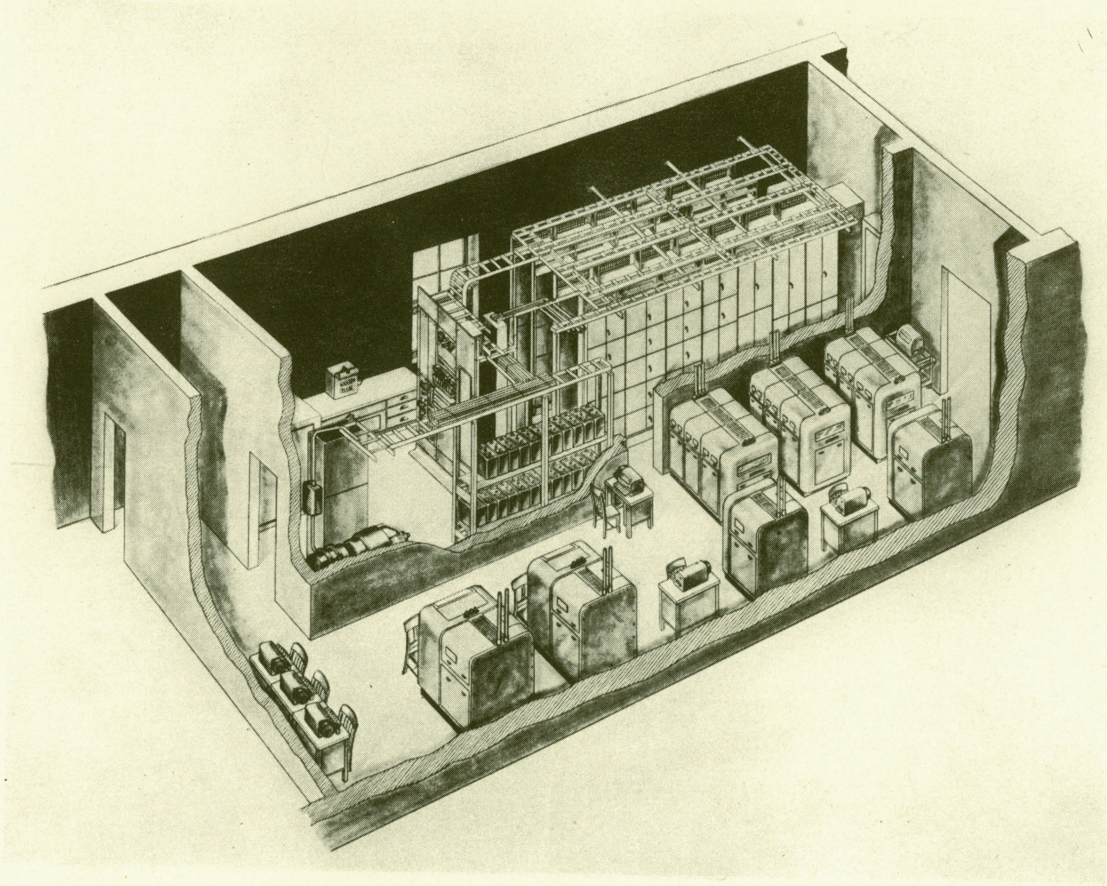

A Bell Telephone Laboratories' Computing Machine

A Bell Telephone Laboratories' Computing Machine—I

This is the first of two articles by Dr. ALT describing a Bell Telephone Laboratories' machine. The second article will treat machine control, operational characteristics, and problems for the machine.

1. Introduction

High-speed digital computing machines which have been developed during the past few years are of two kinds, electronic and electro-mechanical. In the latter class two families have come to prominence. One consists of the machines built by H. H. AIKEN at Harvard University,[1] the other of a series of machines built by the Bell Telephone Laboratories in New York. This article deals with the latter family, and specifically with its largest and youngest member, a machine which has been built in the past two years in duplicate, one unit for the National Advisory Committee for Aeronautics (Langley Field, Va.) and one for the Ordnance Department of the U. S. Army (Aberdeen Proving Ground, Md.).

