Table 1, p. 15–16, *American Practical Navigator* (originally by N. Bowditch), U. S. Hydrographic Office, no. 9, revised edition of 1938, is for Radio Bearing Conversion. For difference of longitude = 1°(0°.5)16°.5 and middle latitude 4°(1°)60° the table gives the correction to be applied to radio bearing to convert to Mercator bearing. This table is to be replaced by the much more extensive new table.

With the new table

(1) Great circle directions can easily be converted to rhumb line directions for plotting radio bearing on a Mercator chart. Since radio waves travel along great circles, such corrections are necessary.

(2) Rhumb lines may be converted to great circle directions. Usually tangents or chords of the great circle, for about 5° of longitude, are sailed. The Table will be of value in planning courses, and will quickly show when the difference between rhumb line courses and great circle courses is significant.

If values are desired within the 5° intervals of the table at higher latitudes they may readily be found by double interpolation. The table may be of service in both marine and air navigation.

Frances W. Wright

Harvard College Observatory

AUTOMATIC COMPUTING MACHINERY

Edited by the Staff of the Machine Development Laboratory of the National Bureau of Standards. Correspondence regarding the Section should be directed to Dr. E. W. Cannon, 418 South Building, National Bureau of Standards, Washington 25, D. C.

Technical Developments

The leading article of this issue of *MTAC*, "A Bell Telephone Laboratories' Computing Machine—I," by Dr. Franz L. Alt, is our current contribution under this heading.

Discussions

Decimal Point Location in Computing Machines

General. For various reasons, the majority of large scale automatic-sequence digital computers in existence or in the design stage use fixed decimal (or binary) point numbers. In all the machines of which the writer has knowledge, the decimal (or binary) point is located at the extreme left so as to make all numbers fall in the range -1 to $+1$. It is the writer's opinion that this choice is not the best, and that location of the decimal (or binary) point several digits away from the extreme left is better in almost every respect. Some of the considerations leading to this opinion are presented in this paper. For the sake of simplicity and definiteness, unless otherwise stated, the discussion refers to a machine in which data are stored in some form of memory which restricts numbers to a specified fixed number of decimal digits. This machine is assumed capable of automatically following a prescribed sequence of instructions which include addition, subtraction, multiplication, division, and transfer operations. The results of the operations are rounded off on the right to the same number of digits as the original data, after positioning the digits in accordance with a single fixed decimal point location. It is further assumed that the machine has no provision for handling excess digits on the left (exceeding capacity of the machine) created by any operation, so that such an occurrence implies an error in programming.

Possibility of Exceeding Capacity. One of the chief advantages claimed for the extreme-left-hand position of the decimal point is that the machine capacity as regards size of individual numbers cannot be exceeded in multiplication, since a product of two numbers, each less than unity, must also be less than unity. As this is not true for the other operations, such as addition or division, this feature would not appear to have much value, since magnitudes of numbers at each step have to be studied by the programmer in any case to avoid exceeding capacity in other operations. Even in multiplication, magnitudes must be considered to correct for the opposite effect caused by this choice of decimal position (the continual decrease in magnitudes of numbers as a result of successive multiplications) by judicious shifting (or division by powers of 10) at suitable stages in the computation.

Although any choice of the decimal point further to the right creates the possibility of obtaining products larger than either factor, thus exceeding limits if the factors are sufficiently large, this is compensated somewhat by the possibility of arranging values (for example by keeping them approximately equal to unity) so that any number of successive multiplications may be programmed without either exceeding limits or losing many significant digits. Furthermore, in division, use of wholly fractional numbers requires that the numerator be smaller than the denominator to keep the quotient within bounds, while much broader limits are permissible if one or more digits are available to the left of the decimal point. The extra magnitude of the checking in division necessitated by choice of the extreme-left decimal point would appear to offset any possible gain in checking of multiplications.

Inherent Accuracy. Another important advantage claimed for use of the extreme-left decimal point location is that this permits maintenance of the greatest number of significant figures in each factor in multiplication and hence leads to greatest accuracy. That this contention cannot be wholly true is easily seen from the following 3-digit example, which compares the result obtained when the decimal point is at the extreme left with that obtained when the decimal point is one digit to the right.

$$.123 \times .111 = .014 \qquad \text{accuracy 2 digits}$$
$$1.23 \times 1.11 = 1.37 \qquad \text{accuracy 3 digits.}$$

Let us examine the question of accuracy in a little more detail. We restrict ourselves to consideration of errors due to the limited number of digits (round-off errors) only and, for simplicity, evaluate the maximum such error instead of the most probable value. We further assume that the programmer has accurate cognizance of the magnitudes of all numbers and can shift them to the most advantageous digital position, subject to the number limitations of the machine, but that no auxiliary operations such as division or multiplication by factors other than powers of 10 are to be employed, so that there is no change in the digits themselves. If numbers have n total digits and the decimal point is m digits from the left, the maximum round-off error in a number of magnitude N is $10^m K/N$ of the number, where

$$K = \tfrac{1}{2} \times 10^{-n}$$

is the error for maximum N. If two numbers M and N are multiplied together, their product will be $P = MN$. The percentage error in P is the sum of the percentage errors in M and N plus additional round-off error in limiting P to the specified number of digits, making the maximum error in P equal to

$$E = 10^m(1/M + 1/N + 1/P)K,$$

where all values are assumed positive.

