agreement with Watson is not complete, and a value known to be in error in Watson is correct in the table under review.

J. C. P. MILLER

23 Bedford Square
London W.C. 1

[1] G. N. WATSON, *Treatise on the Theory of Bessel Functions*, Cambridge, 1922, second ed. 1944 [*MTAC*, v. 2, p. 49–51.]

94[V].—BALLISTIC RESEARCH LABORATORIES, Aberdeen Proving Ground, Md. *Supersonic Flow past Cone Cylinders*. [See Note **113**.]

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, 225 Far West Building, National Bureau of Standards, Washington 25, D. C.

TECHNICAL DEVELOPMENTS

Characteristics of the Institute for Numerical Analysis Computer

In January 1949, members of the staff of the Institute for Numerical Analysis[1] began the development and construction of a high-speed electronic digital computer. As of December 1, 1949, the central computer was approximately eighty per cent completed. The group responsible for the building of this machine is composed of, besides the author, three engineers, three junior engineers, and four technicians. In addition, one mathematician is assigned to the coding and programming of problems to be run on the machine.

Information is stored and processed in the computer in units called words, a word consisting of 41 binary digits. This word length is determined by the number of words which can be stored in the computer's high-speed memory.

Words in the machine sense may represent (1) numerical information, (2) instructions to the computer, and (3) alphabetic information.

In the case of numerical information, one binary digit of a word is used for the sign and 40 binary digits are available for numerical data. Numbers are stored in the memory as absolute value and sign. In the arithmetic unit, negative numbers may be converted to complementary[2] form to keep the operational algorithms relatively simple. Thus, negative numbers involved in addition, subtraction, and compare are complemented upon arrival in the arithmetic unit. In the multiplication, extract, input, and output commands, negative numbers are not complemented.

Numbers may be represented in many different ways. For example, a word may represent a signed-binary number lying somewhere between -2^{40} and $+2^{40}$, or the binary point may be ahead of the most significant digit in which case a word lies in the range -1 to $+1$; the built-in arithmetic operations handle numbers in either of these forms. The word may, on the other hand, represent a signed-ten-decimal-digit number where each decimal digit is represented as a four-digit-binary number. A floating representation may be used where the first digit represents the sign, the next eight digits represent the exponent b, and the next 32 digits represent the significant digits of the number in binary form. A floating decimal representation may also be used giving numbers of eight significant digits ranging in absolute value from about 10^{-50} to 10^{+50}. More than one word can be used to represent a number to effect much greater precision or range of values.

Floating operations have been coded (as will be explained later in this paper) to provide for the addition, subtraction, multiplication, or division of two numbers of the form $a \cdot 2^b$ (with a and b stored in the same address). All four floating operations involve about 87 instructions. These floating operations are performed in about 3, 3, 6, and 14 milliseconds, respectively. Compare these times with 64, 64, and 384 *micro*-seconds, the times required for doing ordinary binary addition, subtraction, and multiplication.

Instructions are subclassified into three classes. These three classes are *command words*, *control words*, and *code words*.

The *command words*, commonly called commands, are explicitly "understood" and "obeyed" by the computer. A command causes the computer to perform a specific operation and gives the necessary information about where to get the needed data and what to do with the results. At present, the command system used by the Institute's computer consists of a set of thirteen commands.[3] Eight of the thirteen are what might be termed basic commands; the other five are variations of these eight. Such a command system is known as a four-five address code, with four addresses generally in the command word and the fifth address in a control counter. The function, or operation, of a particular command is denoted by F, the four addresses of the command by α, β, γ, and δ, and the fifth address by ϵ which normally determines the address of the next command to be obeyed.

The size of the memory determines the length of each address which, in the case of a 512 word memory, is nine binary digits. Thus, 36 binary digits are used to denote the four addresses, α, β, γ, and δ. Four digits are used to denote F; of these four, three are used to define the eight basic command words, and one is used to denote modifications of the eight. There is one spare digit in the command word.

The thirteen operations, or functions, of the Institute's computer, as well as the meanings of the addresses for the various command words, are given in Table 1.

In order to change automatically the course of operation in the calculator when certain bounds have been reached,[4] there are conditional and unconditional transfers of control commands. A conditional transfer is accomplished with compare commands. An unconditional transfer is accomplished with certain special commands (A_1, S_1, and M_1), wherein the fourth address, δ, determines the next command.

In the table, special compare might well be called absolute compare in that the absolute values of the numbers are compared. Since the result of the subtraction in the compare operation is put back into the memory, this command can be used to obtain the absolute value of a number in one operation by comparing the number with zero. The compare command can also be used for an operation called tally, as follows: Assume that it is desired to repeat a routine fifty times stored in memory locations 31 to 37, inclusive.

