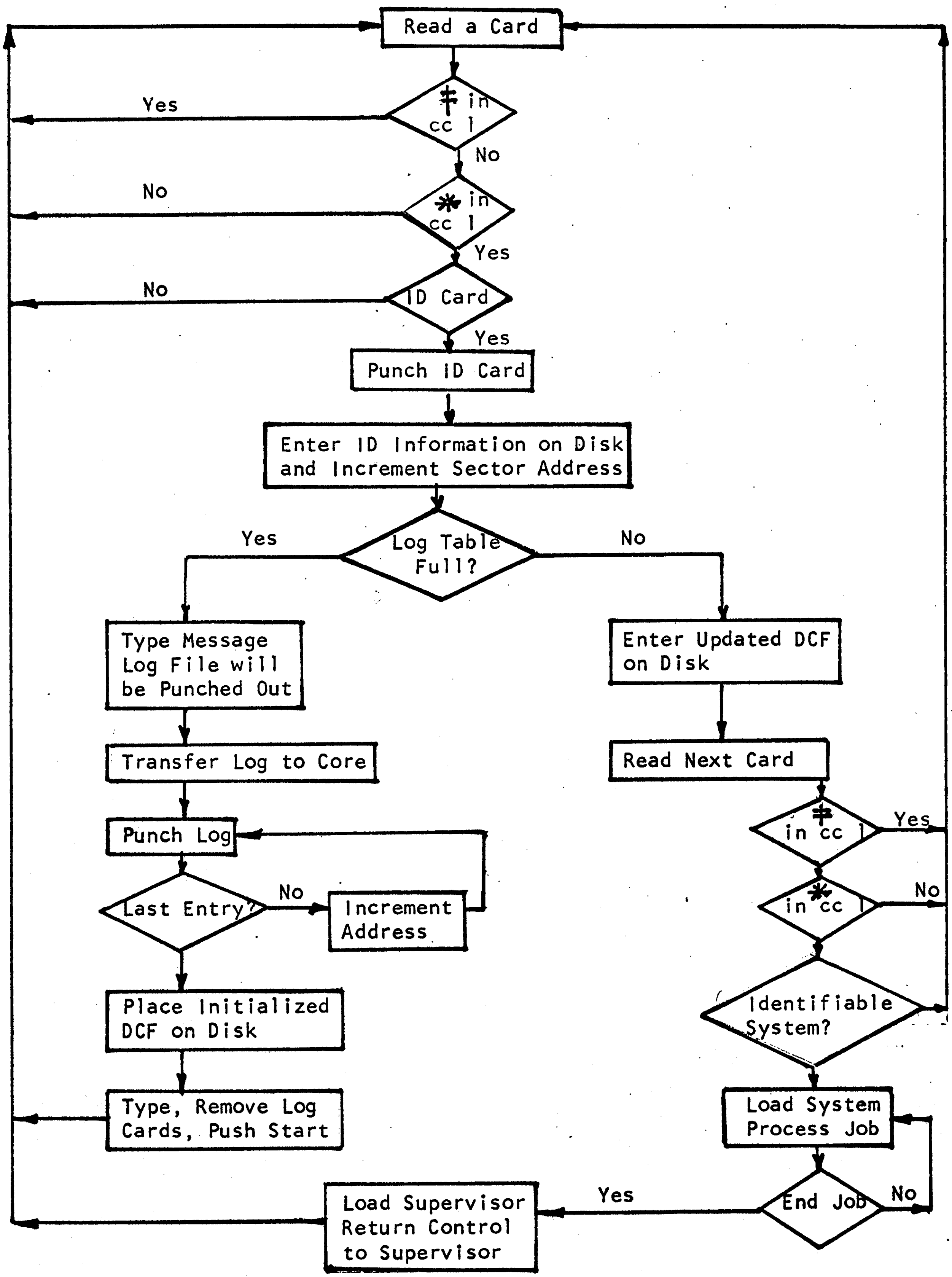


Fig. 1

379

4.25.4

LISTING OF CESL SUPERVISOR OF MONITOR .

COMMENT	ADD.	INSTRUCTION	COMMENTS
READ 1	500	37 02701 00500	READ ID CARD
	12	45 00536 02701	CHECK FOR RECORD MARK IN CC 1
	24	49 00500 00000	IF A RECORD MARK READ NEXT CARD
	36	14 02701 000J4	IS THERE AN * IN CC 1
	48	47 00500 01200	READ NEXT CARD IF NOT *
	60	14 02715 0M944	CHECK FOR ID CC7 AND 8
	72	47 00500 01200	IF NOT ID READ NEXT CARD
	84	39 02701 00400	PUNCH ID CARD
LOG F	96	38 02900 00702	WRITE ID INFO ON DISK
	608	11 02905 00001	INCREMENT SECTOR ADD
	20	14 02905 05400	IS LOG TABLE FULL
	32	46 00728 01300	BRANCH IF HIGH ZERO IS ON
	44	38 02914 00702	WRITE UPDATED CONTROL FIELD ON DISK
	56	37 02701 00500	READ SYSTEM ID CARD
	68	45 00692 02701	CHECK FOR RECORD MARK IN CC 1
	80	49 00500 00000	
	92	14 02701 000J4	CHECK FOR * IN CC 1
	704	47 00500 01200	
	716	49 00980 00000	TO DECODE SYSTEM NAME
LOG PUNCHOUT	28	34 00000 00102	RETURN CARRIAGE
	40	39 02501 00100	TYPE LOG FILE WILL BE PUNCHED OUT
	52	36 02928 00702	TRANSFER LOG FILES TO CORE
E	64	26 02817 03099	3099 IS HIGH ORDER POS OF FIRST LOG ENTRY
	76	45 00800 02737	CHECK FOR END OF ENTRIES
	88	49 00848 00000	
B	800	39 02719 00400	PUNCH ID CARD
	12	11 00775 00J00	INCREMENT ADDRESS TO GET FOLLOWING LOGS
	24	14 00775 K0099	LAST RECORD
	36	47 00764 01300	
OUT	48	15 03019 0000Z	SET A RECORD MARK IN LOC 3019
	60	11 00854 00J00	INCREMENT TO GET ADD. OF NEXT ID
	72	14 00854 J9919	IS THIS LAST ID ENTRY
	84	47 00848 01200	
	96	38 02928 00702	MOVE ID LOG TO CORE
	908	16 02905 05229	
	20	38 02914 00702	PLACE INITALIZED FIELD ON DISK
	32	34 00000 00102	
	44	39 02561 00100	TYPE REMOVE LOG CARDS, PUSH START
	56	49 01672 00000	
	968	48 00000 00000	
SYSTEM	80	14 02715 0N347	LG
	92	47 01016 01200	
	1004	49 00728 00000	CALL LOG PUNCH OUT
	16	14 02715 0N456	MONITOR 1
	28	46 01314 01200	
	40	14 02715 00264	SUBSET
	52	46 01720 01200	
	64	14 02715 00262	SSGO
	76	46 01382 01200	
	88	14 02715 0M544	EDIT
	1100	46 01200 01200	
	12	14 02715 0M371	C1/COGO
	24	46 01450 01200	
	36	14 02715 0M356	COGO
	48	46 01522 01200	

380

4.25.5

FIG. 1A

```
                60    14 02715 0N356       LOAD
                72    46 01868 01200
                84    49 01888 00000
              1200    31 50000 01220
                12    49 50000 0
                20    345003200701365003200702495004601060001990010O
                66    3450078007013650078007024901208010620012020000070
              1314    31 00000 01334
                26    49 00000 0
                34    34000320070136000320070249024025119636113001022
CALL SSGO
              1382    31 500000 01402
                94    49 50000 0
              1402    345003200701365003200702494000801058630504000002
CALL C1/COGO
              1450    31 00000 01470
                62    49 00000 0
                70    340003200701360003200702490750001090000J990010OZ00000
CALL COGO
              1522    31 00000 01542
              1534    49 00000 0
              1542    34001080070136001080070234000940070136000940070 2
              1590    3400080007013600080007024907500010780019900100
              1636    1080002002000010820020040000Z0
              1672    16 00854 03019
              1684    16 00775 03099
                96    41 00000 00000
              1708    49 00500 00000
CALL SUBSET
              1720    31 15000  01744
              1732    49 15000 00000
              1744    31 00031 01356
                56    34 00032 000701
                68    36 00032 00702
                80    31 13000 15108
                92    25 13001 15123
              1804    25 13003 15123
                16    15 13021 00000
                28    16 04105 000M9
                40    49 02402 00000
                52    0000M6N6N907P50Z      ZZ FORX 5 000
              1868    36 00000 00500
              1880    49 00000 0
              1888    14 02715 0N753        PLOT - COGO PLOT
              1900    46 01920 01200
              1912    49 00500 0
              1920    31 00000 01940
              1932    49 00000 0
              1940    340007000701360007000702340005600701360005600702490750 00
              1996    1 08600 200 20000 1 08400 199 00100 Z
             02501    37 02501 00100      LOG FILE WILL BE PUNCHED OUT Z
             02561    37 02561 00100      REMOVE LOG  CARDS, PUSH  START Z
A             2700    00000000000000000090
L1            2900    1 05229 001 02718   WHERE TO START NEXT LOG ENTRY
L2            2914    1 05228 001 02900   UPDATED FIELD FOR ABOVE
L3           02928    1 05229 170 03000   LOAD AND UNLOAD LOG FILE
```

FIG. 1A CONT.

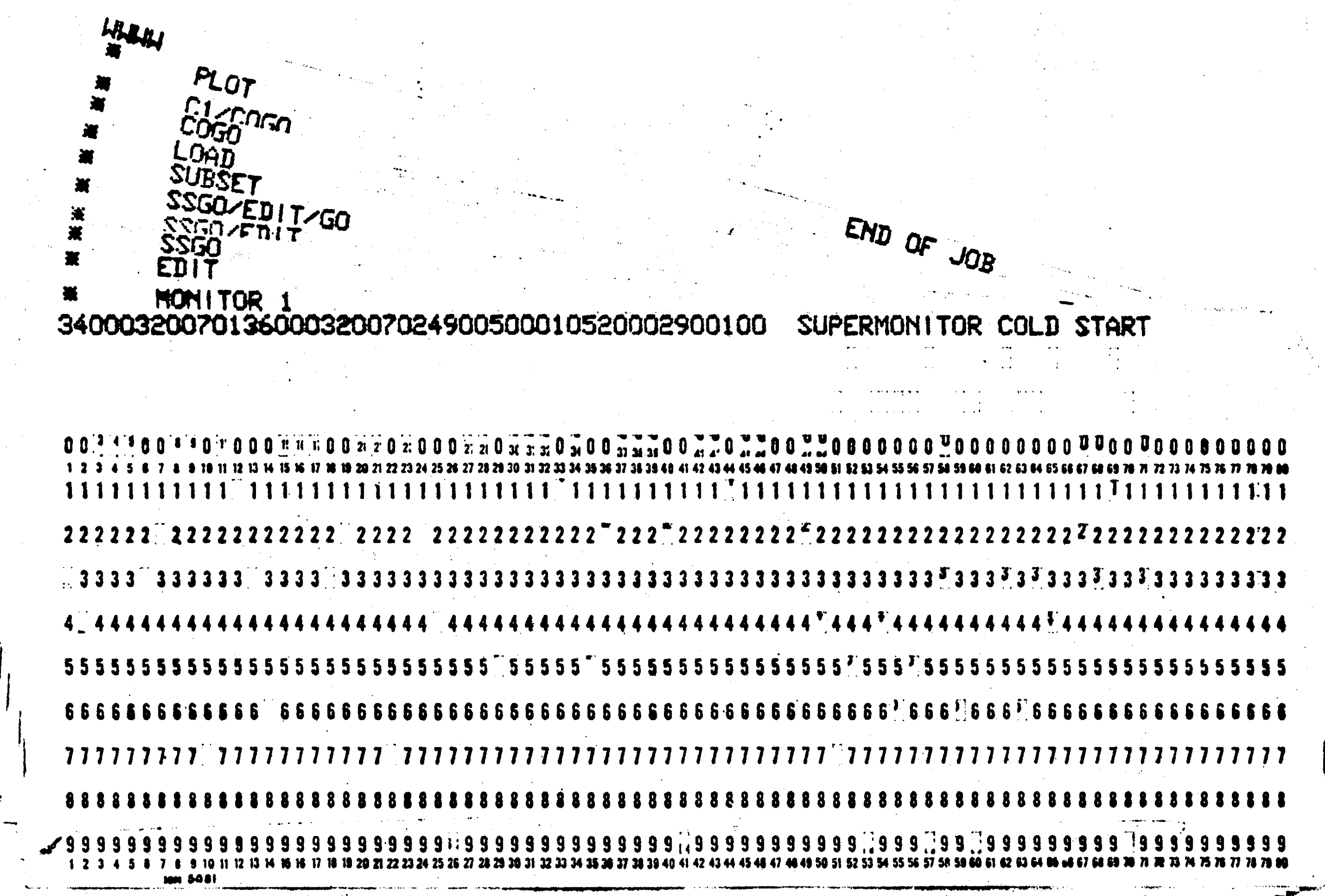


Fig. 2. SYSTEMS CARDS

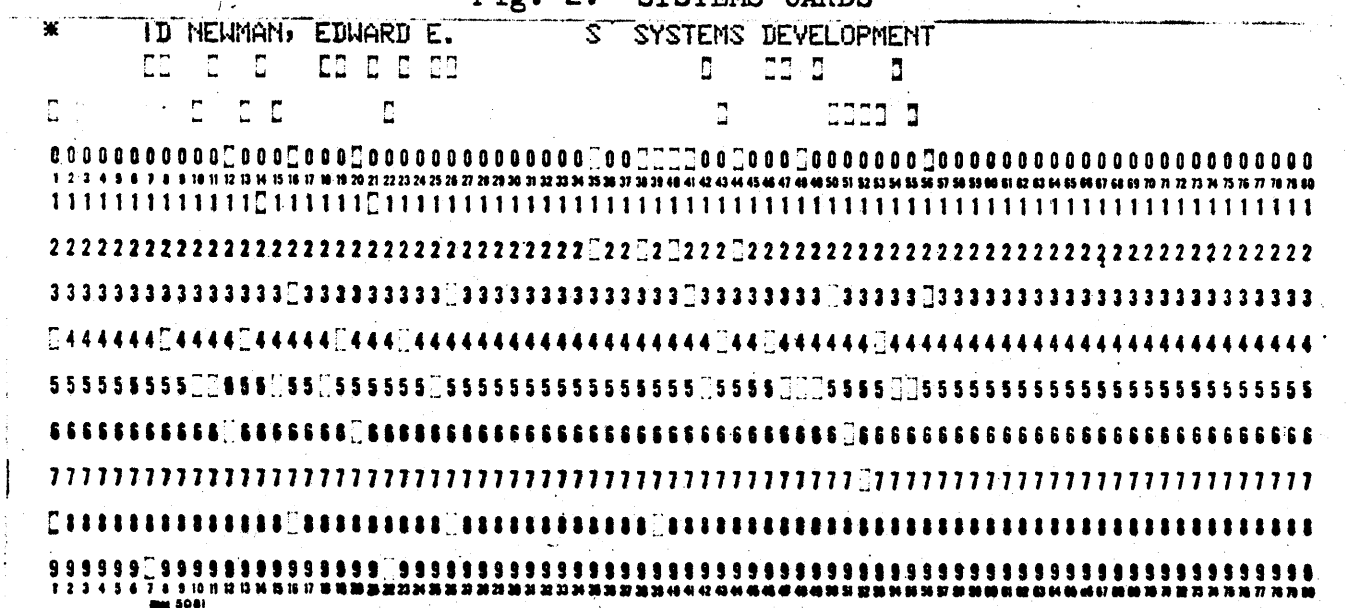


Fig. 3. USER ID CARD

4.25.7

4. * Data Card (when required)
5. End Job Card

The FORMAT for the User ID card is as shown in Fig. 3.

LOG

The 1620 does not have a clock so a system to keep track of the time was devised. To do this a Clock card containing the time and date are read in and entered into the log in the same manner as the ID card. The Format is identical to that of an ID card except the user's name is replaced by Power-On, Clock, or Power-Off and the time and date replace the use. See Fig. 4. One clock card for each hour of the day is pre-punched and entered into the log, on the hour, while the machine is in use. This is accomplished by stacking them in the read hopper along with the jobs being processed. When the log is punched and listed it contains the time power was turned on for the particular day and the names and application of all users to the next hour. The same information with hourly time is repeated until the power is turned off. As was stated previously, the log table is automatically punched out when full. A Log card provides the capability to punch out the log whenever it is desired.

The results of this log were amazing; on two different days 344 starts were logged. One such log is shown in Fig. 5.

EDIT

Edit is a modified version of IBM's FORTRAN with FORMAT PRECOMPILER. It will detect many of the common errors, particularly language, key punching and specification mistakes but it is not exhaustive.

Its suggested use is in conjunction with the SSGO system which does not require Format. But it may be used

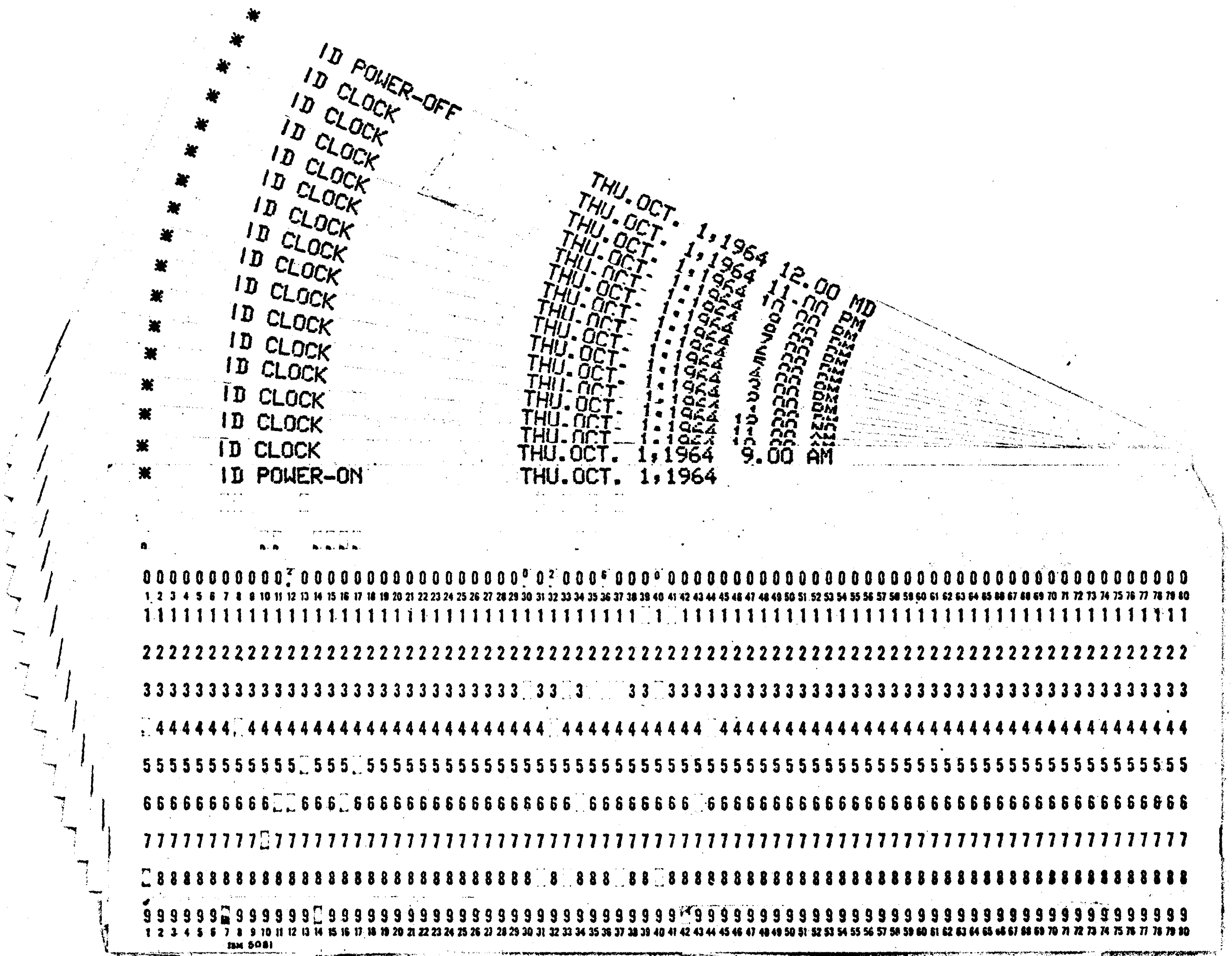


Fig. 4. CLOCK CARDS

DEPARTMENT OF CIVIL ENGINEERING

COMPUTER LABORATORY

RUN LOG

HOURLY ENTRIES FOR THU.MAR.26,1964

TIME	PROJ.	TOTAL ENTRIES
	1.15	
12.01 AM - 1.00 AM	0	15
1.00 AM - 2.00 AM	0	14
2.00 AM - 3.00 AM	0	15
3.00 AM - 4.00 AM	0	17
4.00 AM - 5.30 AM	0	21
8.30 AM - 10.00 AM	2	16
10.00 AM - 1.00 PM	0	12
1.00 PM - 2.00 PM	6	9
8.30 AM - 10.00 AM	2	18
2.00 PM - 3.00 PM	20	21
3.00 PM - 4.00 PM	33	34
4.00 PM - 5.00 PM	28	32
5.00 PM - 6.00 PM	32	35
6.00 PM - 7.00 PM	12	15
7.00 PM - 8.00 PM	30	30
8.00 PM - 9.00 PM	14	16
9.00 PM - 10.00 PM	22	25
10.00 PM - 12.45 AM	4	15
TOTAL NUMBER OF ENTRIES	203	344