For a machine with decimal point at the left, $m = 0$ and M, N, and P must not exceed unity. Minimum E occurs for numbers close to this limit and has the value

$$E \text{ (minimum for } m = 0) = 3K.$$

Maximum E occurs when M and N (and hence P) are smallest. Since, under our assumptions, we may always shift numbers so that the first digit is not a zero, we can insure that M and N are at least .1. For this limit, we have $P = .01$ so that

$$E \text{ (maximum for } m = 0) = (10 + 10 + 100)K = 120K.$$

For a machine with decimal point one digit to the right, $m = 1$, and M, N, and P must not exceed 10. Minimum E occurs for $M = N = \sqrt{10}$ and has the value

$$E \text{ (minimum for } m = 1) = 10(1/\sqrt{10} + 1/\sqrt{10} + 1/10)K = 7.3K.$$

Maximum E occurs when M and N (and hence P) are smallest. Whenever possible, we shift M and N so that the first digit in each is not a zero. When this causes P to exceed 10, we shift the number (say M) with smaller initial digits to the extreme left and allow a single zero digit at the left of the other. With this arrangement, which is always possible, the largest E occurs for $M = \sqrt{10}$, $N = 1/\sqrt{10}$, $P = 1$ and is

$$E \text{ (maximum for } m = 1) = 10(1/\sqrt{10} + \sqrt{10} + 1)K = 45K.$$

We thus see that the maximum error is actually less in this case by a factor of almost 3 as compared to the case where the decimal point is at the extreme left, although the minimum error is not as low. Even for $m = 2$, we get

$$E \text{ (maximum for } m = 2) = 120K,$$

corresponding to M close to 10 and $N = 1$, so that the extreme error, with the decimal point two digits to the right, is no greater than with decimal point at the extreme left.

The above is based on the assumption that the programmer can predict magnitudes very closely and that, in iterated operations, the range of magnitude variation is small. This is generally not the case, and an appreciable spread of values must be allowed for. If we assume uncertainty in each of the factors in a multiplication over a range of 10 to 1, the programmer cannot assure best placement of digits but only that numbers will be at most one digit away from best positioning. Maximum errors will then occur for M and N both one tenth their previous values. For $m = 0$, this corresponds to $M = N = .01$; $P = .0001$, giving

$$E\ (m = 0, \text{ 10 to 1 range}) = (100 + 100 + 10000)K = 10200K.$$

For $m = 1$, the extreme occurs when $M = 1/\sqrt{10}$; $N = 1/10\sqrt{10}$; $P = 0.01$ and is

$$E\ (m = 1,\ 10 \text{ to } 1 \text{ range}) = 10(\sqrt{10} + 10\sqrt{10} + 100)K = 1350K,$$

being almost 10 times as good as the preceding. Similarly, for $m = 2$, $M = 1$; $N = 0.1$; $P = 0.1$ is the extreme case and

$$E\ (m = 2,\ 10 \text{ to } 1 \text{ range}) = 100(1 + 10 + 10) = 2100K.$$

Even for $m = 4$, the extreme is for $M = 10$; $N = 1$, $P = 10$ and has the value

$$E\ (m = 4,\ 10 \text{ to } 1 \text{ range}) = 10000(.1 + 1 + .1) = 12000K,$$

which is not materially greater than for $m = 0$.

Obviously, the greater the range of values in an iterated operation, or the greater the uncertainty in regard to possible magnitudes that may occur, the poorer the choice of decimal point at the extreme left becomes as concerns accuracy, and the greater the number of digits to the left of the decimal point corresponding to minimum extreme error.

Availability of Common Integers and Constants Exceeding Unity. It is certainly highly desirable to have integers such as 1, 2, 3, or 10 and constants such as $\pi = 3.14159$, $e = 2.71828$, $\sqrt{2} = 1.41424$, $\ln 10 = 2.30259$, etc., available for use in programming. With the decimal point at the extreme left, such values cannot be expressed directly but must be divided by a power of 10 or must be given in a different form such as the reciprocal value. This, at best, complicates the programmer's work. As a simple example, consider the usual square root iteration

$$y_{n+1} = (y_n + x/y_n)/2$$

to obtain $y = \sqrt{x}$. The convenient first approximation of $y_0 = 1$ cannot be directly used, nor can the operation be carried out as indicated. It is necessary to change the form to

$$y_{n+1} = \tfrac{1}{2}y_n + \tfrac{1}{2}x/y_n$$

and to take a string of 9's for y_0 to avoid exceeding capacity. This involves one additional initial step (halving of x) and an awkward value for y_0 and still is insufficient if there is any possibility that x itself is a string of 9's, since this would cause y_1 to exceed unity (because of normal round-off's).