TABLE 1

Meanings of the Addresses for the Commands

Command	α	β	γ	δ	F
Add	Address of Augend	Address of Addend	Address of Sum	Address of Next Command if Overflow	A
Special Add	Address of Augend	Address of Addend	Address of Sum	Address of Next Command	A_1
Subtract	Address of Minuend	Address of Subtrahend	Address of Difference	Address of Next Command if Overflow	S
Special Subtract	Address of Minuend	Address of Subtrahend	Address of Difference	Address of Next Command	S_1
Multiply	Address of Multiplier	Address of Multiplicand	Address of Product Rounded Off		M
Special Multiply	Address of Multiplier	Address of Multiplicand	Address of Product Rounded Off	Address of Next Command	M_1
Product	Address of Multiplier	Address of Multiplicand	Address of Most Significant Part of Product	Address of Least Significant Part of Product	P
Compare	Address of Minuend	Address of Subtrahend	Address of Difference	Address of Next Command if Difference is Non-negative	C
Special Compare	Address of Minuend	Address of Subtrahend	Address of Difference of Absolute Values	Address of Next Command if Difference of Absolute Values is Non-negative	C_1
Extract	Address of Extractor (Determines Digits to be Extracted)	Address of Extractee	Address of Extracted and Shifted Result	If Second Digit of δ is {0—Shift Left, 1—Shift Right} Other Digits Tell Number of Places to Shift	E
Input	Address of Incoming Information	Drum Address if used		Selects Input Device	I
Special Input	(Incoming Information Goes to Address ϵ)			Selects Input Device	I_1
Output	Address of Outgoing Information	Drum Address if used		Selects Output Device	O

At the beginning of the routine, let the number "49" be placed in address 10. Suppose that address 1 stores the number "1." Place the command

10, 1, 10, 31, C

in address 38. The first time this command immediately following the routine is obeyed, the number in address 10 will be reduced to "48," and the next command will come from address 31. Each time the routine is repeated, the command in address 38 will be executed, with a reduction of the number in address 10 by one. After the routine has been performed 49 times, the number in address 10 will be zero; the 50th time the difference formed by the compare order will be negative, and the next command will come from address 39 (the value of ϵ). Thus, the desired routine will be repeated exactly fifty times.

In the case of the normal addition and subtraction commands, overflow[5] is automatically detected. An extra digit is provided in the arithmetic unit in the most significant end of the A and M[6] registers so that for normal addition and subtraction commands a "1" in this position will cause the next command to come from address δ instead of address ϵ. In the case of the compare command, the proper result of the subtraction is in the A register, but here, as in the addition and subtraction commands, the most significant digit (overflow digit) is not put back into the memory. In the compare command there is no means of detecting overflow.

The extract command provides for obtaining the logical product of two words[7] and shifting the result an arbitrary amount; it may also be used to delete arbitrary parts of a word. Its primary purpose is to assist automatically in fabricating new commands during the computation and to sort out the exponent from the significant figures in floating-point operations.

Special input is the command used to insert information into the computer when it is first put into operation. A word consisting entirely of zeros in the F portion is used to designate special input. Thus, when the memory is cleared, every word is a special input command, and, with these commands, the destination of the incoming data is determined by ϵ. After ϵ has been increased by "1," it determines the source of the next command. Once started, the calculator will count through its complete high-speed memory and read in information from the teletype tape.[8] After the memory has been filled, the ϵ counter steps to zero, and the calculator obeys the command stored in that position.

The input and output commands may specify the magnetic drum as a source or destination of words to be transferred. The drum itself will be about eight inches in diameter and two feet long and will hold about 8,192 words.[9] Initially only one word at a time can be transferred between it and the high-speed memory. The drum will rotate at 3600 revolutions per minute which means that the average access time for a word will be 8 milliseconds. As soon as possible after the computer is put into operation, counting facilities will be added to the control to enable the high-speed memory to be operated in synchronism with the drum. This will make it possible to transfer a whole vector in one revolution of the drum, thus greatly decreasing the average access time.

In general, the drum will serve as an auxiliary storage for numbers, instructions, and function tables. The drum is used instead of extra tape units because it offers better accessibility to information and requires no manual handling. The drum does not hold as much information as a magnetic tape unit, but its size seems adequate for the purposes mentioned above. For greater storage several drums may be operated in synchronism.[10]

The second class of instructions, *control words*, are not directly obeyed by the calculator, nor are they a direct part of the calculation; yet in various ways, they control the course of the computation or enter into the arithmetic-like operations which are performed upon command words. Control words may serve as parameters which determine the number of repetitions of certain routines; they may be the bounds used to stop certain computational processes; or they may serve as factors in the logical products or extraction operations.

The third class of instructions, *code words*,[11] specify (usually in one word) a whole sequence of procedures for the computer to follow. Thus, one may specify, for example, a scalar multiplication with only one code word. This code word is made up of parameters which specify the common factor (that is, which specify the address in the high-speed memory), the location of the elements of the vector (say, by specifying the address of the first element and the number of terms in the vector), and the location for the result.

When prepared for the computer, a problem consists of a sequence of words called a *main routine.* This main routine is made up of instructions (commands, code words, and control words) together with the numerical constants appropriate to the problem.

The code words in the main routine contain the parameters or addresses necessary to call into action other sequences of instructions, usually called *subroutines.* Subroutines (and code words) exist for such procedures as floating operations, standard iteration formulae, vector operations, integration formulae, and so forth, which are frequently used in the course of doing a computation. A subroutine may itself contain code words which call into use other subroutines. (All of the more frequently-used subroutines will be stored on the magnetic drum, thus being comparatively easily accessible.)

A routine known as the *interpretation routine* keeps track of the place in the main routine, causes segments of the main routine to be brought into the high speed memory, inspects each successive instruction in the main routine to see if it is a command (in which case it is obeyed) or a code word (in which case the interpretation routine extracts an entry[12] from the code word and sends the computer to the appropriate subroutine). A subroutine may itself change the code word being considered by the interpretation routine and, thus, cause the computer to carry out other subordinate activities before going on with the main routine.