385

FIG. 5

4.25.10

DEPARTMENT OF CIVIL ENGINEERING
COMPUTER LABORATORY
RUN LOG

POWER-ON		THU.MAR.26,1964	12.01 AM	
SUHRBIER, JOHN	1	S	9407
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
CLOCK		THU.MAR.26,1964	1.00 AM	
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
SUHRBIER, JOHN	1	S	9407
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
SUHRBIER, JOHN	1	S	9407
JESSIMAN	1	G	9430
JESSIMAN	1	G	9430
GARDNER,D.		S	1673F
SUHRBIER, JOHN	1	S	9407
GARDNER,D.		S	1673F
SUHRBIER, JOHN	1	S	9407
CLOCK		THU.MAR.26,1964	2.00 AM	
SUHRBIER, JOHN	1	S	9407
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
GARDNER,D.		S	1673F
MURRAY,W.E.		1	S
CLOCK		THU.MAR.26,1964	3.00 AM	

386

4.25.11

SUHRBIER, JOHN	1	S	9407	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
MURRAY,W.E.		1	S	
GARDNER,D.		S	1673F	
MURRAY,W.E.		1	S	
GARDNER,D.		S	1673F	
MURRAY,W.E.		1	S	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
CLOCK		THU.MAR.26,1964		4.00 AM
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
MURRAY,W.E.		1	S	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
GARDNER,D.		S	1673F	
POWER-OFF		THU.MAR.26,1964		5.30 AM
POWER-ON		THU.MAR.26,1964		8.30 AM
KIM,JIN HWAN	1	G	DSR 8790	
KIM,JIN HWAN	1	G	DSR 8790	
KIM,JIN HWAN	1	G	DSR 8790	
NEWMAN, EDWARD E.		S		
JOHNSON,WILLIAM F	1	S	DSR NO 9107	
JOHNSON,WILLIAM F	1	S	DSR NO 9107	
JOHNSON,WILLIAM F	1	S	DSR NO 9107	
JOHNSON,WILLIAM F	1	S	DSR NO 9107	
JOHNSON,WILLIAM F	1	S	DSR NO 9107	
JOHNSON,WILLIAM F	1	S	DSR NO 9107	

4.25.12

10(3)

JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
HODNICK, J. M.	I	G	1.15
LEARY D.F.		1	1.15
CLOCK		THU.MAR.26,1964	10.00 AM
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
JOHNSON,WILLIAM F	1	S	DSR NO 9107
FOSTER A. T.	1	GRA	SYSTEMS DEVELOP.
ROESSET J.M.	1	GTA	SPEC. PROJECT
ROESSET J.M.	1	GTA	SPEC. PROJECT
CLOCK		THU.MAR.26,1964	1.00 PM
NEWMAN, EDWARD E.		S	
NEWMAN, EDWARD E.		S	
NEWMAN, EDWARD E.		S	
BAYNE A R	10	3	1.15
SEXAUER,W.L.		1	1.15
GOLDMAN,F.W.		1	1.15
LAZAR, JAY L.	1	2	1.15
TAYLOR, WILLIAM A.	I	1	1.15
SHWIMER, JOEL		1	1.15
CLOCK		THU.MAR.26,1964	2.00 PM
GIBSON, C T			IBM SYSTEMS ENGINEER
TELSON,MICHAEL		1	1.15
TELSON,MICHAEL		1	1.15
TELSON,MICHAEL		1	1.15
TELSON,MICHAEL		1	1.15
TELSON,MICHAEL		1	1.15
TELSON,MICHAEL		1	1.15
SEXAUER,W.L.		1	1.15
SEXAUER,W.L.		1	1.15
SEXAUER,W.L.		1	1.15
HUIE,J.L.		1	1.15
ALAM	1	G	1.15
KINGSNORTH, R C		1	1.15
LEARY D F		1	1.15
LEARY,D,F		1	1.15
LEARY D F		1	1.15
BATOR RICHARD	18	3	1.15
TAYLOR, WILLIAM A.	I	1	1.15
MOORER, JAMES		1	1.15
SEXAUER,W.L.		1	1.15
LAZAR, JAY L.	1	2	1.15

4.25.13

CLOCK THU.MAR.26,1964 3.00 PM

FAIRBURN, DON	1	G	1.09
MUMFORD,J.		1	1.15
TELSON,M.		1	1.15
MUMFORD,J.		1	1.15
ALAM	1	G	1.15
TELSON,M.		1	1.15
BATOR RICHARD	18	3	1.15
KINGSNORTH, R C		1	1.15
GOLKA, JOHN J.		1	1.15
LEARY D F		1	1.15
LEET KENNETH	1	G	1.15
TAYLOR, WILLIAM A.	I	1	1.15
WANEK, GLENN	I	1	1.15
PROKOPY, JOHN C	2	4	1.15
WANEK, GLENN	I	1	1.15
NAINIS	I	1	1.15
LAZAR, JAY L.	1	2	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
KINGSNORTH, R C		1	1.15
LEARY D F		1	1.15
BERNHARDT,L.J.		1	1.15
BERNHARDT,L.J.		1	1.15
GOLKA, JOHN J.		1	1.15
BATOR RICHARD	18	3	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
PROSEK, J. R.	1	3	1.15
SNYDER, M. B.		4	1.15
LAZAR, JAY L.	1	2	1.15

CLOCK THU.MAR.26,1964 4.00 PM

KAVALLIEROS ANTONY	1	G	1.27
KAVALLIEROS ANTONY	1	G	1.27
NEWMAN, EDWARD E.		S	
FAIRBURN DON	1	G	1.27
TENNISON, THOMAS J.		1	1.15
SIKES,P.G.	I	1	1.15
NAINIS	I	1	1.15
ERNST, D.	21	3	1.15
DEDRICK D F	16	2	1.15
CHAPMAN, D G		1	1.15
MATTES,D.A.		1	1.15
HUIE,J.L.		1	1.15
KITTREDGE	18	2	1.15
TENNISON, THOMAS J.		1	1.15
JOSEPH,GUNTER		G	1.15
LEARY D F		1	1.15
PHILLIPS, C.R.	VI	4	1.15
WANEK, GLENN	I	1	1.15
KINGSNORTH, R C		1	1.15

4.25.14

10(5)

KUBO, A. S.	IA	G	1.15	
SEXAUER,W.L.		1	1.15	
SEXAUER,W.L.		1	1.15	
SEXAUER,W.L.		1	1.15	
SEXAUER,W.L.		1	1.15	
SEXAUER,W.L.		1	1.15	
SEXAUER,W.L.		1	1.15	
ALAM	1	G	1.15	
MABRY R. E.	I	G	1.15	
PHILLIPS, C.R.	VI	4	1.15	
DEDRICK D F	16	2	1.15	
BERNHARDT,L.J.		1	1.15	
BERNHARDT,L.J.		1	1.15	
CLOCK		THU.MAR.26,1964		5.00 PM
NEWMAN, EDWARD E.		S		
NEWMAN, EDWARD E.		S		
COOPERBERG		7	OPERATOR	
NEWMAN, EDWARD E.		S		
SEXAUER,W.L.		1	1.15	
MATTES,D.A.		1	1.15	
JOSEPH,GUNTER		G	1.15	
WANEK, GLENN	I	1	1.15	
SHWIMER, JOEL		1	1.15	
GOLDMAN,F.W.		1	1.15	
SEXAUER,W.L.		1	1.15	
LEET KENNETH	1	G	1.15	
HUIE,J.L.		1	1.15	
SEXAUER,W.L.		1	1.15	
SEXAUER,W.L.		1	1.15	
BERNHARDT,L.J.		1	1.15	
JOSEPH,GUNTER		G	1.15	
SEXAUER,W.L.		1	1.15	
DEDRICK D F	16	2	1.15	
DEDRICK D F	16	2	1.15	
GOLDMAN,F.W.		1	1.15	
MATTES,D.A.		1	1.15	
WANEK, GLENN	I	1	1.15	
BERNHARDT,L.J.		1	1.15	
JOSEPH,GUNTER		G	1.15	
JOSEPH,GUNTER		G	1.15	
JOSEPH,GUNTER		G	1.15	
GOLDMAN,F.W.		1	1.15	
CHAPMAN, D G		1	1.15	
GRAASS J H	14	4	1.15	
BERNHARDT,L.J.		1	1.15	
CHAPMAN, D G		1	1.15	
BERNHARDT,L.J.		1	1.15	
CHAPMAN, D G		1	1.15	
CHAPMAN, D G		1	1.15	
CLOCK		THU.MAR.26,1964		6.00 PM
NEWMAN,E.		S		
ROBINSON,J.V.		S	9407	
NEWMAN,E.		S		
BERNHARDT,L.J.		1	1.15	
BERNHARDT,L.J.		1	1.15	

4.25.15

10(6)

CHAPMAN, D G		1	1.15
CHAPMAN, D G		1	1.15
CHAPMAN, D G		1	1.15
LEARY D F		1	1.15
LEARY D F		1	1.15
CHAPMAN, D G		1	1.15
CHAPMAN, D G		1	1.15
CHAPMAN, D G		1	1.15
CHAPMAN, D G		1	1.15
LEARY D F		1	1.15
CLOCK		THU.MAR.26,1964	7.00 PM
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
LEARY D F		1	1.15
LEARY D F		1	1.15
CHAPMAN, D G		1	1.15
CHAPMAN, D G		1	1.15
CHAPMAN, D G		1	1.15
LEARY D F		1	1.15
LEARY D F		1	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
MOORER, JAMES A.		1	1.15
LEARY D F		1	1.15
BERNHARDT,L.J.		1	1.15
LEARY D F		1	1.15
CHAPMAN, D G		1	1.15
LEARY D F		1	1.15
CHAPMAN, D G		1	1.15
GOLKA, J. J.		1	1.15
LEARY D F		1	1.15
WILLIAMS,RE			1.15
WILLIAMS,RE			1.15
BERNHARDT,L.J.		1	1.15
CHAPMAN, D G		1	1.15
DESMOND,BARBARA A.		1	1.15
DESMOND,BARBARA A.		1	1.15
HAUSSLING H J	I	1	1.15
MOORER, JAMES		1	1.15
CLOCK		THU.MAR.26,1964	8.00 PM
SKODON, JOHN G.	I	S	5011
SKODON, JOHN G.	I	S	5011
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
DESMOND,BARBARA A.		1	1.15
CHAPMAN, D G		1	1.15
DESMOND,BARBARA A.		1	1.15
BERNHARDT,L.J.		1	1.15
GOLKA, J. J.		1	1.15
DESMOND,BARBARA A.		1	1.15
WILLIAMS,RE			1.15
GOLKA, J. J.		1	1.15

391

4.25.16

CHAPMAN, D G		1	1.15
BERNHARDT,L.J.		1	1.15
HAUSSLING H J	I	1	1.15

CLOCK THU.MAR.26,1964 9.00 PM

ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
SKODON, JOHN G.	I	S	5011
SKODON, JOHN G.	I	S	5011
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
ENGER,TOM		1	1.15
CHAMPY,J.A.	I	GTA	1.09
HAUSSLING H J	I	1	1.15
GOLKA, J. J.		1	1.15
HAUSSLING H J	I	1	1.15
CHAPMAN, D G		1	1.15
GOLKA, J. J.		1	1.15
HAUSSLING H J	I	1	1.15
CHAPMAN, D G		1	1.15
BATOR RICHARD	18	3	1.15
BATOR RICHARD	18	3	1.15
GOLKA, J. J.		1	1.15
HAUSSLING H J	I	1	1.15
BATOR RICHARD	18	3	1.15
GOLKA, J. J.		1	1.15
GOLKA, J. J.		1	1.15
HAUSSLING H J	I	1	1.15

CLOCK THU.MAR.26,1964 10.00 PM

ENGER,TOM		1	1.15
GARDNER,D.		S	1673F
SKODON, JOHN G.	I	S	5011
SKODON, JOHN G.	I	S	5011
SKODON, JOHN G.	I	S	5011
SKODON, JOHN G.	I	S	5011
HARMAN , JACK	1	S	9407
HARMAN , JACK	1	S	9407
HARMAN , JACK	1	S	9407
HARMAN , JACK	1	S	9407
HARMAN , JACK	1	S	9407
HARMAN , JACK	1	S	9407
LEARY D F		1	1.15
LEARY D F		1	1.15
LEARY D F		1	1.15

POWER-OFF FRI.MAR.27,1964 12.45 AM

NUMBER OF ENTRIES 344

to check source programs in FORTRAN II-D. Since SSGO does not require Format statements, EDIT was modified to by-pass this type of statement.

Each statement in which an error is detected is punched out together with a diagnostic report. It is a useful debugging technique as well as an aid in program development. It is most useful to novice programmers. If no errors are detected the message "NO ERRORS DETECTED" is typed out. Shown in Fig. 6 is a list of the error messages.

<u>SSGO</u>

SSGO is a modified version of a LOAD and GO FORTRAN which was based upon IBM's first 1620 CARD FORTRAN WITHOUT FORMAT. This is a one-pass operation with 3 options. Any, all or none of these options may be exercised by inserting, immediately preceding the Data card, the following:

1. * Trace M to N
   M is the number of the source statement where tracing is to begin and N is the number of the source statement where tracing is to cease.
2. * Map.
   Causes a storage map of the symbol table together with the unused memory to be punched out.
3. * Cards

The original version of this FORTRAN was modified so that the compiler was located in the 2nd Module while the compiled object program was stored in the first 20K of core, thus conserving both time and cards. The Cards card was, therefore, used to obtain a punched object deck. This of course finds little use today since it is more expedient to load on disk any program that might be needed later.

4.25.18

ERROR MESSAGES IN EDIT

UNDEFINED STATEMENT NUMBERS
UNREFERENCED STATEMENT NUMBERS
UNACCEPTABLE FORM TO LEFT OF = SIGN
MULTIPLE = SIGNS
SUCCESIVE OPERATION SYMBOLS OR OPERATORS
MISSING OPERATION SYMBOL OR OPERAND
RIGHT PARENTHESIS ENCOUNTERED BEFORE LEFT PARENTHESIS
MISSING RIGHT PARENTHESIS
MIXED MODE EXPRESSION
NO VARIABLE TO LEFT OF = SIGN
EXPONENTIATION OF A FIXED POINT VARIABLE OR CONSTANT
VARIABLE USED HAS NOT BEEN DEFINED
SUBSCRIPTED VARIABLE HAS NOT APPEARED IN A DIMENSION STATEMENT
ERROR IN SUBSCRIPT
NOT ALL OF THE INDICES ARE LEGAL
SECOND INDEX IS LESS THAN FIRST
THIRD INDEX IS ILLEGAL
STATEMENT NUMBER NOT ACCEPTABLE OR IS MISSING
DO STATEMENT IS INCOMPLETE
DO LOOPS INCORRECTLY NESTED
DO LOOP TERMINATES ON TRANSFER STATEMENT
DECIMAL POINT OMITTED FROM FLOATING POINT CONSTANT
DECIMAL EXPONENT FOLLOWING AN E IS INCORRECTLY EXPRESSED
THE EXPONENT FOLLOWING AN E HAS BEEN OMITTED
FLOATING POINT NUMBER FOLLOWED BY CHARACTER NOT AN E
STATEMENT NUMBER GREATER THAN 9999
STATEMENT NUMBER PREVIOUSLY DEFINED
UNNUMBERED CONTINUE
STATEMENT CANNOT BE REACHED BY PROGRAM
ERROR IN STATEMENT NUMBER(S)
MISSING COMMA AFTER RIGHT PARENTHESIS
INDEX IS ILLEGAL OR MISSING
NON-NUMERIC CHARACTER FOLLOWS RIGHT PARENTHESIS
LEFT PARENTHESIS DOES NOT FOLLOW WORD IF
NO EXPRESSION WITHIN PARENTHESIS
MISSPELLED OR UNACCEPTABLE NONARITHMETIC STATEMENT
STATEMENT CONTAINS UNACCEPTABLE CHARACTER
STATEMENT CONTAINS DECIMAL POINT NOT IN A CONSTANT
DIMENSION STATEMENT IS IN INCORRECT FORM
UNNUMBERED FORMAT STATEMENT
INCOMPLETE FORMAT STATEMENT
VARIABLE IN DIMENSION STATEMENT PREVIOUSLY DEFINED
FORMAT STATEMENT NUMBER MISSING OR INCORRECTLY STATED
FIRST OR LAST CHARACTER IN LIST IS NOT ACCEPTABLE
THREE DIMENSIONS HAVE BEEN SPECIFIED, OR IN INCORRECT FORM
COMMA MISSING, OR LIST IS MISSING OR INVALID
SYMBOL TABLE FULL
FIXED POINT CONSTANT GREATER THAN 9999
FLOATING POINT CONSTANT OUT OF RANGE
VARIABLE NAME LONGER THAN 5 CHARACTERS

Fig. 6

4.25.19

3.5 SSGO DEBUGGING AIDS

Trace

SSGO includes a built-in trace which is a very powerful debugging aid. Tracing is specified by the insertion of a control card after the END statement of the following form:

```
*  TRACE FROM n TO m
```

where n is the statement number where tracing is to begin and m is the statement number where tracing is to cease.

The portion of the program which lies logically between the two limiting statement numbers will be traced each time it is executed. If the trace is selected, object program execution proceeds in normal mode at full speed until control reaches the statement with statement number "n". From this point until control reaches the statement with statement number "m", in addition to the normal programmed output the following output is punched:

1. For each arithmetic statement, the variable name to the left of the = sign followed by the current numeric value of the right hand side. Examples:

   ```
   A = -.13570000
   STA = .10600000E03
   JOB = 8431
   ```

2. For each transfer statement, the type of transfer and the statement number branched to. Example:

   ```
   COMPUTED GO TO
   TRANSFER TO STATEMENT NO. 0006
   ```

3. Reports of progress through DO loops are punched at the end of each loop. Example:

   ```
   0012TH TIME THROUGH DO
   ```

   The DO message assumes the index has an initial value of one and is

For compatibility a Call Exit command was added to this processor. Format statements may be included but they are ignored except the = character in an H field which will cause erroneous compilation. All Print statements have been modified to Punch to speed up the operation.

SSGO/EDIT

SSGO/EDIT is a further effort to conserve time and to simplify the operation procedure. When this system is specified the source statements, as they are encountered, are temporarily stored on disk and processed with the EDIT program. If no errors are detected control is returned to SSGO which reads the source statements from disk instead of from cards and proceeds to compile and execute. If at the completion of the EDIT phase the program was found to contain errors, control is returned to the supervisor rather than SSGO.

SSGO/EDIT/GO

SSGO/EDIT/GO is a special version of SSGO/EDIT.

At the end of the EDIT phase if no errors are detected the message "NO ERRORS DETECTED" is typed. Processing and execution are attempted just as with SSGO/EDIT. If at the completion of the EDIT phase errors were found the message "Turn SW-1 on to compile with errors" is typed. If SW-1 is left off and the start button on the console is depressed control is returned to the supervisor. If the option to continue is selected, an attempt is made to compile and execute.

SUBSET

SUBSET provides a direct call to FORTRAN II-D for a Load and Go operation and essentially replaces 2 MONITOR I CONTROL CARDS:

Processing

As source statements are read and translated by the processor, the object program is built up in lower core starting at location 07500. After processing of the END statement and detection of the DATA card, control is sent to 07500 and the object program is executed. The object program occupies the first 20K module. The processor occupies the second 20K and the third 20K is then reserved for special subroutines. Programs of the order of 75 to 125 statements can be handled assuming no large arrays. For SSGO debugging of programs involving large arrays, the arrays should be temporarily dimensioned for small ranges to conserve space in the symbol table.