Before entering upon a description of this machine, a brief enumeration of its "ancestors" may be in order. The basic idea underlying all machines in this family, that of using telephone switching equipment for computing purposes, was conceived several years before the outbreak of the last war by GEORGE R. STIBITZ, then on the staff of the Bell Telephone Laboratories. The Company decided to test the idea on a small scale by building, for its own use, a machine capable of performing addition, subtraction, multiplication and division of complex numbers. Under the impact of the war-time demand for large-scale computing, the Company next developed the "Relay Interpolator," a machine consisting mainly of about 500 telephone relays, together with some teletype equipment used for transmitting numbers into and out of the machine and for directing the operations. The next machine to be built was the "Ballistic Computer," containing about 1300 relays, more elaborate and more complex than the Relay Interpolator but still, like all the earlier machines, a special-purpose device, designed to carry out only a few special kinds of computations. Finally, in 1944 the Bell Telephone Laboratories undertook to develop the all-purpose computing system with which this article is concerned. This system differs from its predecessors in size—it contains over 9000 relays and about 50 pieces of teletype apparatus, covers a floor space of about 1000 square feet, and weighs about 10 tons—as well as in flexibility, generality of application, reliability, and ability to operate automatically without requiring the presence of human operators.

When the complete description of this machine is written, including instructions for the coding of problems and the operation of the machine, description of all the circuits involved, specifications for the components and instructions for their maintenance, it will fill hundreds of pages. This article will be limited for the most part to those aspects which are of interest to mathematicians in general, rather than to persons operating the machine. It will therefore merely attempt to give a first idea of the methods by which the machine solves computing problems, to describe its capabilities and its limitations, and to make a few comparisons between this machine and other

large-scale computing machines, notably the Harvard machines mentioned above, the punch-card machines of the IBM Corporation, the ENIAC,[2] and some of the projected electronic machines of the future.

The author has had the benefit of informal discussion with H. H. AIKEN, E. G. ANDREWS, J. VON NEUMANN, S. B. WILLIAMS, and many others, too numerous to mention, all of whom have greatly contributed to crystallizing the views set forth in this article.

2. Number Storage. The Bi-quinary System

a. Number Systems.—Every calculating machine stores numbers by setting some elements (such as wheels, shafts, vacuum tubes, or the like) in one of several possible (stable) states. This machine uses relays, which are capable of being maintained in either of two stable states: a relay may be operated ("up") or released ("down"). If operated, it closes a group of electric paths and opens another group; if released, it opens the former group and closes the latter. Each closed path may be used, if desired, to operate other relays.

Since each relay has two stable states, a natural way to apply relays to number storage would be to convert numbers from the decimal to the binary system and store them in the latter form. This method was not used. Instead, numbers are represented in a system called the "bi-quinary system." Each decimal digit is replaced by two digits, of which one (the "quinary" digit) has one of the values from 0 to 4 and the other (the "binary" digit) has one of the two values 0 or 5, such that the sum of the two values is equal to the value of the decimal digit which is to be represented. There are two relays to store each binary digit, and five relays to store each quinary digit; when a number is stored, exactly one relay in each group is operated. There are thus seven relays required for each decimal digit stored. This is a saving as compared with the most obvious system of using ten elements, one for each of the ten values of a decimal digit. (This latter system is in use, for instance, on the ENIAC.) It is, on the other hand, wasteful by comparison with binary system as well as with some other systems that have been devised. For instance, the recently developed "Relay Multipliers" of the IBM Corporation use the bi-quinary system, described above, with the modification that only one relay is used to store the binary digit (the value 0 being stored by having that relay released), and four relays are used to store the quinary digit (the value 0 being stored by having all four released). The Harvard "Mark II" translates every decimal digit into four binary digits and stores these on four elements; this system differs from the straight binary system in that it translates each decimal digit separately, whereas the binary system translates the entire number with all its digits into binary form. The system originally suggested by Stibitz consisted in representing a decimal digit of value d on four relays by expressing the number $d + 3$ in the binary system; this arrangement simplifies the design of the calculator. Various other systems of number representation have been under discussion.

b. Relative Merits of Various Systems.—The question of which system of representation is to be used comes up in the design of any computing machine which uses an inherently binary storage device, such as relay or vacuum tube, i.e., a device capable of maintaining two stable states. It does not exist, for example, in most desk-type machines and most punch-card

machines, which use counter wheels for number storage. All these latter machines use the usual decimal system.

The designers of the bi-quinary system acknowledge that their system requires more storage elements per digit stored than some other systems, but maintain that this drawback is more than compensated for by two advantages: no special equipment is needed to translate the numbers put into the machine from the decimal system to the system used in the machine, or to perform the opposite translation on the output of the machine (the translation between decimal and bi-quinary system is trivial); and the system can be made self-checking, so that a stored number cannot be changed into another number by some transient machine trouble. Checking is accomplished by means of a circuit which is closed when exactly one binary and exactly one quinary relay are "up" for every digit stored, and which is open under all other conditions.