Once the appropriate subroutines are established, the task of coding a complex problem is very much simpler. For example, the problem of solving systems of simultaneous linear equations of orders up to 125 by the elimination method has been worked out with the five main subroutines listed in Table 2. The arithmetic operations performed by these subroutines are of

TABLE 2

Routines for Solving Simultaneous Linear Equations

Name	Code	No. of Instructions	Time for Execution in μsecs.	Purpose
Vector Input	α, γ, n, VI	8	$(n + \frac{1}{16})8000$	n words transferred from addresses $\gamma + i$ on drum to addresses $\alpha + i$ of high-speed memory, $i = 0, 1, \cdots, n - 1$.
Vector Output	α, γ, n, VO	9	$(n + \frac{1}{16})8000$	Converse of VI, that is, transfer from high-speed memory to drum.
Floating Operations	α, β, γ, A	88	3,000	Adds, subtracts, multiplies, or divides the operands and puts result in γ.
	α, β, γ, S		3,000	
	α, β, γ, M		6,000	
	α, β, γ, D		14,000	
Vector Constant Product	α, β, γ, n, VC	10	$6500n + 448$	Multiplies word in β by words in $\alpha + i$; answers go into $\gamma + i$, $i = 0, 1, \cdots, n - 1$.
Vector Subtraction	α, β, γ, n, VS	10	$3500n + 448$	Subtracts words in $\beta + i$ from those in $\alpha + i$, answers go into $\gamma + i$, $i = 0, 1, \cdots, n - 1$.

the floating binary type, and each such arithmetic operation is in itself a subroutine. By using code words the coding for this simultaneous linear equation problem is reduced to laying out a sequence of about thirty instructions. It is convenient to have all routines and constants, and two rows of the matrix stored in the high-speed memory; the figure of 125 is based on a 512 word memory. The approximate time required to solve a set of sixty equations under various conditions is given in Table 3. In fixed-point operation, a division routine replaces the floating routines of Table 2. The largest pivot may be used in each reduction, and scale factors may have to be introduced. Table 2, Table 3, and some of the routines were worked out by ROSELYN LIPKIS of the Machine Development Unit at the Institute for Numerical Analysis.

TABLE 3

Time Required to Solve Sixty Simultaneous Linear Equations[13]

	Fixed Binary Point		Floating Binary Point	
	Synchronized Drum	Non-Synchronized Drum	Synchronized Drum	Non-Synchronized Drum
Computing Time	3.5 min.	3.5 min.	19 min.	19 min.
Transfer Time (to and from the drum)	1.0 min.	30.0 min.	1 min.	30 min.
Totals	4.5 min.	33.5 min.	20 min.	49 min.

H. D. HUSKEY

Institute for Numerical Analysis
Univ. of California, Los Angeles

[1] The Institute is one of four sections of the National Applied Mathematics Laboratories of the National Bureau of Standards. It is located on the campus of the University of California at Los Angeles. The computing machine discussed in this paper is financed by the Air Materiel Command of the United States Air Force.

[2] $-N$ is converted to $2^{42} - N$ or $2^{46} - N$, depending upon whether the size of the memory is 512 or 1024 words, respectively.

[3] These commands are a variation of a set proposed by E. F. MOORE while he was working in the National Applied Mathematics Laboratories of the National Bureau of Standards.

[4] For example, in the integration of the exterior ballistic equation, the procedure must change when a shell has completed its flight; or in a square root iteration, the procedure changes when two successive iterants are sufficiently close together.

[5] Addition, subtraction, and compare may produce results which exceed the capacity of the memory cells.

[6] This extra digit in M is needed for the complement process.

[7] Or it may be used for deleting arbitrary parts of a word.

[8] At a later date, magnetic tape may also be used for inserting information into the computer.

[9] Professor P. MORTON of the University of California at Berkeley is constructing such a drum.

[10] Professor F. C. WILLIAMS of Manchester University, England, has operated a drum in synchronism with a master oscillator.

[11] Code words have been variously termed abbreviated code instructions, quasi-commands, shorthand commands, abbreviated commands, and coded commands. The term "code word" has been selected in preference to these other terms by the author to distinguish more clearly this type of instruction from the explicit commands.

[12] An entry is the address of the command in the subroutine which should be obeyed first.

[13] This table is based on storing three rows of the matrix in the high-speed memory. For more than approximately eighty equations, only two rows can be stored, and input-output is increased by fifty per cent.

Discussions

Statistical Treatment of Values of First 2,000 Decimal Digits of e and of π Calculated on the ENIAC

The first 2,000 decimal digits of e and of π were calculated on the ENIAC by Mr. G. Reitwiesner and several members of the ENIAC Branch of the Ballistic Research Laboratories at Aberdeen, Maryland (*MTAC*, v. 4, p. 11–15). A statistical survey of this material has failed to disclose any significant deviations from randomness for π, but it has indicated quite serious ones for e.