Error Indicators - Translation Phase

During translation, if certain kinds of errors are detected in the source statements, error message of the following form is typed:

```
ZG      A = B*C+5
ERROR NO. 5
```

The statement containing the error followed by the error number is typed. The error numbers and their meaning are as follows:

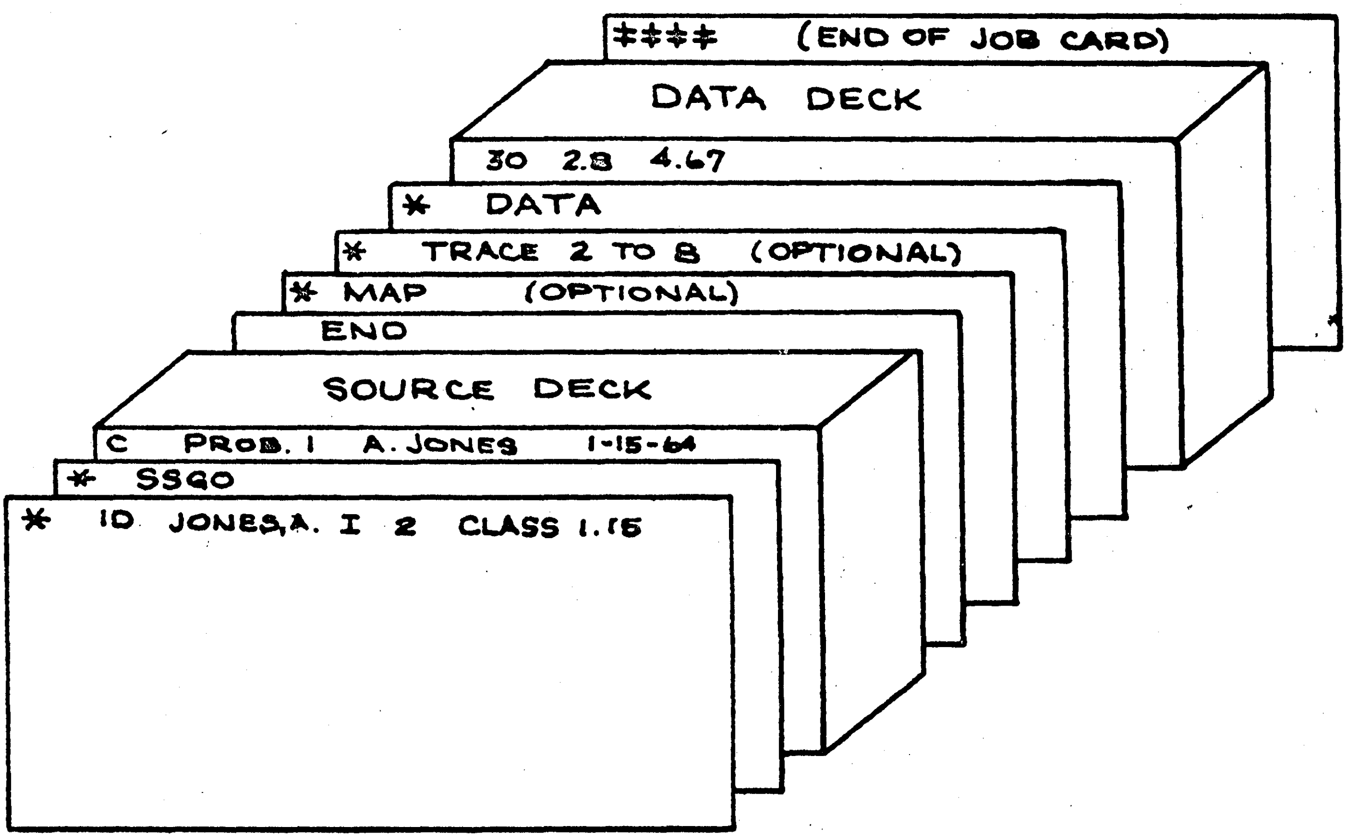
ERROR NO. 1	Incorrectly formed statement
ERROR NO. 2	Subscripted variable not dimensioned
ERROR NO. 3	Constant out of range
ERROR NO. 4	Symbol table full
ERROR NO. 5	Mixed mode expression

After an error is detected, processing continues so that additional errors may be checked for. However, the resulting object program should not be executed as it will be incorrect.

PROCESSING SSGO RUNS

Run Deck

The make-up of a typical SSGO run deck is indicated in the following diagram.



The * DATA control card is necessary for execution even if no input data is required by the program since it is the signal to execute the program.

Fig. 7

* MONITOR

≠≠ FORX 5

SUBSET is provided to increase the compatibility between the 1620 and 7094. Students enrolled in the previously mentioned course, "Computer Approaches to Engineering Problems," offered by the Civil Engineering Department use not only the 1620 but the 7094's. The 7094's are accessible either by direct submission of card decks or through the 1050's or teletype units as time shared remote consoles. 1620 SUBSET programs may be processed and executed on the 7094 by merely changing the ID and systems card to conform to that required at the 7094 Computation Center.

To speed up compilation time with SUBSET or FORTRAN II-D all the Print statements in FORTRAN II-D have been changed to Punch. The Type statements were left unchanged.

Error messages are no longer typed out but instead are punched on cards; the incorrect statement on one card and the coded error message on the following card. During execution the floating point subroutines punch errors but the carriage returns associated with the original typewriter output still occur to provide a warning. The Data card initially required by FORTRAN II-D is still required but the word DATA starts in cc 7 the same as for SSGO and 7094 FORTRAN.

MAKE UP SUBSET RUN DECK

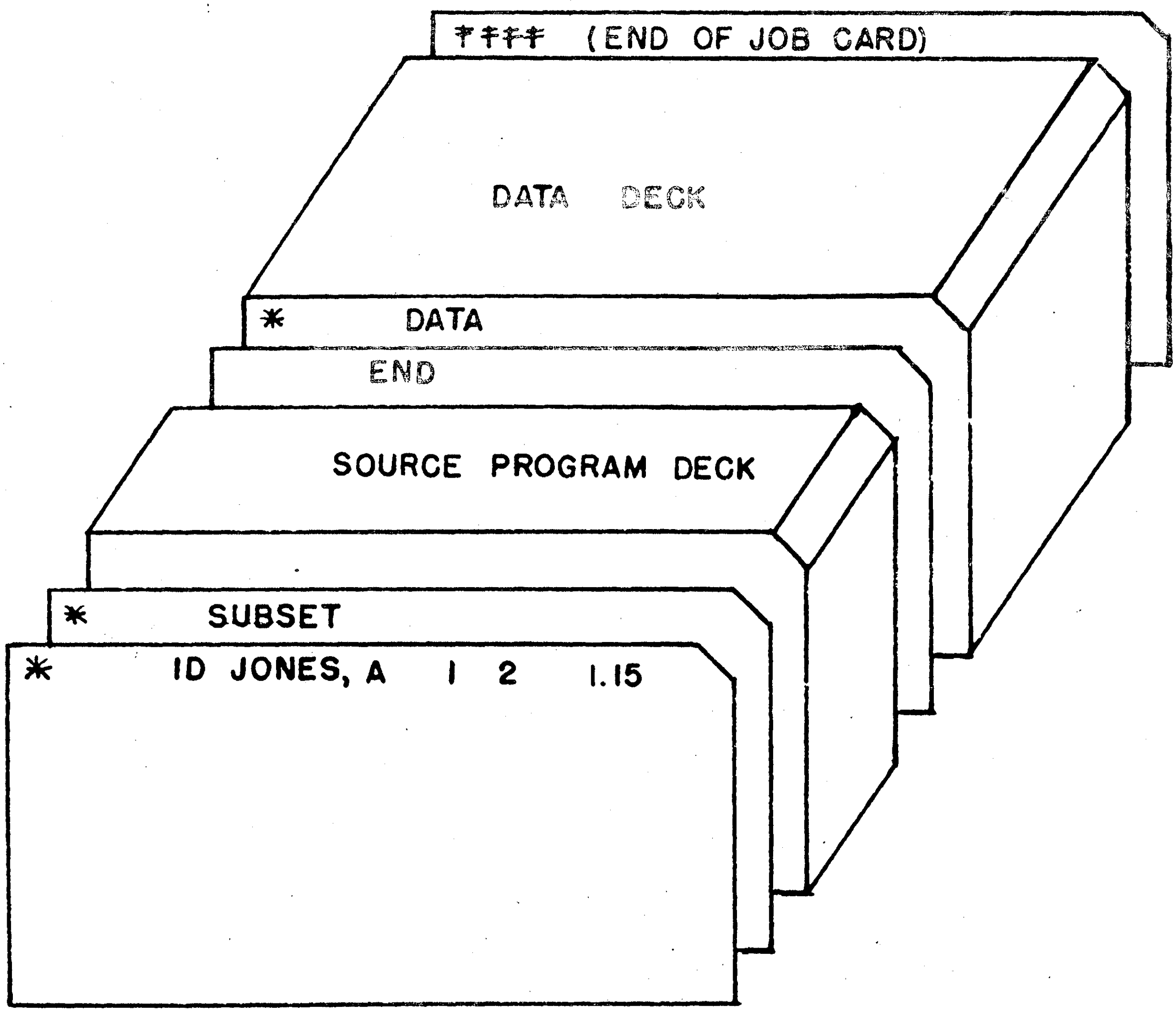


Fig. 8

400

4.25.25

Error Reports - Translation Phase

As source statements are compiled, checks are made for source program errors. If an error is detected, a message of the following form is punched:

SSSS + CCCC ERROR n

where SSSS is the last statement number encountered by the program prior to the error, CCCC is the number of statements following the last numbered statement, SSSS + CCCC is the statement containing the error, and n is the error code number. If an error is encountered before a statement number is encountered, SSSS will be 0000. Comment cards, blank cards, and continuation cards are not included in the statement count. Only one error per statement is detected. The following is a table of error code numbers and their meaning:

Error No.	Condition
1	Undeterminable, misspelled, or incorrectly formed statement.
2	Syntax error in a nonarithmetic statement (exception: DO statements).
3	Dimensioned variable used improperly, i.e., without subscripting, or subscripting, appears on a variable not previously dimensioned.
4	Symbol table full (processing may not be continued).
5	Incorrect subscript.
6	Same statement number assigned to more than one statement.
7	Control transferred to FORMAT statement.
8	Variable name greater than 6 alphameric characters.
9	Variable name used both as a nondimensioned variable name and as a Subroutine or Function name.
10	Invalid variable within an EQUIVALENCE statement.
11	Subroutine or Function name or dummy variable used in an EQUIVALENCE statement (subprogram only).
13	Within an Equivalence list, placement of two variables previously in Common, or one variable previously equivalenced and another either equivalenced or placed in Common.

MAKE-UP OF 7094 FORTRAN RUN DECK

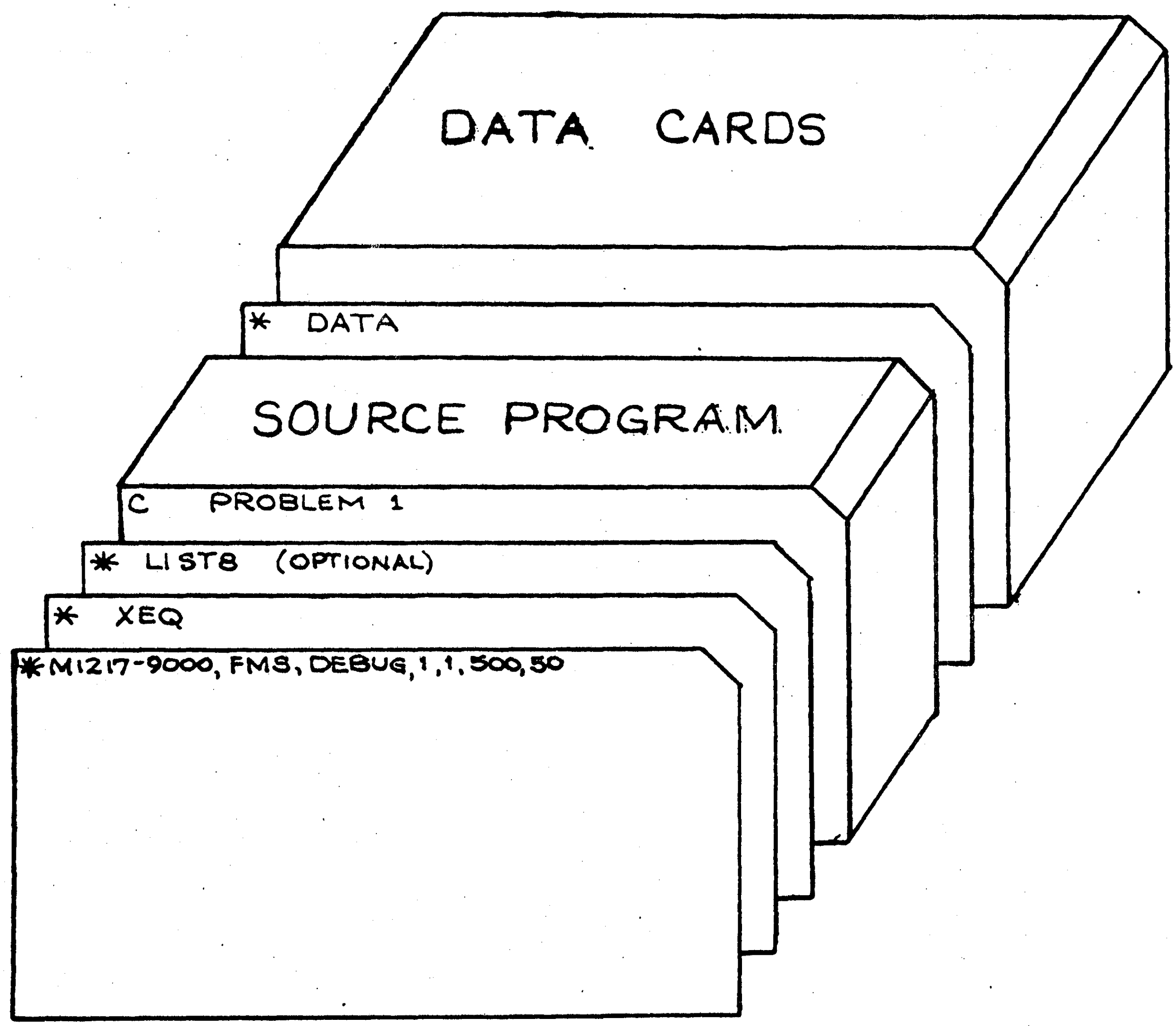


Fig. 9

402

4.25.27

b) Execute Card (Control Card)

* XEQ

This must be the second card in your deck. It informs the processor that your program is to be executed immediately after it is compiled.

c) List Card (Control Card)

* LIST8

This should be the third card in your deck. It informs the processor that a listing of the equivalent machine language object program resulting from the compilation is to be included in your printed output. This listing is helpful in off-line debugging.

d) Source Program Deck

Source statements according to specifications. Last card must be END statement.

e) Data Card (Control Card)

* DATA

f) Data Deck

Data cards for testing your program at execution, prepared in accordance with Format statement specifications.

A typical run deck is shown in the diagram which follows. The above control cards are the only ones which may be included.

Submission of 7094 Runs

Student decks, assembled as described above, should be delivered to Room 1-151 before 4:30 PM and deposited in the IN box for the appropriate subject (such as 1.15). The runs will be batched and delivered to the Computation Center for processing overnight. Results are returned the following

MONITOR 1

MONITOR 1 calls in the Monitor System provided by IBM with the 1311 disks. This card merely transfers control from the supervisor to the Monitor System. Modifications mentioned in connection with SUBSET still hold. The monitor program was, however, modified so that when an end of job card is encountered, control is returned to the supervisor.

LOAD

LOAD is not really a system but simply a means to simulate the load button on the 1622. It provides the capability for stacking machine language programs along with any of the other system jobs. The user must provide his own loader and must terminate his program with a specified branch to return control to the supervisor. The add and multiply tables are brought into core with the supervisor and remain there when LOAD is called.

COGO

The COGO associated with this call is a version of COGO prepared specially for a 60K machine with a special application in mind. It was designed to provide maximum efficiency with this hardware configuration. The system is stored on 600 sectors of disk and is transferred to core when called as a unit so that all commands are always in core and available. A new command FINISH was provided so that COGO jobs could be stacked with any of the other systems jobs.

CI/COGO

CI/COGO is a disk version of COGO designed for a 20K machine. It operates much like the card plug deck system except that the plug is read from disk rather than

from cards. It like the plug deck system is limited, at the present time, to typewriter output and is used primarily for demonstration only.

PLOT

PLOT like COGO is a problem-oriented language and at this time is used to graphically display the results of COGO. This is the very first attempt at the adaptation of graphical displays to problem-oriented languages. At present it contains only a minimum command set but programming is under way to greatly expand the system for a wide variety of applications.

All the systems above mentioned are stored on a single disk pack and are file protected. The Equivalence Table, the Dim Table and the Sequential Program Table all contain appropriate entries and are cognizant of the existence of the various systems thus providing additional protection. All the systems in use have been modified so that as each job is terminated control is returned to the supervisor. Thus the supervisor provides a simplified procedure of operation with adequate flexibility and compatibility for an efficient system requiring a minimum of machine time for a maximum number of users.

TYPICAL ARRANGEMENT OF 1620 RUN DECKS

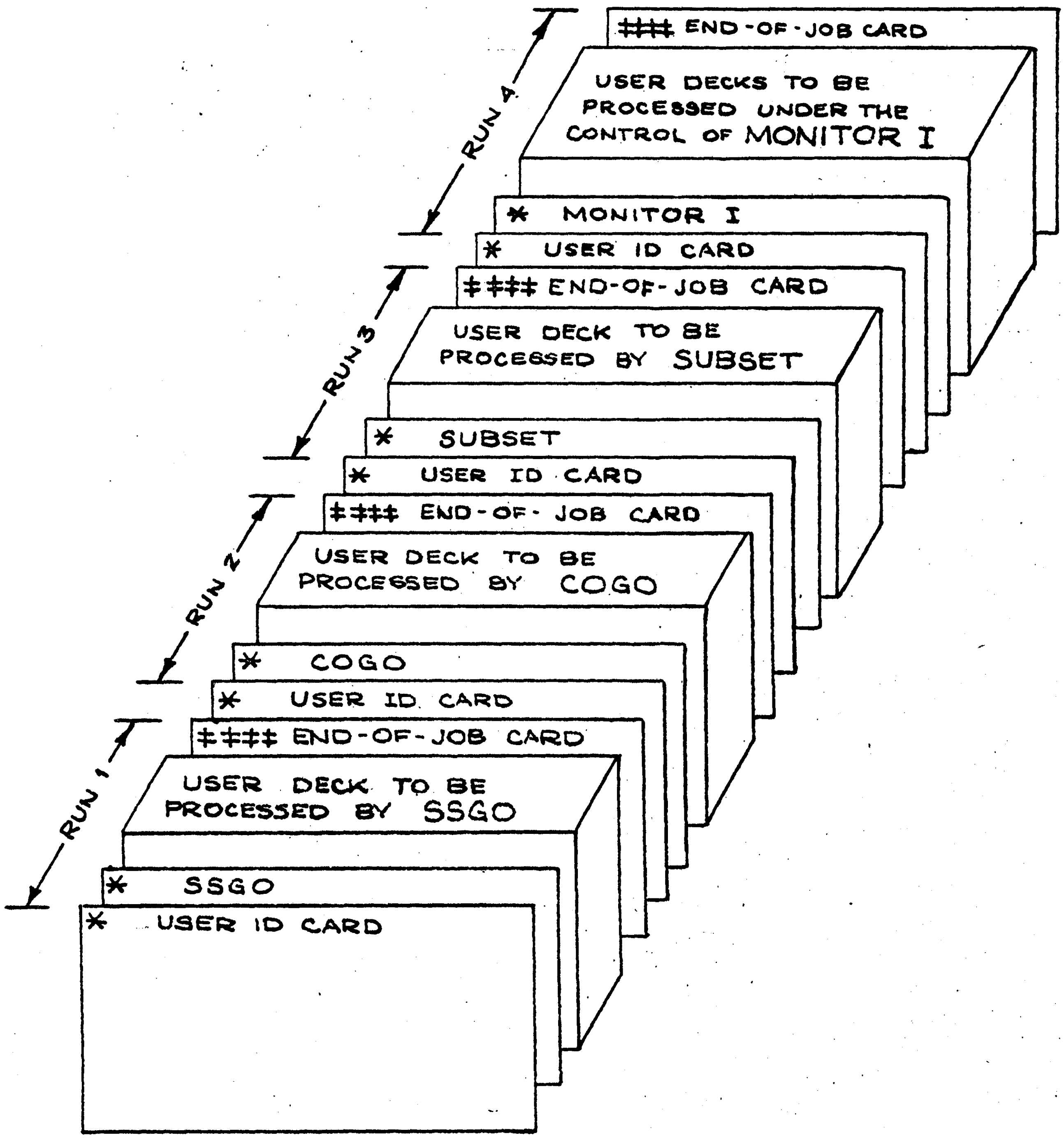


Fig. 10

406

4.25.31

1.2 AVAILABLE SYSTEMS

At the present time, the following systems can be called from disc by the CESL MONITOR via System ID cards as follows:

* EDIT

A program for editing FORTRAN source programs for common programming mistakes.

* SSGO

A simplified load-and-go FORTRAN System especially designed for instructional use.

* SSGO/EDIT

A combination of the first two which will check the program with EDIT and process it with SSGO if no mistakes are detected.

* SUBSET

A subset of FORTRAN II for the 1620 and 7094 designed for debugging 7094 programs on the 1620.

* MONITOR I

Turns control of the machine over to the IBM 1620 Monitor I System. This system in turn can call SPS Symbolic Programming System, FORTRAN II-D, Disc Utility Program, etc., through additional control cards. Use limited to authorized users.

* LOAD

For loading 1620 machine language programs included in stacked input.

* COGO

A problem-oriented language and processor for geometric problems.

* PLOT

A problem-oriented language for preparing plots and graphical displays with the on-line plotter.

Additional systems are under development and will be added from time to time.

USE OF COMPUTERS IN DESIGN OF ELECTRONIC EQUIPMENT

by

Arnold Spitalny
Chief, Computer Branch
Advanced Engineering Section
Norden Division of United Aircraft Corporation
Norwalk, Connecticut

Presented at the Eastern Regional 1620
Users Group meeting in Washington D.C.,
on May 7, 1964

408

22 June 1964

X36018

4.26.0

USE OF COMPUTERS IN DESIGN OF ELECTRONIC EQUIPMENT

(Presented at the Eastern Regional 1620 Users Group meeting in Washington D.C., on May 7, 1964.)

by: Arnold Spitalny
Chief, Computer Branch
Norden Division
United Aircraft Corporation
Norwalk, Connecticut

The theme of this meeting is "Consolidation Thru Systems Integration." The word "Systems" means many things to many people. We are concerned now with a type of system considerably more complex than most of those you are accustomed to dealing with. This is a man-machine system for design and development of electronic equipment. At Norden, we are concerned primarily with military electronic, electro-optical and electro-mechanical equipment, such as radar, displays, navigation systems and vehicle control systems. However, the same methods and techniques are applicable in many other areas.

The first problem in system design is always to define the problem. What are the problems in the military electronics business? Some of the basic trends are indicated in Figure 1.