Mathematical values do not naturally limit themselves to magnitudes below unity, and such an arbitrary limitation may be expected to lead to various programming difficulties, which may be minimized by allowing some range above unity. The most important value that should be included in the permissible range is unity itself. If, for example, cosine or sine values are required in a computation, it is possible to exceed the capacity of a machine with decimal point at the left if the argument should get too close to zero or $\frac{1}{2}\pi$, requiring either some assurance that this will not occur or sacrifice of a decimal digit for values with the accompanying difficulty of introducing additional factors in the programming. It is certainly far more convenient for the programmer to work with natural values than to keep track of various factors introduced to shift values to lie within range of the machine. Of course, some factors are required with any choice of decimal point location, but their number is materially greater for the extreme left location than for locations several digits further to the right. It is only necessary to glance through a mathematical text to verify this statement by observing the relative occurrence of values above, say, 100 as compared to those between 1 and 100.

Problems Involving Extreme Accuracy. A general-purpose computer may be required to solve problems whose solution is desired to greater accuracy than corresponds to the fixed number of digits permitted in a single memory location of the machine. In certain problems, while the solution is not required to excessive accuracy, it is necessary at some intermediate stage to provide much greater precision because of large loss of relative accuracy during the process of computation. Before discussing this matter in regard to decimal machines, it may be noted that, in addition to difficulties similar to those encountered in decimal machines, a binary machine used for problems of extreme accuracy involves questions of precision of conversion between the normally used decimal system and the binary system used by the machine. Since, in general, only integral values convert exactly between these two number systems, a binary machine with binary point at the extreme left requires the programmer to insert powers of 2 as factors as well as powers of 10 in order to avoid conversion errors. For example, 10 would be inserted as $\frac{5}{8}$ with the factor 16 to be remembered by the programmer. While such binary factors can probably be handled expeditiously by high-grade mathematicians, they are certainly more likely to cause errors and require special training for programmers of lower skill.

In order to simplify handling of high-precision problems, we will assume that our machine has available, when called for, a special division process which, in addition to the normal quotient, stores the remainder (shifted to the left up to the decimal point) into a specified memory location. Such a facility as well as a similar one for multiplication will be available in the EDVAC now under construction by the University of Pennsylvania. For a machine of 3 decimal digits with decimal point at the left, this special division will work as follows:

$$.111/.123 = .902 + .000054/.123.$$

The machine will store .902 in one location and .054 in another. For a similar machine with one digit to the left of the decimal point, this operation will be:

$$1.11/1.23 = 0.90 + .0030/1.23,$$

and the machine will store 0.90 in one location and 0.30 in another. In all cases, the second number will always be less than the divisor.

With the aid of this special operation and a corresponding multiplication, and more than one memory location for the results of each operation, almost any degree of accuracy can be achieved. It is to be noted, however, that each value is expressed as a sum of component terms, with suitable factors, each stored as a separate number. Successive operations are, therefore, operations on polynomials and result in similar polynomials. At each stage, care must be taken not to exceed capacity in any term of the polynomial and to "carry over" to the next term any indicated excess. With the decimal point at the extreme left, this is a difficult chore and generally requires extensive modification of procedure with addition of quite a few comparison instructions (branch orders) to check for and correct this condition without momentarily exceeding capacity at some point. With one or more digits to the left of the decimal point, this problem becomes much simpler to take care of since, as will be noted in later numerical examples, all terms other than the first are normally less than unity. It is thus possible to perform the next operation without considering the exceeding of capacity as a result of it, and then to add the integral portion of each term to the preceding (after suitable shift to the right) as a "carry." In fact, this "carry" process may frequently be postponed until after a number of operations are carried out. For example, the evaluation of *e* to any number of digits can be completely carried out on a machine of the type we are discussing, with one digit to the left of the decimal point, with only a single "carry" routine in the entire procedure.

It may be worth while to give a simpler numerical example here. Suppose we had a 3-digit machine and wished to evaluate

$$101/301 + 201/401$$

to 7 digits. Consider first the decimal point at the extreme left. Using special division, we obtain

$$\begin{aligned} .101/.301 &= .335 \text{ quotient} + .165 \times 10^{-3} \text{ remainder} \\ .165/.301 &= .548 \text{ quotient} + .052 \times 10^{-3} \text{ remainder} \\ .052/.301 &= .173 \text{ normal quotient} \end{aligned}$$

so that we store 101/301 in 3 memory locations as

$$.335 + .548 \times 10^{-3} + .173 \times 10^{-6}.$$

Similarly, we get

$$201/401 = .501 + .246 \times 10^{-3} + .883 \times 10^{-6}.$$

We cannot now add terms directly since their sums may exceed unity. It is thus necessary to examine values before summing them. One way of doing this is to compare the complement of .173, which is .827, with .883. Since it is smaller than the latter, we subtract it from the latter obtaining .056 and carry 1 to the next term. Comparing the complement of .548, which is .452, with .246, we find it larger, indicating no carry. We thus add the original terms, obtaining .794, to which we must add .001 carry from the previous addition, requiring another comparison to show that we may directly add to get .795 with no carry. The first terms can, of course, be directly added without any ado since we know their order of magnitude. We thus get

$$101/301 + 201/401 = .836 + .795 \times 10^{-3} + .056 \times 10^{-6}.$$

If additional similar terms were to be added, the work would, of course, increase enormously.