These advantages are extremely important and are well worth the price of additional storage elements needed; but there are systems that would have maintained both advantages and saved at least some of the excessive storage equipment. For example, each decimal digit could be represented on a group of five relays, of which exactly two are operated when a digit is stored. There are exactly ten combinations of two out of five elements, and these can be made to correspond to the ten values of a decimal digit. (Combinations of three out of five elements can be used in the same way, and are, in fact, used in a few places in the machine which we are discussing.) A checking circuit can be devised which is closed only when exactly two relays are up, is open in every other case, and stops the machine when open. This gives the same degree of insurance against error as the bi-quinary system: an error will pass only when a failure of a relay is offset by another relay in the same group being falsely operated. Experience shows that the probability of such an occurrence is remote.

It is desirable to arrange a computing machine in such a way that it stops at once when any part of its equipment runs into trouble. Such an arrangement greatly facilitates the task of locating the source of trouble, a formidable task even under the most favorable circumstances, as anyone knows who has tried it. Our machine fulfills this requirement for the most part, although not entirely. For example, in the case of number storage, any "over-registration" (i.e., any case of more relays being operated than are needed for storing a number) will halt the machine instantly. But an under-registration will stop it only at the time when the number is being transferred into a register or when a stored number is called for. This is, however, one of a small number of cases in which the machine does not halt at once in case of trouble, and it has not been the cause of any undue difficulty in operating the machine.

c. Digital Capacity and the Floating Decimal Point.—As has been said above, each decimal digit of a number is stored on seven relays. The sign of a number, + or −, is stored on two relays, exactly one of which is operated whenever a number is stored. Negative numbers are represented by the minus sign and the absolute amount of the number, not by its complement as is the case on some other machines.

The standard length of numbers stored in the machine is seven significant decimal digits. For comparison, the length of numbers stored on some other

computing machines is as follows: desk machines, usually 8 or 10; standard IBM punch-card machines, up to 8; IBM Relay Multipliers, 6, and for certain operations, 12; H. H. Aiken's "Mark I", 23; "Mark II," 10; ENIAC, 10.

These figures are not strictly comparable, for at least two reasons. One is that, as machines become faster, they will presumably do longer problems in which the accumulated effect of rounding errors becomes more pronounced. Therefore, the faster a machine, the more digits should it provide. The other reason is that for two of the machines listed the numbers given refer to significant digits, while for all the other machines all numbers stored are limited to a given range of decimal positions. The ability of a machine to retain only significant digits and to discard the insignificant (zero) digits in front of a number is sometimes referred to as "floating decimal point."

The Bell Laboratories' machine was the first to use a floating decimal point. The only other existing machine which uses it is the "Mark II." In our machine it works as follows: each number is transformed into a decimal number with seven significant digits, of which the first is immediately to the right of the decimal point, multiplied by an appropriate power of ten. The machine stores the sign of the number, the first seven decimal digits, and the exponent of the power of ten by which the decimal is to be multiplied. (This exponent is briefly called the "exponent of the number.") Exponents are limited to the range from -19 to $+19$.

The "exponent of a number" on our machine consists of a sign ($+$ or $-$), a first digit which is limited to the value 0 or 1, and a second digit between 0 and 9. The sign and tens digit are stored on one of two relays each, and the units digit on two out of seven relays in the bi-quinary system. Together with the 49 relays needed to store the seven significant digits of the number and the two relays needed to store its sign, a total of 62 relays are required to store one number. These, together with a few operating relays, form what is called a *register*. The machine contains 44 registers, of which 30 serve only to store numbers, while the remaining 14 perform various additional functions.

3. Calculation

a. Addition and Subtraction.—For adding two numbers, three groups of relays designated A, B and C, respectively, are provided in the machine. Each group is capable of storing a ten-digit number in the bi-quinary system (the first of the ten digits being limited to values 0, 1, 9). They are called "groups" rather than "registers" because of the absence of sign and exponent. The groups are wired together in such a way that if two numbers are stored on the A and B groups, those relays in the C group which represent the sum of the two numbers are automatically operated.

The details of these wirings are not of interest for our present purpose. The only point that may be interesting is the handling of carries. Each digit provides for two adding circuits: the one mentioned before, and another one which gives a sum higher by 1 (or 0 instead of 9) and which is used in case there is an incoming carry of 1. The addition of the two numbers is done in two steps which are carried out simultaneously. One sets up one of two conditions—incoming carry 0 or 1—for each digit. The other step carries out the addition (i.e., operates 2 relays in each digit of the C group) by using

one of the two circuits in accordance with the incarry condition. In the former step, the carry conditions are determined from a three-way distinction regarding the sums of the two digits in the next-lower position, as follows: sum less than 9, out-carry 0; sum more than 9, out-carry 1; sum equal to 9, out-carry equal to in-carry.

