AD NUMBER
ADB205179
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; 29 SEP 1948. Other requests shall be referred to U.S. Army Research Laboratory, Attn: ARL-APG Security Office, RDRL-LOA-I, Aberdeen Proving Ground, MD 21005.**
AUTHORITY
ARL Form 1, signed by Public Affairs Office, dtd 9 Jul 2012

BRL-673
ADB205179
LIMITED

SEARCH LABORATORIES

REPORT NO. 673

A Logical Coding System Applied to the ENIAC

R. F. CLIPPINGER

ABERDEEN PROVING GROUND, MARYLAND

TABLE OF CONTENTS

	Page
ABSTRACT	3
A LOGICAL CODING SYSTEM APPLIED TO THE ENIAC	4
INTRODUCTION	4
SECTION I	
GENERAL DESCRIPTION OF THE ENIAC	4
PROGRAMMING BY MEANS OF PULSES, SWITCHES AND CABLES	7
SYNCHRONIZED SYSTEM	8
PROGRAMMING THE ENIAC	9
THE ACCUMULATOR	9
THE MULTIPLIER	10
THE DIVIDER AND SQUARE-ROOTER	11
THE READER	11
THE PRINTER	11
THE CONSTANT TRANSMITTER	11
THE FUNCTION TABLES	11
THE MASTER PROGRAMMER	12
SECTION II	
CONVERSION OF DIGIT PULSES TO PROGRAMMING PULSES	12
THE PULSE AMPLIFIERS	14
SECTION III	
DETAILED DESCRIPTION OF THE SIXTY ORDERS	15
READER TO CONSTANT TRANSMITTER	16
CONSTANT TRANSMITTER TO ACCUMULATORS 11 AND 15	16
ACCUMULATOR TO ACCUMULATOR	16
THE TRANSFER ORDERS IN THE FIRST SET	17
FUNCTION TABLE TO ACCUMULATOR ORDERS	18
ACCUMULATORS TO PRINTER	18
THE TEN OPERATIONS	18
SECTION IV	
SUMMARY OF ORDERS	19
SECTION V	
A SUGGESTED TEST PROGRAM	24
SECTION VI	
FLOW DIAGRAM OF TYPICAL PROGRAM	27
SECTION VII	
CODING THE OPERATION BOXES	35
SECTION VIII	
MODIFIED ENIAC	35
DISTRIBUTION LIST	40

DUPLICATING BRANCH
THE ORDNANCE SCHOOL
ABERDEEN PROVING GROUND
MARYLAND

BALLISTIC RESEARCH LABORATORIES
REPORT NO. 673

Clippinger
Aberdeen Proving Ground, Md.
30 July 1948

ABSTRACT

During the first two years of operation, problems were coded for the ENIAC by listing complicated sets of switch settings and cable connections. Meanwhile new machines were being designed with a small but sufficient vocabulary of operations to each of which a number was assigned. Problems were to be coded for these machines by listing a logical sequence of these numbers. Application of such a system to the ENIAC, described in this report, has been possible with only minor electronic changes. Among the advantages of the new system are: 1) reduction of time and difficulty in coding problems; 2) increase in complexity of problem which can be solved by the ENIAC; 3) reduction in time required to change from one problem to another; 4) increase in facility of testing and consequent increase in efficiency of the ENIAC. In addition to the description of the new vocabulary a test procedure is suggested and a representative problem coded.

A LOGICAL CODING SYSTEM APPLIED TO THE ENIAC

INTRODUCTION

In the spring of 1947, J. von Neumann suggested to the author that it would be possible to run the ENIAC in a way very different from the way contemplated when it was designed; a way which had very important advantages to be discussed below. Since that time his suggestion has been worked into a finished regime by J. von Neumann, A. Goldstine, B. Bartik, R. Clippinger, and A. Gehring with contributions by A. Galbraith, J. Giese, K. McNulty, J. Holberton, E. Snyder, E. Schlain, K. Jacobi, F. Bilas, and S. Spear. The role of J. von Neumann in working out the details has been a central one.

Sections 1, 2, and 3 make clear the methods by which the sixty orders are effected. However, the reader who wants a more rapid description of the procedure of programming a problem for the ENIAC may skip these sections and read only Sections 4, 5, 6, and 7.

A. It is hoped by the author that this report will make the task of coding problems so clear and straightforward that physicists, aerodynamicists, applied mathematicians, etc. with no prior experience with computing machines can code their own problems and prepare tests. This is of critical importance since the shortage of personnel at Aberdeen will prevent the Ballistic Research Laboratories from being able to code extra-Ballistic Research Laboratory problems in the foreseeable future.

B. Not only will problem coding for the ENIAC be easier but, using the new code in the systematic methods of J. von Neumann and H. Goldstine[1], it will also be much faster.

C. Problems about four times as long may be coded in the new scheme. A rough measure of this factor is the ratio of the number, 1800, of order positions (see Section 3) in the function tables to the number, 450, of program controls (see Section 1). This gain is achieved at the expense of function table space.

D. Since problems are put on the machine in the new system by setting switches on the function table, problems can probably be changed in an hour instead of a day by the old method where many cables had to be plugged in and out. Furthermore, the switch settings made in the function tables can be systematically and rapidly checked. (See Section 5.)

E. Besides easing the switch settings check, the new method will ease the ENIAC functional testing. On one function table, programs may be set up at will to test any suspected unit. This facility may very well result in causing the ENIAC to deliver results at a greater rate despite the reduction in computing speed by a factor of six. At the time of writing, the ENIAC is being tested about half of the time.

SECTION I

GENERAL DESCRIPTION OF THE ENIAC

There exists a five volume report on the ENIAC prepared by A. Goldstine, H. Huskey, and A. Burks for the Moore School of Electrical Engineering. The report cited and section 1 of this report consist of a description of the machine and a method for using it which we shall call local programming. Under this method of programming, the machine is given its instructions for each new problem by the setting of large

[1] H. H. Goldstine and J. von Neumann, "Planning and Coding of Problems for an Electronic Computing Instrument", Institute for Advanced Study, Princeton, N. J. (1947).

numbers of switches throughout the machine and is informed of the sequence of instructions by the rewiring of variable circuits. The remainder of this report is intended to describe a system of central programming. Such programming is achieved by instructing the machine in a basic vocabulary of complex orders compounded from the machine's primitive operations by local programming and fixed for all problems. The sequence in which these orders are carried out is determined for each problem by the settings of switches on a central organ of the machine.

The principle of central programming for large scale computing machines is not a new one[2]; only its application to the ENIAC is new.

The Electronic Numerical Integrator and Computer (ENIAC) is a high-speed electronic computing machine which operates on discrete variables. It is capable of performing the arithmetic operations of addition, subtraction, multiplication, division, and square rooting on numbers (with sign indication) expressed in decimal form. The ENIAC, furthermore, remembers numbers which it reads from punched cards, or which are stored on the switches of its so-called function tables, or which are formed in the process of computation, and makes them available as needed. The ENIAC records its results on punched cards from which tables can be automatically printed. Finally, the ENIAC is automatically sequenced, i.e., once set up to follow a routine consisting of operations in its repertoire, it carries out the routine without further human intervention. When instructed in an appropriate routine consisting of arithmetic operations, looking up numbers stored in function tables, etc., the ENIAC can carry out complex mathematical operations such as interpolation and numerical integration and differentiation.

The fundamental signals used in the ENIAC are emitted by its oscillator at the rate of 100,000 per second. The interval between successive signals, 10 micro-seconds, is designated by the term pulse time. The time unit in which the operation time for various parts of the ENIAC is reckoned is the addition time. An addition time is 20 pulse times or 200 micro-seconds (1/5000th of a second). An addition time is so named because it is the time required to complete an addition. Other operations require an integral number of addition times; for example, 10 digit multiplication requires 14 addition times.

The ENIAC proper consists of 40 panels arranged in U shape, 3 portable function tables, a card reader, and a card punch. The term unit of the ENIAC is used to refer to one or more panels and associated devices (the portable function tables, for example) containing the equipment for carrying out certain specific related operations.

The units of the ENIAC can be classified functionally into 4 categories: arithmetic, memory, input and output, and governing. The arithmetic units include 20 accumulators (for addition and subtraction), 1 high-speed multiplier, and 1 combination divider and square rooter. There are two primary memory aspects in the ENIAC: memory for numbers and memory for programming instructions. The constant transmitter, 3 function tables, and the 20 accumulators provide numerical memory. The constant transmitter with its associated card reader reads from punched cards numbers that are changed in the course of a computation and makes these numbers available to the computer as needed. Numbers that remain con-

[2] H. H. Goldstine and J. von Neumann, loc. cit.

stant throughout a computation are stored on the switches of the constant transmitter or of the portable function tables and emitted when needed. The accumulators not only function arithmetically, but also can be used to store numbers which are computed in one part of a computation and required in other parts. All units have program controls which contribute to the programming memory in the following ways:

(1) by recognizing the reception of a program imput signal which stimulates the unit to perform

(2) by causing the programming circuits to operate (as specified by the setting of program switches when there are options regarding the operation to be performed)

(3) on the completion of the operation, by emitting a program output signal which, by means of program cable connections to program lines, is brought to other units to cause them to operate. The program cable connections and switch settings are established before the computation begins.

The kind of programming described in points 1, 2, and 3 above is described as local programming memory because it is taken care of locally at each unit for that unit. The master programmer provides a certain amount of centralized programming memory by coordinating the local programming of the other units.

The imput devices for the ENIAC consist of the card reader and the constant transmitter mentioned above in connection with numerical memory. The printer and card punch record computed results.

The governing units of the ENIAC are the initiating unit and the cycling unit. The initiating unit has controls for turning the power on and off, starting a computation, initial clearing, and other special functions. The cycling unit converts 100 kc sine waves emitted by its oscillator into a fundamental train of signals repeated every addition time (i.e. repeated 5000 times per second). These signals include various sequences of pulses. The term pulse is used to refer to a voltage change (either positive or negative) from some reference level and the restoration to the reference level which takes place in a short time, between 2 and 5 micro-seconds.

With a few exceptions digits are communicated from one unit of the ENIAC to another in pulse form. Digit trays stacked above the front panels running from accumulator 1 to the second panel of the constant transmitter are used for this transmission. A digit tray has 11 wires and a ground. Each of ten wires carries the pulses for one place of a 10 place decimal number. To represent the digit n (where $0 \leqslant n \leqslant 9$) in a particular decimal place, n pulses are transmitted over the wire associated with that particular decimal place. The 11th wire is used for the transmission of sign information. No pulses are transmitted for sign plus, and 9 pulses for sign minus (see discussion of complements below). Pulses are transmitted over all 11 conductors simultaneously. Units which are to communicate with one another in the course of a computation have their digit input and/or output terminals connected by means of digit cables to the 12 point terminals on a digit trunk. At any given time, only one 10 digit number with its sign may be transferred over a particular digit trunk. More than 1 unit may listen to this number. Through the use of more than one digit trunk, several different numbers may be transferred simultaneously. The units of the ENIAC transmit numerical information by emitting appropriate numbers of the pulses which they receive from the cycling unit. Addition is performed in accumulators by means of 10 decade counters, one counter for each decimal place of a 10 digit number, and a binary counter for sign plus (P) or minus (M). These counters are advanced one step by each pulse received. The decade counters and PM counter of an ac-

cumulator are so interconnected that provision is made for carry over. Subtraction is performed by adding the negative of the subtrahend to the minuend.

In order to avoid the necessity for cycling counters backwards, the negative of a number is represented as a complement with respect to a power of ten. Let us consider the decimal point to be located at the extreme right of an accumulator. Then the complement with respect to 10^{10} of the positive number stored in an accumulator as $P + \sum_{i=0}^{9} a_i \cdot 10^i$ is formed by transmitting 9 pulses for sign M and by transmitting the digit pulses for $10^{10} - \sum_{i=0}^{9} a_i \cdot 10^i$. Similarly, the complement with respect to 10^{10} of the negative number stored as $M + \sum_{i=0}^{9} b_i \cdot 10^i$ is formed by transmitting no pulses for sign P and by transmitting the digit pulses for $10^{10} - \sum_{i=0}^{9} b_i \cdot 10^i$.

Because the counters in an accumulator are so connected that there is carry over not only from each decade counter to the one on its left but also from the 10th decade counter to the binary counter for sign, the usual arithmetic properties obtain when complements are used in addition and subtraction.

PROGRAMMING BY MEANS OF PULSES, SWITCHES AND CABLES.

Before a computation can be performed on the ENIAC by the old method, not only must the digit input and output terminals of the units be connected into digit trunks for the communication of numerical data, but also the units must be set up so as to recognize when they are to operate and which particular operations are to be performed. Program controls and program trays and cables are used to instruct the ENIAC in the programming requirements for a particular computation.

Each unit of the ENIAC has one or more program controls. These controls are either of the repeat or non-repeat type. Non-repeat program controls have an input terminal for a program signal and a receiver. Repeat controls have both an input and an output terminal for a program signal and a transceiver or some logically equivalent device. Each program control on a unit which is capable of more than one operation or which is capable of performing operations in a variety of ways has a set of program switches.

Receivers and transceivers alike have the following properties: 1) they have two stable states which will be referred to as the normal and abnormal states; 2) when a program input signal is received, they are set into the abnormal state; 3) they are so connected (through the program switches, if any) to the programming circuits that in the abnormal state they cause the programming circuits to function appropriately; and 4) when the required routine has been completed, they are reset to the normal state so that activity in the unit ceases. When the set of instructions either set up on the program switches of a repeat control or built into the programming circuits has been completed, the transceiver of a repeat program control causes a central programming pulse to be emitted as a program output pulse from the program control's output terminal.

The program trays, like the digit trays, contain 11 wires and a ground, and have 12 point terminals at each end, so that as many trays as desired can be connected to form a program trunk. Each of the 11 lines running the length of a program trunk is referred to as a program line. The program trays differ from the digit trays only in that at two foot intervals the program trays have a set of 11 two point program terminals (1 wire and a shield) instead of a 12 point digit terminal. Input and output terminals of program controls are connected to the program lines by means of program cables.

The procedure for instructing the ENIAC in its routine, then, consists, in local programming, of setting program switches on the units so that, when stimulated by a program input pulse, the program controls will cause the units to carry out a set of specific operations. The temporal order in which the operations are to follow one another is determined by the manner in which program pulse input and output terminals are connected to program lines. All program controls whose program pulse input terminals are connected into the same program line start to operate simultaneously when the program line carries a program signal. If one of the program controls thus stimulated is a repeat program control and if its program pulse output terminal is connected to a second program line, all program controls whose program input terminals are connected to this second program line start to operate when the routine set-up on the repeat program control has been completed.