Now let us consider the same problem with the decimal point one digit to the right. As before, we get

$$\begin{aligned} 1.01/3.01 &= 0.33 \text{ quotient} + 1.50 \times 10^{-2} \text{ remainder} \\ 1.50/3.01 &= 0.55 \text{ quotient} + 1.45 \times 10^{-2} \text{ remainder} \\ 1.45/3.01 &= 0.48 \text{ quotient} + 0.52 \times 10^{-2} \text{ remainder} \\ 0.52/3.01 &= 0.17 \text{ normal quotient} \end{aligned}$$

so that we store 101/301 in 4 memory locations as

$$0.33 + 0.55 \times 10^{-2} + 0.48 \times 10^{-4} + 0.17 \times 10^{-6}.$$

Similarly, we get

$$201/401 = 0.50 + 0.12 \times 10^{-2} + 0.46 \times 10^{-4} + 0.88 \times 10^{-6}.$$

Here we may add corresponding terms without worry of exceeding capacity. In fact, since each term is less than unity in value and the machine capacity is 10, we could add 10 terms at a time without fear of exceeding limits. We thus get

$$101/301 + 201/401 = 0.83 + 0.67 \times 10^{-2} + 0.94 \times 10^{-4} + 1.05 \times 10^{-6}.$$

We now take the integral part of each term as a "carry" to the preceding term. If less than 9 terms had been previously added, such carry cannot cause excessive magnitudes and can be very simply carried out in succession. In this case, only one carry occurs and we finally get

$$101/301 + 201/401 = 0.83 + 0.67 \times 10^{-2} + 0.95 \times 10^{-4} + 0.05 \times 10^{-6}.$$

The only disadvantage of the new decimal point location is that it requires more terms for equal accuracy; however, the given example accentuates this difficulty because of the small number of total digits and the particular choice of numerical values. For usual machines of 10 or more decimal digits, the increase in number of terms caused by one digit shift of the decimal point would be negligible, but the saving in carry-over difficulties just as large. Choice of numerical values such as 41/301 + 51/401 would further equalize the number of terms.

SAMUEL LUBKIN

Reeves Instrument Corporation
New York City

BIBLIOGRAPHY, Z–II

1. ARTHUR W. BURKS, HERMAN H. GOLDSTINE & JOHN VON NEUMANN, *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument.* Prepared for the Research and Development Service, Ordnance Department, U. S. Army, June, 1946, i, 53 leaves and table, 21.6 × 27.9 cm. Mimeographed.

This report, consisting of 6 sections and a table giving a proposed instruction code, is the first of two papers dealing with "some aspects of the over-all logical considerations arising in connection with electronic computing machines."

Section 1.0, "Principal Components of the Machine," is a statement of the "main organs" a fully automatic general-purpose computing machine must possess. The authors point out that such a machine must have a memory organ, a control organ, an arithmetic organ, and input and output organs.

The computer must be capable of storing tables of functions, the original data of problems, and the intermediate results of computations; it must also be able to store sequences of instructions for performing specified numerical computations. This storage function is performed by a memory organ. The automatic execution of the orders stored in the memory is in turn effected by the control organ.

A general-purpose calculator must be capable of performing the arithmetic operations. The machine will view certain of these operations as elementary and will have an organ—the arithmetic organ—to perform them automatically upon demand. This organ will add, subtract, multiply, divide and perform other operations such as shifting, frequently occurring. Lastly, the computing machine must possess input and output organs to serve as communication channels between it and the human operator.

Section 2.0, "First Remarks on the Memory," is concerned with the size of the memory organ. Here are discussed briefly the memory requirements of various types of computations, including the numerical solution of total differential equations, of partial differential equations of the parabolic or hyperbolic type, and of problems solved by iterative procedures, such as systems of linear equations and elliptic partial differential equations. It is concluded that an electronic memory organ capacity of 4000 numbers of 40 binary digits each exceeds the capacities required for problems handled by existing machines by a factor of about 10. The machine proposed in the report will have an electronic memory capacity of this size and will have, in addition, a subsidiary memory, also fully automatic, of larger capacity but slower in the transfer of numbers and orders.

In Section 3.0, "First Remarks on the Control and Code," a type of instruction code envisaged by the authors is analyzed. A code must be adequate for the description of any sequence of operations the computing machine may be expected to perform. Decisive considerations in the choice of a code are the simplicity of the equipment it requires, the ease with which the human operator can use it to instruct the machine to work important problems, and the speed with which the machine can interpret and follow it.

Minimum requirements for a code are given. The code must contain instructions for performing the fundamental arithmetic operations, for transfer of data from the memory to the arithmetic organ and back again, and for the modification of memory location numbers of orders. Orders which will integrate the input-output devices with the machine will also be necessary. Finally, since the control organ will not always follow the normal sequence from place n in the memory to place $(n + 1)$ for its next instruction, the code must include control transfer orders.