Let D_n^i be the number of digits i (where $i = 0, 1, \cdots, 9$) among the first n digits of e or of π. The count begins with the first digit left of the decimal point. If these digits were equidistributed, independent random variables, then the expectation value of each D_n^i (with n fixed and $i = 0, 1, \cdots, 9$) would be $n/10$, and the χ^2 would be

$$a_n = \chi_n^2 = \sum_{i=0}^{9} \left(D_n^i - \frac{n}{10}\right)^2 \bigg/ \frac{n}{10}.$$

The system of the D_n^i's (where $i = 0, 1, \cdots, 9$) has 9 degrees of freedom. Therefore let

$$p = P^{(k)}(a) = \frac{1}{2^{\frac{1}{2}k}\Gamma(\frac{1}{2}k)} \int_0^a e^{-\frac{1}{2}x} x^{\frac{1}{2}k-1} dx$$

be the cumulative distribution function of $a = \chi^2$ for k degrees of freedom. Then

$$p_n = P^{(9)}(a_n)$$

is a quantity which would be equidistributed in the interval [0, 1], if the underlying digits were equidistributed independent random variables.

Consider $n = 2000$. In this case, the D_n^i's for e are

(1) 196, 190, 208, 202, 201, 197, 204, 198, 202, 202.

Hence $a_n = \chi_n^2 = 1.11$ and $p_n = .0008$. The D_n^i's for π are

(2) 182, 212, 207, 189, 195, 205, 200, 197, 202, 211.

Hence $a_n = \chi_n^2 = 4.11$ and $p_n = .096$.
The e-value of p_{2000} is thus very conspicuous; it has a significance level of about 1:1250. The π-value of p_{2000} is hardly conspicuous; it has a significance level of about 1:10.

The relevant fact about the distribution (1) appears upon direct inspection. The values lie too close to their expectation value, 200. Indeed their absolute deviations from it are

4, 10, 8, 2, 1, 3, 4, 2, 2, 2,

and hence their mean-square deviation is $22.2 = 4.71^2$, whereas in the random case the expectation value is $180 = 13.4^2$.

In order to see how this peculiar phenomenon develops as n increases to 2000, $a_n = \chi_n^2$ and p_n of e have been determined from D_n^i for the following smaller values of n

n	$a_n = \chi_n^2$	p_n
500	6.72	.33
1000	4.82	.15
1100	5.93	.25
1200	4.03	.093
1300	3.83	.080
1400	4.74	.145
1500	3.69	.070
1600	2.47	.019
1700	3.22	.046
1800	2.85	.031
1900	2.22	.013
2000	1.11	.0008

These numbers show that the abnormally low value of p_n which is so conspicuous at $n = 2000$ does not develop gradually, but makes its appearance quite suddenly around $n = 1900$. Up to that point, p_n oscillates considerably and has a decreasing trend, but at $n = 2000$ there is a sudden dip of quite extraordinary proportions.

Thus something number-theoretically significant may be occurring at about $n = 2000$. A calculation of more digits of e would therefore seem to be indicated. A conversion to a simpler base than 10, say 2, may also disclose some interesting facts.

We wish to thank Miss HOMÉ MCALLISTER of the ENIAC Branch of the Ballistic Research Laboratories for sorting the digital material on which the above analyses are based, and Professor J. W. TUKEY, of Princeton University, for discussions of the subject.

Since the above was written (November 9, 1949), the ENIAC Branch of the Ballistic Research Laboratory very obligingly followed our suggestion and calculated the following 500 additional digits[1] of e. These should replace the last 10 digits of the value of e given in *MTAC*, v. 4, p. 15.

55990	06737	64829	22443	75287	18462	45780	36192	98197	13991
47564	48826	26039	03381	44182	32625	15097	48279	87779	96437
30899	70388	86778	22713	83605	77297	88241	25611	90717	66394
65070	63304	52795	46618	55096	66618	56647	09711	34447	40160
70462	62156	80717	48187	78443	71436	98821	85596	70959	10259
68620	02353	71858	87485	69652	20005	03117	34392	07321	13908
03293	63447	97273	55955	27734	90717	83793	42163	70120	50054
51326	38354	40001	86323	99149	07054	79778	05669	78533	58048
96690	62951	19432	47309	95876	55236	81285	90413	83241	16072
26029	98330	53537	08761	38939	63917	79574	54016	13722	36188

This makes it possible to extend the table of $a_n = \chi_n^2$ and p_n up to $n = 2500$

n	$a_n = \chi_n^2$	p_n
2100	1.94	.0075
2200	2.02	.0088
2300	1.65	.0041
2400	1.70	.0046
2500	1.90	.0070

Thus the values of p_n for $2100 \leqslant n \leqslant 2500$ are still significantly low but higher than the value of p_n at $n = 2000$.

Note that the general size and trend of p_n, as well as its sudden deviation at $n = 2000$, indicate a non random character in the digits of e.

More detailed investigations are in progress and will be reported later.

Los Alamos Scientific Laboratory — N. C. METROPOLIS

Ballistic Research Laboratories — G. REITWIESNER

Institute for Advanced Study
Princeton, N. J. — J. VON NEUMANN

[1] Both e and $1/e$ were computed somewhat beyond 2500 D and the results checked by actual multiplication.