The rapid advance in electronic technology over the past few years has resulted in a very rapid rate of obsolescence for all electronic equipment. The demand for increased capabilities and specialized customer requirements also results in a steadily increasing complexity of equipment. Competition both in cost and technology is applying greater pressure all the time. Reliability requirements of the Military are getting steadily more severe and those of NASA are practically out of this world. The combination of increasing complexity and stricter reliability requirements results in a demand for a wide variety of automatic check-out equipment, manufactured in much smaller volume than the equipment they check out. Most development programs are subjected to frequent changes due both to changes in customer requirements and to engineers thinking of better ways to do things they have already designed.

Once the changes stop and production begins, there is usually only a very short production run before rapid obsolescence takes over and the entire cycle repeats. The end result is a steady increase in the average amount of engineering work required to support a given level of production. This is substantiated by the steady industry-wide increase in the ratio of engineering personnel to manufacturing personnel. The basic problem then, is how to get more engineering work done faster and better.

Figure 2 indicates the two basic approaches to this problem. The obvious way is to hire more engineers. We do not consider that an adequate solution over the long run. It is much better to see if we can find a way to use the engineers we have more efficiently by using computers to assist engineers. Computers have been used many times to assist engineers in a wide variety of individual tasks. We are expanding these tasks and developing new ones to provide an over-all man machine system in which engineers use computers effect-

TREND IN MILITARY ELECTRONIC BUSINESS

1. RAPID OBSOLESCENCE
2. INCREASING COMPLEXITY
3. SPECIALIZED REQUIREMENTS
4. INCREASING COMPETITION
5. RELIABILITY
6. AUTOMATIC CHECK-OUT
7. FREQUENT CHANGES
8. SHORT PRODUCTION RUNS

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 1

4.26 2

HOW TO RAISE ENGINEERING OUTPUT

1. HIRE MORE ENGINEERS
2. USE ENGINEERS MORE EFFICIENTLY
   - USE COMPUTERS TO ASSIST ENGINEERS
   - SYSTEM PLANNING, [redacted]

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 2

ively for all tasks computers can do better than man. Under these conditions, engineers have the opportunity to concentrate their efforts on those intellectual functions man can do far better than computers.

Figure 3 indicates some of the basic steps in developing such a system. The first problem is to identify and select the design tasks for which we should consider the possibility of using computers to assist the engineers. For each of these tasks we must determine a set of design rules, which would be in sufficient detail to consider programming on a computer. Preliminary planning of methods and procedures for each task must proceed far enough to provide a realistic basis for estimates of the cost of fully implementing the task , and of the benefits that would result from it. Evaluation of these costs and benefits indicates that there are some tasks on which a substantial early payoff can be achieved from a reasonable degree of effort. Work is started on these tasks as soon as they are discovered. In this way the program is put on a pay-as-you-go basis while planning and development of the rest of the system continues.

Now that you understand the philosophy behind this program, let's take a look at what has actually been done. Figure 4 is a simplified block diagram of the entire series of engineering activities from initial planning of a new project to release of final documentation to other divisions of the company and to the customers. The engineering tasks that presently involve the greatest proportion of computer usage are marked A. Some of those for which we are presently developing computer applications are identified by a B, and those labeled C require the highest proportion of human decisions. Even in this area, computer techniques have been extensively used to provide engineers with the information they need for their decisions. For example, the first step in developing a test console for failure analysis of radar modules was to code the input-output requirements of every module and analyze the requirements on the 1620 computer. This provided information needed by the engineers to specify signal generators and control and processing circuits.

Computer simulation analysis has also been used extensively to check the theoretical performance of proposed systems prior to commitment of detailed designs. For example, in the case of an automatic radar terrain-following system for aircraft, the terrain characteristics, radar performance, processing of radar informantion, airborne computer, action of the autopilot, and the response of the airplane were all simulated on the computer. Many flights were simulated by using the computer. Conditions in the equipment that might cause airplane crashes were eliminated before any equipment was actually designed. We are now working on development of simulation analysis techniques which could be applied to any electronic system at the block diagram level. This would be based on adaptation of the IBM general-purpose system simulator for the 7090 computer.

STEPS IN DESIGN AUTOMATION

1. IDENTIFY & SELECT DESIGN TASKS
2. DETERMINE DESIGN RULES
3. PLAN METHODS & PROCEDURES
4. ESTIMATE COSTS & BENEFITS
5. PERFORM EARLY PAYOFF TASKS
6. EXECUTE LONG RANGE PLAN

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 3

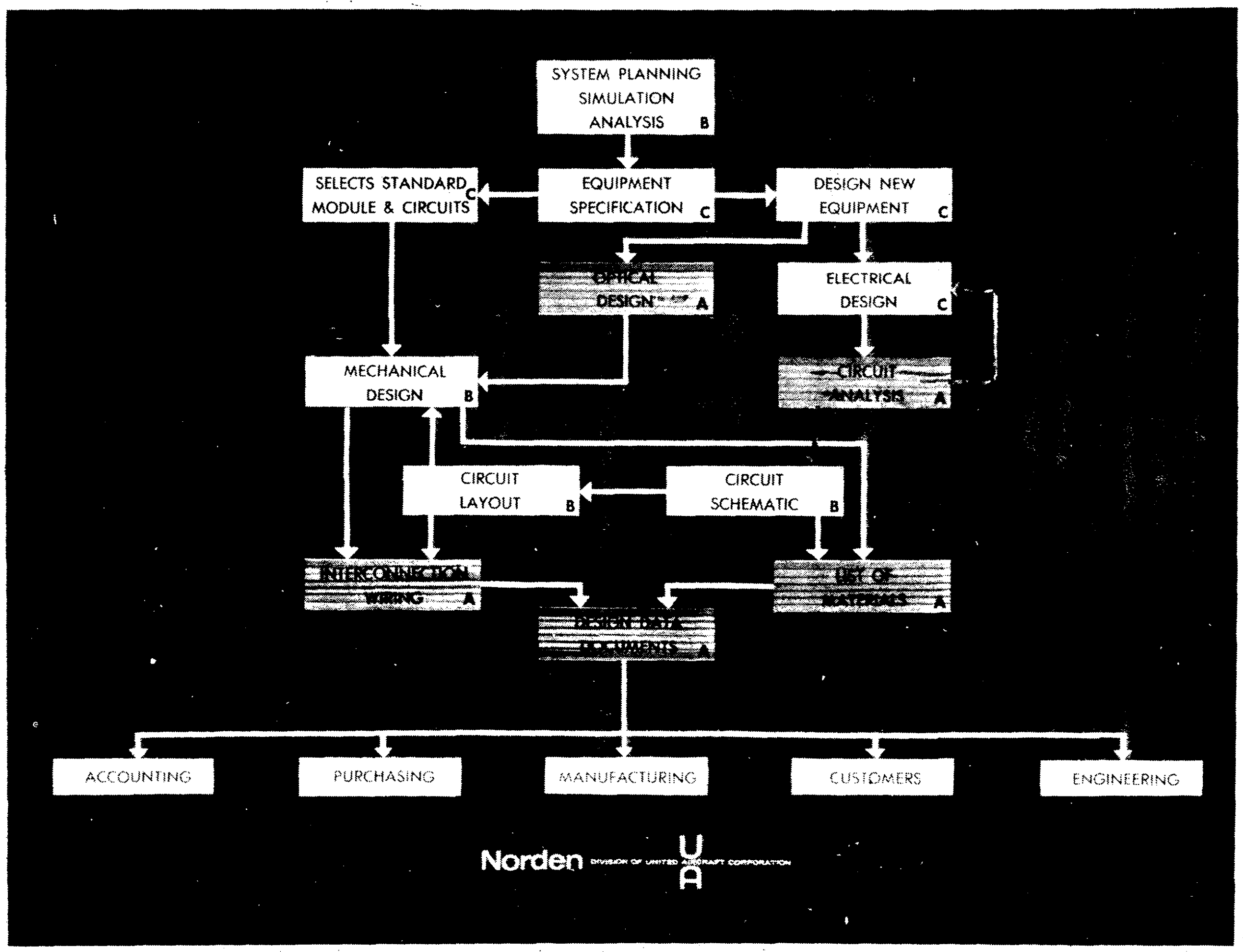

SYSTEM PLANNING
SIMULATION
ANALYSIS B
SELECTS STANDARD
MODULE & CIRCUITS C
EQUIPMENT
SPECIFICATION C
DESIGN NEW
EQUIPMENT C
OPTICAL
DESIGN A
ELECTRICAL
DESIGN C
MECHANICAL
DESIGN B
CIRCUIT
ANALYSIS A
CIRCUIT
LAYOUT B
CIRCUIT
SCHEMATIC B
INTERCONNECTION
WIRING A
LIST OF A
A
ACCOUNTING
PURCHASING
MANUFACTURING
CUSTOMERS
ENGINEERING
Norden

Figure 4

Once the simulation analysis has provided assurance that the planned system will do the job satisfactorily, it is time to specify the detailed equipment and proceed with the design. Whenever possible, standard modules and circuits which have been previously developed and built should be utilized, rather than develop completely new equipment. The computer files data from previous designs which can help to select such equipment. However, there will be many cases in which completely new designs are necessary to meet the new requirements.

The critical bottleneck in electronic system development is usually design, development, and test of new electronic circuits. The traditional method requires building a laboratory breadboard of the circuit and conducting extensive tests and modifications in the laboratory to determine how the circuit would perform under varying conditions. This is a very expensive and time-consuming procedure and never gives optimum design, since the tendancy is to stop at the first thing that appears to work.

Norden and IBM have jointly developed a set of computer programs for analysis of electronic circuits which provides the basis for a real break-through in this critical problem area. With these programs it is possible to predict the performance of many alternative circuits and select the best one before starting to build anything in the laboratory. In addition to predicting the voltage outputs which could be measured in the laboratory, the computer programs also provide sensitivity of all outputs to all circuit parameters, thereby providing much greater insight into the design problem and into the accuracy requirements of the components.

A key feature of the circuit analysis program is the use of a very simple user-oriented input language. This permits preparation of input data directly from schematic circuit diagrams or from equivalent circuit diagrams. There is no need to derive any equations or prepare any new computer program instructions when a completely new and different circuit is analyzed for the first time. Once the new circuit has been described to the computer for purposes of circuit analysis, many other things can be done with the same basic data.

We have extended the basic circuit analysis to include a wide variety of reliability analysis techniques. Tools now available to the reliability engineers for evaluation of new proposed circuit designs include stress analysis, derating analysis, margin analysis, mean time-between-failure calculation, and failure effect analysis. These can all be performed before the circuit is actually built and reliability improvement recommendations can be incorporated in the original system design.

At this point the basic design has been completed but a lot of detailed work is required before the system can be built. A schematic diagram of the final computer-checked design is needed to record the design and to provide information for engineers. We are working on development of techniques for computer printout of schematic diagrams. This will eliminate some routine drafting work. More important, it will provide absolute assurance that the circuit in the diagram corresponds exactly to the circuit analyzed.

The next problem is mechanical design or layout of the circuit. The problems here vary depending on the type of manufacturing technology to be used. In conventional transistor circuitry, this is usually the problem of laying out a printed circuit board.

At Norden we are now concerned increasingly with developing integral circuits, such as that in Figure 5. The layout design problem of these circuits is far more complicated. It involves development of a set of masks for the various etching, diffusion, and deposition steps required for manufacture of a complex semi-conductor integral circuit within a single silicon chip. Norden is presently working on an Air Force contract, developing techniques for automatic design and manufacture of such circuits. This involves use of the same basic circuit input definition which was used for circuit analysis, extended to include geometric as well as electronic information.

Norden has used computer techniques to assist in optical and mechanical design as well in electronic design. The 1620 optical design package has been particularly helpful in developing optical designs for new display techniques. However, we have had to develop some of our own optical design programs to eliminate excessive use of the typewriter and provide data in the coordinates and format most useful to the optical designer.

The mechanical design work at Norden includes such things as design of servo gear trains and critical inertial gimbal platforms as well as structures and housings for electronic equipment. Computer techniques have been developed to assist in mechanical layout and design, and in vibration analysis. However, a lot more remains to be done in this area. Once the electronic modules and the structure for housing them have been designed, the next problem is to route all of the interconnection cabling. This is sometimes referred to as the problem of "Where do all the wires go?" In the Norden design system the basic interconnection wiring data is key punched and fed into the computer. The computer assigns terminal numbers, selects cables, determines wiring length, sorts data, and prints out final wiring lists, which are the shop instructions for wiring the equipment.

The basic manufacturing definition of electronic equipment is in the form of a list of materials. Detailed lists of material must be prepared for every assembly and sub-assembly in the system. These lists of materials contain considerable detailed reference information about each part, as well as its basic part number designation.

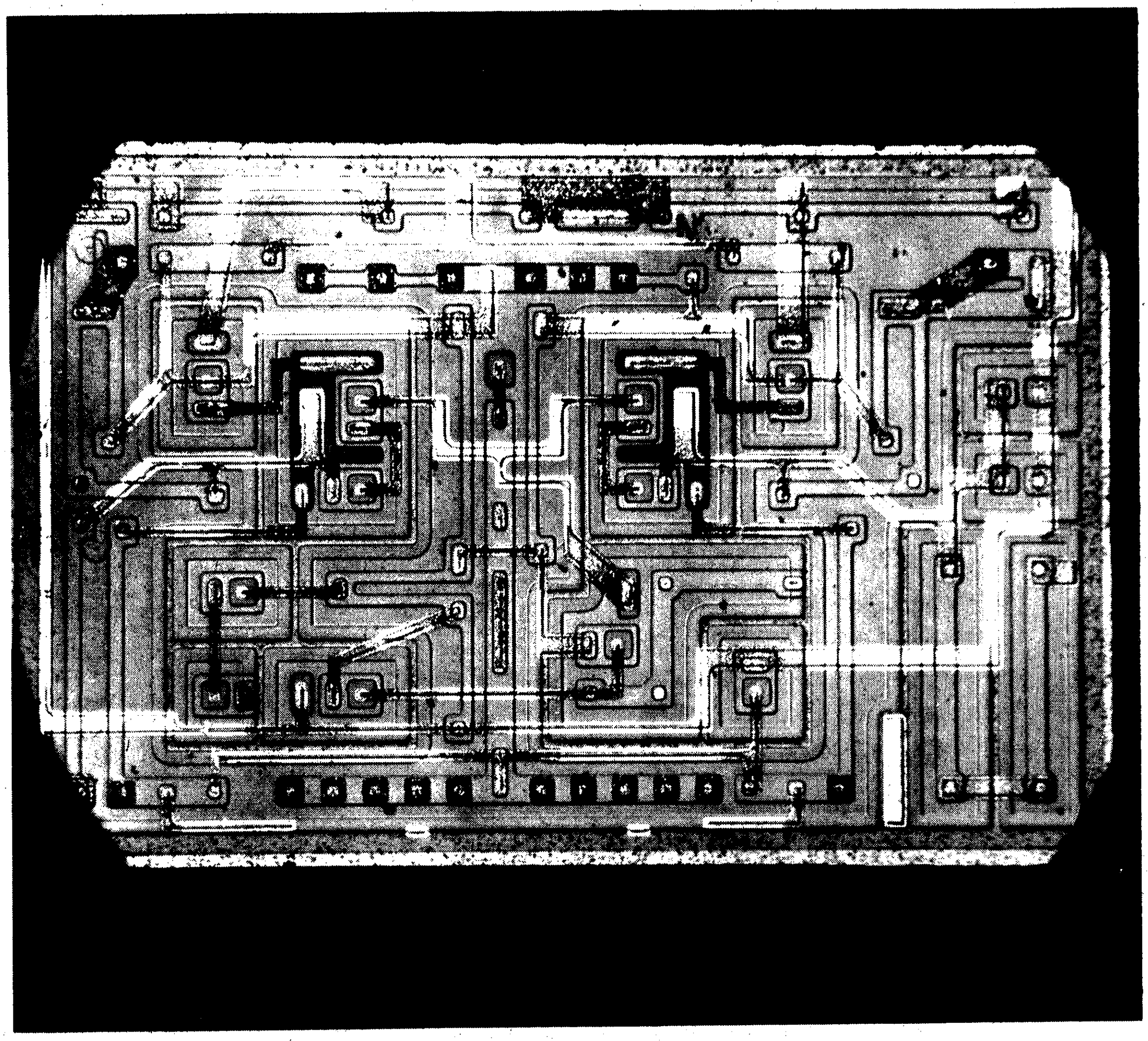


Figure 5

This detailed printed information is maintained on disk files in the 1620 computer for all standard Norden parts. The engineer then has only to designate the part numbers and quantities which are key-punched for computer input. The computer extracts other reference data from the disk and prints out the detailed lists of materials. It also records and stores the list of material information on disk file for later use in generating a wide variety of other design data documents, such as the unique parts lists, lists of electronic components, tube complement lists, non-standard parts lists, and many others. By generating all this detailed information from the same basic system definition, we have greater assurance than ever before that the information is correct and consistent, as well as having it available much earlier, in more complete form, and at lower costs. This design data is the basic output product of engineering which is used by accounting, purchasing, manufacturing, and our customers.

Now that you have seen the over-all system, you may want to know what does all this really mean? How much difference does it make in a complicated program?

Figure 6 indicates a complete equipment currently under development at Norden, the Module Analyzer Test Console (MATC) for ground support of airborne radar equipment. It is something like a fancy tube tester, except that instead of testing tubes it tests complete modules of radar equipment. Several hundred different modules can be accommodated, including all its own internal modules. The control card for the module under test selects the appropriate input signals which are applied to the module, and the appropriate processing circuits for examining the module output voltages, so that the source of failure or malfunction can be isolated to a particular component within the module. This equipment was developed concurrently with the computer-aided design system. Portions of this system were used on it as they became available.

The first step in design of this equipment was tabulation and computer analysis of the input and output signal requirements of all modules to be tested. This computer analysis assisted the engineers in determining the signal source specifications and the specifications for various signal processing circuits in the MATC equipment. The same module data cards later were used for automatic design of the control cards which select the appropriate signal sources and processing circuits to be connected with each module when it is tested. The complexity of this equipment is indicated by the fact that there are approximately 35,000 parts, of which 4,500 are unique items including about 2,000 electrical components. Two separate units or test benches have been designed, each requiring about 10,000 interconnection wires. All of the interconnection wiring, lists of materials, and other design data for this program were prepared on the 1620 computer, as well as analysis of most of its circuits.

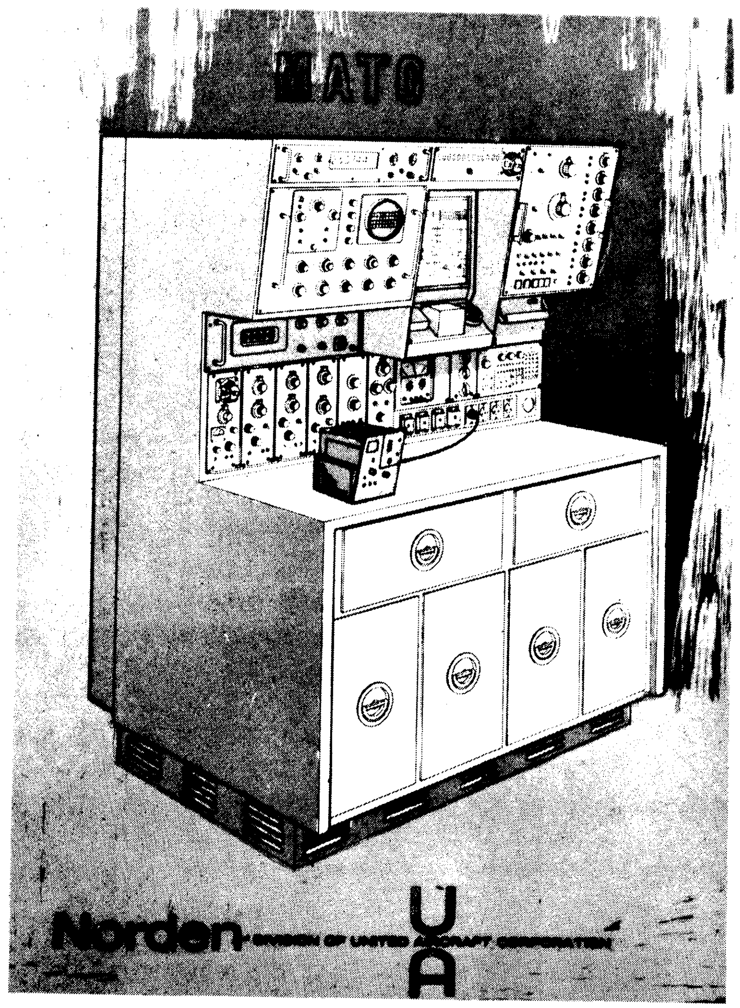


Figure 6

Design and development of a system of this complexity by traditional methods would normally take at least two or three years. This equipment was designed and developed in 1-1/2 years with partial use of experimental computer-aided design techniques. A new system of comparable complexity could be designed starting now in less than one year, with no increase in the average manning level. When all parts of the computer-aided design system are completed and data is flowing smoothly from one computer operation to another, this time will be considerably shortened.

The over-all results of the program are summarized in Figure 7. The time required to get from an engineering idea to its incorporation in a finished product is substantially reduced. The time required to process changes to an existing design is cut to an extent that makes a substantial difference in whether or not it is practical to incorporate many last-minute changes. The total cost of an engineering effort is substantially reduced by shortening the engineering time cycle, without increasing the number of engineers. The resulting products delivered to our customers are greatly improved as a result of providing more thorough preliminary analysis and reliability checking of designs. Finally, the engineering records resulting from this design process are far more complete and accurate than they ever were before.