The ideal memory unit, described in Section 4.0, "The Memory Organ," would be capable of storing an indefinitely large number of 40-binary digit numbers or words in such manner that each word would be available within a few micro-seconds. It is stated that it appears physically possible to achieve indefinitely large memory capacity only by the use of a hierarchy of memories, each of which has greater capacity than the preceding but which is less quickly accessible for the transfer of stored information. A possible hierarchy would consist of a primary high-speed memory unit of relatively small capacity, a secondary memory, slower but of larger capacity, and a vast quantity of dead storage, i.e., storage not integrated with the computing machine.

A suitable primary or high-speed memory device appears to be a cathode-ray tube storing electrical charges on a dielectric plate inside the tube. Such a tube is effectively a myriad of electrical capacitors which can be connected into the memory circuit by means of an electron beam. The Princeton Laboratories of the Radio Corporation of America are engaged in the development of the Selectron, a storage tube of this type. For the high-speed memory it is planned to use 40 Selectrons in parallel, each tube storing one digit of a 40-binary digit number. Such parallel use of Selectrons provides a faster memory and appears to involve simpler techniques than the use of Selectrons in series. The parallel-type memory causes the computer to be essentially of the parallel type; for example, in performing an addition all corresponding pairs of digits are added simultaneously. (In a serial computer, these pairs of digits are added serially in time.) In fact, the authors point out that once a given component has been chosen as the elementary memory unit, the nature of the balance of the computing machine is more or less determined.

The secondary memory can consist of storage on teletype tapes, magnetic wire or tapes, movie film, or similar media controlled by the Control Organ. The automatic integration of this storage medium with the computer is achieved by introducing appropriate orders into the code. The secondary storage medium must be such that the operator can place words on the medium and can read words put on it by the computing machine. It serves as a part of the input-output system of the machine. The dead storage is an extension of the secondary storage medium, differing only in not being immediately available to the computing machine. It may consist of a library of tapes that can be introduced into the machine by the human operator when desired and which will then be automatically controlled by the computer.

Section 5.0, "The Arithmetic Organ," includes a discussion of the features the authors consider desirable for the arithmetic part of a computer. Tentative conclusions are reached as to which of the arithmetic operations should be built into the machine. The remaining operations must be programmed. A schematic drawing of the arithmetic unit is described.

The planning of the arithmetic organs of a computing machine leads naturally to a consideration of the number system to be adopted, to a determination of whether the machines shall store numbers in terms of significant figures or decimals (i.e., "floating" versus fixed decimal point), to a study and evaluation of rounding-off procedures, and to the analysis of iterative processes that can be used as substitutes for built-in arithmetic operations. These matters are treated clearly and in detail (21 leaves of a 53 leaf report). Perhaps the most interesting characteristic of this section is the manner in which arithmetical and statistical analysis is correlated with the design of the arithmetic organ. The proposed computing machine is described as a 40-binary digit, fixed-binary-point machine with a double precision multiplier, in which addition, subtraction, multiplication, division, shifting, and taking the absolute value are "built-in" arithmetic operations.

Section 6.0, "The Control," describes the function of the Control Organ. An order code for the operations to be performed within the computer is proposed and given in Table I at the end of the section. Orders relating to operations involved in getting data in and out of the computer are not discussed. The control routine in the execution of the built-in orders is analyzed and the general manner in which the control organ functions is described. It is indicated how decoding or many one-function tables serve as switches by means of which the control selects a specified Selectron memory location and decodes the orders, which are given in coded binary form. How binary counters and flip-flops (trigger circuits) enable the control to take order pairs in sequence from the memory and shift itself from one sequence of control orders to another is described briefly.

The merits of various alternative checking systems are weighed against their cost in equipment. The method of localizing errors is discussed. The proposed computer has a feature the authors consider advantageous in the light of their experience with the ENIAC. Namely, the circuits are designed in such a manner that if the timing clock is stopped between pulses, the computer will retain all its information in trigger circuits so that the computation may proceed unaltered when the clock is started again. The operator will then be able to put the machine through an operation step by step, checking the results by means of the indicating lamps connected with the flip-flops. It is clear that this will be an extremely useful feature of the computer.

The report concludes with a discussion of the function of the control organ in the execution of input-output orders. It is apparent that the authors' views on input-output organs are not as definite as those on the other organs described in the report.

This report presents a clear picture of the factors that were taken into account by a competent group in planning an automatic-sequence electronic digital computing machine, and is thus interesting and useful to anyone who is active in the large-scale electronic computer field. Its main value lies in the fact that the authors have succeeded in giving a general picture of the type of computer under consideration, and in providing a background of the logical considerations underlying their decisions concerning the kind of computer that would be most useful for general mathematical computation.

MDL

2. L. J. Comrie, "Calculating—past, present and future," *Future, the Magazine of Industry, Government, Science, Arts*, 1947, no. 1, "Overseas Issue," p. 61–69, 15 illustrations. 20.6 × 29.5 cm. Published in England but printed in Holland.