In addition to the circuits for adding A and B and putting the sum on C, described above, there is a set of wires which will similarly operate those of the B-relays which represent the sum of two numbers stored on the A and C relays.

If the machine receives an order to add two (positive) numbers, it first compares their exponents. The number with the greater exponent (in case of equal exponents, the first of the two numbers) is transferred to the A group of relays. Of the ten digits for which this group is equipped, those from the third to the ninth correspond to the seven significant digits of a storing register. The number with the smaller exponent is transferred to the B relays, after being shifted to the right by a number of decimal places equal to the difference between the two exponents. That is, if the difference between the two exponents is 1, the number is placed onto the fourth to tenth place of the B relays; if the difference of exponents is 2, the first six significant digits of the number are placed onto the fifth to tenth place of the B relays, while the seventh significant place is discarded; etc. The machine then closes the adding circuit mentioned above, which places the sum of the two numbers onto the C relays. The first seven significant digits of the sum (which may be located in digits 3 to 9 or in digits 2 to 8 of the C relays) are then transferred to a register designated for this purpose by the operator, after having first been rounded in the seventh place.

It will be noted that nine places in the A, B, and C relays would have been sufficient to accomplish this operation, namely, the seven places of the larger addend, plus one place on the left for carries and one on the right for rounding. The tenth was added mainly for simplification of the circuits, especially in connection with division.

To subtract two numbers, the machine converts the one with the larger exponent into its complement and stores this on the A relays. The other number is put onto the B relays, shifted as needed, and the sum appears on the C relays. This is the complement of the desired result, and the machine converts it back into a true number. If the exponents are equal, the first of the two numbers is converted to a complement and put on the A relays, regardless of the relative size of the two numbers. If the first happens to be smaller, the result is represented by its absolute value, rather than by its complement, and there will be a digit "0" in the left-hand position when the two numbers are added. The presence of this digit gives the signal to the machine to omit converting the result back to its complement, and to reverse the sign of the answer. The rounding and transferring out of the result occurs as in addition.

If one or both numbers involved in an addition or subtraction are negative, the machine determines from the combination of sign whether the absolute amounts should be added or substracted, and proceeds accordingly.

All complements are taken with respect to 9999999, and an "end around carry" into the right-hand position is simulated on addition in order to effect the necessary correction by 1 in this position.

If the result of an addition or subtraction has one or more zeros in the left-hand position—this happens if the two addends have opposite sign and agree in the first few places—the machine will automatically shift all digits of the answer to the left until the first significant digit occupies the left-hand position of the storage register, and will correct the exponent of the answer accordingly. This feature, briefly called "zero shift," can be canceled at the discretion of the operator. The rounding of the answer can also be canceled if desired. If the operator has not elected to cancel either the rounding or the zero shift, the machine will automatically omit rounding in case a zero shift is performed. This is necessary for obvious reasons. If the answer of a computation is zero, the calculator reads out a minus sign, seven zero digits, and an exponent of -19 (the smallest possible exponent within the capacity of the machine); if the operator has specified that the "zero shift" be canceled, the exponent read out is that of the larger of the two addends, and the sign may be either $+$ or $-$ depending on the signs of the addends.

b. Multiplication, Division, Square Root.—Multiplication is performed by repeated addition. The multiplicand is entered on the A relays, and the C relays are initially set to hold the number 0. By means of the adding circuits described above, the machine now adds A to C, records the sum on B and clears the C relays. It then adds A to B, records the result on C and clears B. It keeps on adding alternatingly A + C onto B and A + B onto C, the number of such additions being determined by the size of the first multiplier digit. After one digit of the multiplier is exhausted, the machine goes on to the next, and so on.

The reader will have noticed in the description of addition that the machine provides for two adding circuits, A + B = C and A + C = B, of which only the first was used in addition and subtraction. The purpose of the second is to make possible the multiplication process just described. If a multiplier digit is 5 or more, the machine will perform repeated subtractions instead of additions (the number of subtractions being equal to the complement of the multiplier digit) supplemented by one addition in the next-higher decimal place.

Throughout those successive additions at least eight significant digits are retained. When the multiplication process is finished, any initial zeros will be shifted off just as in addition. As a rule there is at most one initial zero, but there may be more than one if one of the two factors had initial zeros. This condition may come about as a result of a previous operation in which the zero shift was canceled. If no zero shift is necessary, the answer will be rounded in the seventh place. This is accomplished automatically by adding a 5 in the eighth place and then dropping that place. Zero shift, rounding, or both may be canceled at the discretion of the operator.