RESULTS OF PROGRAM

FAST REACTION TIME

[redacted]

IMPROVED PRODUCT

IMPROVED RECORDS

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 7

NETWORK ANALYSIS ON THE 1620

by

Martin J. Goldberg
Computer Applications Engineer
Advanced Engineering Section
Norden Division of United Aircraft Corporation
Norwalk, Connecticut

Presented at the Eastern Regional 1620 Users Group meeting in Washington D.C., on May 7, 1964.

422

22 June 1964

x36017

4.27.0

NETWORK ANALYSIS ON THE 1620

(Presented at the Eastern Regional 1620 Users Group meeting in Washington D.C., on May 7, 1964.)

by: Martin J. Goldberg
Computer Applications Engineer
Norden Division
United Aircraft Corporation
Helen Street
Norwalk, Connecticut

ABSTRACT

A computer program for analyzing electronic circuits on a 40K 1620 with 1311 disc has been developed under a cooperative agreement between the Norden Division of United Aircraft Corporation and IBM.

The Program is divided into four sections: language interpreter, DC analysis, AC analysis, and transient analysis. The analysis sections are limited to linear circuits, although the transient analysis section can approximate some non-linearities. The language interpreter section interprets input data on the circuit to be analyzed. The data is written in a user-oriented language so that the Program can be used by individuals not familiar with programming, and also so that man-machine communication is more efficient. The primary input is an equivalent circuit schematic. Input data cards, prepared from the schematic, describe the circuit topology and parameter values.

From the above data, each analysis section sets up the proper matrix equations describing the network and performs the desired analysis.

The DC analysis section obtains the DC steady-state solution for all node voltages and currents. In addition, there are several optional routines such as worst-case analysis of node voltages, computation of standard deviations of node voltages, determination of sensitivities of node voltages to network parameters and the like.

The AC analysis section obtains the frequency response of the network i.e., magnitude and phase of node voltages and of currents versus frequency.

The transient analysis section obtains the time response of node voltages and currents. It is capable of accepting time varying input sources and also has the capability of switching parameter values from one state to another. This allows for simulation of change of state of transistors, diodes, and other switching elements.

The Program has had a significant impact on circuit design procedures at Norden. Engineers are relieved of routine calculations and are left with more time to apply to creative tasks. A high degree of confidence that a circuit will meet specifications is obtained before manufacturing. This is especially important in the case of integrated circuits. More reliable circuits are obtained since statistical analyses which used to be too time-consuming to perform at all now are done on the computer. In addition, over-design is reduced since the designer can readily see the effect of reduced tolerances on performance.

Norden has already extended the capability of the Program beyond that of the version to be released by IBM, and is working on further extensions such as adding a non-linear capability.

1. INTRODUCTION:

This report will discuss the main features of the Network Analysis Program developed jointly by the Norden Division of the United Aircraft Corporation, and I.B.M. Details of operating procedures and the mathematical theory will not be dealt with at length. The objective here is to describe the usefulness and significance of the program.

A specific sample circuit will be discussed from preparation of input data to obtaining the output from the computer.

The Network Analysis Program is used as a tool by the designer of electronic circuits. It can also be used to review the design of existing circuits for reliablity or optimization.

The Program is written in FORTRAN II for a 40K, 1620 with 1311 Disk Storage Drive, 1622 Card Read Punch, Automatic Divide and Indirect Addressing. The Program also requires the Monitor I System.

2. Advantages of Network Analysis Program.

Figure 1 lists some of the advantages of the Program over conventional design techniques of hand calculations, breadboarding and testing. Points 1 and 2 (figure 1) are obvious advantages. Point 3 emphasizes that conventional techniques cannot give accurate statistical information on the effects of component variations on circuit performance. Because the designer does not have a high degree of confidence that the circuit will perform satisfactorily under all extremes of component variation and environment, he is forced to over-design.

The Network Analysis Program can perform complicated statistical analyses on the circuit, analyses much too time consuming to perform by hand calculation. With this tool the designer can optimize the circuit since the program readily shows the effects of tolerances on performance.

Alternative circuit configurations (point 4) can easily be studied on the computer and the best configuration chosen. In fact, designs incorporating new or expensive components can be studied even if the actual components are not available for breadboarding.

It is often possible to obtain insight into the operation of a circuit by use of the Program (point 5). Also, the effects of parameters, such as transistor current gain, can be determined. On a breadboard one cannot vary the gain of a transistor directly.

ADVANTAGES OF NETWORK ANALYSIS PROGRAM

1. ROUTINE CALCULATIONS ELIMINATED
2. MORE TIME FOR CREATIVE DESIGN
3. EFFECTS OF COMPONENT VARIATIONS
4. ALTERNATIVE DESIGN STUDIED
5. INSIGHT INTO COMPLEX INTERACTIONS
6. BREADBOARDING & TESTING REDUCED

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 1

The factors mentioned contribute to reducing the need for breadboarding (point 6), and even when breadboarding is still required the amount of testing is reduced.

3. Brief Description of Analysis Procedure

Before discussing the details of the Program, it is appropriate to obtain an overall picture of the procedure and flow of information involved in analyzing a circuit by use of the Program. Figure 2 is a flow-chart describing the procedure.

From the specifications and requirements on the circuit, the designer develops a circuit schematic describing the configuration he plans to use. An equivalent circuit schematic is then prepared from the original circuit schematic. The equivalent circuit schematic contains equivalent circuit representations for transistors and diodes. The equivalent circuit of a transistor or diode can be thought of as the model used by the Program to simulate them. The Program will accept any equivalent circuit chosen by the designer.

The input data cards are prepared by referring to the equivalent circuit schematic, using a user-oriented language to describe the circuit.

The Network Analysis Program is divided into four main sections: language interpreter, DC analysis, AC analysis, and transient analysis. The language interpreter translates the input data cards, setting up lists of information for the analysis sections. It then calls on the proper analysis section as specified by the user, and the results of the analysis are computed.

The user-oriented language used to prepare the input Data cards is probably one of the most important features of the Program. It allows the Program to be used by engineers not familiar with the machine or with programming. Also, it allows easy and fast communication between the circuit designer and the computer. Very little of the full potential of the Program would be realized without the capabilities provided by the user-oriented language.

4. Basic Network Branch Defination

The basic entity in the matrix approach used to set up and solve the matrix equations of the network is the network branch. Figure 3 shows the basic branch with a definition of the important variables. As shown, the branch is composed of three network elements: a passive element (resistor, capacitor or inductor); voltage source; and current source. The termination points of the branch are defined as nodes. We may then imagine a network as composed of many branches.

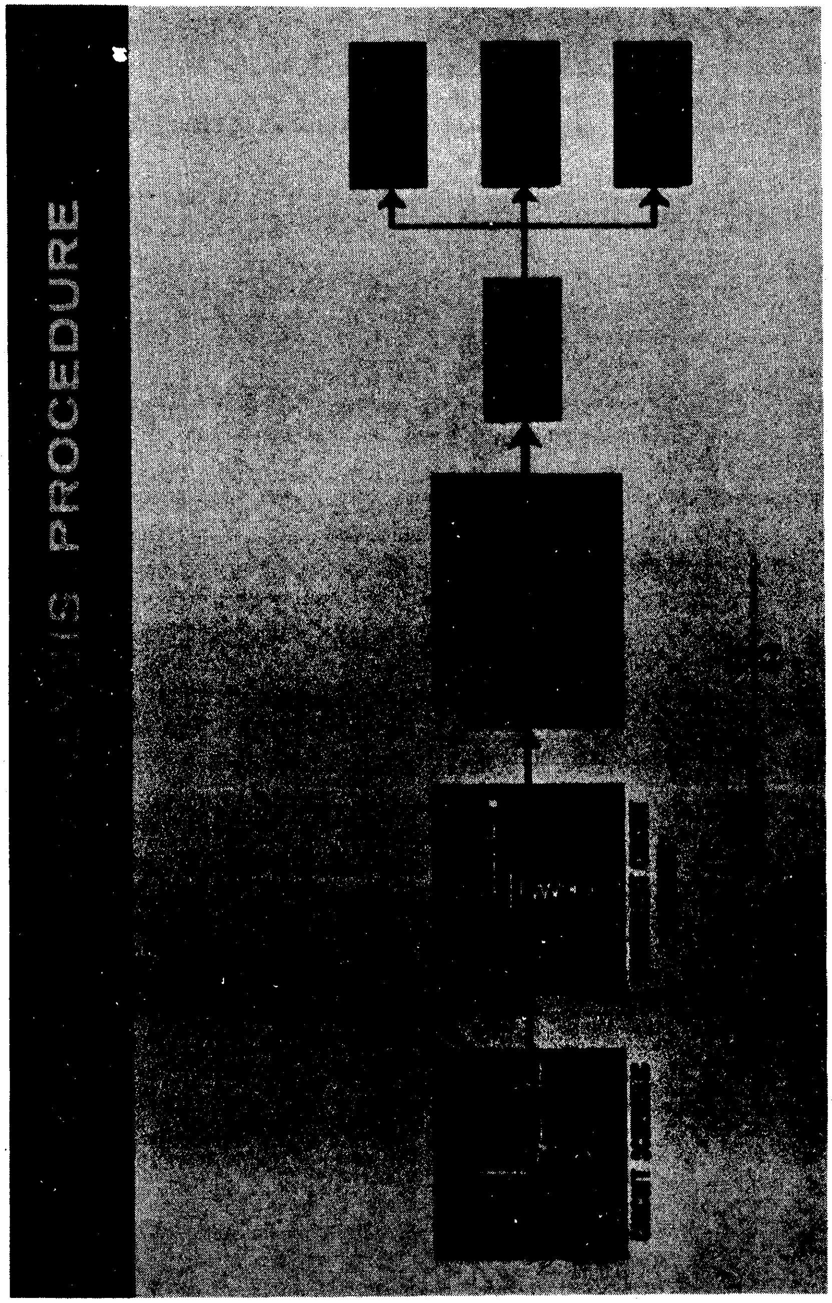
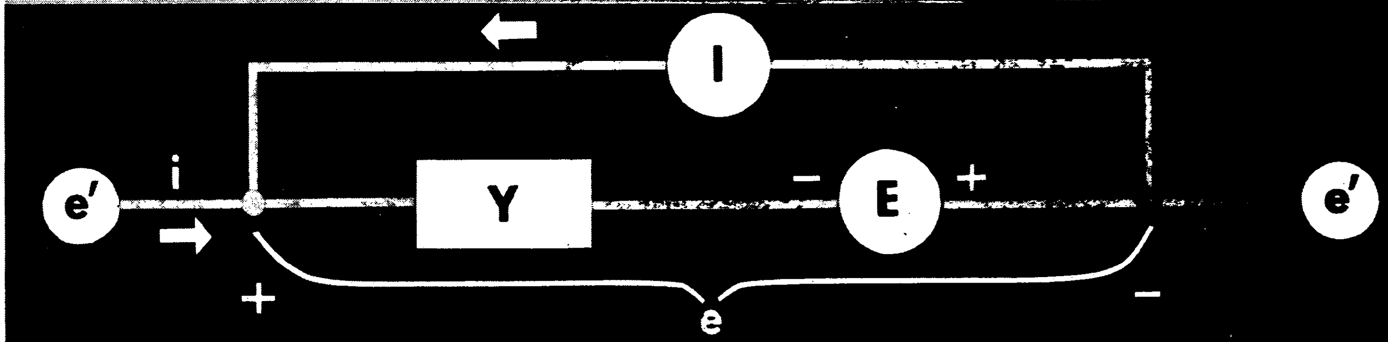


Figure 2

427a
4.27.6

BASIC BRANCH DEFINITION



Y = CONDUCTANCE OF PASSIVE ELEMENT

E = VOLTAGE SOURCE

I = CURRENT SOURCE

i = BRANCH CURRENT

e = BRANCH VOLTAGE

e′ = NODE TO DATUM VOLTAGE

BRANCH DATA CARD:

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 3

Then the variables defined in figure 3 may be thought of as vectors or column matrices with elements corresponding to each branch of the network. The primary output of each of the analysis sections is the node voltage vector, e', and the branch current vector, i.

The arrows on the diagram (figure 3) define the positive convention for current flow and the positive convention for the voltage source with polarity as shown. This convention should be kept in mind until later when preparation of input data cards for a specific example is described.

Also, shown in figure 3 is a typical branch data card prepared in the user-oriented language. It contains all necessary information on a branch. The B1 indicates the data is for branch number 1. The N = (0,1) indicates that the branch is connected to nodes 0 and 1 and that the assumed direction of current is from node 0 to node 1. Then the values for the three branch elements are given.

5. Nodal Equations of Network*

Figure 4 shows the nodal equations which are set up automatically by the program. These same equations would be set up by using the conventional manual technique of summing the currents at each node using Kirchhoff's current law.

The only item which has not been defined is matrix A. It is sufficient for our purposes to say that it defines the topology of the network, that is, which nodes each branch is connected to and the assumed current direction for the branch. It is developed from the nodal data supplied for each branch.

The e' vector of the node voltages is solved for. This is done by inverting the triple product (A^t YA) and multiplying it times the right hand side of the equations, where the superscript t indicates the transpose of A. This gives a nominal solution for the node voltages. New solutions corresponding to a change in value of one or more parameters are obtained by a very fast update technique. This is a method of modifying the inverse of (A^t YA) rather than starting the solution over from the beginning. An example of the use of this technique will be shown later.

6. Preparation of Input For Sample Problem.

A specific example will help to clarify the preceding discussion and should show the relative ease with which a problem can be set up for analysis by use of the user-oriented language.

* Technical Report TR 00.855, March 30, 1962. Machine Analysis of Networks and Its Applications. Franklin H. Branin, Jr. IBM Data Systems Division Poughkeepsie, New York.

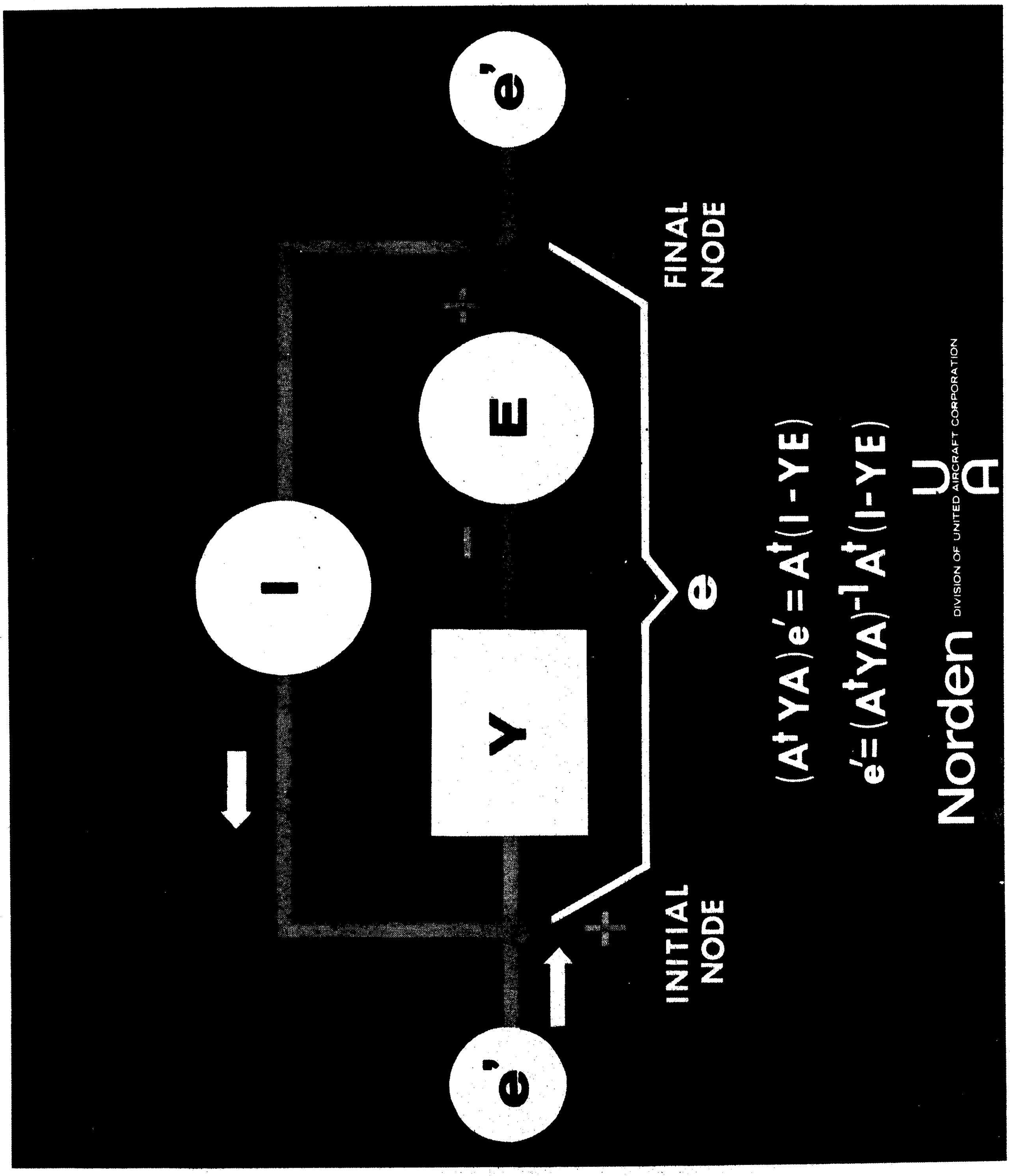

e'
I
E
Y
e'
+
-
+
+
FINAL NODE
INITIAL NODE
e
$(A^t YA)e' = A^t(I - YE)$
$e' = (A^t YA)^{-1} A^t (I - YE)$
Norden
DIVISION OF UNITED AIRCRAFT CORPORATION
UA

Figure 4

Figure 5 is the schematic of a one transistor amplifier. From the circuit schematic the equivalent circuit schematic (figure 6) is prepared. The equivalent circuit of the transistor is enclosed within the dotted lines. Numbers enclosed in squares are branch numbers and the circled labels are node designations. The arrows indicate assumed current directions. The term G(4,3) indicates that there is a transconductance from branch 4 to branch 3. The transconductance is used to model the gain of the transistor and is related to the current gain.

The language interpreter has a dual function of interpreting specific data on the circuit and also interpreting certain control funtions such as which analysis is to be performed, which optional outputs are to be obtained and which input-output devices are to be used. Thus there are two types of input: control statements and data statements.

The input cards required to perform a DC analysis of the sample circuit are shown in figure 6. The first card, DCNODE, is an example of a control statement and indicates that a DC analysis is to be performed. The next card is an example of a data statement. The B1 indicates that the data is for branch number one of the circuit. The N = (0,1) is nodal data indicating that the branch is connected between nodes 0 and 1 and since node 0 is mentioned first, the assumed direction of current flow is from node 0 to node 1. The R = 1. (.01) indicates that the nominal value of resistance is 1 ohm and that the tolerance is 1 percent. The tolerance is specified as a decimal percent tolerance enclosed in parenthesis. The E = -30. indicates that there is a 30 volt voltage source in the branch and it is negative according to the basic branch definition of section 4. Data on the other branches is prepared in a similar manner. Note that the resistance of branch 3 is expressed as R = 5.E + 06 which is equivalent to $5. \times 10^6$ ohms.

Skipping to the card labeled T1 we have another type of data statement for presentation of data on transistor number one. The B = (4,3) means that the transconductance is from branch 4 to branch 3. Next, the value of the transconductance is given. It has a nominal value of .0364. The tolerances are specified in an optional form in which the minimum is .0361 and the maximum is .0367.

Figure 7 shows additional control statements and is a continuation of figure 6. The SENSITIVITY statement indicates that an optional output consisting of the sensitivities of each node in the circuit with respect to each parameter is to be computed. These outputs are discussed further in section 7. The POWER control statement asks for the power dissipation in each branch to be computed. The STATISTICAL ANALYSIS statement calls on a worst-case analysis and computation of the standard deviation of each node voltage. The MODIFY control statement and the statement immediately following it indicate that the resistance in branch 7 is to be modified from 35K to 40K in 5 equal steps.