In this popular article, Dr. Comrie presents an interesting survey of the development of calculating machines from the use of small stones as counters up to the modern ENIAC built at the Moore School of Engineering, University of Pennsylvania. He shows the progression from the early, primarily commercial machines to those designed and built by scientists for the solution by numerical evaluation of equations classified as insoluble by the pure mathematician. The conception of the ENIAC as an electronic brain—a false impression that seems to have captured the public imagination—is swept aside by the author. He considers the machine in no way a substitute for the human mind, which must still instigate the machine's every move; the rapid calculation of problems heretofore impracticable of solution is seen as its chief function.

Charles Babbage's "analytical engine" is described as one of the most important forerunners of the modern calculators. This machine, with a store for numbers, a mill where calculations were to be performed, and a control to link these two units, was never completed because of production difficulties and lack of financial support. However, it has served as a stimulus to the development of the present IBM Sequence-Controlled Calculator, the Bell Telephone Relay Machine, and the ENIAC. It is pointed out that the development of the Differential Analyzer by Dr. BUSH at the Massachusetts Institute of Technology gave further impetus to the new trend in calculating, which emphasizes speed and the elimination, as far as possible, of human intervention.

The Harvard IBM machine, "the first of the really big machines brought into being by the war to produce actual numerical results," consists largely of specially assembled HOLLERITH parts and has as its "brain" a newly designed tape control. Professor H. H. AIKEN, who collaborated with IBM engineers in the creation of this machine is now at work on an improved calculator of this type. Another war-time development was the Bell Telephone Laboratories Relay Machine, designed by STIBITZ & WILLIAMS. It is organized on very much the same basis as a desk calculator and bears a resemblance to a telephone exchange. The ENIAC, designed to meet the needs of the U. S. Army Ballistics Research Laboratory at Aberdeen, Maryland, attains much greater speed by the use of electronic instead of mechanical parts. Although it is much too large and contains far too little memory space, it has served to pave the way for the current EDVAC machines now being designed at Philadelphia and Princeton, and for the general-purpose British ACE, also in a design stage.

This entertaining article, containing more than a dozen excellent illustrations of historical as well as modern machines, emphasizes the pressing need for continued support of the development of the new calculators.

MDL

3. HERMAN H. GOLDSTINE, and JOHN VON NEUMANN, *Planning and Coding of Problems for an Electronic Computing Instrument*, Institute for Advanced Study, Princeton, New Jersey, 1947, 69 pages, 21.6 × 27.9 cm.

To the steadily increasing group of engineers and mathematicians who are becoming electronic-machine conscious, two reports prepared by the Institute for Advanced Study for the Army Ordnance Department have proved of invaluable aid. The first,[1] which outlined the features of a high-speed computing machine possessing a large selectron-tube memory, carried a table of 21 orders, or codes,[2] of such a nature that the ability of the machine to execute them would render it a powerful tool for solving highly complex mathematical problems. This table, with several modifications, is repeated in the second report—the volume under review—which constitutes a manual on the application of these codes to actual problems.

When first faced with the task of setting up a problem on a high-speed machine, the coder's situation is like that of a navigator in unfamiliar waters. A route must be charted consisting of a series of steps for the control of the machine to follow from the starting point to the successful solution of the problem. Unfortunately, these steps (except in the case of the simplest problems) do not lie in a straight, unbroken, one-way path, but usually meander through a tortuous maze of loops, bifurcations, retracings, and jumps, as the computations may entail any combination of iterative processes, alternative procedures, and various modifications conditioned by intermediate results obtained during the process.

In the first section of their manual, the authors have laid down the general principles for the construction of such charts, or "flow diagrams," with amazing skill and inventiveness. They list and carefully analyze the various turns and twists that the future pilot-coder is likely to encounter and point out the location of dangerous reefs he must avoid. In order to designate on the flow chart all possible digressions from the straight path that may occur in the course of a problem's solution, they invent a set of comprehensive signals to be attached at strategic points.

The authors then proceed to illustrate these principles by application to actual problems. The first two of the five examples coded in the second section are too simple to require flow diagrams. They serve, however, to introduce the reader to several elementary coding principles, thus enabling him to follow the more difficult situations exhibited by the succeeding problems, each of which is accompanied by its flow diagram.

The flow diagram for Problem 3, which calls for the computation of the values of a quotient of two polynomials in u for a set of given values of u, has a loop that must be traversed repeatedly until all the required divisions are performed. Problem 4 entails a much more formidable flow diagram. The machine under consideration in this report computes with binary numbers of absolute value less than unity. It is essential that the machine be instructed to determine the magnitude of the quotient of two such numbers when it occurs, so that the proper scaling factor may be introduced to render the quotient suitable for further manipulation. Problem 4 illustrates how such automatic sensing of size can be coded. By the time the reader has covered the various steps contained in Problem 4, he is prepared to follow without difficulty the diagram and coding of the last problem in the section, that of finding the square root of a number by the process of iteration.