SYNCHRONIZED SYSTEM.

All units of the ENIAC operate in synchronism with one another, i.e., all units that start to operate at the same time complete their operations either at the same instant or at times that differ by an integral number of addition times. The phrase "complete an operation" covers not only finishing the numerical processes involved in the operation but also the emission of a program output signal.

The basis of this synchronization is the fundamental train of pulses emitted by the cycling unit and delivered to all units of the ENIAC by means of a set of connected trays called the sychronizing trunk. These trays are physically the same as the digit trays. The central programming pulse (CPP) emitted by the cycling unit in pulse time 17 of every addition time cycle plays a major role in such synchronization, since the program output pulse which a repeat program control emits upon the completion of a program results from allowing a CPP to pass. The units of the ENIAC, moreover, have been so designed that in order to complete their operations they require the pulses of either one addition time cycle or of an integral number of addition time cycles.

Even though the electromechanical devices used with the ENIAC, the reader and the card punch, do not take an absolutely definite number of addition time cycles to complete their operations, these units have been integrated into the synchronized system since they have been provided with program controls which emit a CPP as a program output pulse. Units of the ENIAC can even operate in parallel with the card reader since the reader does not emit a program output pulse signifying the completion of reading until it has received as an interlock pulse a program output pulse from some other unit of the ENIAC to indicate that the sequence carried on in parallel with reading has been completed.

Because the units of the ENIAC operate in synchronism with one another and because multiple digit and program trunks have been provided, the operator can schedule parallel operations when planning the set-up of a problem. For example, the multiplier can be operating while several accumulators are performing additions and subtractions and while the divider is finding a quotient. Naturally, the scheduling of parallel

operations requires that the operator plan for the use of separate digit trunks for the various operations and, in some cases, requires that attention be given to the number of addition times needed for the operations.

It is the sacrifice of the possibility of scheduling parallel operations which, surprisingly, leads to the considerable advantages of the system to be described in sections 2-7.

PROGRAMMING THE ENIAC.

The detailed description of the sequence of operations to be followed by the ENIAC is what we call a set-up table. As formerly carried out it consisted of a set of sheets of paper with about twenty-seven columns, (one for each accumulator and function table, the constant transmitter, the master programmer, the reader, and the printer) on which were listed chronologically downward the program control settings and input and output pulses for each operation. As we briefly describe each unit we shall sometimes show a typical element of a set-up table for that unit.

THE ACCUMULATOR

The accumulator front panel contains, from bottom to top: program trays and sockets; the program control neons; the input, output sockets; the program controls; digit input and output sockets; digit trays and sockets; and finally the counter neons. Typical parts of it are shown in Figure 1.1. Program pulses are routed from a program line to a program input socket 11, 21, etc. by a cable.

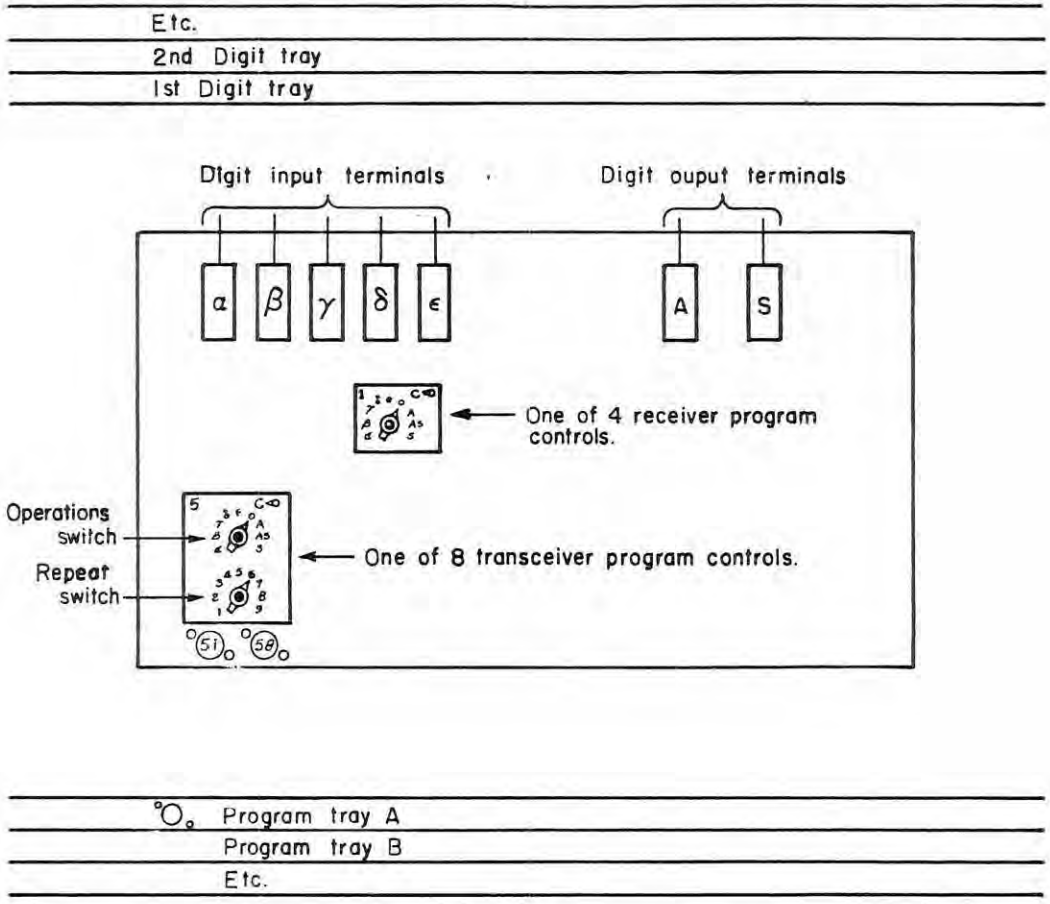

FIG. 1.1

In Figure 1.2, the numbers and associated letters in the top row show how the digit tray sockets and the accumulator input and output sockets are cabled together. For example, in accumulator 7 the δ input socket is connected to the second digit tray and the subtract output socket is connected to digit tray 3.

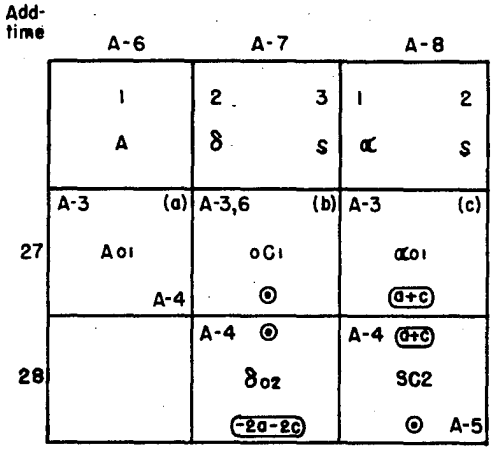

FIG. 1.2

In each row and column the letter and number in the upper left corner indicate the program tray and wire which carry the CPP which stimulates the operation indicated in that box. The next number, if any, indicates with which program input socket the aforementioned wire is connected. The symbols in the center of the box indicate firstly whether the accumulator is to receive through the α, β, γ, δ or ϵ input socket, or transmit positively or negatively or both, or do nothing; secondly, whether the accumulator should hold the manner or clear it when finished; and thirdly, how many times the operation should be done. The letter and number at the lower right corner indicate the tray and wire which should carry away the output pulse emitted by the accumulator upon completion of the operation. For example, in Figure 1.2, in the twenty-seventh add-time CP pulse carried by program tray A, wire 3 stimulates accumulators 6, 7, and 8. Accumulator 6 transmits its contents, a, once on digit tray 1 and retains it; accumulator 7 does nothing, then clears; accumulator 8 receives once from digit tray 1 and holds. Thus a, b, and c are replaced by a, 0, and a + c. At the end of add-time 27, accumulator 6 emits a pulse which is carried on program tray A, wire 4, and stimulates accumulator 8 to send -a-c twice to accumulator 7, then clear. This example illustrates set-up tables in general and reminds the reader that addition, subtraction, and transmission are accomplished in one add-time.

THE MULTIPLIER

The multiplier has a front panel similar to that of an accumulator and is associated with four accumulators called the multiplier (11), multiplicand (12), left-hand partial product (13), and product accumulators (15). An element of the set-up table for the multiplier is more complicated simply because there are more things to control. The symbols shown in Figure 1.3 indicate that in add-time 56, accumulators 11 and 12 receive into channels α and β respecitvely, then in the next 7 + 3 add-times the 7-digit multiplier is multiplied by the 10-digit multiplicand, and in the 12th add-time the answer rounded off to six places is transmitted positively and negatively from accumulator 15 which then clears. The multiplier and multiplicand retain their numbers. At the end of the 11th add-time the CPP is emitted from the multiplier and is carried

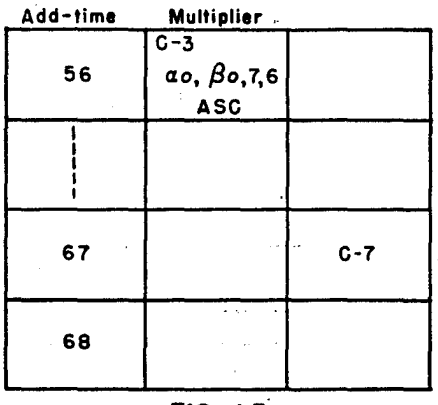

FIG. 1.3

away on wire 7 of tray C to stimulate some accumulator to receive the product. The decimal is at the left, i.e., only the 10 most significant figures of a 20 digit product are retained.

THE DIVIDER AND SQUARE-ROOTER

The divider is similar to the multiplier, with the two differences that the associated accumulators are 4-quotient, 5-numerator, 7-denominator, 9-shifter, and that division requires an indefinite time averaging approximately 130 add-times. The divider is also the square-rooter but accumulator 4 is not associated with square-rooting and the number appearing in accumulator 7 is twice the square-root. In both cases the number in accumulator 5 is replaced by the remainder.

THE READER

The reader is a piece of IBM equipment which reads eight ten-digit numbers from a card into relay switches called the constant transmitter. Reading takes about 2500 add-times and like every other operation is stimulated by a CPP and terminates with a CPP.

THE PRINTER

The eight accumulators 1, 2, 15, 16, 17, 18, 19, and 20 are connected to a piece of IBM equipment called the printer which prints the numbers stored in these accumulators on a card when the printer is stimulated. This requires about 1500 add-times.

THE CONSTANT TRANSMITTER

Any of the eight numbers from an IBM card sent by the reader to one of the eight positions A, B -- H in the constant transmitter may be sent out on a digit tray in one add-time as a whole, or only the left or right five digits may be sent. For example, the symbols in figure 1.4 indicate that in add-time 1021, the left five digits of the first number are sent out on digit tray one, followed in the next add-time by the right five digits of the second number, and in the next add-time by all ten digits of the fourth number. The numbers transmitted remain in the constant transmitter until a new card is read.

Add-time	A I	B I	D I
1021	D-11 A_L D-9		
1022		D-9 B_R E-1	
1023			E-1 D_{LR} E-2

FIG. 1.4

THE FUNCTION TABLES

The 3 function tables each contain a portable bank of switches for 200 six-digit numbers which may be used as 100 twelve-digit numbers and an associated panel of program controls. There is a switch to

govern whether the function table looks up F(x-2), F(x-1), F(x), F(x+1) or F(x+2), where x is the number from 0 to 99 sent to it in the second add-time, and a switch to determine the number of times the function is transmitted. For example, from figure 1.5, in the 21st add-time a CPP from tray K, wire 2, stimulates function table I. In the next add-time the pulse NC stimulates some accumulator to send an argument from one to ninety-nine to function table I. In the 25th and 26th add-times F(arg-1) is transmitted.

	FT I
21	K-2 A-1 NC-2
22	
23	
24	
25	
26	C-5

FIG. 1.5

In the next section we shall see that the function tables provide the memory for orders as well as numbers.

THE MASTER PROGRAMMER

The master programmer consists of ten 6-stage ring counters called steppers. Each stepper has three inputs and six outputs. The inputs are the direct input, the input, and the clear direct input. N pulses, CPP or digit, to the direct input immediately advance the stepper N stages. A CPP sent to the input causes a CPP to be emitted one add-time later from the output socket associated with that stage of the stepper in which it finds itself; a pulse to the clear direct input immediately clears the stepper to the first stage. The master programmer was designed to permit repeating a sequence of operations an arbitrary fixed number of times and then do another sequence of operations. However, the conditional transfer order to be discussed in sections 3, 4, 5, 6, and 7 has obviated this use of it and replaced it with another which we shall discuss in the next section.

SECTION II

CONVERSION OF DIGIT PULSES TO PROGRAMMING PULSES

Reduced to simplest language the basic notion of von Neumann's programming method is to build complicated routines from a small (not necessarily minimal) set of elementary orders. In our case, the number of these elementary orders is sixty. Von Neumann's largest single contribution to the application of his methods to the ENIAC was to point out how 60 distinct two-digit numbers can be converted into 60 distinct CPP's, each of which can in turn stimulate one of the sixty desired orders.

It would have been possible to use a vocabulary of 51 orders without changing the ENIAC at all. However, by the addition of four small units, it became possible to speed the process up considerably and have 60 orders; therefore, these units were constructed. The first of them is a ten-stage stepper which, associated with the ten six-stage steppers of the master programmer, constitute the converter. This device operates as follows: one digit, α, of a two-digit number, $10\alpha + \beta$, is fed into the direct input of the ten-stage stepper, setting it to stage α; the other digit, β, is fed to the direct inputs of the ten six-stage steppers

setting them to stage β. A CPP to the ordinary input of the ten-stage stepper causes a CPP to go from the α th output of that stepper to the ordinary input of the α th stepper. One add-time later, a pulse $\alpha\ \beta$ is emitted from stage β of stepper α.

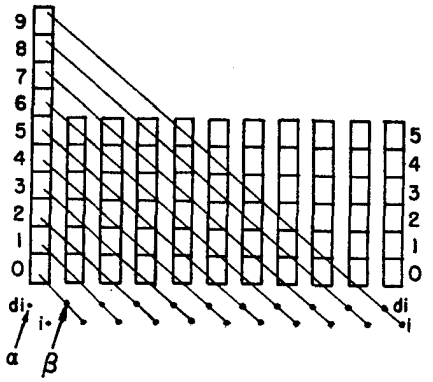

FIG. 2.1

Since α can be 0, 1, ---, 9 and β can be 0, 1, ---, 5, there are sixty distinct pulses which can be obtained by converting the corresponding two-digit number. As a matter of practice, the pair of digits $\alpha\beta$ comes from the function table, but not directly. Instead, a twelve-digit number is sent six times from the function table to a device we shall call the order selector. It is a set of twelve gates controlled in pairs by a six-stage ring. The order selector allows the first pair of digits to pass the first time, the second pair the second time, etc., until all six pairs of digits from one row of the function table have been converted, at which time a pulse from the sixth stage of the selector ring causes the function table argument to be increased by one and clears the selector ring back to stage one.