Notes on Numerical Analysis—2
Note on the Condition of Matrices

1. The object of this note is to establish the following theorem.

THEOREM. *Let A be a real $n \times n$ non-singular matrix and A' be its transpose. Then AA' is more "ill-conditioned" than A.*

This theorem confirms an opinion expressed by Dr. L. FOX[1] based on his practical experience. The term "condition of a matrix" has been used rather vaguely for a long time. The most common measure of the condition of a matrix has been the size of its determinant, ill-conditioned matrices being those with a "small" determinant. With this interpretation imposed, the theorem is clearly correct. More adequate measures of the condition of a matrix have been proposed recently by JOHN VON NEUMANN & H. H. GOLDSTINE[2] and by A. M. TURING.[3] Their definitions concern all matrices, not just the ill-conditioned ones, characterized by very large condition numbers. The following two of these definitions will form a basis for the proof of the above-mentioned theorem:

The P-condition number is $|\lambda_{\max}|/|\lambda_{\min}|$, where $\lambda_{\max}$ and $\lambda_{\min}$ are the characteristic roots of largest and smallest modulus.[2]

The N-condition number is $N(A)N(A^{-1})/n$, where[3]

$$N(A) = (\sum_{i,k} a_{ik}^2)^{\frac{1}{2}}.$$

2. Proof of the theorem in the P case:

Let λ_i be the characteristic roots of A and μ_i those of AA' (which are in general distinct from the squares of the absolute values of λ_i). E. T. BROWNE[4] has shown that

$$\mu_{\min} \leqslant \lambda_i\bar{\lambda}_i \leqslant \mu_{\max}.$$

From this it follows that

$$1 \leqslant \left|\frac{\lambda_{\max}}{\lambda_{\min}}\right| \leqslant \left|\frac{\lambda_{\max}}{\lambda_{\min}}\right|^2 \leqslant \frac{\mu_{\max}}{\mu_{\min}},$$

which implies the required result.

3. Proof of the theorem in the N case:

It is known that $N(A)$ is the square root of the trace of AA' and therefore equal to $(\sum \mu_i)^{\frac{1}{2}}$. The numbers μ_i are all positive since AA' is symmetric and positive definite. Since the characteristic roots of $A'A$ and AA' are the same and since the characteristic roots of the inverse of a matrix are the reciprocals of those of the original matrix, it follows that

$$N(A^{-1}) = (trA^{-1}(A^{-1})')^{\frac{1}{2}} = (tr(A'A)^{-1})^{\frac{1}{2}} = (\sum \mu_i^{-1})^{\frac{1}{2}}.$$

The N-condition number of A is therefore

$$\frac{1}{n}(\sum \mu_i)^{\frac{1}{2}}(\sum \mu_i^{-1})^{\frac{1}{2}}.$$

In a similar way it can be shown that the N-condition number of AA' is

$$\frac{1}{n}(\sum \mu_i^2)^{\frac{1}{2}}(\sum \mu_i^{-2})^{\frac{1}{2}}.$$

The theorem follows from the inequality

$$\sum \mu_i^2 \sum \mu_i^{-2} \geqslant \sum \mu_i \sum \mu_i^{-1},$$

which is in fact true for all real and positive numbers. (It is, indeed, true when the first power on the right is replaced by an arbitrary power r and the second power on the left by a power $s > r$.) The proof of the inequality is as follows:

$$\begin{aligned}
\sum \mu_i^2 \sum \mu_i^{-2} - \sum \mu_i \sum \mu_i^{-1} \\
&= n + \sum_{i \neq j} \mu_i^2 \mu_j^{-2} - n - \sum_{i \neq j} \mu_i \mu_j^{-1} \\
&= \sum_{i<j} (\mu_i^2 \mu_j^{-2} + \mu_j^2 \mu_i^{-2}) - \sum_{i<j} (\mu_i \mu_j^{-1} + \mu_j \mu_i^{-1}) \\
&= \sum_{i<j} \{(\mu_i \mu_j^{-1} + \mu_j \mu_i^{-1})(\mu_i \mu_j^{-1} + \mu_j \mu_i^{-1} - 1) - 2\} \geqslant 0,
\end{aligned}$$

since

$$\mu_i \mu_j^{-1} + \mu_j \mu_i^{-1} \geqslant 2, \qquad \text{and} \qquad \mu_i \mu_j^{-1} + \mu_j \mu_i^{-1} - 1 \geqslant 1.$$

There is equality if and only if

$$\mu_1 = \mu_2 \cdots = \mu_n.$$

OLGA TAUSSKY

NBSMDL

[1] In a course of lectures given at the British Admiralty by himself and D. H. SADLER in 1949.

[2] J. VON NEUMANN & H. H. GOLDSTINE, "Numerical inverting of matrices of high order," Amer. Math. Soc., *Bull.*, v. 53, 1947, p. 1021–1099. (These authors consider symmetric matrices only, but it is reasonable to apply the definition to the general case.)

[3] A. M. TURING, "Rounding-off errors in matrix processes," *Quart. Jn. Mech. Appl. Math.*, v. 1, 1948, p. 287–308.

[4] E. T. BROWNE, "The characteristic equation of a matrix," Amer. Math. Soc., *Bull.*, v. 34, 1928, p. 363–368.

BIBLIOGRAPHY Z-XI

1. E. G. ANDREWS, "The Bell Computer, Model VI," *Electrical Engineering*, v. 68, 1949, p. 751–756, 7 figs., 5 tables. 22.2 × 29.5 cm.