The third, and last, section of the manual deals with various systems of digital notation and discusses the possible input and output speeds of data. These are related to the projected speed of binary-decimal conversion within the machine. There follow two problems concerning binary-to-decimal and decimal-to-binary conversion, although it would seem that the latter operation could well be delegated to several subsidiary converters. The section ends with two problems illustrating the coding of double-precision arithmetic processes. These problems show in detail how the machine may be instructed to add, subtract, and multiply two numbers containing 78 binary digits each, instead of the ordinary 39 digits. A brief outline for coding double-precision division is also given. It is pointed out that double precision will take 6 to 8 times longer than ordinary precision operations.

As in the case of the earlier manual[3] issued by the Computation Laboratory at Harvard, the present manual is of greatest benefit in so far as it contains principles and illustrations sufficiently general to apply to any high-speed sequence machines about to be constructed. It does not detract from the great value of these reports that the codes contained in them may perhaps be revised in character and number to simplify either the construction of the machine, the task of the coder, or both.

The authors themselves admit the lack of finality in their coding system.

What they attempt to do in their report is to indicate that (to quote their own words) "coding . . . has to be viewed as a logical problem and one that represents a new branch of formal logics." In this they succeed admirably, as well as in formulating the laws of this new branch. The series of forthcoming reports on the same subject, which are promised the reader in the Preface, will be eagerly awaited.

MDL

[1] A. W. BURKS, H. H. GOLDSTINE, & J. VON NEUMANN, *Preliminary Discussion of the Logical Design of An Electronic Computing Instrument*, see no. 1 in this Bibliography.
[2] These do not include the necessary input-output orders.
[3] HARVARD UNIV., Computation Laboratory, *A Manual of Operation for the Automatic Sequence Controlled Calculator*, 1946, see *MTAC*, v. 2, p. 185–187.

4. D. R. HARTREE, "The application of the differential analyzer to the evaluation of solutions of partial differential equations," *Proceedings of the First Canadian Mathematical Congress, Montreal, 1945.* Toronto, Univ. of Toronto Press, 1946, p. 327–337. 14.6 × 20.96 cm.

In this short technical paper, Professor Hartree first mentions the possibility of a quantitative solution on the differential analyzer of certain partial differential equations of applied mathematics that are inconvenient or impossible to solve in terms of tabulated functions. He then briefly describes this machine, which was developed by Dr. Vannevar Bush at the Massachusetts Institute of Technology primarily to solve ordinary differential equations by mechanical means. Its units, which may be interconnected by shafts and gearing in many different ways depending on the specific problem, translate the required mathematical operations into mechanical terms. Chief among these units are integrators capable of performing integrations continuously with respect to any variable, adding units, input and output tables, and in some cases a multiplier. The machine is used most effectively, says the author, when a number of solutions of a single equation or a solution of a series of equations of the same type is required, all of which would necessitate comparatively minor changes in the machine setup.

More specifically, Professor Hartree discusses the application of this machine to the solution of a two-point boundary problem in which an estimate of some quantity must be made at the beginning of the integration and trial solutions worked through until the required condition at the other end of the range of integration is satisfied. In the case of simple heat conduction in one dimension in a slab of a uniform solid whose thermal properties are constant, a solution is obtained by replacing the original partial differential equation with an approximately equivalent ordinary differential equation. This may be done by working in finite differences in the time variable and integrating continuously in the temperature distribution variable, although several hundred trial runs are necessary before an accurate solution is thus reached. Alternatively, the author outlines a more recent method that is less time-consuming but more demanding on the machine. It consists essentially in integrating in time and replacing the space derivative by finite differences. This is effected by repeatedly selecting three equally spaced points in the field and replacing the original heat equation with a finite difference equation involving these points. A simultaneous solution of the resulting equations is therefore necessary but may be accomplished in only one machine run. Either of these methods is applicable, as the field of integration in this particular problem is open in the time direction. If the field of integration were surrounded by a closed boundary, it would be more efficient to apply Southwell's relaxation method, which does not come within the scope of the machine.

Several extensions of the heat conduction equation, to which both of these methods might be applied, are also mentioned. A list of nineteen references is appended to the article.

MDL

News

Eastern Association for Computing Machinery.—The first meeting of this group was held on September 15, 1947, at Columbia University, New York City. An interesting description of the pilot model of the EDVAC, illustrated with slides, was given by Dr. T. K. Sharpless of the Moore School of Electrical Engineering. The following officers were elected:

J. H. Curtiss of the National Bureau of Standards, President;
J. W. Mauchly of the Electronic Control Company, Vice-President;
Edmund C. Berkeley of the Prudential Insurance Company of America, Secretary;
R. V. D. Campbell of the Raytheon Manufacturing Company, Treasurer.

The officers were instructed by the committee of the whole to appoint eight additional members of an Executive Committee, which is to write a set of by-laws for the Association. The by-laws will be submitted for approval within three months to those interested in the formation of the Association.