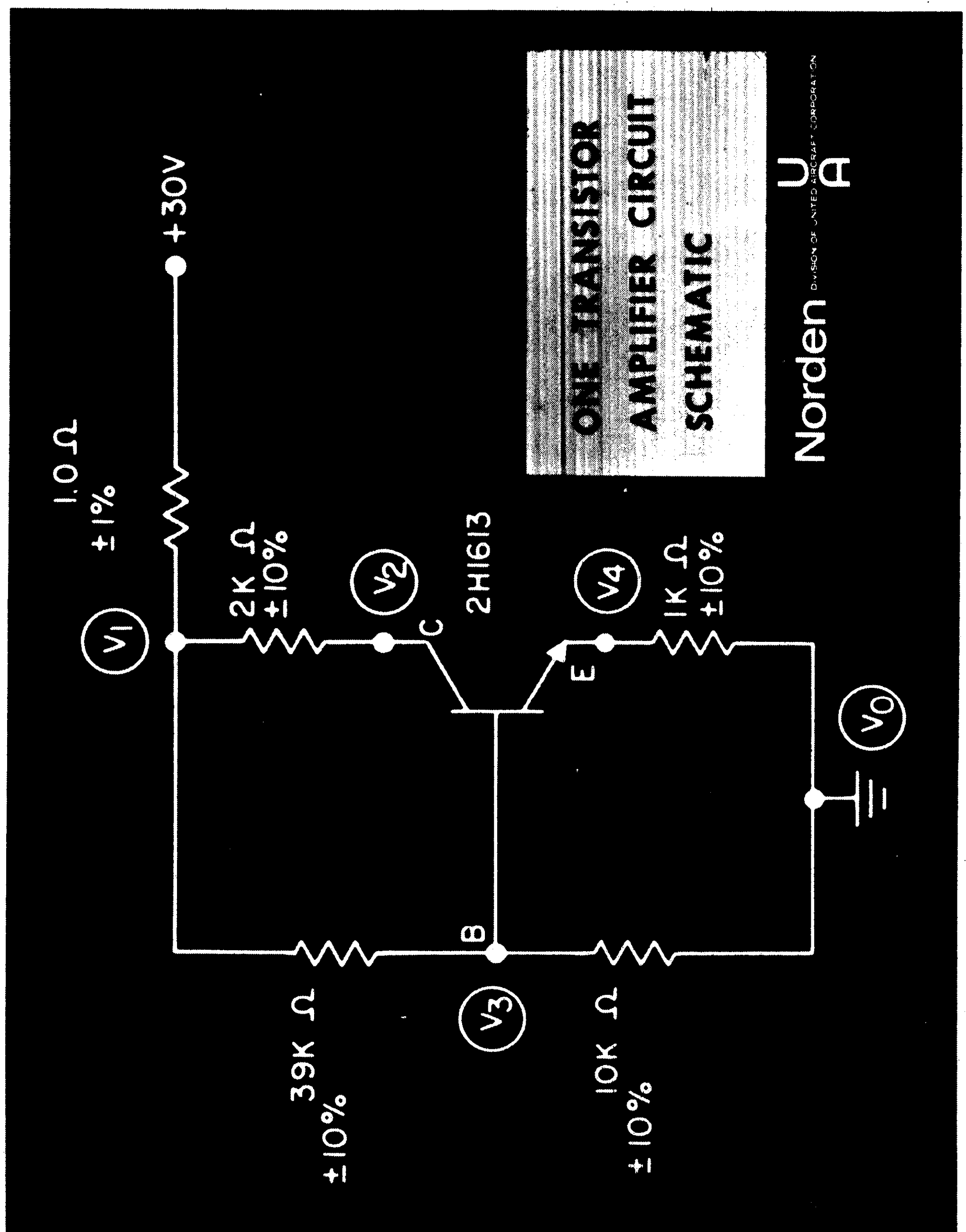
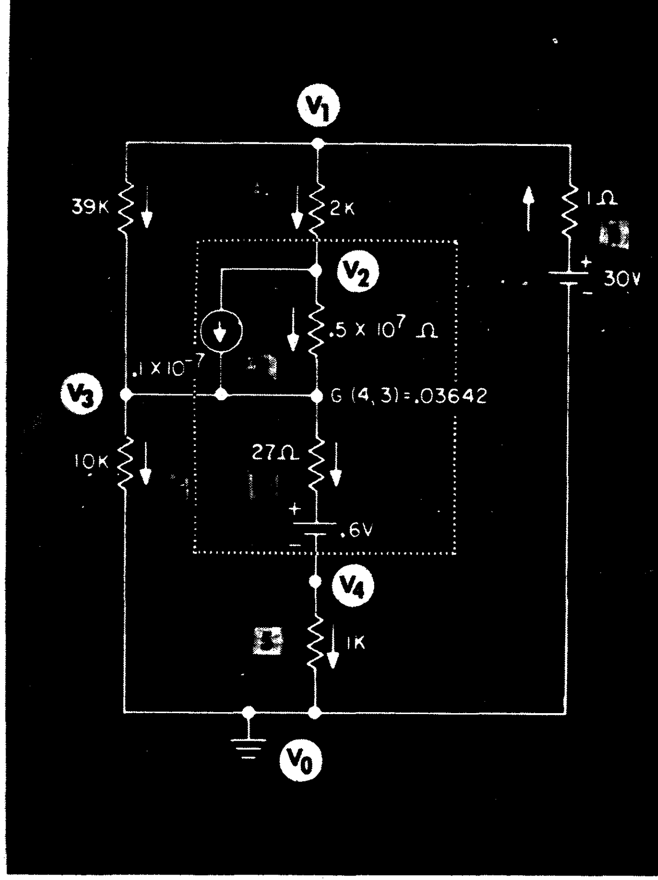


Figure 5

4.27.11

INPUT DATA PREPARATION



DCNODE

B1	N = (0,1)	R = 1.(.01)	E = 30.
B2	N = (1,2)	R = 2000.(.10)	
B3	N = (2,3)	R = 5.E + 06	I = −.01E − 06
B4	N = (3,4)	R = 27.	E = −.6
B5	N = (4,0)	R = 1000.(.10)	
B6	N = (3,0)	R = 10000.(10)	
B7	N = (1,3)	R = 39.E3(.10)	
T1	B = (4,3)	G = .0364(.0361	.0367)

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 6

4.27.12

433

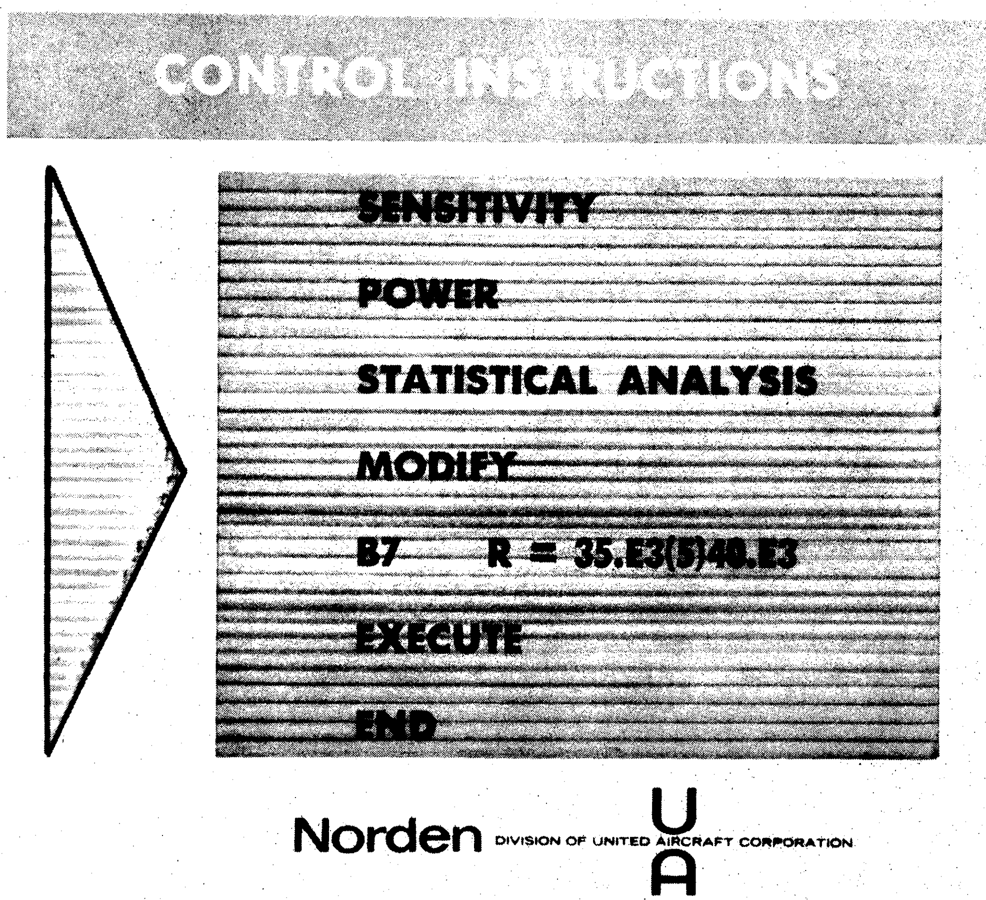

CONTROL INSTRUCTIONS
SENSITIVITY
POWER
STATISTICAL ANALYSIS
MODIFY
B7 R = 35.E3(5)40.E3
EXECUTE
END
Norden DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 7

It is implied that the node voltages at each step are to be computed. These new solutions are obtained at great speed by making use of the update technique discussed in Section 5. The EXECUTE statement causes all previous control statements to be executed in the order in which they appeared. The END statement returns control to the monitor system in the 1620.

From the above discussion it should be evident that preparation of input is relatively simple using the user-oriented language. In fact the rules for preparing the input cards from the equivalent circuit schematic can easily be taught to an aid or keypunch operator.

7. Discussion Of Output Of Sample Problem.

Figure 8 shows the output of the node voltages and the sensitivities for the sample problem. There are four nodes in the problem, not counting the ground or datum node. The values of each node voltage are outputed from left to right.

The sensitivities are the percent change in each node voltage with respect to a positive one percent change in each parameter of the network. The first set of sensitivities are with respect to the resistance in branch 1. They read from left to right for nodes 1 through 4 respectively. The sensitivities with respect to the other parameters such as transconductances, voltage and current sources are outputed in a similar form. The sensitivities are used to detect critical parameters in the network. For example, scanning the first column of sensitivities one can see that node 1 is most sensitive to the voltage source, E1. The sensitivities are also used in the worst-case and standard deviation computation.

The method used in obtaining the partial derivatives, from which the sensitivities are computed, is a relatively efficient method. They are obtained from explicit matrix formulas obtained by taking the partial derivatives of the nodal equations described in Section 5. This is a much faster technique than some others which require obtaining new solutions for the node voltages with the parameters increased by small amounts.

Figure 9 shows the results of a stress analysis which was called for by the POWER control statement. Here the current, voltage and dissipated power for each branch are outputed. Also, shown are the results of the worst-case and standard deviation(1-sigma) computations for each node voltage. Thus, the nominal, worst-case maximum and minimum and standard deviation for each node voltage is listed. These are based on the tolerances of the parameters and the computed sensitivities. From this output is is evident that the designer can readily see the effects of component tolerances on the circuit and can optimize the circuit by selecting the proper tolerances.

4.27.14

```
NO. BRANCHES =   7
NO. NODES =  4


 NOMINAL NODE TO DATUM VOLTAGES
     2.999E+01     2.062E+01     5.493E-00     4.764E-00


 SENSITIVITIES
 PERCENT CHANGE IN NODE VOLTAGES FOR A ONE PERCENT CHANGE IN PARAMETERS

 R 1
     1.770E-04    -1.683E-04    -1.747E-04    -1.961E-04
 R 2
     4.690E-07    -4.530E-01    -2.385E-03    -2.677E-03
 R 3
     1.172E-03     3.493E-00    -5.962E-00    -6.693E-00
 R 4
     3.623E-06     1.050E-02     2.809E-03    -2.313E-02
 R 5
     1.342E-04     3.891E-01     1.040E-01     1.431E-01
 R 6
     1.192E-04    -3.564E-01     6.993E-01     7.850E-01
 R 7
     1.572E-04     4.076E-01    -7.997E-01    -8.978E-01
 G   4, 3
     1.171E-03    -3.491E-00     5.958E-00     6.689E-00
 E 1
     9.999E-01     9.509E-01     9.868E-01     1.107E-00
 E 4
     1.690E-05     4.900E-02     1.310E-02    -1.079E-01
 I 3
     2.502E-09    -7.456E-06     1.272E-05     1.428E-05
```

Figure 8

STRESS ANALYSIS

BRANCH	CURRENT	VOLTAGE	POWER
1	.5312E-02	-.2999E+02	.1593E-00
2	.4685E-02	.9371E+01	.4391E-01
3	.4685E-02	.1513E+02	.7089E-01
4	.4764E-02	.7286E-00	.3471E-02
5	.4764E-02	.4764E+01	.2270E-01
6	.5493E-03	.5493E+01	.3017E-02
7	.6282E-03	.2450E+02	.1539E-01

WORST CASE ANALYSIS AND STANDARD DEVIATION

NODE	NOMINAL	MAX	MIN	SIGMA
1	.2999E+02	.3299E+02	.2699E+02	.9998E-00
2	.2062E+02	.2649E+02	.1475E+02	.8795E-00
3	.5493E+01	.7187E+01	.3799E+01	.2809E-00
4	.4764E+01	.6426E+01	.3102E+01	.2739E-00

Figure 9

NODE VOLTAGES FOR MODIFIED INPUT DATA

R 7 = 3.500E+04
NODE VOLTAGES

2.999E+01	1.968E+01	5.985E-00	5.234E-00

R 7 = 3.600E+04
NODE VOLTAGES

2.999E+01	1.993E+01	5.853E-00	5.105E-00

R 7 = 3.700E+04
NODE VOLTAGES

2.999E+01	2.017E+01	5.728E-00	4.982E-00

R 7 = 3.800E+04
NODE VOLTAGES

2.999E+01	2.040E+01	5.608E-00	4.864E-00

R 7 = 3.900E+04
NODE VOLTAGES

2.999E+01	2.062E+01	5.493E-00	4.752E-00

R 7 = 4. E+04
NODE VOLTAGES

2.999E+01	2.083E+01	5.382E-00	4.644E-00

Figure 10

Figure 10 shows the output obtained as a result of the MODIFY control statement. Sets of node voltages for each value of R7 are obtained. The five solutions are obtained in about the time it takes to get one nominal solution, since the high speed update technique is used to modify the original nominal solution.

8. Discussion of AC and Transient Analyses

The AC and transient analyses will not be discussed in as much detail as the sample DC problem. The outputs obtained from these analyses are shown graphically in figure 11. In the AC analysis the primary output is the frequency response of a network. This consists of the magnitude and phase of each of the node voltages as a function of frequency. The input is prepared in a similar manner to the DC analysis, except that additional input such as values of capacitors, inductors and mutual inductances must be given. Also, the range of frequencies over which the output is to be obtained is specified in a manner similar to the method used in the sample problem for modifying a resistance.

The primary output of the transient analysis is the time response of the node voltages of the network. The transient analysis is capable of handling time varying sources which may be inputs to a circuit. Thus, input waveforems such as square waves, sawtooths, sine waves, etc., can be handled. A certain amount of non-linear capability exists in that switching of parameters from one value to another is possible. Therefore, change of state of transistors and diodes can be simulated. The switching is accomplished automatically and the times at which switching occurs can be made dependedt on computed quantities such as a certain node voltage. For example, switching of certain parameter values from one value to another can be made to occur when a specified node reaches, say, 10 volts.

9. Program Extentions

Norden has extended the Network Analysis Program beyond the vv version to be released by IBM. Some of these extensions are shown in figure 12.

The Failure Effect Analysis searches for failures of components which may be over-stressed when certain other components fail. For example, each resistor in a network is alternately shorted and opened and the corresponding stress levels in the other components are checked to see that they do not exceed recommended values beyond which a failure may occur.

The Derating Analysis compares the stress levels computed in the Stress Analysis (described in the sample problem) with recommended rated values which are stored for each standard part on the 1311 disk file. Percent operating to recommended and percent operating to rated stress are then computed. Also, the failure rate of each component and for the entire circuit is computed.

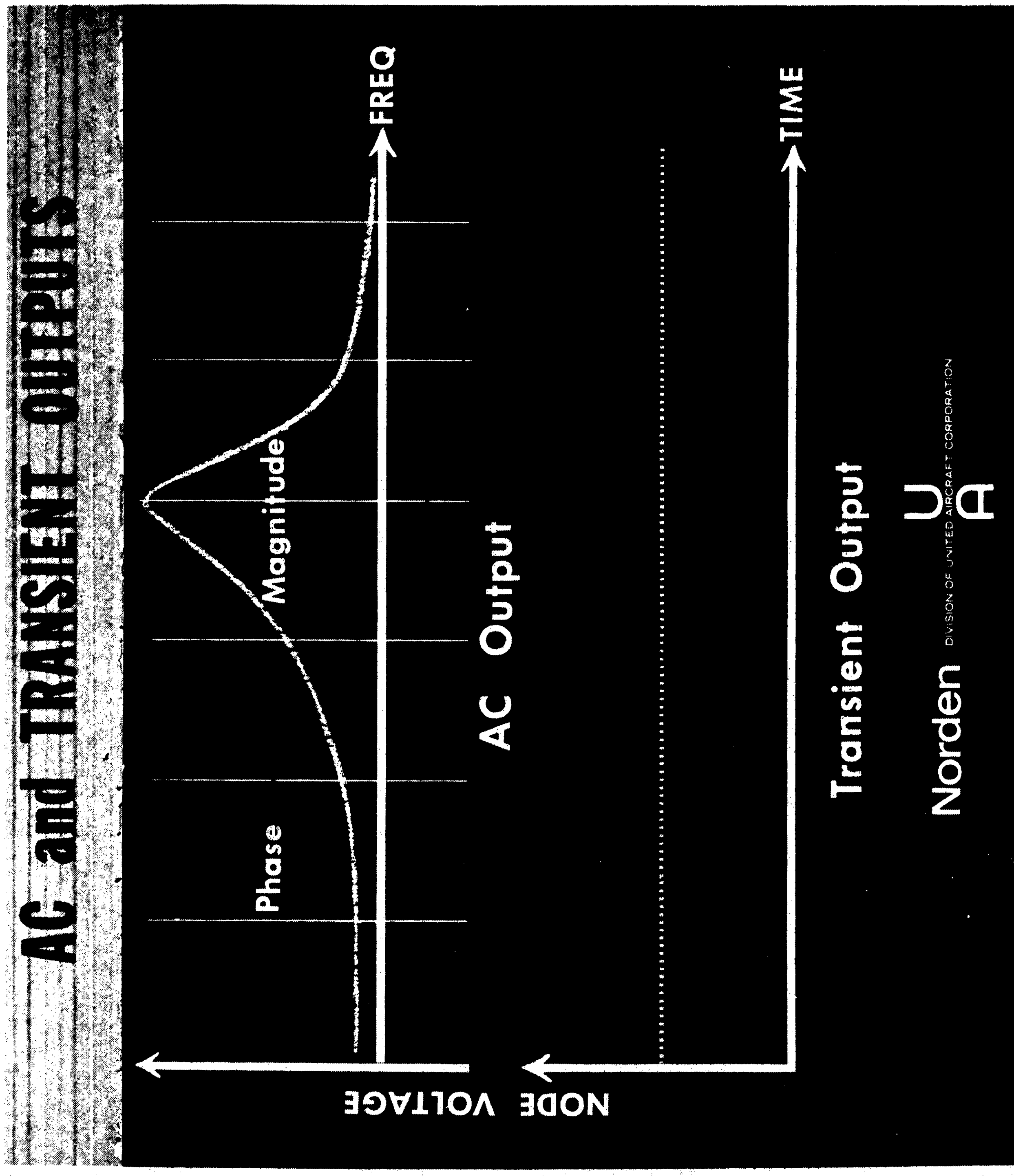

AC and TRANSIENT OUTPUTS
NODE VOLTAGE
FREQ
Magnitude
Phase
AC Output
TIME
Transient Output
Norden
DIVISION OF UNITED AIRCRAFT CORPORATION

Figure 11

PROGRAM EXTENSIONS

[redacted]

DERATING ANALYSIS

DEGRADATION ANALYSIS

EQUIVALENT CIRCUIT STORAGE

[redacted]

Norden DIVISION OF UNITED AIRCRAFT CORPORATION

441

Figure 12

The Degradation Analysis accounts for the effects of aging or some other criterion on a circuit. For example, the nominal tolerances on the parameters are adjusted to account for degradation with aging and the worst-case and standard deviations are computed with the adjusted tolerances.

One of the disadvantages of the program as it presently exists is the necessity of preparing an equivalent circuit schematic including the equivalent circuit models of the transistors and diodes. Norden is extending the Program to include storage of the equivalent circuit so that the user need only specify the type transistor or diode he is using. The necessary data would then be pulled automatically from storage. This will greatly facilitate the setting up of a problem for the computer.

One of the most important limitations of the Program is the lack of non-linear capability. This problem makes itself evident in cases where it is necessary to account for the non-linear characteristics of transistors and diodes. At present, the user must select an operating point for a transistor from which the parameter values for the equivalent circuit model are selected. If after analysis, the operating point does not turn out to be the same as that which was expected, the model will not be accurate. This problem also occurs in the transient case when transistors change state and it may be necessary to account for the non-linear transition, say, from the off state through the active to the saturated state. Norden is working on extending the program to account for the above non-linearities.

AN EXPERIMENTAL PERSONALIZED ARRAY TRANSLATOR SYSTEM

by

H. Hellerman

Watson Research Center

IBM Corporation

443

4.28.0

AN EXPERIMENTAL PERSONALIZED ARRAY TRANSLATOR SYSTEM

by

H. Hellerman

IBM Watson Research Center
Yorktown Heights, New York

ABSTRACT: A system designed for intimate man-machine interaction in a general purpose problem-solving environment is experimentally operational. The system utilizes an array-oriented symbolic source language containing powerful statement types. These include Numeric, Boolean, Relational, and Selection operators on operands which can be entire arrays. The system also permits simple specification of test and argument arrays in single statements.

The completely symbolic operating system includes display and entry of program and data. Sequence control is aided by an interrupt switch which allows the user to interact with the program during execution. In addition to normal stored program sequencing, the system provides trace options and the ability to enter any statement for immediate execution.

Present implementation of the system is with an interpretive translator on an IBM 1620 computer.

Research Paper
RC-1091
December 19, 1963

444

4.28.1

DEBUGGING IN THE FOR II SYSTEM

by

R. D. Burgess

Mechanical Technology, Inc.

Debugging in the FOR II System

This paper briefly describes the 1620 FOR II programming system. Machine language features pertinent to FOR II, difference between FORTRAN with FORMAT and FOR II, indexing in FOR II, subroutine linkage and construction of the object deck are covered. Stress is placed on debugging techniques such as use of the Symbol Table, Instruction Register No. 2, and general debugging aids. Knowledge of FORTRAN with FORMAT is assumed.