There are three function tables and we may desire to use any argument with any one of the three function tables. This is done by putting a three-digit address in accumulator 6 in the right three digit positions (called positions 1, 2, 3). The third digit, i, is first sent to a six-stage stepper called the function table selector which emits a pulse one add-time later to stimulate the corresponding function table F_i. The function table itself then stimulates accumulator six to transmit the first two digits $10\,\delta + \gamma$ for an argument.

ORDER / BASIC SEQUENCE		F.T. SEL.	ORDER SEL.	10 STAGE STEPPER	MASTER PROG.		ACC. 6		ACC. 20		F.T. I	F.T. II	F.T. III	SC
	C-5	IS	DUAL OF	D-6.	C-5	BEGINS	BASIC	SEQ.			NC	→	J-4	
	D-6 is DUAL OF E-1	E-1 i #1 E-8 di E-9 i #2 1 → E-11 2 → F-1 3 → F-2	F-3 CPP1 F-4 G_i-8	 F-5 cd1 E_i-10	 F-5 cd1	D-6 is DUAL OF J-2	J-2 A01 E-9 J-4 A01 F-3 F-4 ϵC1 F-5 002 G-8		C-5 006 E-10		E-11 A-2 NC1	F-1 A-2 NC1	F-2 A-2 NC1	F-3 001 F-5

Figure 2.2

It then transmits $F_i(10\delta + \gamma - 2)$ to the order selector whence a certain pair of digits, $10\beta + \alpha$, of $F_i(10\delta + \gamma - 2)$ goes to the convertor to be converted to a pulse to stimulate the corresponding operation. This whole basic sequence is summarized in detail in Figure 2.2. The basic sequence precedes every order; however, in the case of most orders it is programmed to start before the preceding order has finished. Thus, though the basic sequence takes seven add-times, the basic sequence plus any one of many of the orders also takes seven add-times.

The pulses which stimulate the sixty orders are listed in the following table.

STAGE \ STEPPER	A	B	C	D	E	F	G	H	J	K
5	S-5	L-11	H-5	S-10	L-4	E-3	H-10	G-2	C-10	V-7
	05	15	25	35	45	55	65	75	85	95
4	S-4	L-10	H-4	S-9	L-3	H-7	H-8	V-1 or E-7	D-11	V-6
	04	14	24	34	44	54	64	74	84	94
3	S-3	L-9	H-3	S-8	L-2	H-6	V-3	E-4 or E-6	D-10	V-8 or V-4
	03	13	23	33	43	53	63	73	83	93
2	S-2	L-8	H-2	S-7	L-1	C-9	O-2	H-11	D-9	H-0 or V-2
	02	12	22	32	42	52	62	72	82	92
1	S-1	L-7	H-1	V-9	C-2	L-6	O-1		D-8	C-8 or C-3
	01	11	21	31	41	51	61	71	81	91
0	C-1	S-6	C-7	E-5	S-11	L-5	B-1	V-5	D-7	C-11
	00	10	20	30	40	50	60	70	80	90

Table 2.I

THE PULSE AMPLIFIERS

Often it is desirable to have many different pulses stimulate the same operation (as well as distinct ones). This may be accomplished with the pulse amplifiers (eight units, 11 amplifiers per unit) using the one-way pulse transmission property of a vaccum tube. In the following table is shown the hierarchy of CPP pulses obtained with the pulse amplifiers.

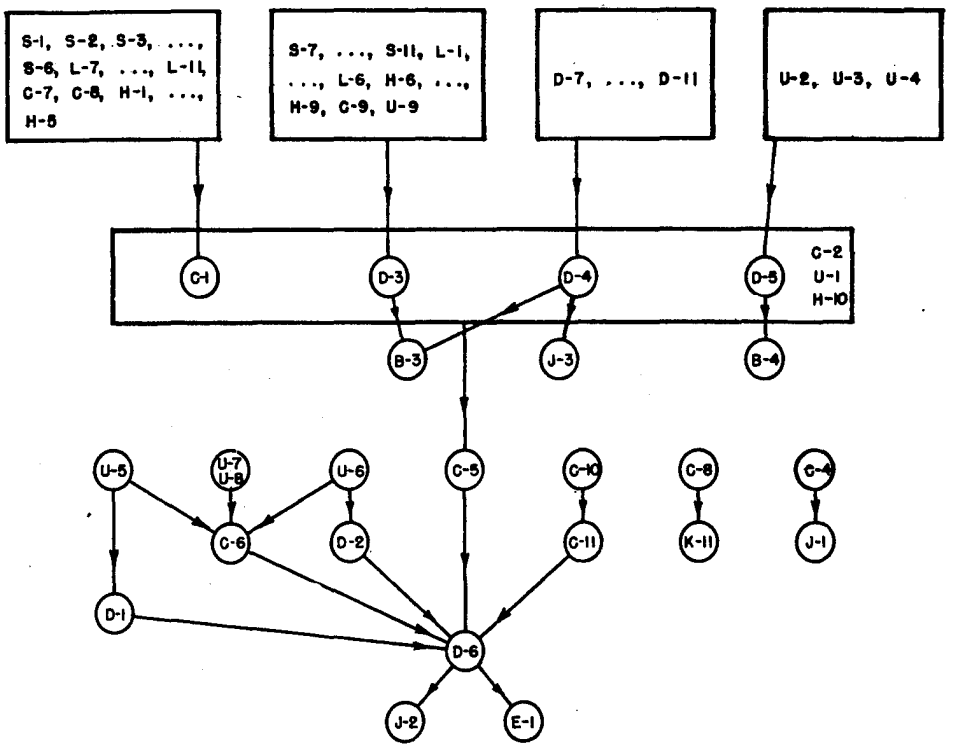

TABLE 2. II

SECTION III

DETAILED DESCRIPTION OF THE SIXTY ORDERS

We have seen that there are five kinds of storage places used in the ENIAC. Numbers may be moved between these places but not arbitrarily. From a card in the reader numbers may only be sent to the constant transmitter; from the constant transmitter numbers may only be sent to one of the twenty accumulators; likewise numbers in the function tables can only go to the accumulators; numbers in accumulators may be sent to other accumulators or from certain special accumulators to the printer; from the printer the numbers may only be taken by carrying away the IBM cards. It is accordingly clear that a considerable number of the 60 available orders must provide for moving numbers about. In fact there are 50 orders used for this purpose. Before describing them we should remark that actually 67 orders are set up on the ENIAC making two sets of orders possible; 53 are common to both sets. Of the dual orders we shall name those of the first set 73, 74, 91, 92, 93, 94, 95 and those of the alternate set 73a, 74a, 91a, 92a, 93a, 94a, 95a.

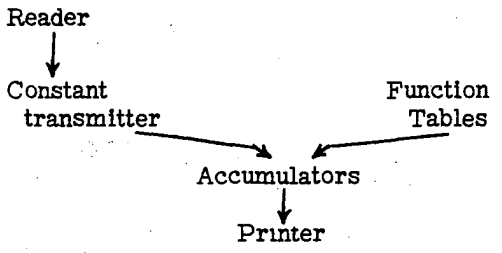

Figure 3.1

READER TO CONSTANT TRANSMITTER

The order which sends the eight ten-digit numbers from the reader to the constant transmitter we call the read order, order 62 (code name: Rd). It requires a little over a half second to complete.

CONSTANT TRANSMITTER TO ACCUMULATORS 11 AND 15

Since there are ten ten-digit numbers in the constant transmitter (In addition to the eight coming from the reader, there are two fixed ten-digit numbers set with switches.) and twenty accumulators, two hundred orders would be required to make it possible to send any number immediately to any accumulator. It would be possible to have one order which caused all ten numbers to go to a specific set of ten accumulators. However, this would mean that all ten accumulators had to be clear before the order could be given. The compromise agreed upon for these two sets of 60 orders is a set of five "constant transmitter" orders, 80-84 (code names: AB, CD, EF, GH, JK), each of which sends two numbers, A&B, C&D, E&F, G&H, or J&K to accumulators 11 and 15 respectively. J&K store the two ten-digit arbitrary constants.

Consider order 80 for example: From table 2.I we see that order 80 starts with pulse D-7, which via the pulse amplifiers blossoms into pulses D-4, B-3, C-5, J-3, D-6, J-2, and E-1.

ORDER															
80 - C.T.		F.T. SEL.		ACC. 6		ACC. 11		ACC. 15		ACC. 20	F.T. I	F.T. II	F.T. III		C.T.
(A_{LR}, B_{LR})	G-5 IS DUAL OF D-6					D-7 IS DUAL OF D-4 D-4 IS DUAL OF C-5, J-3, & B-3									
	D-6 is DUAL OF E-1	E-1 i #1 E-8 di E-9 i #2 1 → E-11 2 → F-1 3 → F-2	D-6 is DUAL OF J-2	J-2 A01 E-9		J-3 $OC1_{T-6}$ T-6 δ 01		B-3 β01							D-7 B_{LR} *1 *1 A_{LR}

Figure 3.2

From Figure 3.2 we see that pulse D-7 causes the constant transmitter to send out B, then A on digit tray one (see digit tray hook-up). B-3 causes accumulator 15 to receive B from digit tray one, adding it to its previous contents; J-3 causes accumulator 11 to clear, then receive A from digit tray one. C-5 causes the basic sequence for the next order to start while A and B are moving from the constant transmitter to accumulators 11 and 15. This is a 7 add-time order.

ACCUMULATOR TO ACCUMULATOR

There are, in the first and alternate sets of 60 orders, 39 and 38 orders respectively which move numbers from one accumulator to another. We shall discuss first the 34 orders in this category common to both sets.

There are sixteen orders 01, 02, 03, 04, 05, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, controlling accumulators 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 17, 18, 19 respectively which we may name "accumulator X listen", "X_i", orders. Of these, order 0.1 is typical. Order 01 starts (see tables 2.I and 2.II) with S-1, which leads to C-1, C-5, D-6, J-2, and E-1. S-1 (see figure 3.3) causes accumulator 1 to clear and then

receive from digit tray two; C-1 stimulates a dummy in the constant transmitter whose only purpose is to delay J-1 which causes accumulator 15 to transmit positively onto digit tray two then clear; C-5, of course, starts the basic sequence for the next order. The effect, then, or order X_1 is for accumulator X to clear, then receive from accumulator 15 which then clears.

ORDER

			ACC. 1					ACC. 15				CT
01 - ACC. 1_1		S-1 is DUAL OF C-1	S-1 AC1 0-8 0-8 α01					J-1 AC1			C-1 is DUAL OF C-5	C-1 001 J-1

Figure 3.3

Order 20 (code name: 13_1) differs from these orders in that accumulator 13 does not clear before receiving from accumulator 15.

There are likewise 16 orders, 31, 32, 33, 34, 35, 40, 41, 42, 43, 44, 45, 51, 52, 53, 54, 64 controlling accumulators 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 17, 18, 19 respectively, which we may name "accumulator X talk" or "X_t" orders. The effect of X_t is for accumulator X to send its number positively to accumulator 15 with both accumulators holding. This leads to addition in accumulator 15.

Order 50 (code name: 13_t) is similar except that accumulator 13 clears after sending.

In the alternate set of orders, orders 91a, 92a, 73a and 74a (code names: 20_1, 20_t, 6_1, 6_t) are used for accumulator 20 listen, accumulator 20 talk, accumulator 6 listen, and accumulator 6 talk respectively.

In the first set of orders, accumulator 6 is treated essentially like four smaller accumulators: respectively digits 1, 2, 3; digits 4, 5, 6; digits 7, 8; and digits 9, 10, 11. Order 92 (code name: 6 (11, 10, 9)) adds the numbers in accumulator 6 (11, 10, 9) and accumulator 15 (11, 2, 1) and leaves the sum in both. Order 93 (code name: 6 (8, 7)) adds the numbers in accumulator 6 (8, 7) and accumulator 15 (2, 1), leaving the sums in accumulators 6 (8, 7) and 15 (2, 1). All these accumulator orders take 7 add-times. To control accumulator 20, order 91 (code name: 18↔20) is provided which takes 9 add-times and interchanges the numbers in accumulators 18 and 20 via accumulator 15.

THE TRANSFER ORDERS IN THE FIRST SET

Any order which sends a number to accumulator 6 (3, 2, 1) we shall call (with J. von Neumann) a transfer order because the next order will automatically be taken from a different part of the function table, i.e. the control is transferred from one address to another. Order 73 (code name: 6R3) is the simple transfer for the first set. It causes accumulator 6 (3, 2, 1) to clear and receive the number in accumulator 15 (3, 2, 1) and then clears accumulator 15. We might call order 74 (code name: 6R6) the transfer with a rider. It causes accumulator 6 (6, 5, 4, 3, 2, 1) to clear, then receive the number in accumulator 15 (6, 5, 4, 3, 2, 1) and then clears accumulator 15. The purpose of this order is to make it possible to store an extra address in accumulator 6 (6, 5, 4). Both of these transfer orders take 13 add-times.

FUNCTION TABLE TO ACCUMULATOR ORDERS

The first system has four function-table-to-accumulator orders and the alternate system has five, of which two are transfers. Two of these orders are common to both sets. The first is order 72 (code name: FT). Accumulator 11 is cleared, then the function table transmits 12 digits and 2 signs as F (address stored in accumulator 8 (3, 2, 1)). Accumulator 11 receives the left sign and the left 6 digits at far left, and accumulator 15 receives the right sign and the right 6 digits at far left. The argument in accumulator 8 is then increased by one. The time consumed is 13 add-times.

Order 70 (code name: N2D) we call the next two digits order. It operates to send accumulator 15 (2, 1) the two digits following the order 70 in the function table. Fourteen add-times are required.

In the first system the orders 94 and 95 (code names: N4D, N6D) operate similarly to send the next four and next six digits to accumulator 15 in 20 and 26 add-times respectively.

The alternate system contains an order 93a (code name: N3D8) which sends, in 20 add-times, an address from the next two order positions in the function table to accumulator 8 (3, 2, 1) to be used as address for order 72. The first digit in the next two order positions is added to the digit in accumulator 8 (4); accumulator 8 (3, 2, 1) is cleared before receiving.