Controlled from remote stations, this new digital computer of the relay type reduces punched-tape instructions to a minimum. With novel control features similar to those used in recent automatic dial-telephone developments, this "upper-class" computer possesses six "intelligence levels." Subordinate levels are capable of solving problems such as complex-number multiplication without special guidance.

Author's summary

2. W. R. Ashby, "Design for a brain," *Electronic Engineering*, v. 20, 1948, p. 379–383, figs.

An ideal "thinking machine" must possess negative feedback—or the ability to look after itself cleverly by correcting all deviations from a central, optimal state. The author describes the homeostat, an electro-magnetic machine which is capable of selecting its own arrangement of feedbacks. The designer has merely provided it with plenty of variety so that, if the basic conditions are altered, it can adapt itself. The author points out that the homeostat is still too larval to be a serviceable synthetic "brain."

The merits of the ENIAC, the ACE, and the homeostat are compared in playing the game of chess. The latter needs no detailed instructions—but rather a method by which it is informed of the occurrence of illegal moves and mates. Such a machine, if perfect, could in the opinion of the author eventually play with a subtlety and depth of strategy beyond that of its designer.

There are two apparent disadvantages to this machine. First, the machine will develop a temperament which will probably be manifested in a form too complex for the designer's understanding. More serious in implication is the selfishness of the machine; it will judge the appropriateness of an action by how it is affected by the feedback.

NBSMDL Edith T. Norris

3. Isaac L. Auerbach, J. Presper Eckert, Jr., Robert F. Shaw, & C. Bradford Sheppard, "Mercury delay-line memory using a pulse rate of several megacycles," I.R.E., *Proc.*, v. 37, 1949, p. 855–861, figs.

A mercury delay line memory system for electronic computers, capable of operating at pulse repetition rates of several megacycles per second, has been developed. The high repetition rate results in a saving in space and a reduction in access time.

Numerous improvements in techniques have made the high repetition rate possible. The use of the pulse envelope system of representing data has effectively doubled the possible pulse rate; the use of crystal gating circuits has made possible the control of signals at high pulse rates; and a multichannel memory using a single pool of mercury has simplified the mechanical construction, reduced the size, and made temperature control much easier.

The memory system described makes possible a significant increase in the over-all speed of an electronic computer.

Author's summary

4. Kay Howard Barney, "The binary quantizer," *Electrical Engineering*, v. 68, 1949, p. 962–967, figs.

The binary quantizer is a new device for translating a time-varying voltage in an analogue computer into a binary number—the size of each being directly proportional to the instantaneous value of the quantity being measured. This makes possible the use of digital techniques in an analogue computer. Although primarily designed for computing work, this apparatus may find use where quantization of a continuously varying function is desired, as in pulse code modulators, automatic metering devices, and recording machines employing typed or printed numbers.

5. WARREN H. BLISS, "Electronic digital counters," *Electrical Engineering*, v. 68, 1949, p. 309–314, figs.

Recent advancements in all branches of science have created demands for high-speed counting devices. The development of the electronic digital counters, which are used for many purposes in the large computers, has made such fast computations possible. These counters, which use the binary system, are described in detail in this article.

Author's summary

6. D. R. HARTREE, *Calculating Instruments and Machines*, Univ. of Illinois Press, Urbana, Ill., 1949, 138 + ix pages, 68 figs., $4.50. 17 × 25.3 cm.

Professor Hartree has written for the slightly mathematical reader an able summary of existing and projected computers and of some of the mathematical problems which can be solved with their help. His interpretation of the subject is authoritative, as he has personally contributed to many of the phases of the field of computing machinery. Everyone has been intrigued by the Hartree Differential Analyser built of Meccano parts at the University of Manchester in England. More recently Professor Hartree has solved problems on several of the large-scale digital machines, so that he speaks from personal knowledge and experience when he discusses these computers.

In some respects the book has a distinctly English flavor; the terms "instrument" and "machine" are used to denote "analogue" and "digital" computers, respectively. In addition, there are numerous references to developments in England which may be less familiar to American readers than the parallel developments in this country.

After an introductory chapter discussing the distinction between instruments and machines, the author devotes two chapters to the various differential analyzers and to their uses in solving ordinary and partial differential equations. Chapter 4 concludes the section on instruments with a rapid sketch of devices for solving systems of linear algebraic equations, for finding the roots of polynomials, and for integration. (It was noted that no mention was made of the TRAVIS-HART electrical root-finder.)

The section on digital machines opens with a much needed chapter on terminology. A chapter is devoted to BABBAGE's astonishing Analytical Engine of one hundred years ago, and another to existing computers—mechanico-electric, relay, and electronic. Chapter 8 contains schematics and discussions of machines now under construction or in the planning stage.

The book closes with an especially valuable chapter on the methods of numerical analysis, emphasizing the "machine's-eye view" of these methods. The author mentions iterative methods and the solution of algebraic, ordinary and partial differential equations, with useful hints based on his own experience.

Our thanks are due to Professor Hartree for a pleasing and a useful book.

G. R. STIBITZ

393 South Prospect St.
Burlington, Vermont

7. Harry D. Huskey, "The status of high-speed digital computing systems," *Mechanical Engineering*, v. 70, 1948, p. 975–977, bibl.