SECTION 5

APPENDICES

Table of Contents

EASTERN REGION 1620 USERS GROUP

WASHINGTON, D. C., MAY 6 - 8, 1964

REGISTRATION

ALMAN, John E.
Boston University
700 Commonwealth Ave.
Boston, Mass.

ANDERSON, Jay Martin
Bryn Mawr & Haverford College
Bryn Mawr, Pa.

ARENA, Angelo
IBM
White Plains, N. Y.

ARNOLD, Stewart
J.M. Patterson State
P.O. Box 6199
Montgomery, Ala

AUSTIN, Milton T.
Mauchly Associates, Inc.
P.O. Box 475
Ft. Washington, Pa

BABER, Gaye M.
National Ed. Assoc.
1201 16th St., N. W.
Washington, D. C.

BAILEY, C.
IBM Dept. 808
P. B. #6
Endicott, N. Y.

BAKER, A. Doyle
Kentucky Utilities
120 South Limestone
Lexington, Ky.

BANTA, E. S.
Wesley College
Dover, Dela.

BERGER, Kenneth R.
National Ed. Assoc.
1201 16th St., N. W.
Washington, D. C.

BECK, W. H.
Baltimore Gas & Electric
Baltimore 3, Maryland

BEST, A. H.
The Glidden Co.
P. O. Box 389
Jacksonville, Fla.

BICKFORD, Paul A.
O.U. Medical Research
800 N.E. 13th Street
Oklahoma City, Okla.

BLAICHER, Herb
Jersey Central - N.J. P & C
Madison Ave.
Morris Plains, N. J.

BLADES, C. L.
Baltimore Gas and Electric
Baltimore 3, Md.

BOCKHOFF, J. W.
Gen. Am. Trans. Co.
2501 N. Natchez Avenue
Chicago, Ill.

BOLLACASA, Dario
General Electric
P. O. Box 1072
Schenectady, N. Y.

BONNETTE, Mrs. Robert W.
Norwich University
Northfield, Vt.

BRADY, Kenneth W.
Long Island Lgt. Co.
250 Old Country Road
Mineola, Long Island, N. Y.

BRAUN, Carol A.
Harrison Radiator Div. GMC
1400 N. Goodman St.
Rochester, N. Y.

BRIDGEMAN, Keene
General Dynamic/Electronics
Rochester, N. Y.

BRIESEMEISTER, R. E.
Consolidated Edison
4 Irving Place
New York, New York

BRIGHT, Harold F.
George Washington Univ.
Washington, D. C.

BRITTAIN, MA1 USN, Paul
U. S. Naval Academy
Annapolis, Maryland

BROWN, Charles C.
Albemarle Paper Mfg. Co.
Roanoke Rapids, N. C.

BRZOZOWSKI, W.
Comp-Tron Co.
1035 N. Calvert St.
Baltimore, Md.

BURGESS, Robert D.
Mechanical Tech. Inc.
968 Albany Shaker Road
Latham, New York

BURLEIGH, W. F.
Tidewater Oil Co.
Delaware City, Dela.

BURROWS, W. A.
Dravo Corp.
Pittsburgh, Pa.

BUSCH-PETERSEN, Bent
Houdry Process & Chem. Co.
1339 Chestnut St.
Philadelphia, Pa.

CALLAHAN, E. R.
E. I. DuPont
Textile Fibers Dept.
Chattanooga, Tenn.

CANNER, Arnold K.
Transitron Electronic Corp.
168 Albion Street
Wakefield, Mass.

CARLSON, C. E.
New York State Dept. of
Public Works
Albany 5, New York

CARR, Wendell
US Pub. Health Service
1901 Chapman Ave.
Rockville, Maryland

CASLIN, James C.
US Air Force
318 Whitmore Ave.
Dayton, Ohio

CHANCEY, Harold R.
Hercules Powder Company
Kenvil, New Jersey

CHAPUT, Robert L.
Comm. of Mass
100 Nashua St.
Boston, Mass.

CHURCH, John O.
General Foods Corp.
555 S. Broadway
Tarrytown, N. Y.

CIPOLLA, Jo Ann
E. I. DuPont de Nemours & Co.
Wilmington, Dela.

CLARK, Stanley A.
Public Service Co. of N. H.
1087 Elm Street
Manchester, New Hampshire

COOK, Leroy L.
U.S. Public Health Service
1901 Chapman Ave.
Rockville, Maryland

COOPER, Richard A.
Riley Stoker Corp.
9 Neponset St.
Worcester, Mass.

COX, Ed
Memphis Light Gas & Water
P. O. Box 388
Memphis, Tenn.

CROMWELL, John E.
Sun Pipe Line Co.
1608 Walnut Street
Philadelphia, Pa.

DABE, Rodney
Consoer Townsend
360 E. Grand Ave.
Chicago, Ill.

DAVIDSON, C. H.
Univ. of Wisconsin
Madison, Wisconsin

DAVIDSON, E. A.
IBM
425 Park Ave.
New York, New York

DAVIDSON, James N.
Long Island Lighting Co.
Hicksville, New York

DAVIES, R. W.
Baltimore Gas & Electric
Gas and Electric Building
Baltimore 3, Maryland

DAVIS, Russell H.
State Roads Comm. of Md.
300 W. Preston Street
Baltimore, Maryland

DELHOMMER, Harold
Petroleum Industrial Computer Center
132 Demanade Blvd.
Lafayette, Louisiana

DICKINSON, Frank N.
Biometrical Services - USDA
Beltsville, Maryland

DOUGHERTY, Edward
Broome Technical Community College
Upper Front Street
Binghamton, N. Y.

DRISCOLL, Thomas L.
US Army Trans. Research Comm.
Fort Eustis
Newport News, Va.

EISEMANN, Dr. Kurt
Catholic University
Michigan Ave. & 4th St., N. E.
Washington, D. C.

EISENBERG, Jerry
IBM
1120 Conn. Ave., N. W.
Washington, D. C.

ELLIS, Robert W.
Houdry Process & Chem. Co.
1528 Walnut Street
Philadelphia, Pa.

ELROD, J. C.
Georgia Experiment Station
Experiment, Ga.

EVANS, John H.
Southwestern Public Service
Box 1261
Amarillo, Texas

FALES, Henry N.
General Electric
100 Plastics Ave.
Pittsfield, Mass.

FARINA, Mario
General Electric
600 Main St.
Johnson City, N. Y.

FELL, Henry W.
Princeton-Penna Accelerator
Forrestal Rech. Center
Route #1
Princeton, New Jersey

FINNEGAN, Katherine
Remington Arms Co.
939 Barum Avenue
Bridgeport, Conn.

FISCHER, Michael
Worthington Corp.
Harrison, N. J.

FITCH, Harold L.
Long Island Lighting Co.
250 Old Country Road
Mineola, Long Island, New York

FLAGG, Donald
Budd Company
Hunting Park Ave.
Philadelphia, Pa.

FLEISCH, Sylvia
Boston University
700 Commonwealth
Boston, Mass.

FOLSE, Paul
Tampa Electric Co.
P. O. Box 111
Tampa, Fla.

FOX, J. Baine
Washington & Lee Univ.
Lexington, Va.

GANGI, Vincent J.
The Armstrong Rubber Co.
475 Elm St.
West Haven, Conn.

GARDNER, Donald
General Foods Corp.
555 S. Broadway
Tarrytown, New York

GARZANELL, James W.
Solvay Process-Allied Chem.
P. O. Box 271
Syracuse, New York

GELSI, Hector, P.
Chemical Construction
320 Park Ave.
New York, New York

GERRISH, Everett
Orange and Rockland Utilities
125 High Street
Boston, Mass.

GLIMN, Arno
IBM
3424 Wilshire
Los Angeles, Calif.

GOLDBERG, Martin
United Aircraft Corp.
Helen Street
Norwalk, Conn.

GOLDEN, J. T.
IBM
1730 Cambridge St.
Cambridge, Mass.

GOLDMAN, Norman
Boston University
700 Commonwealth Ave.
Boston, Mass.

GOLDREICH, Estern
Young & Rubicam
285 Madison Ave.
New York, New York

GOTTHEIM, Harold
N.Y.S. Dept. of Publ. Works
State Campus
Albany, New York

GRANT, Jay W.
IBM
San Jose, Calif.

GREEN, Bill
Arkansas Power & Light
6th & Pine
Pine Bluff, Arkansas

GROFT, Garth
York Div. Borg-Warner
Richland Ave.
York, Pa.

GROVE, Dr. Richard E.
Randolph-Macon College
Ashland, Va.

GUION, Richard L.
U.S. Dept. of Agriculture
14th & Ind. Ave., N. W.
Washington, D. C.

HAMILTON, E. Michael
George Washington University
300 North Washington Street
Alexandria, Va.

HAMILTON, G. F.
Atlas Steels Co.
P. O. Box 20
Tracy, Quebec, Canada

HANAN, Sandra
Rockland State Hospital
Orangeburg, New York

HARALAMPA, George S.
New England Electric System
441 Stuart St.
Boston, Mass.

HAPLING, Reginald T.
A.F. Institute of Technology
Wright-Patterson AFB
Ohio

HELLERMAN, Herbert
IBM
Yorktown Hts., New York

HELTZEL, William
Newport News Shipbuilding &
Dry Dock Co.
Newport News, Va.

HENDERSON, Frederick R.
Rochester Institute of Tech.
Rochester, New York

HENDRICKSON, MA3 USN, Lee
U. S. Naval Academy
Annapolis, Maryland

HEWLETT, R. F.
Colorado School of Mines
Holfrn, Volotsfo

HIṢE, William B.
George Washington Univ.
300 N. Washington St.
Alexandria, Va.

HOFFMAN, Edward R.
St. Regis Paper Co.
Nyack, New York

HOFFMAN, L. L
Princeton Univ.
Guggenheim Labs.
Princeton, New Jersey

HOFFMAN, Thomas R.
Union College
Schenectady, New York

HOKE, Tom
Oklahoma Gas & Electric
P. O. Box 1498
Oklahoma City, Okla.

HOULIHAN, Joseph
Riley Stoker Corp.
9 Neponset St.
Worcester, Mass.

HUBBARD, J. C.
Baltimore Gas & Electric
Gas & Electric Building
Baltimore 3, Md.

HURT, Harry
U. S. Navy
Pensacola, Fla.

HUSSON, S. S.
IBM
Poughkeepsie, New York

IRONS, R. C.
Naval Air Station
Pensacola, Fla.

IVESTER, Robert
Southern Engineering Co.
1000 Crescent Ave., N. E.
Atlanta, Ga.

JACKSON, Arthur F.
A. and T. College
Greensboro, N. Carolina

JARDINE, D. A.
Dupont of Canada
Box 5000
Kingston, Ontario, Canada

JASPEN, Dr. Nathan
National League of Nursing
10 Columbus Circle
New York, New York

JENSEN, Clayton, E.
Virginia Military Institute
Lexington Va.

JOHNSON, James E.
Seton Hall Univ.
South Orange Ave.
South Orange, N.J.

JURNACK, Steve
Speer Carbon Co.
St. Marys, Pa.

KANE, J. L.
Guggenheim Labs.
Princeton University
Forrestal Research Center
Princeton, N. J.

KARR, Len
Burns and Roe, Inc.
160 West Broadway
New York, New York

KELLMAN, Sidney
Naval Air Eng. Center
Phila. Naval Base
Philadelphia 12, Pa.

KELLER, John F.
Loyola University
New Orleans, La.

KENNECOTT, Robert L.
Chas. H. Sells, Inc.
409 Manville Road
Pleasantville, N. Y.

KENT, John D
Worthington Corporation
P. O. Box 69
Buffalo, New York

KINDRED, Alton R.
Manatee Junior College
Bradenton, Fla.

KLEIN, Donald
Villanova University
Villanova, Pa.

KNEBEL, Martin
Burns and Roe, Inc.
160 West Broadway
New York, New York

KNIGHT, Mr. & Mrs. D. M.
St. Regis Paper Co.
Rt. 59-A
W. Nyack, New York

LAPOSATA, Samuel M.
Ordance Research Lab.
P. O. Box 30
State College, Pa.

LARSON, Greta
Vapor Corporation
6420 W. Howard
Chicago, Ill.

LASKA, Gene
Rockland State Hospital
Orangeburg, N. Y.

LATTERNER, Thomas S.
Hercules Powder Co., Inc.
Rocky Hill, New Jersey

LEE, Ralph E.
Missouri School of Mines
Rolla, Mo.

LEUR, John F.
Niagara Mohawk
1100 Electric Bldg.
Buffalo, N. Y.

LIEBMAN, Judith
General Electric Comp.
Ithaca, New York

LIPPO, Veikko A.
West Penn Power Co.
Cabin Hill Drive
Greensburg, Pa.

LIPSON, A. L.
Virginia Electric & Power
7th & Franklin Sts.
Richmond, Va.

LITTLE, Joyce M.
Goucher College
Dulaney Valley Road
Towson, Maryland

LLOYD, Diana A.
FMC Corporation
P. O. Box 8
Princeton, New Jersey

LOPEZ, Steve
IBM
112 E. Post Road
White Plains, N. Y.

LOUIS, Jene
Long Island Lighting Co.
175 Old Country Road
Hicksville, New York

LOWRIMORE, G. R.
Hercules Powder Co.
Radford Army Ammunition Plant
Radford, Va.

LUSSOW, Patricia
General Electric Comp.
Ithaca, New York

MAC LEOD, D. H.
Stetson University
DeLand, Fla.

MARTIN, C.
IBM
425 Park Avenue
New York, New York

MARTIN, Samuel F.
US Aeronautical Charts
2nd Arsonal St.
St. Louis, Mo.

MC GEHEE, Lt. T. L.
Naval Air Station
Pensacola, Fla.

MC LAUGHLIN, James J.
Army Materials Res. Agency
Watertown, Mass.

MEADS, M. E.
Baltimore Gas & Electric
Electric Building
Baltimore, Maryland

MERIWEATHER, P. Wayne
NASA - MSFC
Huntsville, Ala.

MILLER, Adelaide
IBM
1120 Conn. Avenue, N. W.
Washington, D. C.

MINKIN, Jean A.
Institute for Cancer Research
7701 Burholme Avenue
Philadelphia, Pa.

MOORHEAD, Patricia
John Hancock Mutual Life Ins.
200 Berkeley Street
Boston, Mass.

MONAHAN, Jeanette
Rohn & Haas
5000 Richmond St.
Philadelphia, Pa.

MOORE, E. T.
A. and T. College
Greensboro, N. C.

MORRISSON, Thomas E.
Speer Carbon Co.
47th and Packard Road
Niagara Falls, New York

MUNROE, M. E.
University of New Hampshire
Durham, New Hampshire

NEWMAN, Edward
MIT
77 Massachusetts Ave.
Cambridge, Mass.

NICHOLS, R. E.
Virginia Electric & Power Co.
7th and Franklin Sts.
Richmond, Va.

NORTHROP, David T.
Fairchild Dumont Labs.
750 Bloomfield Ave.
Clifton, N. J.

NUNBERG, Bennett
Leesona Moos Labs.
Lake Success Park
Great Neck, New York

OGILVIE, Norman
Airbourne Instruments Lab.
Melville, New York

OLIVER, James R.
Univ. of Southwestern La.
Box 1333 USL Sta.
Lafayette, La.

O'Neill, Brother B. E.
Manhattan College
Manhattan College Pkwy.
New York, New York

ORTH, Jr., E. J.
Southern Services, Inc.
P. O. Box 2641
Birmingham, Ala.

ORTH, Richard
Worthington Corporation
Harrison, New Jersey

OWENS, John D.
New York Univeristy
University Heights
New York 53, New York

PACINO, Salvatore
General Nuclear Eng. Corp.
P. O. Box 10
Dunedin, Fla.

PALMIERI, Anthony J.
Grumman Aircraft Eng. Corp.
South Oyster Bay Road
Bethpage, Long Island, N. Y.

PARSONS, Jr. G. G.
Lublin, McGaughy & Assoc.
220 W. Freemason St.
Norfolk, Va.

PAQUIN, Nancy
U. S. Public Health Service
1901 Chapman Ave.
Rockville, Md.

PARKER, E. H.
Baltimore Gas & Electric
Gas and Electric Building
Baltimore 3, Maryland

PATERNITI, Richard
IBM
112 E. Post Road
White Plains, N. Y.

PAYNE, Gregson
F.T. Nicholls State College
Box 121
Thibodaux, La.

PEARCE, D. J.
E. I. DuPont
Chattanooga, Tenn.

PETERS, Carl B.
Union County Tech. Inst.
423 Park Ave.
Scotch Plains, N. J.

PETERSON, Andrew I.
Univ. of Bridgeport
Park Ave.
Bridgeport, Conn.

PETERSON, H. C.
Air Reduction, Inc.
Murray Hill, N. J.

PIXLEY, David C.
Bausch & Lomb, Inc.
St. Paul St.
Rochester, N. Y.

PLEINES, W. W.
Baltimore Gas & Electric
Gas and Electric Building
Baltimore 3, Maryland

POINTER, John P.
Tenn. Polytechnic Inst.
Box 37
Cookerville, Tenn.

POWELL, E. R.
Pan American World Airways
Patrick AFB, Fla.

PRATT, Richard L
Air Force Inst. of Tech.
Wright Patterson AFB, Ohio

RANE, Dr., D. S.
Pennsylvania Military College
14th and Chestnut Sts.
Chester, Pa.

RAVER, Richard E.
W. R. Grace & Co.
Clarksville, Maryland

REMILEN, Charles H.
Eastman Kodak
Lake Ave.
Rochester, New York

ROSENBARKER, I. E.
IBM
P. B. #6
Endicott, N. Y.

RICHARDSON, F. W.
Jones & Lamson Mach. Co.
60 Clinton St.
Springfield, Vt.

RICKER, Guy W.
Jersey City State College
Boulevard
Jersey City 5, New Jersey

RIGGS, Heath K.
Univ. of Vermont
Burlington, Vermont

RILEY, R. B.
A.D. Margison & Assoc., Ltd.
1155 Leslie St.
Don Mills, Ont. Canada

RODGERS, J. A.
Baltimore Gas & Electric
Gas and Electric Building
Baltimore 3, Maryland

RUSSELL, James H.
Gilbert Associates, Inc.
525 Lancaster Ave.
Reading, Pa.

RUVOLIS, Leo
York Junior College
Country Club Road
York, Pa.

SABINE, Theodore R.
Operations Research, Inc.
1400 Spring St.
Silver Spring, Md.

SALEK, Franklin
Hamden Testing Services, Inc.
87 Valley Road
Montclair, New Jersey

SAMSON, Stephen L.
US Atomic Energy Comm.
376 Hudson St.
New York, New York

SAMSTAG, Jay
U. S. Public Health Service
1901 Chapman Ave.
Rockville, Md.

SANDIN, E. M.
Baltimore Gas & Electric
Gas and Electric Building
Baltimore 3, Md.

SAWYER, Dr., John W.
Wake Forest College
Box 7401
Winston Salem, N. C.

SCALETTI, Henry M.
United Shoe Mach. Corp.
Balch St.
Beverly, Mass.

SCHIFFMAN, Leonard
Berger Associates, Inc.
P. O. Box 1943
Harrisburg, Pa.

SCHLAGER, Dr. Gunther
The Jackson Laboratory
Bar Harbor, Maine

SCHNECKER, Joe
Bethlehem Steel Co.
Bethlehem, Penna.

SCHRODEL, C. S
Sun Oil Co.
1608 Walnut St.
Philadelphia, Pa.

SCOTT, Thomas J.
Sun Oil Co.
1608 Walnut St.
Philadelphia 3, Pa.

SEEWALD, Milton
Texaco, Inc.
P. O. Box 509
Beacon, N. Y.

SHEEHAN, Daniel M.
Virginia Polytechnic Inst.
Blackburg, Va.

SHIONO, Ryonosuke
Univ. of Pittsburgh
Pittsburgh 13, Pa.

SHORE, Maynard N.
E. R. Squibb & Sons
Georges Road
New Brunswick, N. J.

SILVERMAN, Joan
IBM
555 Madison Ave.
New York, New York

SINANIAN, Ed
IBM
425 Park Ave.
New York, New York

SMITH, Walter J.
U. S. Naval Academy
Annapolis, Md.

SMITH, Welborn H.
Airbourne Instruments Lab.
Melville, New York

SOUCY, Robert
Transitron Electronic Corp.
168 Albion St.
Wakefield, Mass.

SPARVERO, Luke J.
FMC Corporation
Dunham Road
Meadville, Pa.

SPITALNY, Arnold
United Aircraft Corp.
Helen St.
Norwalk, Conn.

SPRINGER, Berl M.
Southwestern Public Service
3rd & Polk Street
Amarillo, Texas

STACKE, Wanda
A T & T
32 Ave, of Amer.
New York, New York

STAIANO, E. F.
Bucknell Univ.
Lewisburg, Pa.

STANSBURY, James C.
Halcon International, Inc.
2 Park Avenue
New York, New York

STEINHART, R. F.
IBM
425 Park Ave.
New York, New York

STONE, Eleanor
Brandeis University
Waltham, Mass.

STORRER, R. L.
Norwich Pharmacal Co.
Box 191
Norwich 1, N. Y.

SULLIVAN, Peter J.
United Illumination Co.
80 Temple Street
New Haven, Conn.

SVIHLA, Edward B.
Automatic Electric Labs.
P. O. Box 17
Northlake, Ill.