In the alternate system the transfers 94a and 95a (code names: N3D6, N6D6) send in 20 or 26 add-times one or two addresses from the next two or three order positions in the function table to accumulator 6. In order 94a, as in 93a, the first digit in the next two order positions is added to accumulator 6 (4).

ACCUMULATORS TO PRINTER

The last transmission order, order 61 (code name: Pr), causes the next IBM card to move into position in the printer and then the numbers in accumulators 1, 2, 15-20 to be printed on the card and held in the accumulators. One second is required.

THE TEN OPERATIONS

It is not quite true that each of the fifty transmission orders we have just described accomplishes transmission only. All the "X_t" orders, for example, are also addition orders; the transfer orders perform the important operation of shifting the control. The following ten orders, however, are more explicity operative.

Order C. Order 00 (code name: C) simply clears accumulator 15 and takes 7 add-times.

Halt. Order 71 stops the ENIAC. It is the only order which does not stimulate the basic sequence for the next order.

M. Order 63 replaces the number in accumulator 15 by its complement via accumulator 13 in 7 add-times; for example, to form a-b, send b to accumulator 15, use order 63, then send a to accumulator 15.

D.S. Order 65 replaces the plus or minus sign in accumulator 15 by the plus sign. This is not the same as the change sign order 63 even if the number is negative since it does not affect the first ten digits. This order is added to facilitate the use of accumulators to hold two numbers: an arbitrary number and a number of known sign. This order takes 7 add-times.

X. Order 30 is the multiplication order. To use it, the multiplier must have been previously put in accumulator 11, the multiplicand in accumulator 15, and any number to be added to the product in accumulator 13. After multiplication the product plus the contents of accumulator 13 are placed in ac-

cumulator 15 and the multiplier and multiplicand remain in accumulators 11 and 12; accumulator 13 clears. This order takes 15 add-times. The factors are treated as if they had ten digits each. Only ten places of the partial products are kept. It follows that the answer may be incorrect in the 10th place from the left, by 14 for maximum error or 7 for the average error.

÷ Order 55 causes the number in accumulator 15 to be divided by the number in accumulator 7. During this process the number in accumulator 5 is replaced by the remainder and the quotient goes to accumulator 15. Accumulator 13 must be previously cleared. Assuming that the decimal point in the numerator and denominator is at the extreme left, the quotient has its decimal point two places to the right, and therefore the numerator must not exceed the denominator by more than a factor of 100. Order 55a is arranged so that the quotient is not shifted and therefore the numerator must not exceed the denominator. The time required for division is variable but is of the order of 130 add-times.

$\sqrt{\ }$ Order 60 differs from order 55 only in that the operation is different and accumulator 4 is not involved formally. The radicand must not exceed .2499999999.

Sh and Sh'. Orders 90 and 85 are called shift and shift' respectively. Each takes two order positions in the function tables and requires 20 add-times to complete. The shift order causes the number in accumulator 15 to be shifted as indicated by the two digits in the next order position. The first of these two digits is 0 or 9 according as the shift is to the left or right and the second digit defines the magnitude of the shift, from 1 ---5 inclusive. In the shift' order 85, the number in accumulator 15 is shifted as in order 90, but also the complementary amount in the complementary direction and put in accumulator 12. Thus 85 04 would replace a in accumulator 15 by 10^{4}a and put 10^{-6}a in accumulator 12.

C.T. The most important order, from the point-of-view of control, is order 75, the conditional transfer. The effect of the order is to shift the control from the address in accumulator 6 (3, 2, 1) to accumulator 6 (6, 5, 4) if the number in accumulator 15 is positive or zero. If the number is negative, the program continues with address in accumulator 6 (3, 2, 1). Sensing the sign of the number in accumulator 15 is accomplished by sending the pulses from the P-M line of the add output socket of accumulator 15 to a two-stage stepper called P-M discriminator no. 2. Shifting the control is accomplished by sending the number from accumulator 6 to accumulator 10 through a shifter which shifts 6, 5, 4 to 3, 2, 1, clears 3, 2, 1, and leaves the other digits alone; the number is then returned to accumulator 6. In section 6 we shall consider more carefully the use of this order. It is immediately clear that because of it the machine can make any choice of sequences of operations that can be made to depend on the sign of a number. Most computations involve more or less simple inductions which in turn depend on decisions based on the sign of a number.

SECTION IV

SUMMARY OF ORDERS

A summary of orders is found on the following four pages.

ORDERS FOR 60 WORD VOCABULARY - November 13, 1947

Code Number	Code Name	Order	Contents of Affected Accs. Before	After	Add. Times
00	C	Clear Acc. 15	(15)a	(15)0	7
01,02,03 04,05,10 11,12,13 14,15, 21-25	x_1	Acc. x clear, and then receive from acc. 15 which transmits and then clears ($x \neq 6,13,15,20$)	(x)a (15)b	b 0	7
20	13_1	Acc. 13 hold and receive from acc. 15, which transmits, then clears	(13)a (15)b	a + b 0	7
31-35, 40-45, 51-54,64	x_t	Acc. x transmit-hold to acc. 15 ($x \neq 6,13,15,20$)	(x)a (15)b	a a + b	7
50	13_t	Acc. 13 transmit-clear to acc. 15	(13)a (15)b	0 a + b	7
30	X	No round-off, 10 place multiplier. The number, a in acc. 11 is multiplied by the number d, in acc. 15, added to the number c in acc. 13, and stored in acc. 15. At the end acc.'s 11,12,13 hold a, 0, and d respectively.	(11)a (12)b (13)c (15)d	a d 0 ad + c	15
55	÷	Divide and round-off to 10 places. The number c in acc. 15 is divided by the number b in acc. 7, and the quotient is stored in acc. 15, the remainder in acc. 5. Acc. 13 must be clear at the beginning. The alternate division order uses a special + 2 shifter at 4α that takes data arriving on line 9 and puts it into counter 11. For this reason, denominator must always exceed numerator and, if decimal points of numerator and denominator are at extreme left, decimal point of quotient is also at extreme left.	First digit place at left of b must be zero. (5)a (7)b (13)0 (15)c	 rem. b 0 c/b	 approx. 75
60	$\sqrt{}$	Round-off to 10 places. The square root of the number c, in acc. 15 is put into acc. 15 and the remainder in acc. 5. Acc. 13 must be clear at the beginning. A +1 shift of $\sqrt{}$ is made by sending twice square root to acc. 15 five times. This means that if decimal point of radicand is at extreme left, decimal point of $\sqrt{}$ is also at far left.	(5)a (13)0 (15)c $c \leq .24\ 99\ 9999\ 99$	rem. 0 c	 approx. 75
63	M	(complement of no. in acc. 15)	(13)0 (15)a	0 -a	7
65	DS	Drop the sign of the number in acc. 15. Note that if number in acc. 15 is originally -a, it will become (10^{10}-a); if it is originally, a, it will remain, a.	(13)0 (PM counter of 15) (counters 1-10) a	0 + a	7
90	Sh	Shift no. in acc. 15 as described below. Shift order takes two positions: 1st position: code symbol 90 2nd position: 10j +k where j indicates direction (0 to left, 9 to right) k indicates number of places ($1 \leq k \leq 5$)	(13)0 (15)a	$\frac{10^k \cdot a \text{ if } j=0}{(13)0 \quad (15)10^{-k} \cdot a \text{ if } j=9}$	20

ORDERS FOR 60 WORD VOCABULARY - November 13, 1947 (Continued)

Code Number	Code Name	Order	Contents of Affected Accs. Before	After	Add. Times
71	Halt	Do not get next order (all other orders include getting next order) and therefore cease computing.			
80 81 82 83 84	AB CD EF GH JK	Clear acc. 11, then constant transmitter send 1st. no. mentioned to acc. 11 and 2nd no. mentioned to acc. 15.	(11)x (15)y Const. Trans (A)a (B)b	a y + b a b	 7
70	N2D	Send next pair of instruction digits to acc. 15. This order occupies two order positions. 1st position: code symbol 70; 2nd: $10\alpha+\beta$ where $0< \alpha < 9, 0\leq \beta \leq 9$	(13)0 (15)x	(13)0 (15)x + $10\alpha + \beta$)	 14
72	F.T.	Clear acc. 11. F.T. transmit 12 digits and 2 signs from F.T. and line specified in acc. 8 (3,2,1). Acc. 11 receive left sign and 6 digits. Acc. 15 receive right sign and 6 digits. Increase arg. in acc. 8 by 1. Nos. are received at far left of 11 and 15.	(8) places 3,2,1 a (11)x (15)y	 a + 1 L(a) R(a)	 13
61	Pr.	Move next card into printing position and print contents of accs. 1, 2, 15-20. Move next card into read position.	Move and print		60 cards per min.
62	Rd.	Store information from card on relays of ENIAC reader.			100 cards per min.
75	C.T.	Examine no., a, in acc. 15 and then clear acc. 15. 1) If number in acc. 15 is negative, continue with next order on line specified at 6(3,2,1) 2) If positive, transfer 6(6,5,4) → 6(3,2,1) and continue with FIRST ORDER of line now specified in 6(3,2,1)	 Acc. 6 11-7 x 6-4 a 3-1 b Acc. 13 0 Acc. 15 a	If a ≥ 0 Acc. 6 11-7 x 6-4 0 3-1 a Acc. 13 0 Acc. 15 0 If a < 0 Acc. 6 11-7 x 6-4 a 3-1 b	 14 8
85	Sh'	Shift no. located in acc. 15 as described below. Shift' order takes 2 positions: 1st position: code symbol 85 2nd position: 10 j + k The symbols j and k are interpreted in the following way: The no. which is in acc. 15 at the start of this order is shifted as described in the ordinary shift order (code symbol 90). But, furthermore, the no. which is in acc. 15 at the start of this order is also operated on by a shift of 100-(10j +k), in the sense of the ordinary shift order. The no. resulting from this latter shift is placed in acc. 12.	<u>If j=0</u> (12)a (13)0 (15)b <u>If j=9</u> (12) a (13) 0 (15) b N.B. The sign of the no. stored in acc. 15 before this order is given is preserved with the nos. that result in accs. 12 and 15 from this order.	 $10^{-(10-k)}$.b 0 10^{k}.b $10^{(10-k)}$.b 0 10^{-k}.b	 20

ORDERS FOR 60 WORD VOCABULARY - November 13, 1947 (Continued)

Code Number	Code Name	Order	Contents of Affected Accs. Before	After	Add. Times
93a	N3D8	Clear acc. 8(3,2,1). Send next 3 instruction digits to acc. 8(3,2,1). This order consumes 3 order positions. 1) 1st position: code symbol 93 2) 2nd position: $10\alpha+\beta$ In general, it is assumed $\alpha = 0$. But if $\alpha \neq 0$, α is added to whatever decade 4 of the acc. 8 holds before this order is given. $\beta \rightarrow$ acc. 8(3) 3) 3rd position: 10 $\gamma \rightarrow 8(2)$ $\delta \rightarrow 8(1)$	Acc. 8 11-4 x 3-1 y Acc. 13 0	Acc. 8 11-4 $x+\alpha$ 3-1 $10^2\beta+10\gamma+\delta$ Acc. 13 0	20
94a	N3D6	Clear acc. 6(3,2,1). Then send next 3 instruction digits to acc. 6(3,2,1). Take next order from FIRST ORDER POSITION of line now indicated in acc. 6(3,2,1). This order consumes 3 order positions. 1) 1st order position: code symbol 94 2) Remarks made above in N3D8 apply here if acc. 6 is substituted wherever acc. 8 is mentioned in N3D8.	Acc. 6 11-4 x 3-1 y Acc. 13 0	Acc. 6 11-4 $x+\alpha$ 3-1 $10^2\beta+10\gamma+\delta$ Acc 13 0 Next order is taken from F. T. 20 $\beta+1$ ($\beta = 0,1,2$, for Table 1,2,3, resp.) 1st pos. of line $(10\gamma+\delta)$	
95a	N6D6	Clear acc. 6(6-1). Then send next 6 instruction digits to acc. 6(6-1). Take next order from FIRST ORDER POSITION of line now indicated in acc. 6(3,2,1). This order consumes 4 order positions. 1) 1st position: code symbol 95 2) 2nd position: $10\alpha+\beta$ $\alpha \rightarrow 6(6)$, $\beta \rightarrow 6(5)$ 3) 3rd position: $10\gamma+\delta$ $\gamma \rightarrow 6(4)$, $\delta \rightarrow 6(3)$ 4) 4th position: $10\epsilon+\rho$ $\epsilon \rightarrow 6(2)$, $\rho \rightarrow 6(1)$	Acc. 6 11-7 x 6-4 y 3-1 z Acc. 13 0	Acc. 6 11-7 x 6-4 $10^2\alpha+10\beta+\gamma$ 3-1 $10^2\delta+10\epsilon+\rho$ Acc. 13 0	26
91a	20_1	Same as x_1 (see common orders)	(20) a (15) b	b 0	7
92a	20_t	Same as x_t (see common orders)	(20) a (15) b	a a + b	7
73a	6_1	Same as x_1 (see common orders) <u>BUT</u> take next order from FIRST ORDER POSITION of line specified at 6(3,2,1)	(6) a (15) b	b 0	9
74a	6_t	Same as x_t (see common orders).	(6) a (15) b	a a + b	7

ORDERS FOR 60 WORD VOCABULARY - November 13, 1947 (Continued)