After the multiplication is finished, the product is transferred to a register designated by the operator. The sign of the product is supplied by the machine in accordance with the signs of the two factors in the usual manner. The exponent of the product is determined by the machine as the sum of the exponents of the factors, corrected by the amount of any zero shift that may have been performed.

The result of the multiplication is accurate to seven significant digits (provided both factors had this accuracy). There is no possibility of utilizing

the subsequent digits of the product. It appears that it would have been possible and desirable to design the machine in such a way that fourteen digits of the product could be used if desired.

Just as multiplication is accomplished by repeated addition, so division and extraction of square roots are accomplished by repeated subtraction. We shall omit a detailed description of these processes here, since the general principles on which they are based are well enough exemplified by the case of multiplication. It may be mentioned that both processes are carried in the machine to seven significant digits, and that there is no possibility of utilizing the remainder after division or square root.

4. Input and Output

a. Input Systems.—The "input" for a computational problem (i.e., the information available before the start of the computation) consists of two kinds of elements: numbers, and "orders." The latter are the instructions as to the operations that are to be performed on the numbers.

The "output" or result of computation consists of numbers only. It has been proposed—facetiously by some, in earnest by others—to build a "thinking" machine whose output would be orders rather than numbers. From a small input such a machine would automatically produce the system of orders required for the solution of a complex problem on a computing machine. Perhaps such a machine will be developed in the future, but all existing machines produce only numbers.

An important characteristic of any computing machine is the method by which orders and numbers are transferred ("read") into the machine. It seems that most designers of computing machines agree that the time required for the input of one number should be of the same order of magnitude as the time required for an arithmetic operation, or perhaps longer by one order of magnitude. If the input time is outside these limits, an unbalanced machine results, because one component of the machine spends an unduly large proportion of its time waiting for another component. The input time for an order should be no longer than the time required to carry out the order. Punch-card machines are faster than desk machines and therefore need a faster input method. Such a method is realized in the feeding of cards, which proceeds at speeds in the neighborhood of $\frac{1}{2}$ second per card (the exact speed varying with the type of machine), and is well in balance with the computing speed of these machines. Note that the speed is $\frac{1}{2}$ second per card, not per number, and that one card usually holds several numbers. The Harvard machines use punched cards for most input data, and perforated tape for the rest. Again, the speed of input is well in balance with the computing speed. The ENIAC uses a card feed like that of standard punch-card machines. This is far out of balance with the computing speed of the machine; a computation is carried out in a few milliseconds, whereas reading of numbers requires $\frac{1}{2}$ second.

The Bell Laboratories' machine uses teletype devices for reading and perforating of paper tape (with small modifications) for its final stage of input, both for numbers and orders. Its speed turns out to be approximately in balance with the computing speed of the machine, or perhaps slightly too slow. It takes about two seconds to read and transfer into the

machine either a (seven-digit) number or an order of average length. Additions and multiplications, on the other hand, can be carried out in 0.3 and 1.0 seconds, respectively. Thus, the input speed of numbers is adequate while that of orders is slightly too slow.

b. The Tape-Reading Process.—The tape used is $\frac{7}{8}$ inch wide and has room for six holes across the tape. These six holes are designated by the numbers 0 to 5. Decimal digits are represented on tape by combinations of three holes not including the hole numbered 0. There are exactly ten possible combinations of three out of the five holes 1 to 5, and these ten combinations represent the ten possible values of a decimal digit. Orders to the machine are written in a code consisting of the first twenty letters of the alphabet and these are represented on tape by all combinations of three holes out of six. (The representation of the first ten letters is identical with that of the ten decimal digits.) Some combinations of two or four holes are used for checking purposes; other two-hole combinations are used to represent the sign and exponent of a number; some combinations of four holes and most one- or five-hole combinations are used to designate special signals to the machine. Any such combination of between one and five holes, all lying in a row across the tape, is called a "code." Tapes are fed into standard teletype devices called "transmitters," which move the tapes and read one code at a time. The reading time is not determined by the mechanical construction of the transmitter but by the amount of work the machine does between codes. The transmitter moves the tape the distance between two codes; this requires about .05 seconds. Then the tape is stopped, a set of six sensing fingers comes up to the tape, and those fingers which find holes in the tape penetrate them, thereby closing certain electrical contacts and breaking others. The electric paths thus closed set up relays corresponding to the holes in the tapes. There follows a long chain of relay operations, until finally a signal is sent to the tape transmitter which causes it to advance the tape in preparation for reading the next code. For the purpose of this paper it is not necessary to enumerate all the steps performed, but it is important to realize that they happen serially, each step closing the electric path which will operate one or more relays in the next step. This chain of steps (many of which are introduced for the sole purpose of checking and safeguarding the proper functioning of the machine) is time-consuming and raises the total time elapsed between code readings to an average of .2 seconds. The exact reading time varies within wide limits, depending on the codes and on the context in which they come. There is nothing in the machine to synchronize the operations of its various parts; each relay operation engenders the next, just as fast as relays will click.