A brief outline of the history of various computing devices precedes a discussion of technical aspects of many of the current large-scale computers being built all over the world. The author discusses the rôle of computing in science emphasizing the continuing necessity for hand computers and desk machines as well as the high-speed computers. He goes on to point out that the current computers can substitute as a universal model for a large class of model experiments.

Edith T. Norris

NBSMDL

8. T. Kilburn, "The University of Manchester universal high-speed digital computing machine," *Nature*, v. 164, 22 Oct. 1949, p. 684–687.

The author describes the binary high-speed electronic computer being developed under the direction of Professor F. C. Williams, with the active assistance of the Telecommunications Research Establishment, Malvern.

9. Warren S. McCulloch, "The brain as a computing machine," *Electrical Engineering*, v. 68, 1949, p. 492–497, bibl.

The author, a medical doctor, employs electrical engineering terminology to show how the brain may be likened to a digital computing machine consisting of ten billion relays called neurons. To carry the analogy further, the performance of the brain is governed by inverse feedback; subsidiary networks secure invariants, or ideas; predictive filters enable us to move toward the place where the object will be when we get there; and complicated servomechanisms enable us to act with facility and precision. Disorders of function are explained in terms of damage to the structure, improper voltage of the relays, and parasitic oscillations.

Author's summary

10. J. Howard Parsons, "Electronic classifying, cataloging, and counting systems," I.R.E., *Proc.*, v. 37, 1949, p. 564–568, figs.

The determination of the distribution of a series of physical events according to magnitude is important in the study of the associated physical laws. Previous methods of determining this magnitude distribution were slow and cumbersome. Three new electronic systems which operate on events at a very high rate have been developed by the author while at Oak Ridge National Laboratory, and these are described.

The analyzers can be used to determine the magnitude distribution of any series of physical events, if the characteristic under observation can be translated into a proportional voltage pulse. Two applications are discussed, and the advantages of the analyzers over other systems are shown.

Author's summary

11. T. Pearcey, "Modern trends in machine computation," *Australian Journal of Science*, Supplement, v. 10, Feb. 21, 1948, 20 p., 18 figs.

The object of this paper is "to indicate the general structure and organization of high-speed computing machines and the avenues which have been opened up for elaborate high-speed calculations and to suggest the advantages and disadvantages of the various systems available." The author briefly discusses the requirements for an ideal high-speed computer, the basic organic structure, number representation systems, machine components, and simultaneous versus parallel operation. He briefly outlines computer developments which were then current and predicts that in the future these machines may possibly tackle the symbolic procedures of pure mathematics provided the premises are supplied to the machine. The uses of these machines are not limited to the solution of mathematical problems; for example, they could be used to set up efficient, speedy filing systems in industry.

EDITH T. NORRIS

NBSMDL

12. TATSUJIRO SHIMIZU & YOICHI KATAYAMA, "Solution of non-linear equations by punched card methods," *Math. Japonicae*, v. 1, 1948, p. 92–97.

This paper discusses first the solution of simultaneous algebraic equations with particular reference to finding the initial approximation to the roots of the equation using punched card techniques. After the initial difficulty of obtaining the first approximation has been surmounted, closer approximations can easily be found using the NEWTON or other iterative techniques. The author specifically applies his method to the solution of an ordinary differential equation by reducing it to the solution of a system of simultaneous equations.

EDITH T. NORRIS

NBSMDL

13. M. V. WILKES, "Electronic calculating-machine development in Cambridge," *Nature*, v. 164, Oct. 1, 1949, p. 557–561, illustrs.

A very brief description of the EDSAC is presented, and the solution of a typical problem on the machine, i.e., the evaluation of AIRY's integral is discussed.

14. M. V. WILKES, "Programme design for a high-speed automatic calculating machine," *Jn. Sci. Inst. and Phys. in Industry*, v. 26, 1949, p. 217–220.

Problems intended to be solved with the aid of a high-speed digital calculating machine must first be reduced to a series of arithmetical operations. These, together with various auxiliary operations required for such purposes as keeping count of the number of times a particular routine has been repeated, must then be expressed in the code appropriate to the machine. The paper contains an account of some of the details of this process, with special reference to the EDSAC, a high-speed electronic machine in the University Mathematical Laboratory at Cambridge. Simple examples are given. These have been specially designed to illustrate the use of conditional orders and the way in which arithmetical operations may be performed on orders.

Author's summary

15. M. V. WILKES, "Progress in high-speed calculating machine design," *Nature*, v. 164, 27 Aug. 1949, p. 341–345.

An outline is presented of computer developments (chiefly those in England) which were discussed at a conference on high-speed automatic digital calculating machines held in the University Mathematical Laboratory, Cambridge, on June 22–25, 1949, to mark the completion of the EDSAC. [For an account of this conference see *MTAC*, v. 4, p. 51–53.]

16. R. WILSON WILLIAMS, "A survey of some recent advances in computing devices," *Science Progress*, v. 37, 1949, p. 42–52, bibl.

The paper presents a concise survey of recent trends in the design and use of computing devices. Emphasis is laid on those developments involving electronic and electro-mechanical principles.

NEWS

Institute for Numerical Analysis.—A course on electronic computing machines, with special emphasis on the NBS INA Computer, was initiated October 28, 1949, at the Institute for Numerical Analysis. This course, which meets once a week under the direction of Dr. H. D. HUSKEY, is for the purpose of acquainting interested persons with the possibilities and limitations of high-speed digital computers of this type. The course is open to personnel of other governmental and industrial agencies as well as to UCLA and INA staff members.