Code Number	Code Name	Order	Contents of Affected Accs. Before	After	Add. Times
94	N4D	Send next 4 instruction digits to acc. 15. This order takes 3 order positions. 1) 1st position: code symbol 94 2) 2nd position: $10\alpha+\beta$ where $\alpha \rightarrow$ 15(4), $\beta \rightarrow$ 15(3) 3) 3rd position: $10\gamma+\delta$ where $\gamma \rightarrow$ 15(2), $\delta \rightarrow$ 15(1)	(13) 0 (15) a	(13) 0 (15) a + $10^3\alpha$ + $10^2\beta$ + $10\gamma+\delta$	20
95	N6D	Send next 6 instruction digits to acc. 15. This order takes 4 instruction positions. 1) 1st position: code symbol 94 2) 2nd position: $10\alpha+\beta$ where $\alpha \rightarrow$ 15(6), $\beta \rightarrow$ 15(5) 3) 3rd position: $10\gamma+\delta$ where $\gamma \rightarrow$ 15(4), $\delta \rightarrow$ 15(3) 4) 4th position: $10\epsilon+\zeta$ where $\epsilon \rightarrow$ 15(2), $\zeta \rightarrow$ 15(1)	(13) 0 (15) a	0 a+$10^5\alpha$+$10^4\beta$ +$10^3\gamma$+$10^2\delta$ +$10\epsilon+\zeta$	26
91	18↔20	Accs. 18 and 20 interchange contents	(15) 0 (18) a (20) b	(15) 0 (18) b (20) a	9
92	6(11,10,9)	Acc. 15(11,2,1) transmit-clear to Acc. 6 (11, 10,9). Then Acc. 6(11,10,9) transmit-hold to Acc. 1(11,2,1). Then Acc. 1(11,2,1) transmit-clear to 15(11,2,1).	Acc. 6 11-9 x 8-1 y Acc. 13 0 Acc. 15 10-3 0 11,2,1 a	 x + a y 0 0 x + 2	7
93	6(8,7)	Acc. 15(2,1) transmit-clear to Acc. 6(8,7). Acc. 6,(8,7) transmit-hold to Acc. 1(2,1). Then, Acc. 1(2,1) transmit-clear to Acc. 15(2,1).	Acc. 6 11-9 x 8,7 y 6-1 z Acc. 13 0 Acc. 15 11-3 0 1,2 a	 x y + a z 0 0 y + a	7
73	6R3	Clear 6(3,2,1) and then 15(3,2,1) transmit-clear to 6(3,2,1). Get next instruction from FIRST ORDER POSITION of line now specified 6(3,2,1)	Acc. 6 11-4 x 3-1 y Acc. 13 0 Acc. 15 3-1 b	 x b 0 0	13
74	6R6	Clear 6(6-1) and then 15(6-1) transmit-clear to 6(6-1). Get next instruction from FIRST ORDER POSITION of line now specified at 6(3,2,1).	Acc. 6 11-7 x 6-4 y 3-1 z Acc. 13 0 Acc. 15 6-4 a 3-1 b	 x a b 0 0 0	13

SECTION V

A SUGGESTED TEST PROGRAM

The ENIAC is complicated. It has about 20,000 tubes and thousands of switches and plug-in contacts. Since any of these or other things may fail, it is not surprising that the duration of an average run without some failure is only a few hours. For example, a power failure (of which there were 9 in November and December, 1947) may spoil four or five tubes. Some of these failures are not clear-cut failures like a short, but borderline failures which may cause a tube to operate improperly once in a hundred to a thousand times. It is therefore important to be able to test the ENIAC rather thoroughly, rapidly and systematically, and, as far as possible, without touching any cables or switches. Testing of this kind is a science in itself and testing procedures will constantly be changed and improved. The procedure described in this section may accordingly be considered as only one stage in such a development. It is a program, set on the function tables which tests each unit an appropriate number of times and prints the errors found on IBM cards. The nature of the errors should localize the point of failure usually to within 10 or 20 tubes (if it is a tube failure) which can be replaced and examined at leisure without holding up the machine. Of course, the fact that part of the function table switches which describe the problem to be run must be changed to carry out the test is a defect, since switch errors or bad contacts may be made in setting the problem back. This defect is partially corrected by comparing the orders set on the function table with those set on cards before starting the problem. This latter test uses only a few lines of a function table.

Function table test. To test the function tables the number in the first row and first function table is sent to accumulator 11 or 15, 10^n times, then shifted and subtracted from the correct number in the constant transmitter; if the difference is not zero, it is printed and then the next number is sent to accumulator 11 or 15, etc. The whole test takes about fifteen minutes if n = 3.

Multiplier test. A set of about 120 IBM cards is provided, each having two numbers, a_i and b_i, and the product c_i. The ENIAC reads a card, multiplies a_i by b_i, 10^n times, shifts the sum n places to the right, compares it with c_i and prints the difference if it is not zero; the next card is then read, etc. This test takes about a minute if n = 2.

Divider and Square-rooter test. Divisions and a square root are used.

Accumulator test. To test the accumulators, minus 0000000001 is sent from the constant transmitter to accumulator 1 whence it is sent to accumulator x and back to accumulator 15. Its sign is then changed and it is added to the number in accumulator 1; if the sum is not zero, it is printed. This is repeated m times, then the number is printed and then the value of x is changed. At present it is planned to test fourteen of the accumulators since the others are checked some other way. However, it may be desirable to check the others the same way. The testing time, if m is 1000, is about 5 minutes.

Constant transmitter. Two tests are provided for the constant transmitter. The first consists of sending minus 0000000001 from A_{LR} m times, adding m and printing if not zero, then repeating for B_{LR}, etc. This only takes a few seconds. The second consists of reading the card 9999999999, -9999999999, 9999999999, -9999999999, etc., adding A_{LR} to B_{LR}, C_{LR} to D_{LR}, etc., and printing if not zero. This is repeated fifty times, then the signs are changed, and it is repeated fifty times. The 9's are then reduced to

8's and the test proceeds. This tests the reader and the relay switches in the constant transmitter. About ten minutes are required for this test.

The whole program takes about thirty minutes for the values of m and n suggested, then probably another thirty minutes would be consumed by about four people to set and reset the 3600 switches required to run the program and then put the problem back on, and check the orders reset on the function tables.

Thus 1 to 1-1/2 hours will be required to run this functional test if all goes well. If a defective part is discovered, more time will be required to replace it and retest that unit (it is easy to start and stop the program at any two arguments in the function table).

A shorter, more elementary test which may be run every few hours to seek out major defects has been programmed. Since it only takes about 30 lines it may often be left on the function tables throughout the problem.

TABLE 5.I TEST PROCEDURE FOR ENIAC 10 March 1948

F.T. α

F.T. α test	00	94	0X	XX	11	00	00
	01	01	02	94	*01	00	63
	02	03	94	03	00	63	04
	03	70	01	63	05	00	00
	04	95	α0	9α	05	74	00
	05	72	90	94	32	02	44
	06	90	94	31	01	41	35
	07	11	35	63	33	03	33
	08	75	94	0α	05	73	00
	09	41	61	70	01	11	35
	10	90	02	03	70	01	34
	11	04	95	α1	3α	12	74
	12	34	75	94	0α	04	74
MULTIPLY TEST	13	94	01	12	63	05	94
	14	01	00	63	04	00	00
	15	62	95	α2	0α	16	74
	16	81	63	03	80	30	33
	17	03	33	65	63	75	44
	18	01	45	02	34	22	33
	19	61	00	00	00	00	00
	20	95	α2	3α	21	74	00
	21	70	01	34	04	34	75
	22	95	α2	0α	16	74	00
	23	95	α2	7α	24	74	00
	24	70	01	35	05	35	75
	25	94	01	00	63	04	94
	26	0α	15	73	00	00	00
ACC. TEST	27	62	70	13	90	05	03
	28	94	0α	32	15	00	00
	29	94	*09	99	33	03	00
	30	95	α6	2α	31	74	00
	31	45	73	00	00	00	00
	32	80	01	31	63	44	14
	33	94	0α	60	73	00	00
	34	81	02	32	63	44	14
	35	94	0α	60	73	00	00
	36	82	04	34	63	44	14
	37	94	0α	60	73	00	00
	38	83	05	35	63	44	14
	39	94	0α	60	73	00	00
	40	84	10	40	63	44	14
	41	94	0α	60	73	00	00
ACC. TEST	42	80	11	41	63	44	14
	43	94	0α	60	73	00	00
	44	81	12	42	63	44	14
	45	94	0α	60	73	00	00
	46	82	13	43	63	44	14
	47	94	0α	60	73	00	00
	48	83	21	51	63	44	14
	49	94	0α	60	73	00	00
	50	84	22	52	63	44	14
	51	94	0α	60	73	00	00
	52	80	23	53	63	44	14
	53	94	0α	60	73	00	00
	54	81	24	54	63	44	14
	55	94	0α	60	73	00	00
	56	82	25	64	63	44	14
	57	94	0α	60	73	00	00
	58	83	24	91	91	54	63
	59	44	14	94	0α	60	73
	60	44	65	63	75	33	01
	61	44	61	00	00	00	00
	62	95	α6	6α	63	74	00
	63	33	90	05	90	95	63
	64	75	70	01	63	33	03
	65	94	0α	30	73	00	00
	66	95	α7	1α	67	74	00
	67	33	90	95	63	75	70
	68	01	63	90	05	33	03
	69	70	02	45	15	94	0α
	70	29	73	00	00	00	00
	71	70	01	93	95	α7	4α
	72	77	74	00	00	00	00
C.T. TEST	73						
	74	70	19	63	92	00	00
	75	70	25	93	00	00	00
	76	62	95	α8	2α	77	74
	77	80	02	44	01	83	24
	78	91	44	25	82	24	44
	79	23	81	22	44	61	00
	80	70	01	63	93	63	75
	81	94	0α	76	73	00	00
	82	95	α8	5α	83	74	00
	83	70	01	92	75	94	0α
C.T. TEST	84	75	73	00	00	00	00
÷ & √ Test	85	62	80	10	44	01	81
	86	55	63	31	61	00	82
	87	63	60	61	00	82	01
SHIFT TEST	88	62	80	21	44	13	43
	89	85	01	45	85	91	45
	90	63	51	01	81	21	43
	91	85	02	45	85	92	45
	92	63	44	02	43	85	03
	93	45	85	93	45	63	51
	94	03	82	21	43	85	04
	95	45	85	94	45	63	44
	96	22	43	85	05	45	85
	97	95	63	51	23	70	85
JLR & KLR TEST	98	24	33	20	50	61	00
	99	84	01	44	02	61	71

30 60 61 00 01 00

*These numbers determine the number of iterations of each of the tests and can be changed at will.

SECTION VI

FLOW DIAGRAM OF TYPICAL PROBLEM

Suppose that we wish to have the ENIAC compute the supersonic airflow past a body of revolution ABC, without yaw. If the Mach number M(velocity in units of sound velocity) is sufficiently large, the nose

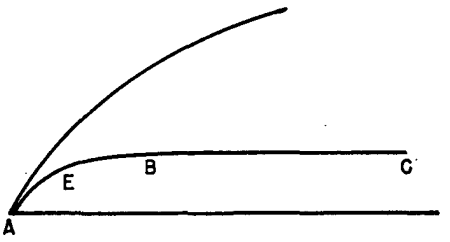

FIG. 6.1

angle sufficiently small on the curve ABC concave toward the axis, it is believed that there will exist an attacked shock-wave front (surface of discontinuous pressure, density, and velocity) and the velocity will be everywhere supersonic. It is then reasonable to assume that replacing a small section AE of the nose contour AB by a straight line will have little influence on the flow at a distance from AE. This assumption does, however, have the following effects:

1. If the characteristic (curve making angle $\omega = \csc^{-1} M$ with streamlines at each point, or curve of propagation of weak disturbances) from E, making an acute angle with the velocity vector, hits the shock wave at a point F, then the shock front is straight from A to F.

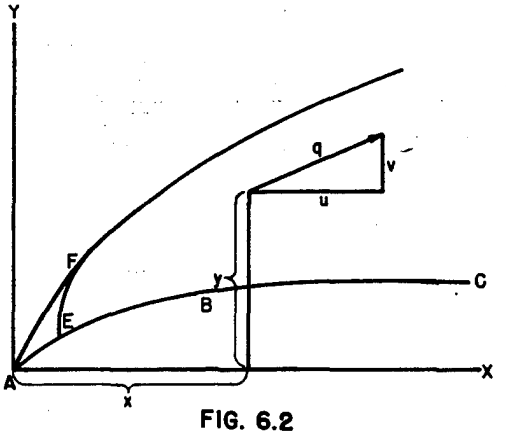

FIG. 6.2

2. In the region AEF, the partial differential equations defining velocity distribution can be satisfied by assuming that the velocity is constant on straight lines through A[3]. Thus u and v are functions of a single parameter in AEF and the partial differential equations become ordinary. It is the solution of these equations for x, y, u, v, and z along EF which we shall choose as an illustrative problem.

We have then a system of ordinary equations for x, y, u, v (see Figure 6.2) and z (stream function) in terms of some parameter along EF. We shall chose v/u, tangent of the inclination of the velocity vector, as this parameter.

[3]G.I. Taylor and J. W. Maccoll, The Air Pressure on a Cone Moving at High Speeds", Proc. Roy. Soc. A139, 278-311 (1938).

For our purposes there is no need to derive the equations, which are:

6.1

$$\begin{cases} \dfrac{dx}{dt} = (a^2 - u^2)\,F & q^2 = u^2 + v^2 \\ \dfrac{dy}{dt} = (-uv - \sqrt{a^2(q^2 - a^2)}\,)F & a^2 = \dfrac{\gamma-1}{2}(1 - q^2) \\ \dfrac{du}{dt} = -\dfrac{y\,u}{x + yt} & (1 - q^2)^{\frac{1}{\gamma-1}} = A \\ \dfrac{dz}{dt} = yA\left(u\dfrac{dy}{dt} - v\dfrac{dx}{dt}\right) & \\ v = ut & \\ F = -\dfrac{y}{a^2 v}\,\dfrac{u}{(x + yt)}\,\dfrac{(a^2 - v^2)x + y(uv + \sqrt{a^2(q^2 - a^2)})}{uv + \sqrt{a^2(q^2 - a^2)}} & \end{cases}$$

γ, the ratio of specific heats, is taken as 1.4 in the computations. There are initial and terminal conditions to be satisfied by x, y, u, v, z.

Clearly, at E

6.2

$$\begin{cases} x = x_E \\ y = y_E \end{cases}$$

and z may be taken

6.3 $$z = 0$$

Furthermore, the velocity vector must be parallel to AE:

6.4
$$\begin{cases} u = \dfrac{q_s}{\sqrt{1 + t_s^2}} & q_s \text{ is a parameter} \\ v = \dfrac{q_s t_s}{\sqrt{1 + t_s^2}} & t_s = \text{slope of AE.} \end{cases}$$

From the principles of conservation of mass, energy, and momentum follow the equations of Rankine and Hugoniot which must be satisfied at the shock-wave. Eliminating pressure and density from these equations we find the terminal condition which determines the value of t at the shock wave: namely, that if

6.5 $$\phi(x,y,u,v) = (ux + vy)\left[(y^2 + x^2)\,u - x(ux + vy)\right]\frac{\gamma+1}{\gamma-1} - x\left[x^2 + y^2 - (ux + vy)^2\right]$$ and u_w, v_w, x_w, y_w and t_w be the values of u, v, x, y, and t at the shock-wave, then $\phi(u_w, v_w, x_w, y_w) = 0$. Finally the velocity and Mach number in front of the shock-wave are determined by

6.6
$$\begin{cases} q_1 = \dfrac{u_w x_w + v_w y_w}{x_w} \\ M_1 = \sqrt{\dfrac{\frac{2}{\gamma-1} q_1^2}{1 - q_1^2}}. \end{cases}$$

If α and β are numbers, constant on characteristics, α is some constant on EF and we shall take β arbitrarily as a linear function of t

6.7
$$\beta = B\,\frac{t - t_s}{t_w - t_s}$$

where B is some integer, for example, the number of cards printed.