As a rule, eleven codes are needed to represent a seven-digit number with sign and exponent on tape. Orders vary in length, but an average is nine codes per order. Thus, the average reading time for both numbers and orders is in the neighborhood of 2 seconds.

c. Output.—The numerical output of the machine is recorded by perforating numbers on a tape and/or by printing on a sheet of paper. Both operations are performed by standard teletype devices called "reperforator" and "page printer" respectively.

When printing, appropriate orders must be given to the machine to determine the arrangement of numbers on a page, the number of decimal

places to be printed, etc. It is a little unfortunate that these orders have to be interspersed with the computing program, because this detracts from the general applicability of many programs. It might have been better to concentrate all the printing work in a separate auxiliary machine that would operate from tapes put out by the main machine. As it is, occasionally a program will have to be changed merely because the magnitude of the numbers involved changes and the printing orders have to be modified to accommodate more decimal places or the like.

The tape on which the results of the computation are perforated runs through a transmitter in which the numbers previously perforated may be read back into the machine if this is desired. This arrangement makes it possible to store intermediate results for later use in the same computation. In practice it turns out that this storage function is far more important than the function of holding answers. In fact, the machine could be improved by having more storage tapes. It seems to be the consensus of opinion that three storage tapes is a desirable minimum for any large computing machine.

d. Distribution of Input among Tapes.—The tapes which contain the input for any problem are classified into three groups called Problem, Routine, and Table Tapes. As a rule a problem may use up to five routine tapes (designated A to E), up to six table tapes (designated A to F), and one problem tape.

When the tape input system for this machine was designed, it was intended that the routine tapes should contain all the orders, the table tapes should contain numerical information of a general nature, comparable to function tables used in manual computing, and the problem tape should contain numerical information specific to the problem being solved (i.e., initial conditions for differential equations, coefficients of a system of algebraic equations, etc.). Gradually, as the machine developed, the distinction between the kinds of tapes became blurred. As matters stand now, most orders are contained in the routine tapes, but some may be put into the problem tape; some numerical information may be put into routine tapes, and most of it is distributed between the table and problem tapes at will, to fit the needs of a particular problem. Certain information remains restricted to the problem tape, especially that which is to be printed as the heading of the answer sheet; and even this last remaining limitation has proved to be a drawback rather than a useful distinction between tapes.

As a result of the experience gained in the operation of this machine, it appears that the problem tape is an unnecessary complication of the machine, and that its function had better be distributed between the other two types of tapes. The distinction between order tapes and numerical tapes, on the other hand, appears useful. Most of the projected future machines contemplate the use of a single input tape only, and this seems to entail the considerable drawback that an entire new tape has to be made up for each problem to be solved. A division of the input as indicated makes it possible to combine a single order tape with many different numerical tapes, or (less frequently) one numerical tape with various order tapes, and thus to work a large number of problems without having to produce a large amount of new tape. Whether or not it is desirable to have several numerical input tapes, as on our machine, is partly a question of the price paid for them. They can be used to good advantage by putting the numerical information in groups that

are likely to be changed or left undisturbed simultaneously, so that groups of problems can be handled by changing only a single short tape without disturbing the others. For instance, in solving differential equations, one tape may contain the length of an integration step, the desired accuracy and similar constants referring to the numerical process selected; another tape, the initial conditions; a third tape, certain parameters appearing in the equations; etc. As a rule, in going from one problem to the next only one group of numbers will be changed at a time, so that the changeover can be effected with a minimum amount of new tape. On our machine, moreover, the distribution of the numerical input among several tapes has the great advantage of reducing the time required to find any desired place in the tapes. This is of primary importance in this relatively slow machine, but will be a lesser consideration in future electronic machines.

e. Arrangement of Routine Tapes.—The routine tapes are usually made into closed loops. This arrangement was made because most computations, especially large-scale ones, involve repetitive elements. The transmitters into which the routine tapes are placed can move the tapes in one direction only. Occasionally open-end routine tapes are used in cases in which the course of a computation is straightforward without ever returning to an earlier sequence of orders.