TAYLOR, Carlis
National Naval Med. Center
Bethesda, Maryland

TAYLOR, CDR., Dean
U. S. Naval Academy
Annapolis, Maryland

TAYLOR, James S.
Data Corporation
7500 Old Xenia Pike
Dayton, Ohio

TEEPLES, Thomas C.
George Washington Univ.
300 N. Washington St.
Alexandria, Va.

THOMAS, Edward
Engineering Physics Co.
5515 Randolph Road
Rockville, Md.

THOMAS, Roy W.
Southwestern Public Service
Box 1261
Amarillo, Texas

THOMPSON, E. E.
Arkansas Power & Light
6th and Pine
Pine Bluff, Arkansas

THOMPSON, William B.
Photo & Repro. Div.
General Aniline & Film Co.
Binghamton, N. Y.

TIERS, John
Bethlehem Steel
Bethlehem, Pa.

TRANTUM, J. W.
Gen. Am. Trans. Co.
7501 N. Natchez Ave.
Chicago, Ill.

TURNER, Don N.
Nat's Aero & Space Adm.
600 Independence Ave.
Washington, D. C.

VOYTOVICK, Sharon M.
Syracuse University Res. Corp.
P. O. Box 26
Syracuse, New York

WAGNER, Thomas F.
IBM
425 Park Ave.
New York, New York

WANG, Francesca
Stevens Inst. of Tech.
Hoboken, N. J.

WEAVER, John
University of Dela.
Newark, Dela.

WEINER, Myrna S.
Brandeis University
Waltham, Mass.

WEISS, Charles
USAF Aeronautical Chart
St. Louis, Mo.

WELLS, Frank J.
Long Island Lgt. Co.
175 Old Country Road
Hicksville, New York

WHELLER, R. C.
Airborne Instruments Lab.
Melville, New York

WHITACRE, C. G.
Operations Research Group
Edgewood Arsenal, Md.

WHITE, Robert R.
L.A. Dept. Water & Power
207 S. Broadway
Los Angeles, Calif.

WILLIAMS, Charles S.
Washington & Lee Univ.
Lexington, Va.

WILLIAMS, D. D.
Baltimore Gas and Electric
Gas and Electric Building
Baltimore 3, Md.

WILSON, George
Black & Veatch
1500 Meadowlake Pkwy.
Kansas City, Mo.

WINGERSKY, Boty G.
University of N. H.
Durham, New Hampshire

WINGERSKY, Marilyn S.
University of N. H.
Durham, New Hampshire

WINGERT, Joseph T.
Trenton Junior College
101 W. State St.
Trenton, N. J.

WOLFF, Jack M.
Brooklyn College
Brooklyn 10, New York

WOODRUFF, George H.
N. J. State Highway Dept.
1035 Parkway Ave.
Trenton, N. J.

WRIGHT, Donald L.
George Washington Univ.
37th and O Streets, N. W.
Washington, D. C.

WRIGHT, Lawrence
Sprague Electric
Marshall St.
North Adams, Mass.

WYNN, W. Phil
George Washington Univ.
300 N. Washington St.
Alexandria, Va.

YACKULICS, Charles W.
The Armstrong Rubber Co.
475 Elm Street
W. Haven, Conn.

ZIMMERMAN, Leo F.
E.I. DuPont de Nemours & Co.
Savannah River Lab.
Aiken, S. C.

ADDITIONS

DAVIS, Richard B.
Orange and Rockland Utilities
125 High Street
Boston, Mass.

KRENZER, Theodore J.
Gleason Works
1000 University Ave.
Rochester, N. Y.

QUIGLEY, James J.
Orange & Rockland Utilities
10 North Broadway
Nyack, New York

STUDENTS

CARLSON, Dennis
Wesley College
Dover, Dela.

BEALLY, J. Rodger
Wesley College
Dover, Dela.

CLEMENT, Ronald
Wesley College
Dover, Dela.

LIBERMAN, Nick
Wesley College
Dover, Dela.

MORROW, Robert
Wesley College
Dover, Dela.

WILSON, M. A.
Wesley College
Dover, Dela.

PROGRAMS TEAMS CHAIRMEN

1620 USERS GROUP - EASTERN REGION

Mr. Frank J. Wells, Chairman
Electric Utility Team
1620 Users Group - Eastern Region
Long Island Lighting Company
175 Old Country Road
Hicksville, Long Island, New York

Area 516 - Phone: WE 1-6300

Dr. James R. Oliver, Chairman
Operations Research Team
1620 Users Group - Eastern Region
USL Computer Center
Box 133 - USL Station
Lafayette, Louisiana 70506

Dr. Frank N. Dickinson, Chairman
Statistics and Mathematics Team
1620 Users Group - Eastern Region
Agricultural Research Center
Biometrical Services - U.S.D.A.
Beltsville, Maryland

Mr. J. J. Owen, Chairman
1710 Users Sub-Group
Union Bag-Camp Paper Corporation
P.O. Box 570
Savannah, Georgia

Area 912 - Phone: AD 6-5771

Mr. Thomas Scott, Chairman
Structural and Civil Engineering Team
1620 Users Group - Eastern Region
Sun Oil Company
1608 Walnut Street
Philadelphia 3, Pennsylvania

Phone: KI 6-1600

Mr. William Heltzel, Alternate Chairman
Structural and Civil Engineering Team
1620 Users Group - Eastern Region
Newport News Shipbuilding & Drydock Company
Newport News, Virginia

Mr. Arnold K. Canner, Chairman
General Data Processing Team
1620 Users Group - Eastern Region
Transitron Electronic Corporation
168 Albion Street
Wakefield, Massachusetts

Area 617 - Phone: 245-4500

Programs Teams Chairmen (Cont.)

Mr. N. Goldman, Chairman
Education Institutions Team
1620 Users Group - Eastern Region
Boston University
700 Commonwealth Avenue
Boston, Massachusetts

Phone: CO 7-1200

Mr. T. H. Korelitz, Chairman
Chemical Engineering Team
1620 Users Group - Eastern Region
The Badger Company
363 Third Street
Cambridge, Massachusetts

Mr. C. S. Schroedel, Acting Chairman - 5/64
Process and Chemical Engineering Team
1620 Users Group - Eastern Region
Sun Oil Company
1608 Walnut Street
Philadelphia, Pennsylvania

Phone: KI 6-1600

Mr. K. W. Brady, Chairman
Rate Engineering Team
1620 Users Group - Eastern Region
Long Island Lighting Company
250 Old Country Road
Mineola, New York

Area 516 - Phone: PI 7-1000
Ext. 422

Mr. Arnold Spitalny, Chairman
Electronics Team
1620 Users Group - Eastern Region
United Aircraft Corporation
Norden Division
Norwalk, Connecticut 06852

Area 203 - Phone: 838-4471

Mr. Martin Goldberg, Alternate Chairman and Secretary
Electronics Team
1620 Users Group - Eastern Region
United Aircraft Corporation
Norden Division
Norwalk, Connecticut 06852

Area 203 - Phone: 838-4471

Mr. C. Bailey, Chairman
Plotting Team
1620 Users Group - Eastern Region
IBM Corporation
Endicott, New York

Mr. E. E. Newman, Alternate Chairman
Plotting Team
1620 Users Group - Eastern Region
Department of Civil Engineering - Room 1-179
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

May 28, 1964

Report of Meeting of Programs Teams Chairmen
1620 Users Group - Eastern Region
Arlington, Virginia - May 7, 1964

At a meeting of eight of the eleven Programs Teams Chairmen who ran sessions at this meeting of the 1620 Users Group - Eastern Region, the following items were discussed as indicated:

1. It was suggested that the attendants at these sessions be asked to introduce themselves so that a spirit of comradery could be fostered particularly among those who wanted to have themselves considered as "permanent" members of the Teams. The question of the continuity of the Teams' efforts between meetings was discussed in this context.

2. It was requested that rosters of the attendants at these sessions be compiled with indications of those who wanted themselves to be considered as "permanent" members of the Teams.

3. It was requested that a meeting secretary be appointed for each Programs Team to take notes on discussion topics and papers whose authors did not want them published in toto. It was pointed out that the authors of papers presented at these meetings are not requested to have them published in the Proceedings.

4. A survey was asked for from each Programs Team Chairman of the interests which the "permanent" members of his Team have in the sessions of other Programs Teams. It was pointed out that this information would be useful to the Program Chairman of the meeting in scheduling the Programs Teams sessions. In this context, it was suggested that the Programs Teams, in general, could be organized as far as their interests are concerned along lines suggested by the program groupings in the latest quick index.

5. The structure of the meeting, namely programming and operating systems during the first day and a half followed by applications systems and programs during the remaining day and a half, was reviewed with the Programs Teams Chairmen with the suggestion that only those topics in programming and operating systems of exclusive interest to a particular Programs Team be included in that Team's session agenda.

6. The matter of continuity in Team activities and leadership between meetings was discussed. All of the Programs Teams Chairmen at this meeting agreed to continue their leadership at least until the next meeting of the Users Group.

These items were subsequently reviewed with the three Programs Teams Chairmen not present at this meeting.

D. D. Williams
Program Chairman
1964 Spring Meeting
1620 Users Group - Eastern Region

DDW:sk

SURVEY OF TYPE 1620 PROGRAMMING SYSTEMS

1620 Users Group - Eastern Region Meeting
Washington, D.C.
May 6, 7 and 8, 1964

System	Machine Requirements	Detail Page
IBM Corporation		
Statement Languages		
1. FORTRAN with Format (Card) - FO-004, Ver II	20K	1
2. FORTRAN with Format (Tape) - FO-004, Ver II	20K	2
3. FORTRAN II (Card) - FO-019	40K Indirect Addressing Automatic Divide	3
4. GOTRAN (Card) - PR-011	20K	
5. GOTRAN (Tape) - PR-010	20K	
Symbolic Languages		
1. Symbolic Programming System (Card) - SP-009	20K	4
2. Symbolic Programming System (Tape) - SP-008	20K	5
3. 1620/1710 SPS (Card) - SP-020	20K	6
4. 1620/1710 SPS (Tape) - SP-021	20K	6
5. 1710 SPS II (Card) - SP-007	20K	
6. 1710 SPS II (Tape) - SP-006	20K	
7. SPS III (Card) - SP-027	20K Indirect Addressing	
8. SPS III (Tape) - SP-028	20K Indirect Addressing	
Statement and Symbolic Languages		
1. Monitor I (Card) - PR-025 a. SPS II-D b. FORTRAN II-D	20K 1311 Disk Drive Indirect Addressing Automatic Divide	
2. Monitor I (Tape) - PR-026 a. SPS II-D b. FORTRAN II-D	20K 1311 Disk Drive Indirect Addressing Automatic Divide	

System	Author Users	Machine Requirements
Statement Languages		
1. AFIT Improved FORTRAN (Card) - 01.1.010	Pratt Air Force Institute of Technology	20K
2. Complex FORTRAN (Tape) - 06.0.008	Maskiell McGraw-Edison Company	20K
3. FORTRAN 90 Compiler and Subroutines (Card) - 02.0.014	Borst Bausch & Lomb Optical Company	20K Automatic Divide Indirect Addressing Move Flag
4. FORTRAN 1/2 (Tape) - 01.6.037	Anderson Allen-Bradley Company	20K
5. FORTRAN I for Magnetic Tape System (Card) - 02.0.018	Wenrick ACF Industries, Inc.	20K Automatic Divide Indirect Addressing 2 Tape Drives Move Flag
6. FORTRAN I for Magnetic Tape System with Automatic Floating Point (Card) - 02.0.025	Wenrick ACF Industries, Inc.	60K Automatic Divide Indirect Addressing 2 Tape Drives Move Flag
7. FORGO (Load & Go FORTRAN) (Card) - 02.0.008	McClure University of Wisconsin	40K Indirect Addressing Automatic Divide
8. FARGO (Card) - 02.0.027	Heyworth University of Alberta	60K Indirect Addressing Automatic Divide 1 Magnetic Tape Drive Move Flag
9. FOR-TO-GO (2 Pass FORGO) (Card) - 02.0.009	McClure University of Wisconsin	40K Indirect Addressing Automatic Divide
10. Load & Go FORTRAN (Card) - 01.1.009	Mailloux University of Alberta	40K Indirect Addressing
11. Load & Go FORTRAN (Tape) - 01.1.008	Mailloux University of Alberta	40K
12. NOVATRAN I (Tape) - 02.0.012	Northrop Fairchild Camera & Instr.	20K
13. UTO FORTRAN (Card) - 02.0.024	Lee University of Toronto	20K

	System	Author	Machine Requirements
14.	UTO FORTRAN Processor and Subroutines (Tape) - 02.0.017	Field University of Toronto	20K
15.	Automatic Floating FORTRAN Processing System - 01.6.076	Brittain Sunstrand Aviation	Automatic Divide Indirect Addressing Automatic Floating Point Move Flag
16.	PDQ FORTRAN (Card) - 2.0.031	Maskiell McGraw-Edison Company	20K Automatic Divide
17.	Multiprocessing FORTRAN (Tape) - 1.1.005	Carr Indiana Tech. College	20K
18.	60K Load & Go FORTRAN (Card) - 1.1.019	Mailloux University of Alberta	60K Indirect Addressing
19.	NCE Load & Go FORTRAN (Tape) - 2.0.032	Rumrill & Fowler Newark College of Engineering	20K Automatic Divide Indirect Addressing Model 1
20.	NCE Load & Go FORTRAN (Card) - 2.0.029	Rumrill & Fowler Newark College of Engineering	20K Automatic Divide Indirect Addressing Model 1

System	Author	Machine Requirements
Symbolic Languages		
1. Fast Assembler (Card) - 02.0.028	Heyworth University of Alberta	60K Automatic Divide Indirect Addressing 2 Magnetic Tape Drives Move Flag
2. OSAP Assembly System (Card) - 01.1.012	Numerical Computation Laboratory Ohio State University	20K Indirect Addressing
3. SPS I for Magnetic Tape System (Card) - 02.0.021	Wenrick ACF Industries, Inc.	60K
4. SPS II for Magnetic Tape System (Card) - 01.6.077	Wenrick ACF Industries, Inc.	60K 3 Magnetic Tapes Automatic Floating Point
5. LAMP (Card) - 1.1.001	Matthys IBM Corporation	20K
6. Relocatable Assembly System (Card) - 1.1.020	Richardson Australian AEC	60K Indirect Addressing
7. AFIT SPS (Card) - 1.1.023	Pratt Air Force Institute of Technology	40K Indirect Addressing Automatic Divide Move Flag
8. FAST Assembler (Card) - 2.0.028	Mailloux & Davis University of Alberta	60K 2 Magnetic Tapes Indirect Addressing Automatic Divide Move Flag

FORTRAN with Format (Card)

	Program	Author	Machine Requirements
1.	Card FORTRAN Compressor - 01.3.007	Brush IBM Corporation	Indirect Addressing
2.	Dynamic Trace - 01.4.011	Lesson IBM Corporation	Indirect Addressing Move Flag
3.	FORCOM Subroutine - 01.6.051	Pope Utah State University	Indirect Addressing
4.	Format FORTRAN Object Deck Compressor - 01.2.009	Givner Brooklyn College	
5.	FORTRAN Format Checker - 01.1.017	Font Dept. of Public Works (Puerto Rico)	
6.	Gamma Function Subroutine - Positive Arguments - 07.0.048	Pachon Automatic Electric Labs, Inc.	20K Indirect Addressing Automatic Divide
7.	Plot Subroutine - 06.0.119	Vanschaik IBM Corporation	No Automatic Divide
8.	Plot Subroutine - 01.6.056	Reynolds Sprague Electric Company	
9.	Random Number Subroutine 07.0.021	Fink IBM Corporation	
10.	Subroutine for Plotting with a CAL-COMP Plotter - 01.6.068	Fassino Todd Shipyard Corp.	Automatic Divide Special Instructions
11.	Link Subroutine - 13.0.002	Poore Louisiana PI	20K
12.	Plot Subroutine - 13.0.001	Poore University of Kentucky	20K Indirect Addressing
13.	FORTRAN Compressor - Loader - 1.2.007	Sabath Eastman Kodak	
14.	FORTRAN Memory Dump - 1.3.004	Umland Dow Chemical Company	60K Indirect Addressing
15.	FORTRAN Compressor and 75 Column Dump - 1.3.008	Moore IBM Corporation	Indirect Addressing

	Program	Author	Machine Requirements
16.	FORTRAN Compressor and Multi-Programmer - 1.6.045	Gardner General Foods	20K
17.	FORCOM - 1.6.051	Pope Utah State University	Indirect Addressing
18.	FORSTAD - 1.6.054	Anderson General Electric	
19.	Modified CDS FORCOM Subroutine - 1.6.096	Webster Victoria Electricity Commission	Indirect Addressing
20.	FORTRAN Pre-Compiler - FO-006	IBM Corporation	
21.	Plot Subroutine - MP-14X	IBM Corporation	Automatic Divide CAL-COMP Plotter

FORTRAN with Format (Tape)

	Program	Author	Machine Requirements
1.	Bessel Function Subroutine - 07.0.037	Sundvall General Nuclear Eng. Corp.	20K Automatic Divide
2.	CAL-COMP 565 Plotter Subroutine Pen - 13.0.005	Michel Argonne National Lab.	Indirect Addressing
3.	Dynamic Trace - 01.4.012	Lesson IBM Corporation	Indirect Addressing Move Flag
4.	Random Number Subroutine - 07.0.022	Fink IBM Corporation	
5.	Random Number Subroutine - 07.0.009	Sanders Abbott Laboratories	
6.	FORTRAN Lister - 1.1.007	Pratt Air Force Institute of Technology	
7.	FORTRAN Pre-Compiler - FO-005	IBM Corporation	

FORTRAN II (Card)

Program	Author	Machine Requirements
1. Dump Digit-X Unload Memory Program - Self Load, FORTRAN II and Format Subroutine - 01.6.092	Orth Southern Services, Inc.	60K
2. FORCOM Subroutine - 01.6.074	Webster State Electricity Commission of Victoria (Australia)	
3. FORTRAN II Symbol Table Punch - 01.6.067	McIlvain Catalytic Construction Co.	60K Move Flag
4. FORTRAN II Diagnostician - 01.6.019	Burgeson IBM Corporation	Indirect Addressing
5. Format Statement Preselector - 01.6.086	Morgan Washington State Highway Department	
6. Relocatable Cos, Sin, and Exp. Subroutines - 07.0.038	White Mayo Clinic Computer Facility	Automatic Floating Point
7. Relocatable Plot Subroutine - 13.0.004	White Mayo Clinic Computer Facility	
8. Random Exponential Number Generator Subprogram - 06.0.111	Bray Boeing Scientific Research Labs.	
9. Random Normal Number Generator Subprogram - 06.0.108	Bray Boeing Scientific Research Labs.	Move Flag
10. Subroutine for Plotting with CAL-COMP Plotter - 01.6.065	Fassino Todd Shipyard, Inc.	Move Flag
11. 1620 SORT-Library Function - 01.6.069	Anderson Sandia Corporation	Floating Point Hardware
12. Symbol Table on Cards - 01.3.011	Christensen Sylvania Electric System - West	TNF Instruction
13. Automatic Floating Point Subroutines - LM-024	IBM Corporation	Automatic Floating Point

Program	Author	Machine Requirements
14. Relocatable TNS and TNF Subroutines - 1.6.063	White Mayo Clinic Computer Facility	Move Flag
15. ALTER II	Levin Hazeltine Research Division	

Symbolic Programming System (Card)

	Program	Author	Machine Requirements
1.	Additional Instructions Macro Subroutine - 01.1.002	Petersen Washington State Highway Department	Indirect Addressing
2.	Input-Output Subroutines - 01.6.028	Dykstra General Foods Research Center	20K
3.	Multiple Precision Floating Point Arithmetic Subroutine - 01.6.041	Hoffman Boeing Company	60K Indirect Addressing Automatic Divide "Move Flag" Instruction
4.	Relocator Program - 01.2.005	Glanz Purdue University	20K Indirect Addressing "Move Flag" Instruction
5.	SPS Modifier for Magnetic Tape Operation - 01.6.039	Holline Bell Telephone Labs.	
6.	Multi-Purpose SPS Compressor - 1.1.006	Bate Wolf Research and Development	
7.	Float - 1.6.083	Ginsburg U.S. P.H.S.	Indirect Addressing Automatic Divide Move Flag

Symbolic Programming System (Tape)

Program	Author	Machine Requirements
1. Assembly Program for SPS Subroutines - 01.1.003	Pratt Air Force Institute of Technology	20K
2. Floating Point Input/Output Subroutines - 01.6.023	Pratt Air Force Institute of Technology	20K

1620/1710 SPS (Card)

	Program	Author	Machine Requirements
1.	Floating Point Conversion Subroutines - 1.6.053	Pratt Air Force Institute of Technology	
2.	Label Reference Indexer - 1.1.014	Pratt Air Force Institute of Technology	Indirect Addressing
3.	Label Reference Indexer Without Indirect Addressing - 1.1.013	Pratt Air Force Institute of Technology	
4.	L105 Floating to Fixed - 1.6.048	Engelhardt IBM Corporation	Indirect Addressing
5.	L106 Floating to Fixed - 1.6.049	Engelhardt IBM Corporation	Indirect Addressing
6.	Object Deck Analyzer - 1.6.060	Bezreh John Hancock	Indirect Addressing
7.	Symbol Table Punch - 1.3.010	Bradshaw Christian Brothers College	Indirect Addressing
8.	Storage and Label Analyzer - 1.6.093	Bowman IBM Corporation	Indirect Addressing

1620/1710 SPS (Tape)

	Program	Author	Machine Requirements
1.	Label Reference Indexer - 1.1.022	Pratt Air Force Institute of Technology	