The procedure is to assume some value for q_s; integrate the equations with initial conditions by the method of Heun at equal steps in Δt up to the point where ϕ changes sign; integrate backwards a shorter step Δt to the point determined by linear interpolation where $\phi = 0$. Let us call this complete process a "run".

Then q_1 is computed, compared with its given value q_{1_g}, and if $\left|q_1 - q_{1_g}\right| >$ some given number S the initial velocity q_s is adjusted by linear interpolation. New runs are then made (i.e., the process is reiterated) until $\left|q_1 - q_{1_g}\right| < S$. Next Δt is taken equal to $\frac{t_w - t_s}{n}$, where n is some integer and the integration is repeated, printing the values of z, α, β, v, u, -x, y every 2^kth step until the shock-front is passed. One last step backwards is taken and the values of z, α, β, v, u, -x, y on the shock-front are printed. Finally the two functions k_ρ and k_p of the velocity just behind the shock-front are computed and printed.

Before considering the flow diagram of this problem, let us consider the major detail in it, namely one step of Heun integration. Given a system of ordinary differential equations

6.8 $$\frac{dp^h}{dt} = f^h(p^1, p^2, \ldots, p^H, t) = f^h(p,t),$$

where $h = 1,2, \ldots, H$

and $p = \left[p^1, p^2, \ldots, p^H\right]$,

and the values p^h_j of p^h at $t = t_j$,

the procedure of Heun to find $p^h_{j+1} = p^h(t_j + \Delta t)$

is to apply to the formulae:

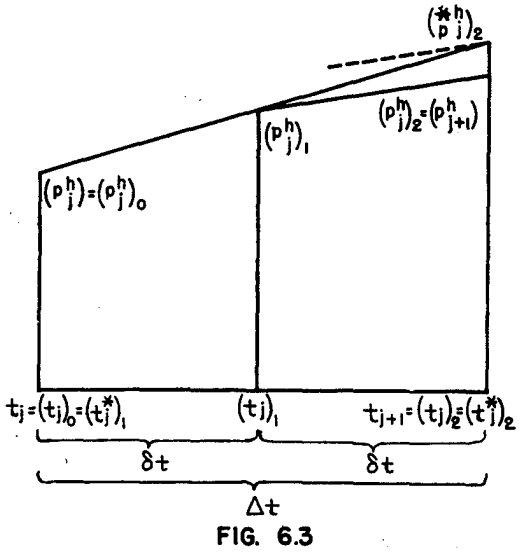

FIG. 6.3

6.9 $$p^{*h}_{j+1} = p^h_j + \Delta t\, f^h(p_j, t_j)$$

6.10 $$p^h_{j+1} = p^h_j + \Delta t \left[f^h(p_j,t_j) + f^h(p^*_{j+1}, t_{j+1})\right]/2$$

Emphasizing the one step from t_j to t_{j+1}, let us introduce δt and a subscript 1 which is 1, 2, or 3 so that

6.11 (Continued next page)
$$\begin{cases} \delta t = \Delta t/2 \\ p^h_j = (p^h_j)_0 \\ p^h_j + \delta t\, f^h(p_j, t_j) = (p^h_j)_1 \end{cases}$$

6.11
$$\begin{cases} p^h_j + \Delta t\, f^h(p_j, t_j) = (p^{*h}_j)_2 \\ \delta(p^{*h}_j)_o = 0 \\ (t^*_j)_1 = (t_j)_o \end{cases}$$

$$(t^*_j)_2 = (t_j)_2 = (t_j)_o + \Delta t$$

The following inductively defined sequences include equations 6.9 and 6.10:

6.12
$$\begin{cases} \delta(p^{*h}_j)_1 = f^h\left[(p^*_j)_1, (t^*_j)_1\right]\delta t \\ (p^{*h}_j)_1 = (p^h_j)_{1-1} + \delta(p^{*h}_j)_{1-1} & p^h_j = (p^h_j)_o \\ (p^h_j)_1 = (p^{*h}_j)_1 - \delta(p^{*h}_j)_{1-1} + \delta(p^{*h}_j)_1 & p^h_{j+1} = (p^h_j)_2 \end{cases}$$

which must be evaluated for $1 = 1$ and $1 = 2$ to complete one step of Heun integration.

We may now begin to form a flow diagram of the problem. See von Neumann and Goldstine's report referred to in section 1. Essentially this problem is a triple induction; a Heun integration loop to be circled twice; a "run" loop to be repeated until ϕ changes sign; and a "set of runs" loop to be repeated until the free stream velocity is satisfactory. We shall assign the induction (bound) variables 1, j, and i to these loops. The values of 1 are 1 and 2; of j are 1, 2,...,J where J is the smallest integer such that $\phi_j \geqslant 0$; of i are 1, 2,..., I where I is the smallest integer such that

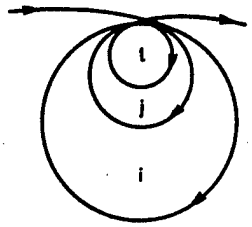

FIG. 6.4

$$\left|(q_1)_i - (q_1)_g\right|^2 - s^2 \geqslant 0.$$

Actually, however, the problem is slightly complicated by the fact that there is always one last step and one last set of steps (which we shall call a last run) with a printing induction to which we assign the bound variable m. To distinguish between an ordinary step and a last step we shall store a quantity

$\epsilon_s = \begin{cases} +1 \text{ ordinary step} \\ -1 \text{ last step.} \end{cases}$ Similarly we shall use $\epsilon_p = \begin{cases} 0, \text{ non-print run} \\ 2^k - m + 1, \text{ print run} \end{cases}$ to distinguish between a print run and a non-print run. With these definitions we may now develop the flow diagram as follows, showing in the alternative box at the exit to each loop, the variable whose change in sign switches the control out of the loop.

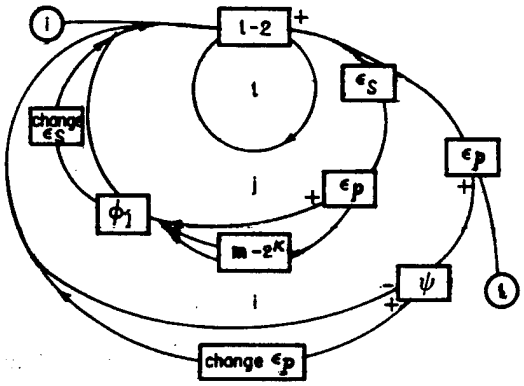

FIG. 6.5

Each of the decisions indicated by an alternative box is, of course, executed by a conditional transfer (order 75).

We are now ready to make a few decisions about the quantities we must store and then we can draw a more complete flow diagram showing what quantities are to be stored where in each operation box.

First. Since we plan to vary the initial conditions until we get the desired free stream velocity, we must save not only $\left(q_s\right)_i$ but $\left(q_s\right)_{i-1}$. However, five figures will suffice so we may put them both in one accumulator, say D1. Furthermore both q_s's are positive and therefore we may store ϵ_s in D1 also.

Second. The other quantity we need in adjusting the initial conditions is $\left(q_1\right)_{i-1}$. This we may put in D2.

Third. Since we plan to take a last step backwards of magnitude Δt to be determined by interpolation with respect to ϕ_j we must store ϕ_j after each step. However we only need its sign until it changes sign at which time it is small; it will therefore suffice to store $10^n \phi_j$ where n is suitably chosen. We shall store $-10^3 \frac{\Delta t}{2}$ in the same accumulator which we may call D3.

Fourth. The quantities 1-2 and ϵ_p being two or one digit numbers, we may put them in accumulator 6(10, 9) and 6(8, 7) respectively.

With these remarks about the bound variables, the following flow diagram (figure 6.6) will be clear. The boxes marked # are either substitution boxes which explain where the induction variables are changed or assertion boxes which state the conditions for leaving the alternative box by the unusual route.

A clear notion of the quantities held in each accumulator during each constancy interval (interval where no change in storage occurs) may be obtained from the storage table 6.I.

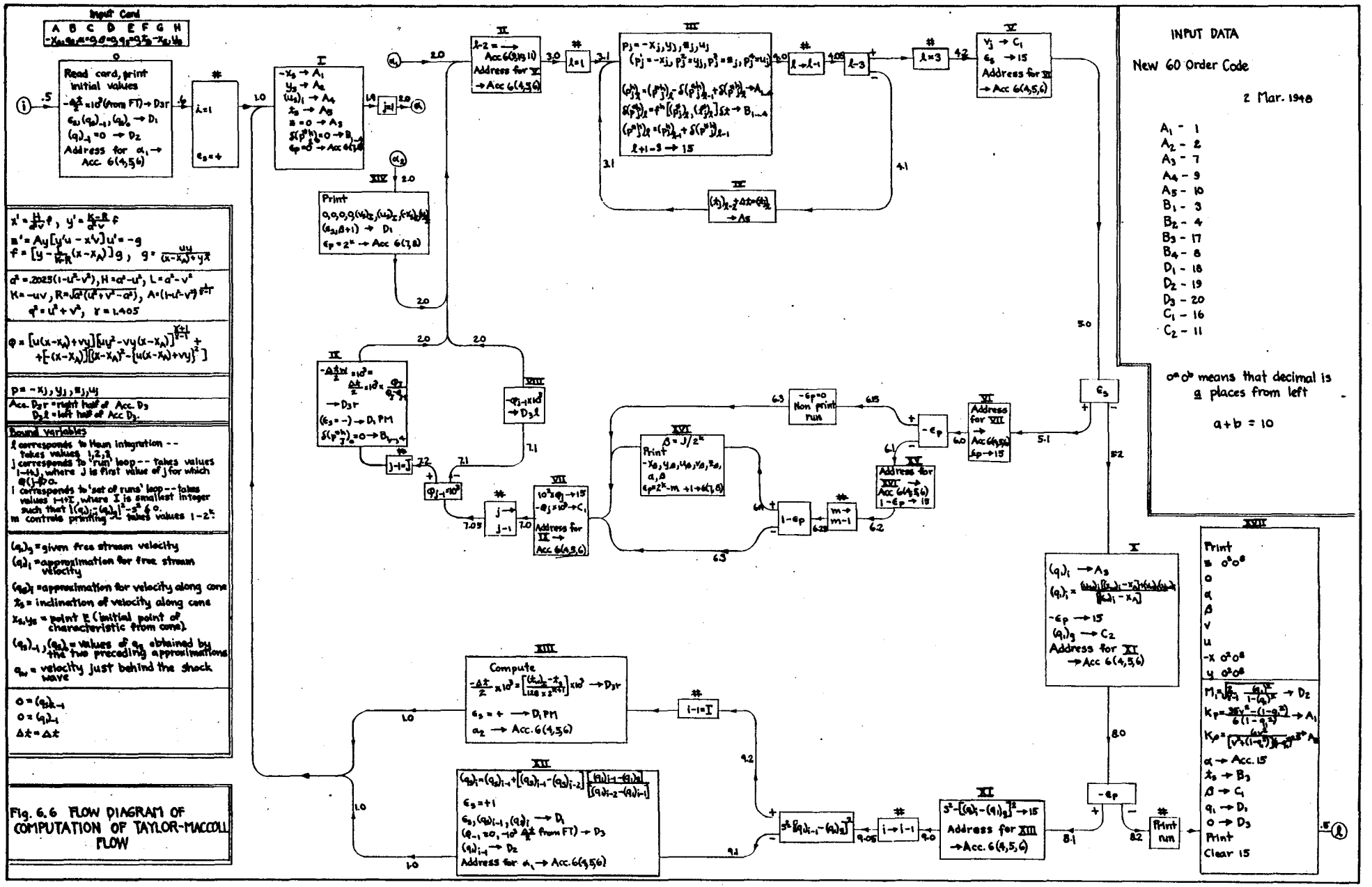
Input Card
A B C D E F G H
INPUT DATA
New 60 Order Code
2 Mar. 1948
A1 - 1
A2 - 2
A3 - 7
A4 - 9
A5 - 10
B1 - 3
B2 - 4
B3 - 17
B4 - 8
D1 - 18
D2 - 19
D3 - 20
C1 - 16
C2 - 11
o^a o^b means that decimal is a places from left
a+b = 10
Read card, print initial values
Address for α1 → Acc. 6(4,5,4)
Address for V → Acc 6(4,5,6)
Address for VII → Acc 6(4,5,4) fp→15
Address for XV → Acc 6(4,5,4) 1-εp → 18
Address for XIII → Acc. 6(4,5,6)
Address for XI → Acc 6(4,5,4)
Address for II → Acc 6(4,5,6)
Address for X → Acc 6(4,5,4)
Print run
Clear 15
Fig. 6.6 FLOW DIAGRAM OF COMPUTATION OF TAYLOR-MACCOLL FLOW