Each routine tape is divided into sections. Six sections are provided for, and any larger number can be obtained by means of a relatively simple programming device. As a rule, the machine reads orders consecutively as they appear on the tape, carrying out each order as it is read. It is possible, however, to give an order which will cause the tape transmitter to "hunt" for the beginning of any desired section of the same routine tape, skipping over all intervening orders. Other orders shift the control from one routine tape to another. While hunting, the tape moves about four times as fast as when reading orders.

f. Arrangement of Table Tapes.—The table tapes of this machine are open-end tapes, and the transmitters into which they are fed can move the tapes in either direction. Each table tape is divided into sections called "pages," and each page into sub-sections called "blocks." Each block can contain an unlimited number of numbers. The use of pages is optional, but the use of blocks is mandatory. In order to find a certain number on a table tape, we have to know the page, if any, the block in which it is located, and its place within the block. Pages and blocks are characterized by four-digit numbers and arranged in ascending order of these numbers. For instance, to find the first number in block 1234 of page 5678 of Table Tape B, we give an order to the machine to move Table Tape B to the beginning of page 5678. Next, we order it to move the same tape to the beginning of block 1234. When this block has been found, the machine will at once read the first number in the block and store it in a special storing register called the "table register." The number can now be used by the machine just as any other number in a storing register. As soon as the machine receives an order to clear the table register, it will automatically read the next number on the table tape and put it into the table register. Thus, numbers are taken off the table tape in the order in which they appear in a block. (It is, of course, possible to read off numbers and clear them without using them and subsequently to call in the same block of the table tape and read off the numbers

previously cleared. Thus, at the cost of some time, numbers may be taken off in any order.)

Since the table tapes can move in either direction, blocks or pages may be called for in any desired order. However, the time required to find a given place on the tape is usually long and is one of the major limitations of this machine. As an indication of the order of magnitude involved, it may be stated that each number covers about one inch of tape (ranging from 1.1 inch for a seven-digit number to 0.5 inch for a one-digit number) and that the "hunting" speed is two inches per second. Thus, if a tape is to contain 2000 numbers, the average hunting distance between two points selected at random is 1000 inches and the hunt requires about 8 minutes. If many such random hunts are needed in a problem, the time requirement becomes prohibitive. However, it is usually possible to arrange the input numbers on the table tapes in approximately the order in which they are needed, and such an arrangement reduces the hunting time materially. It should also be mentioned in this connection that the hunting can be overlapped by other machine operations.

Instead of characterizing a block of data on a table tape by a single block number, it is possible to characterize it by two block numbers which bracket in all block numbers to which the data apply. This feature is used especially when the block numbers represent the arguments of functions stored in the table tape. For instance, suppose that we wish to store on a table the function $\sqrt[3]{x}$. For each value of x we perforate on the table tape a block number equal to x (or a linear function of x, with some suitable arrangement for the decimal point), the value of $\sqrt[3]{x}$ and interpolation coefficients, say, up to the third order. If the machine, in the course of a computation, has obtained the value of x accurate to seven places, it can then be instructed to find block x (to four places) on the table tape, read out the value of the function and of the interpolation coefficients, and by means of interpolation find the exact value of the function for the given x. It may be desired to perforate the value of function and interpolation coefficients, not for every value of x but only for a few selected values, and to take care of the interval between these by interpolation. Thus, we may use one block for all values of x between zero and 0100, another block from 0101 to 0200, etc. Such intervals are called "hyphenated blocks." They may vary in length. For instance, we may continue with blocks 0201–0400, 0401–0700, 0701–1200, etc. Evidently it is desirable to use short blocks in regions in which the function or one of its derivatives changes abruptly, as is the case for $\sqrt[3]{x}$ in the neighborhood of the origin. Also, hyphenated block numbers may be interspersed with simple ("common") ones.

When the machine receives an order to find a given block number x on a certain table tape, it starts to move the tape forward (i.e., in the direction of increasing block numbers) until it encounters a block number. If this is a common block number, it will then move the tape forward if the block number is smaller than x, move the tape backward if the block number is larger than x, and in case it is exactly equal to x, it will read the number following the block number into the table register. If, on the other hand, the machine encounters a "hyphenated block number," it will move the tape forward or backward if x is outside the interval indicated by the block

number, and read the next number into the table register if x is inside that interval (boundaries included).