Institute of Mathematical Statistics.—The forty-first meeting of the Institute of Mathematical Statistics and the twelfth annual meeting was held in New York City on December 27–30, 1949, in conjunction with the meeting of the American Statistical Association, the American Association for the Advancement of Science, the American Mathematical Society, the Econometric Society, the Psychometric Society, the Mathematical Association of America, the Association for Computing Machinery, and the American Psychological Association.

On Friday afternoon, December 30, 1949, a special session of the meeting was devoted to computation at which the following papers were presented:

"Idiosyncrasies of automatically-sequenced digital computing machines" by IDA RHODES, NBS.

"Problem solving on large-scale automatic calculating machines" by W. D. WOO, Harvard University.

"A statistical application of the UNIVAC" by JOHN MAUCHLY, Eckert-Mauchly Computer Corporation.

Discussion by JAMES MCPHERSON, Bureau of the Census, and EMIL SCHELL, Office of the Air Comptroller.

In the first paper, the speaker stressed the fact that the so-called "brain" machines are in reality morons whose every move must be directed. Hence, problems to be put on these machines must be programmed in minute detail; the programmer must foresee every difficulty which might arise in the problem solution. Dr. Woo described the design features of the Mark III calculator, supplementing his discussion with slides. In the UNIVAC discussion, Dr. Mauchly brought out the fact that the UNIVAC was designed primarily for statistical applications; in this machine the emphasis is not on speed for its own sake but rather on rapid input and output speeds needed for Bureau of the Census problems. A detailed discussion of random number generation on the BINAC was also given. During the discussion period following these talks, typical problems to be solved on the UNIVAC were listed, and the time-saving features of this machine were emphasized.

International Business Machines Corporation.—A Seminar on Scientific Computation was held at Endicott, New York, Nov. 16–18, 1949 by the International Business Machines Corporation. The program for the seminar was as follows:

Wednesday Morning, November 16, LEON BRILLOUIN, IBM, Chairman

"The dynamics of nuclear fission" by DAVID L. HILL, Vanderbilt University.

"Monte Carlo calculations" by WILLIAM WOODBURY, Northrop Aviation Co.

"Modifications of the Monte Carlo Method" by HERMAN KAHN, The RAND Corporation.

Wednesday Afternoon, November 16, FRANK HOYT, Argonne National Laboratory, Chairman

"Analyzing exponential decay curves" by ALSTON HOUSEHOLDER, Oak Ridge National Laboratory.

"Eigenvalue problems related to the diffusion equation" by DONALD A. FLANDERS, Argonne National Laboratory and GEORGE SHORTLEY, Operations Research Office, Department of the Army, Johns Hopkins University.

Demonstration of the IBM Card-Programmed Electronic Calculator.

Thursday Morning, November 17, VERNER SCHOMAKER, California Institute of Technology, Chairman.

"Stochastic methods in statistical and quantum mechanics" by GILBERT KING, A. D. Little Company.

"Calculations of resonance energies" by GEORGE KIMBALL, Columbia University.

"Cam design calculations on the card-programmed electronic calculator" by E. A. BARBER, IBM.

Thursday Afternoon, November 17, ARTHUR ROSE, Pennsylvania State College, Chairman

"Distillation theory" by JOHN BOWMAN, Mellon Institute for Industrial Research.

"Calculation of multiple component systems" by STUART R. BRINKLEY, U. S. Bureau of Mines.

Friday Morning, November 18, A. H. TAUB, University of Illinois, Chairman

"The parabolic equation" by L. H. THOMAS, Watson Laboratory.

"Solutions of the wave equation" by PAUL HERGET, Director, Cincinnati Observatory.

"Sampling methods applied to differential equations" by JOHN CURTISS, National Applied Mathematics Laboratories, National Bureau of Standards.

Friday Afternoon, November 18, W. J. ECKERT, Watson Laboratory, Chairman

"On the distribution of KOLMOGOROV's statistic for finite sample size" by Z. W. BIRNBAUM, University of Washington.

"Specialized matrix calculations on the card-programmed electronic calculator" by H. R. J. GROSCH, Watson Laboratory.

Swedish State Board of Computing Machinery. BARK, the Swedish relay machine [*MTAC*, v. 4, p. 52–53] built in Stockholm by C. C. R. A. PALM and collaborators under the Swedish State Board of Computing Machinery is now completed and under trial running. A description of the machine will be given in a future issue of *MTAC*. The mechanical differential analyzer, built in Gothenburg by S. EKELÖF [*MTAC*, v. 3, p. 328] and collaborators has also been completed. Preliminary planning for an electronic computing device is going on under the direction of Ekelöf and Palm.

OTHER AIDS TO COMPUTATION

BIBLIOGRAPHY Z–XI

17. A. E. CARTER & D. H. SADLER, "The application of the National Accounting Machine to the solution of first-order differential equations," *Quart. Jn. Mech. Appl. Math.*, v. 1, 1948, p. 433–441. [*MTAC*, v. 3, p. 548.]

Details are given of the solution on the National machine of the first-order differential equation $y' = f(x, y)$ by the MILNE-STEFFENSEN method.