TABLE 6.I STORAGE TABLE FOR COMPUTATION OF TAYLOR-MACCOLL FLOW

C.I.	Print ACC. 1 A1	Print ACC. 2 A2	ACC. 3 B1	ACC. 4 B2	ACC. 6 (6,5,4)	ACC. 6 (8,7)	ACC. 6 (11,10,9)	ACC. 7 A3	ACC. 8 B4	ACC. 9 A4	ACC. 10 A5	ACC.11 C2	Print ACC. 15	Print ACC. 16 C1	Print ACC. 17 B3	Print ACC. 18 D1	Print ACC. 19 D2	Print ACC. 20 D3
.5	K_p	K_ρ	—	—	—	—	—	—	—				0	β	t_s	q_1	M_1	0
.6	$-x_A$	q_g			α										$(q_s)_o$	ϵ_s, $(q_s)_{-1}$, $(q_s)_o$	$(q_1)_{-1} = 0$	$-\frac{\Delta t}{2} \cdot 10^3$ (rt)
1.0																ϵ_s, $(q_s)_{i-1}$, $(q_s)_i$	$(q_1)_{i-1}$	$-\frac{\Delta t}{2} \cdot 10^3$ (rt)
2.0	$-x_s$	y_s	$\delta(-x_{j-1})_o = 0$	$\delta(y_{j-1})_o = 0$		ϵ_p		$z = 0$	$\delta(u_{j-1})_o = 0$	$(u_s)_1$	t_s				$\delta(z_{j-1})_o = 0$	(ϵ_s, $\beta + 1$)		$-\phi_j \cdot 10^3$ (left) and $-\frac{\Delta tw}{2} \cdot 10^3$ (rt)
3.0					Address for V		1-3 = -2											
3.1							1-3				$(t_j)_{i-1} + \Delta t = (t_j)_{i+1}$							
4.0	$(-x_j)_1$	$(y_j)_1$	$\delta(-x_j)_1$	$\delta(y_j)_1$			1+1-3	$(z_j)_1$	$\delta(u_j)_1$	$(u_j)_1$					$\delta(z_j)_1$			
4.05	$(-x_j)_{i-1}$	$(y_j)_{i-1}$	$\delta(-x_j)_{i-1}$	$\delta(y_j)_{i-1}$			1-3	$(z_j)_{i-1}$	$\delta(u_j)_{i-1}$	$(u_j)_{i-1}$					$\delta(z_j)_{i-1}$			
4.1							1-3											
4.2	$(-x_j)_2$	$(y_j)_2$	$\delta(-x_j)_2$	$\delta(y_j)_2$			0	$(z_j)_2$	$\delta(u_j)_2$	$(u_j)_2$					$\delta(z_j)_2$			
5.0			—	—	Address for VI				—				ϵ_s	v_j	—			
5.1																$\epsilon_s = +1$		
6.0					Address for VII								$-\epsilon_p$					
6.1						ϵ_p												
6.2					Address for XVI	2^{k-m-1}							$1 - \epsilon_p$					
6.25						2^{k-m}												
6.4						1												
6.15						0												
6.3						ϵ_p												
7.0					Address for IX								$\phi_j \cdot 10^3$	$-\phi_j \cdot 10^3$				
7.05													$\phi_{j-1} \cdot 10^3$	$-\phi_{j-1} \cdot 10^3$				
7.1													—	$-\phi_j \cdot 10^3$				
7.2														$-\phi_j \cdot 10^3$				
5.2																$\epsilon_s = -1$		
8.0					Address for XI			$(q_1)_i$				$(q_1)_g$	$-\epsilon_p$					
8.1						$\epsilon_p = 0$												
9.0					Address for XIII								$S^2 - \left[(q_1)_1 - (q_1)_g\right]^2$					
9.05								$(q_1)_{i-1}$					$S^2 - \left[(q_1)_{i-1} - (q_1)_g\right]^2$					
9.1								$(q_1)_{i-1}$					$S^2 - \left[(q_1)_{i-1} - (q_1)_g\right]^2$					
9.2								$(q_1)_I$					$S^2 - \left[(q_1)_I - (q_1)_g\right]^2$					
8.2					$\epsilon_p = 1$													

SECTION VII

CODING THE OPERATION BOXES

It is now a simple matter to consider the operation boxes, one at a time, and write the succession of orders which will cause the machine to carry out the computations described in that operation box. For the convenience of the reader we have written these orders both by name and by number for one example. (See Tables 7.I and 7.II) The reader can learn the code by making a table twenty columns wide and translating the code numbers into the movement of numbers among these columns.

Finally, it is advisable to prepare a test run by giving all the variables numerical values and carrying out one typical computation for testing occasionally that the Eniac is operating properly. Such a test run has been computed for this problem but it does not seem necessary to include it in this report.

SECTION VIII

MODIFIED ENIAC

Two additions to the ENIAC have been ordered which will greatly increase its memory and flexibility. The first is called the converter. It is a device for converting any two-digit number to a program pulse in one add-time. This will make available a hundred pulses for orders instead of sixty and will, in addition, free the master programmer for other uses. The converter has already been delivered and has already been used to modify the sixty-order code in such a way as to save space in the function tables for shifts (i.e. there are 19 two-digit shift orders instead of one four-digit shift order) and to make it possible to take orders from cards as well as from the function tables.

We shall not describe this modified code in detail since there will be a better code available by the time this report is available. The better code will use not only the converter but the other addition called the register.

The register is really a set of one hundred ten-digit registers with three operations: it can clear its old argument and receive a new argument in one add-time; it can receive a number, clear that register designated by its current argument and send the number to that register, increasing its argument by one; it can send out (and also keep) the number designated by its current argument. The latter two operations consume three add-times.

With the converter and register it will be possible to achieve among others the following results which will be incorporated in a new code to be described when completed:

1. Increase the speed of the ENIAC considerably
2. Increase the efficiency of the orders; that is, the amount of computation per unit space in the function tables.
3. Store orders in the register and thereby make it possible to control computations with cards and to modify orders for inductive processes.
4. Use the additional memory to solve problems involving more numbers, e.g., solve a system of thirty ordinary non-linear differential equations.
5. Use the card control to facilitate testing and demonstrating the ENIAC.

A first form of this new code should soon be available and the ENIAC should be ready to use this code by December.

TABLE 7.I INPUT DATA — AXIAL FLOW — NEW 60 ORDER CODE

F.T. I

	00	Rd	AB	2ℓ	11ℓ	1ℓ	GH	
Seq	01	18ℓ	11ℓ	19ℓ	18↔20	EF	18ℓ	
0	02	11ℓ	11ℓ	CD	16ℓ	11ℓ	Pr	Initial
	03	C	N4D	04	86	8ℓ	F.T.	sequence
	04	Sh	95	18ℓ	18↔20	17ℓ	18ℓ	
L	05	19ℓ	N6D	01	80	09	6R6	
	06							
	07							Seq III
	08							
⌜	09	EF	10ℓ	18ℓ	Sh	05	9ℓ	Compute
Seq	10	10ℓ	11ℓ	N6D	01	Sh′	9ℓ	u_{n+1}
I	11	16ℓ	13ℓ	10ℓ	X	√	Sh	
	12	91	7ℓ	9ℓ	÷	9ℓ	GH	
	13	2ℓ	11ℓ	1ℓ	3ℓ	4ℓ	1ℓ	
L	14	8ℓ	17ℓ	6(7,8)	M	6(7,8)	C.T.	
	15							
	16							
	17							
⌜	18	N2D	02	M	6(9,10,11)	C	N6D	
Seq II	19	02	01	00	6R6	C	C	
	20	9ℓ	11ℓ	10ℓ	X	Sh	01	CT on
Seq	21	16ℓ	N6D	20	10	22	6R6	ϵ_s - last step
V	22	10ℓ	C.T.	C	7ℓ	14ℓ	AB	
	23	C	1ℓ	M	11ℓ	Sh	91	
X	24	7ℓ	16ℓ	11ℓ	2ℓ	X	÷	Compute
	25	Sh	01	9ℓ	7ℓ	J15	8ℓ	q_{n+1}
	26	AB	11ℓ	N6D	23	70	28	CT on $-\epsilon_p$
	27	6R6	C	C	C	C	C	
	28	6(7,8)	M	C.T.	C	16ℓ	17ℓ	
XVII	29	N6D	10	Sh	05	M	16ℓ	
	30	16ℓ	2ℓ	Sh	91	18ℓ	18↔20	
	31	1ℓ	Sh	91	19ℓ	9ℓ	18ℓ	
	32	19ℓ	Sh	01	1ℓ	2ℓ	CD	
	33	C	11ℓ	Pr	C	17ℓ	16ℓ	
	34	7ℓ	3ℓ	7ℓ	11ℓ	7ℓ	X	
	35	16ℓ	12ℓ	M	D5	Sh	91	
	36	7ℓ	N4D	04	67	8ℓ	F.T.	
	37	Sh	99	11ℓ	11ℓ	12ℓ	X	
	38	Sh	98	÷	√	9ℓ	16ℓ	
	39	11ℓ	16ℓ	X	19ℓ	7ℓ	Sh	
	40	91	11ℓ	11ℓ	Sh	98	√	

F.T. II

	00	1ℓ	33ℓ	1ℓ	2ℓ	4ℓ	2ℓ	
	01	7ℓ	17ℓ	7ℓ	9ℓ	8ℓ	9ℓ	
	02	9ℓ	11ℓ	10ℓ	X	Sh	01	
	03	16ℓ	11ℓ	X	M	14ℓ	16ℓ	
	04	11ℓ	16ℓ	X	M	16ℓ	16ℓ	
L	05	14ℓ	D5	11ℓ	N4D	20	25	
	06	Sh′	94	12ℓ	X	11ℓ	14ℓ	
	07	11ℓ	14ℓ	14ℓ	16ℓ	M	X	
	08	√	Sh	91	5ℓ	C	16ℓ	
⌜	09	11ℓ	16ℓ	10ℓ	X	11ℓ	9ℓ	Integration
	10	X	13ℓ	7ℓ	11ℓ	13ℓ	7ℓ	Step
	11	5ℓ	12ℓ	14ℓ	Sh	9ℓ	÷	
	12	14ℓ	11ℓ	5ℓ	9ℓ	11ℓ	12ℓ	Heun
	13	9ℓ	10ℓ	X	X	9ℓ	M	
	14	12ℓ	11ℓ	9ℓ	5ℓ	11ℓ	12ℓ	
	15	Sh	91	÷	13ℓ	12ℓ	7ℓ	
	16	13ℓ	12ℓ	16ℓ	÷	16ℓ	11ℓ	
	17	7ℓ	AB	C	1ℓ	M	11ℓ	
	18	Sh	91	13ℓ	2ℓ	11ℓ	12ℓ	
	19	5ℓ	7ℓ	2ℓ	10ℓ	X	7ℓ	
	20	9ℓ	12ℓ	5ℓ	9ℓ	12ℓ	X	
	21	÷	5ℓ	2ℓ	7ℓ	11ℓ	2ℓ	
	22	AB	C	11ℓ	M	1ℓ	11ℓ	
	23	16ℓ	13ℓ	12ℓ	16ℓ	13ℓ	X	
	24	11ℓ	2ℓ	Sh	91	11ℓ	11ℓ	
	25	5ℓ	X	Sh	91	11ℓ	14ℓ	
	26	X	Sh	01	M	14ℓ	9ℓ	
	27	X	Sh	01	9ℓ	16ℓ	11ℓ	
	28	10ℓ	X	Sh	01	11ℓ	11ℓ	
	29	X	13ℓ	16ℓ	11ℓ	16ℓ	X	
	30	M	D5	11ℓ	5ℓ	12ℓ	11ℓ	
	31	Sh	92	√	Sh	01	13ℓ	
	32	12ℓ	5ℓ	13ℓ	X	X	11ℓ	
	33	2ℓ	X	12ℓ	16ℓ	11ℓ	12ℓ	
	34	16ℓ	10ℓ	X	Sh	01	12ℓ	
L	35	11ℓ	13ℓ	14ℓ	11ℓ	13ℓ	14ℓ	
	36	12ℓ	X	13ℓ	14ℓ	12ℓ	11ℓ	
⌜	37	14ℓ	12ℓ	11ℓ	9ℓ	X	13ℓ	
	38	16ℓ	12ℓ	11ℓ	16ℓ	13ℓ	11ℓ	
	39	12ℓ	X	Sh	01	12ℓ	18↔20	
	40	10ℓ	Sh	05	Sh	91	M	

F.T. III

	00		Initial		Pulse			
⌜	01	N6D	20	92	02	6R6	C	
	02	6(7,8)	M	C.T.	C	N6D	16	CT on $-\epsilon_p$
⌜	03	02	04	6R6	C	C	C	
	04	N2D	01	M	6(7,8)	M	C.T.	CT on $-(\epsilon_{p-1})$
L	05	N4D	02	09	6R3	C	C	
	06							
	07							
	08							
⌜	09	N6D	22	B2	10	6R6	C	
	10	AB	C	1ℓ	M	11ℓ	14ℓ	
	11	14ℓ	11ℓ	11ℓ	X	B1	9ℓ	
	12	X	13ℓ	16ℓ	11ℓ	2ℓ	X	
	13	17ℓ	17ℓ	11ℓ	17ℓ	X	M	
	14	8ℓ	5ℓ	2ℓ	11ℓ	2ℓ	X	
	15	4ℓ	4ℓ	5ℓ	11ℓ	14ℓ	X	
	16	5ℓ	14ℓ	M	11ℓ	2ℓ	X	
	17	11ℓ	16ℓ	X	13ℓ	4ℓ	11ℓ	
	18	9ℓ	X	11ℓ	17ℓ	X	12ℓ	
	19	N4D	04	B8	B1	F.T.	Sh	
	20	94	11ℓ	11ℓ	12ℓ	X	Sh	
	21	01	M	5ℓ	Sh′	94	C	
L	22	12ℓ	16ℓ	16ℓ	M	C.T.	C	
⌜	23	3ℓ	4ℓ	B1	17ℓ	18↔20	18ℓ	
	24	Sh	05	Sh	95	16ℓ	18ℓ	
L	25	18↔20	N4D	00	18	6R3	C	
	26							
	27							
⌜	28	7ℓ	3ℓ	18↔20	18ℓ	Sh	05	
	29	Sh	92	11ℓ	16ℓ	Sh	93	Compute $-\frac{\Delta s_w}{2}$
	30	12ℓ	18ℓ	Sh	95	Sh	02	
	31	M	12ℓ	7ℓ	12ℓ	X	÷	
	32	M	Sh	91	18ℓ	18↔20	18ℓ	
	33	M	D5	M	18ℓ	3ℓ	7ℓ	
	34	3ℓ	4ℓ	B1	17ℓ	N4D	00	
L	35	18	6R3	C	C	C	C	
	36							
⌜	37	N6D	25	12	38	6R6	C	
	38	7ℓ	M	11ℓ	11ℓ	11ℓ	X	CT on
	39	M	B1	C.T.	C	18ℓ	Sh	$\epsilon^2-(q_{n-1}-q_n)^2$
	40	95	Sh	05	D5	10ℓ	18ℓ	

TABLE 7.I (Continued)

F.T. I

41	Sh	01	X	X	81	8t			
42	X	X	51	161	18↔20	3t			
43	181	4t	191	11t	31	N2D			
44	35	Sh′	94	12t	111	19t			
45	X	121	3t	Sh	94	M	III		
46	12t	41	N2D	60	Sh′	9.:			
47	12t	111	5t	X	Sh	91			
48	71	9t	÷	11	14t	3t			
49	111	8t	X	Sh	91	71			
50	N2D	60	Sh′	94	12t	111	L		
51	14t	X	÷	21	EF	171	IV		
L 52	CD	161	11t	Pr	C	Halt			
53									
54									
Γ 55	CD	161	11t	141	9t	111			
XIV 56	10t	X	Sh	01	171	18↔20			
57	18t	81	2t	Sh	91	181	First		
58	18↔20	9t	101	1t	Sh	91	Print		
59	191	11	21	14t	Pr	C			
60	8t	181	18↔20	GH	21	11t	Γ		
61	11	N2D	10	Sh	05	16t	Seq		
62	181	N2D	2^k	(7,8)	C	81	XV		
L 63	171	N4D	00	18	6R3	C			