The storage tape, mentioned in Section 4c above, has the same arrangement as a table tape. This tape runs through a reperforator, from there into a bin, and out of it into a tape transmitter. On orders from the machine the reperforator puts page or block numbers onto the tape, as well as the intermediate results which are to be stored. (The orders must be such that page and block numbers will be perforated in ascending order.) The storage tape transmitter can subsequently be ordered to find a given location (page and block) on the storage tape, and it can move the tape in either direction to find that location. A special safeguarding device ensures that the tape will not be torn if the transmitter hunts for a block which is located close to the reperforator, or which, by mistake, has not yet been perforated on the tape. If the storage tape is used only for final results, it is not necessary to put page or block numbers on it.

We mentioned before that the table tapes of this machine are open-end tapes of arbitrary length in which the block numbers are arranged in order of magnitude but otherwise arbitrary. In contrast, the Harvard machines use looped tapes in which the arguments must be equidistant. The latter arrangement enables the machine to find any desired location much faster, because the distance over which the tape must be moved is determined by the value of the argument alone and not by the contents of the tape. On the other hand, this arrangement has the drawback that the machine operator is limited in the way in which he organizes the numerical input.

g. Arrangement of the Problem Tape.—The problem tape is an open-end tape which can move in one direction only. Among its functions are to signal the beginning and the end of a problem or of certain subdivisions of a problem, to provide a check on the correct loading of the routine and table tape associated with the given problem, to provide for the printing of headings on the answer sheet. Additional functions which may or may not be assigned to it are to give certain orders which are needed only once during a problem (usually at the start), and to supply part or all of the numerical input for the problem.

The problem tape may be divided into sections called "PCS" sections, and these may be subdivided into minor sections called "CCS" sections. These designations are not particularly suggestive; they are abbreviations for "problem cycle section(s)" and "computing cycle section(s)." A "PCS" may contain several "CCS" and in addition two special kinds of subsections, called "SWR" (switch to routine) and "SWP" (switch to printer) sections. The former contains orders similar to those normally stored on routine tapes, the latter contains instructions to the printer, usually concerning the printing of headings. The "CCS" contain numbers; they are similar to table tapes except that they are not subdivided into pages and blocks. Such a subdivision would be pointless, since the problem tape can move only in one direction and therefore all information has to be taken off in the order in which it is used.

The problem tape exerts supreme control over the progress of a problem. When the machine is started the problem tape is the first to move. Typically, it will contain a few orders which direct the machine to make sure that the proper routine and table tapes have been loaded into the proper transmitters.

It may give additional orders (as indicated above) at this time. It will then direct the printer to print the desired headings. There may then follow the first CCS section. Whenever the problem tape comes to the beginning of a CCS, it delegates the control of the problem to the first routine tape. This tape now begins to move, and the machine reads from it the orders pertaining to the computation. In the course of the computation it may draw numbers from the CCS of the problem tape (in the order in which they appear there) and from the table tapes (in any desired order). The control may pass to other routine tapes. Finally it will come to an order that restores the supreme control of the problem tape. When this order is received, the routine tapes cease to move, the machine checks to see that it has received from the problem tape an indication that the present CCS section is finished, and then moves the problem tape to the beginning of the next section. If this is again a CCS, control is at once delegated to the routine tapes; if it is a SWR or SWP section, control rests with the problem tape for the length of the section. This process continues until the machine arrives at the end of the problem tape.

(To be concluded)

FRANZ L. ALT

Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland

[1] *MTAC*, v. 2, p. 185f.
[2] *MTAC*, v. 2, p. 97f, and 366.

The Square Root Method for Solving Simultaneous Linear Equations

The square root method for solving a system of linear equations was probably first developed by BANACHIEWICZ[1] in 1938, but it was independently developed by DWYER.[2] The growing popularity of the method in the U. S. stems from the papers by Dwyer and the simple explanation of the method given by DUNCAN & KENNEY.[3] The method is directly applicable to solving a system of linear equations only when the matrix of the coefficients of the unknowns is symmetric. However, if this is not the case, the system represented symbolically by the matrix equation $MX = N$, where M is the matrix of the coefficients of the unknowns, X is the column matrix of the unknowns, and N is the column matrix of the right hand members of the equations, can be transformed into another matrix equation with a symmetric matrix as the coefficient of X. This is accomplished by premultiplying both sides of the equation by the inverse of M. The additional computations required to symmetrize the matrix M, before applying the square root method, makes it doubtful that this method is as efficient as other more direct methods when the coefficients of the unknowns in the original equations do not form a symmetric matrix.

The NBSCL has applied the square root method for solving systems of normal equations arising in least square solutions and has found the method very efficient. With a standard calculating machine a good computer can solve ten equations in ten unknowns carrying about eight significant digits