The Institution of Electrical Engineers.—On March 7, 1947, a discussion was held on "Recent Developments in Calculating Machines" at the Institution of Electrical Engineers, Measurements Section, London, England. Professor D. R. Hartree opened the meeting with a brief mention of the most important analogue and digital machines, illustrating his remarks with photographs of the newest differential analyzer at M. I. T., of the two Harvard machines, and of the ENIAC. He also suggested improvements in the ENIAC to make it a general-purpose machine, namely, a larger memory and the elimination of manual programming. Dr. Turing further suggested that an adequate memory could be used to eliminate the need for any manual programming.

Mr. John Todd, now a member of the staff of the Institute of Numerical Analysis, although admitting the importance of the new machines, pointed out that the Admiralty Computing Service had been able to handle all the work submitted to it using only desk computers. (See *MTAC*, v. 2, p. 289f.) Great care should be exercised, said Mr. Todd, in the formulation and preparation of problems for computation.

In answer to Dr. Hartree's suggestions for a computing machine of a more general nature than the ENIAC, Dr. Comrie stated that the EDVAC, which has been declassified, does meet Professor Hartree's requirements. He also heartily endorsed the proposal of Todd and Erdélyi for an Institute of Practical Mathematics,[1] which would be of great value for the effective utilization of the projected machines.

After a short talk on iconoscope storage devices by Professor F. C. Williams, of Manchester, Professor Hartree, citing problems of quantum mechanics and of supersonic motion, pointed out that there would be no lack of work for the new machines. He remarked that many essentially nonlinear problems, heretofore impracticable, could be handled by these machines, necessitating new mathematical techniques and theories. A typical electronic machine, once the design problem had been overcome, would probably cost around £20,000. Professor Hartree admitted that the problem of checking had so far received too little attention. Running two machines in parallel would effectively eliminate any danger of machine error, but the more serious type of error, arising in the formulation of a problem, could only be avoided by more comprehensive checking.

Mathematical Association of America.—On September 2, 1947, at the concluding session of the summer meeting at Yale University, a symposium on computing machines was held, with lectures by HOWARD H. AIKEN of Harvard University and JOHN VON NEUMANN of the Institute for Advanced Study at Princeton University. Supplementing his lecture with slides, Dr. Aiken described the essential design features of the two electro-mechanical digital computing machines, Mark I and Mark II, which were built under his direction. He stated that the successful operation of the Mark I computer has proved the feasibility of big automatic calculators for large-scale computing, and that in the next few decades vast improvements may be expected in this field.

Dr. von Neumann, who is directing a research project at the Institute for Advanced Study for the construction of an improved electronic digital computing machine, spoke on some general aspects of the new high-speed computing field. In discussing the need for large-scale computing he stated that nearly all of the progress in solving the non-linear partial differential equations of aerodynamics so far has been brought about through the use of the wind tunnel, which may be regarded as a crude analogue calculator. He then spoke of the advantages of digital computers over analogue computers, explaining that the latter are completely inadequate to solve the difficult types of problems that digital computers could handle. He concluded with some speculations on the influence large-scale computers will have on future studies in pure mathematics, expressing his hope that the new machines will enable mathematicians to attack successfully the field of non-linear problems, where the classical methods of analysis, unaided by high-speed computing, have so far been unproductive.

Recent Developments in Mathematical Computing in France.—The Laboratoire de Calcul Mécanique, at Paris, is conducting a program of research in mathematical computing sponsored by the Centre National de la Recherche Scientifique (see *MTAC*, v. 2, p. 251). This project, which is under the direction of Dr. PIERRE LOUIS COUFFIGNAL, comprises two parts—one an investigation of computational methods, and the other the design and construction of an electronic digital computer. Although the design of the machine is not yet complete, certain guiding principles have been outlined, and the development of necessary components has been undertaken.

[1] A. ERDÉLYI & JOHN TODD, "Advanced instruction in practical mathematics," *Nature*, v. 158, Nov. 16, 1946, p. 690–692. Dr. Erdélyi is also now in this country, at California Institute of Technology, preparing papers of HARRY BATEMAN for possible publication. —EDITOR.

OTHER AIDS TO COMPUTATION

BIBLIOGRAPHY Z–II

1. GEORGES BAUDOUIN, "Principe d'une règle à calcul présentant une échelle logarithmique de grande longueur," Acad. d. Sci., Paris, *C.R.*, v. 224, Jan. 1947, p. 96–97.

Last paragraph: "Ainsi, avec une règle de dimensions restreintes, soit 20 cm. sur 4 cm., *on peut*, grâce au découpage de l'échelle logarithmique de 10 à 100 en dix parties superposées, *obtenir une précision environ dix fois plus grande qu'avec les règles ordinaires*." See *Math. Rev.*, v. 8, 1947, p. 289, E. LUKACS.

2. The GLOBE-HILSENRATH *Azimuth Computer. Instructions for the use of the Globe-Hilsenrath Azimuth Computer*. A. M. Messer & Co., 18 40th St., Irvington, N. J. 1945. 12 p. 12.7 × 17.7 cm. Instrument and Instructions $8.75. Distributor for School and College trade: Yoder Instruments Co., East Palestine, Ohio.

This computer consists of a white vinylite sheet, 23.3 × 24 cm., carrying on each of its faces a printed grid, and a transparent plastic protractor with movable radial arm which can