F.T. II

41	111	18↔20	12t	X	Sh	01			
42	121	7t	M	7t	121	71			
43	12t	171	9t	X	Sh	01			
44	12.1	4t	M	2t	12t	21	III		
45	12t	81	14t	X	Sh	01			
46	121	3t	M	11	12t	11			
47	12t	31	5t	M	X	Sh			
48	91	121	8t	M	161	12t			
49	91	12t	81	N2D	01	6(3,10,11)	L		
50	C.T.	C	11t	11t	Sh	93			
51	10t	101	N4D	01	09	6R3	Γ		
52							XIII		
53									
54									
55									
56							L		
57									
58									
59									
60	181	81	2t	Sh	91	181			
61	18↔20	181	51	1t	Sh	91			
62	191	9t	181	161	171	81			
63	161	7t	Sh	91	11	21			
64	CD	C	11t	Pr	141	N2D			
65	10	Sh	05	16t	141	5t			
66	181	18↔20	181	Sh	01	21			
67	17t	161	19t	Sh	01	11			
68	14t	181	N2D	2^k	(7,8)	C			
L 69	N4D	02	09	6R3	C	C			
70									
90	C	C	C	XX	XX	XX	$-\frac{p_2}{p_1}\times 10^{-3}$		
91							$2/(\gamma-1)$		
92							$(\gamma+1)/(\gamma-1)$		

F.T. III

41	Sh	05	DS	161	10t	M			
42	181	121	AB	111	7t	81			
43	8t	M	12t	Sh	91	71			
44	11t	M	8t	÷	111	121	III		
45	X	Sh	01	181	Sh	95			
46	181	181	At	191	18↔20	N4D			
47	04	B6	81	F.T.	Sh	95			
48	181	18↔20	N6D	01	00	09			
49	6R6	C	C	C	C	C	L		
50									
51	18↔20	EF	M	10t	101	N2D	Γ		
52	2^{k+1}	Sh′	92	12t	111	N4D	XIII		
53	12	80	Sh′	94	12t	X			
54	71	10t	÷	Sh	94	M			
55	181	18↔20	18t	DS	181	N6D			
56	05	50	09	6R6	C	C	L		

TABLE 7.II INPUT DATA — AXIAL FLOW — NEW 60 ORDER CODE

1 Mar 1948

		F.T. I							F.T. II							F.T. III							
																		Initial		Pulse			
L	00	62	80	02	44	01	83		00	31	33	01	32	34	02		00						
SEQ 0	01	24	44	25	91	82	24		01	40	53	10	42	41	12	⌐VI	01	95	20	92	02	74	00	
	02	44	23	81	22	44	61	Initial	02	42	14	43	30	90	01		02	93	63	75	00	95	16	CT on $-\epsilon_p$
	03	00	94	04	86	11	72	Sequence	03	22	44	30	63	21	52	⌐	03	02	04	74	00	00	00	CT on $-(\epsilon_p - 1)$
	04	90	95	24	91	53	24		04	14	52	30	63	22	52	XV	04	70	01	63	93	63	75	
L	05	25	95	01	80	09	74		05	51	65	14	94	20	25	L	05	94	02	09	73	00	00	
	06								06	85	94	45	30	14	51		06							
	07								SEQ 07	44	21	51	52	63	30		07							
	08								08	60	90	91	05	00	52		08							
⌐	09	82	13	54	90	05	12	Compute	III 09	44	32	43	30	14	42	Integration	⌐	09	95	22	82	10	74	00	
SEQ I	10	43	14	70	01	85	92	u_{s_i}	10	30	20	40	14	50	10	Step		10	80	00	31	63	44	21	
	11	45	20	43	30	60	90		11	35	15	51	90	92	55		VII	11	51	14	44	30	11	42	
	12	91	10	42	55	12	83		12	21	44	05	42	14	45	Heun		12	30	20	52	14	32	30	
	13	02	44	01	03	04	10		13	12	43	30	30	42	63			13	23	53	14	53	30	63	
L	14	11	23	93	63	93	75		14	15	44	12	35	14	45			14	41	05	32	14	32	30	
	15								15	90	91	55	20	45	10			15	04	34	35	14	51	30	
	16								16	50	15	52	55	22	44			16	05	51	63	14	32	30	
	17								17	10	80	00	31	63	44			17	14	52	30	20	34	14	
⌐	18	70	02	63	92	00	95		18	90	91	20	32	14	45			18	42	30	14	53	30	15	
SEQ II	19	02	01	00	74	00	00		19	05	40	02	43	30	10			19	94	04	88	11	72	90	
⌐	20	42	14	43	30	90	01	CT on	20	42	15	35	12	45	30			20	94	44	14	45	30	90	
SEQ	21	22	95	20	10	22	74	ϵ_s - Last Step or not	21	55	05	32	10	44	02			21	01	63	35	85	94	00	
V	22	54	75	00	40	21	80		22	80	00	44	63	31	14		L	22	45	22	52	63	75	00	
⌐	23	00	31	63	44	90	91		23	52	20	45	22	50	30		⌐	23	03	04	11	23	91	54	
X	24	10	52	14	32	30	55	Compute	24	14	32	90	91	44	14		VIII	24	90	05	90	95	52	24	
	25	90	01	42	10	84	11	q_w	25	35	30	90	91	14	51		L	25	91	94	00	18	73	00	
	26	80	14	95	23	70	20	CT on $-\epsilon_p$	26	30	90	01	83	21	42			26							
	27	74	00	00	00	00	00		27	30	90	01	12	52	14			27							
⌐	28	93	63	75	00	52	23		28	43	30	90	01	14	44		⌐	28	40	03	91	54	90	05	
XVII	29	70	10	90	05	63	54		29	30	20	52	14	52	30		IX	29	90	92	14	52	90	93	
	30	22	32	90	91	24	91		30	63	65	14	35	15	44			30	15	54	90	95	90	02	Compute $-\Delta tw/2$
	31	31	90	91	25	42	24		31	90	92	60	90	01	20			31	63	45	10	45	30	55	
	32	51	90	01	01	02	81		32	45	05	50	30	30	14			32	63	90	91	24	91	54	
	33	00	44	61	00	53	22		33	32	30	15	52	14	45			33	63	65	63	24	33	10	
	34	40	03	40	14	40	30		34	22	43	30	90	01	15			34	03	04	11	23	94	00	
	35	15	45	63	65	90	91		35	44	20	51	14	50	21		L	35	18	73	00	00	00	00	
	36	10	94	04	87	11	72		36	45	30	20	51	15	44			36							
	37	90	94	44	14	45	30		37	21	45	14	42	30	20		⌐XI	37	95	25	12	38	74	00	CT on
	38	90	92	55	60	04	52		38	52	15	44	22	50	14			38	40	63	44	14	44	30	$\epsilon^2 - (q_{w_i} - q_g)^2$
	39	14	52	30	21	40	90		39	45	30	90	01	15	91			39	63	41	75	00	54	90	
	40	01	14	44	90	92	60		40	54	90	05	90	91	63		XII	40	95	90	05	65	13	54	
	41	90	01	30	30	11	41		41	14	91	45	30	90	01			41	90	05	65	24	43	63	

TABLE 7.II (Continued)

		F.T. I							F.T. II							F.T. III						
	42	30	30	05	24	91	33		42	15	53	63	40	45	10		42	54	15	80	14	40	11
	43	24	34	25	44	03	70		43	45	23	42	30	90	01		43	41	63	64	90	91	10
	44	35	85	94	45	14	51		44	15	34	63	32	45	02	XII	44	44	63	41	55	14	45
	45	30	15	33	90	94	63		III	45	45	04	51	30	90	01		45	30	90	01	54	90	95
	46	45	04	70	60	85	92			46	15	33	63	31	45	01		46	54	24	41	25	91	94
	47	45	14	35	30	90	91			47	45	03	35	63	30	90		47	04	86	11	72	90	95
	48	10	34	55	01	51	33			48	91	15	41	63	52	45		48	24	91	95	01	80	09
	49	14	41	30	90	91	10			49	12	45	11	70	01	92	L	49	74	00	00	00	00	00
	50	70	60	85	94	45	14		L	50	75	00	44	44	90	93		50						
	51	51	30	55	02	82	23		IV	51	43	13	94	01	00	73	f^*	51	91	82	63	43	13	70
L	52	81	22	44	61	00	71			52							XIII	52	2^{k+1}	85	92	45	14	94
	53									53								53	12	80	85	94	45	30
	54									54								54	10	43	55	90	94	63
⌜	55	81	22	44	21	42	14			55								55	24	91	54	65	24	95
XIV	56	43	30	90	01	23	91			56							L	56	05	50	09	74	00	00
	57	54	11	32	90	91	24	First		57														
	58	91	42	24	31	90	91	Print		58														
	59	25	01	02	51	61	00			59														
	60	41	24	91	83	02	44		⌜	60	54	11	32	90	91	24								
	61	01	70	10	90	05	52		SEQ	61	91	54	05	31	90	91								
	62	24	70	2^k	93	00	11		XVI	62	25	42	24	52	23	41								
L	63	23	94	00	18	73	00			63	22	40	90	91	01	02								
	64									64	81	00	44	61	21	70								
	65									65	10	90	05	52	21	35								
	66									66	24	91	54	90	01	02								
	67									67	53	22	64	90	01	01								
	68									68	51	24	70	2^k	93	00								
	69								L	69	94	02	09	73	00	00								
	70									70														
	71																							
	72																							
	73																							
	74																							
	75																							
	76																							
	77																							
	78									90	00	00	00	xx	xx	xx	$-\Delta t/2 \times 10^3 \beta$ side (send in arg. 86)							
	79									91							$2/(\gamma - 1)$							
	80									92							$(\gamma + 1)/(\gamma - 1)$							
	81																							
	82																							
	83																							

DISTRIBUTION LIST

No. of Copies	
4	ORDTB - Bal Sec
10	British, of interest to: Prof. D. R. Hartree Cambridge University Prof. J. W. Maccoll Cambridge University, England
4	Chief, Bureau of Ordnance Navy Dept Washington 25, D.C. Attn: Re3
2	Commanding Officer Naval Ordnance Laboratory White Oak Silver Spring 19, Maryland Of interest to Dr. Seeger Dr. F. J. Weyl
1	Commanding Officer Naval Ordnance Test Station China Lake, California Attn: Reports Unit Of interest to: Dr. A. L. Bennett
1	Commanding Officer Naval Proving Ground Dahlgren, Va.
1	Office of Naval Research Navy Dept. Washington 25, D.C. Attn: Armament Br. (NR 463)
1	Superintendent of Postgraduate School U. S. Naval Academy Annapolis, Md.
1	Director, Naval Research Laboratory Anacostia Station Washington 20, D.C.
1	Prof. R. W. Feller Cornell University Ithaca, N. Y.
1	Prof. W. Prager Brown University Providence 12, R. I.
1	Prof. G. F. Carrier Brown University Providence 12, R. I.

No. of Copies	
1	Prof. Stefan Bergman Harvard University Cambridge 38, Mass.
1	Dr. C. Ferrari Cornell Aeronautical Laboratory Buffalo, N. Y.
1	Mr. B. W. Augenstein North American Aviation Co. Inglewood, California
1	Bell Aircraft Co. Niagara Falls, N. Y. Attn: Mr. Paul Emmons
1	Dr. M. V. Morkovin University of Michigan Ann Arbor, Michigan
1	Mr. Edmund C. Berkeley Prudential Insurance Co. of America Newark, N. J.
1	Dr. Fred Bennett Ballistic Research Laboratory Aberdeen Proving Ground, Md.
1	Prof. Lipman Bers Syracuse University Syracuse 10, N. Y.
1	Prof. A. Gelbart Syracuse University Syracuse 10, N. Y.
5 + 10 abstracts	Commanding General, AMC Wright Patterson, Air Force Base Dayton, Ohio Attn: MCREOR - of interest to: Mr. G. Guderley
1	NACA Flight Propulsion Laboratory Cleveland, Ohio
1	NACA Ames Laboratory Moffett Field, California
2	NACA, Langley Field, Va. Of interest to: Mr. A. Ferri
1	Mr. Hans Kraft General Electric Co. Schenectady, N. Y.

DISTRIBUTION LIST (Continued)

No. of Copies	
1	Dr. Fritz John New York University New York 12, N. Y.
1	Prof. J. J. Stoker New York University New York 12, N. Y.
1	Prof. K. O. Friedrichs New York University New York, N. Y.
1	Dr. John Green University of Calif. at Los Angeles Los Angeles 24, California
1	Stanford University Dept. of Mathematics Palo Alto, California
1	Dr. N. Edmonson Applied Physics Laboratory Johns Hopkins University Silver Spring, Md.
1	Dr. U. Fano National Bureau of Standards Washington, D. C.
1	Dr. J. H. Curtiss National Bureau of Standards Washington, D. C.
1	Dr. A. E. Ruark Johns Hopkins University Baltimore, Md.
1	Dr. I. Schoenberg University of Pennsylvania Philadelphia, Pennsylvania
1	Dr. D. H. Lehmer University of California Berkeley 4, California
1	Dr. J. Machly Electronic Control Co. Philadelphia, Pa.
1	Prof. M. H. Martin University of Maryland College Park
1	NACA 1724 F. Street, N. W. Washington, D. C.

No. of Copies	
1	Dr. H. Liepman Guggenheim Aeronautical Laboratory California Institute of Technology Pasadena, California
1	Prof. C. C. Lin Mass. Institute of Technology Cambridge, Mass.
1	Gen. Paul Libessart 16 Rue Octave Faulielet, Paris, France Thru Military Attache
1	Dr. Ernest Newmann Gas Turbine Laboratory Mass. Institute of Technology Cambridge, Mass.
1	Mr. E. Nilson United Aircraft East Hartford, Conn.
1	Dr. Nicholas Metropolis Las Alamos Scientific Lab. Box 1663, Santa Fe, N. M.
1	Mr. J. Baker 419 Commonwealth Ave. Cambridge, Mass.
5	Prof. Howard Aiken Computation Laboratory Harvard University Cambridge 38, Mass.
1	Dr. E. A. Eckhardt Watson Computing Laboratory 612 West 116th St. New York 27, N. Y.
10	Office of Naval Research Math. Branch Washington 25, D. C. Attn: Code N424
5	Moore School of Electrical Engineering University of Pennslyvania Philadelphia, Pennsylvania
2	Operation Evaluation Group (OP 34 H8) Room 3827 Department of the Navy Washington 25, D.C. Attn: Mr. McGrew