

153[F].—A. GLODEN, *Factorisation Table for the Numbers $N^3 + 1$, $N = 500$* . Six typewritten pages. Deposited in the UMT FILE.

The table is an extension of CUNNINGHAM's¹ table to $N \leq 200$. Of its 500 numbers 147 are completely factored. All unknown factors exceed 600000.

A. GLODEN

¹A. J. C. CUNNINGHAM, *Binomial Factorisations*. V. 6, London 1923, p. 140–141.

154[F].—F. GRUENBERGER, *Lists of Primes*. Two sheets tabulated from punched cards. Deposited in the UMT FILE.

The list of primes is extended from 50039981 to 50060033. There are 1131 primes between these limits. This is a continuation of a list given in UMT 148 [*MTAC*, v. 6, p. 167].

F. GRUENBERGER

Univ. of Wisconsin
Madison, Wis.

155[F].—R. J. PORTER, *Tables of Irregular Negative Determinants of exponent $3n$* . Typewritten manuscript on deposit in the UMT FILE.

The table gives the values of $D < 50000$ for which there is a determinant $-D$ which is irregular with an exponent of irregularity which is divisible by 3. [See DICKSON's *History*¹ for definition of these terms.]

The table is arranged by thousands. There are 11, 17, 21, \dots , 43 D 's in the first, second, \dots , 50th thousand, a total of 1718 D 's altogether. Most of these have exponent 3. Only $D = -17561$ has an exponent 6. Thirteen however have exponent 9. These are $-D = 3299, 6075, 11907, 17739, 23571, 24300, 27675, 29403, 33075, 35235, 41067, 46899$, and 47628. All other D 's have exponent 3.

The list was constructed by making extracts from some hundreds of the writer's series of determinants of class-number $3k$. To each determinant in these series belongs a class which has the property of duplicating into its own opposite; e.g., the determinant 21481 has a class (149, 71, 178) which duplicates into (26522, 8117, 2485) and thence by reduction to (2485, $-662, 185$), (185, $-78, 149$) and (149, $-71, 178$).

These extracts are filed in numerical order with their corresponding A values (e.g., 149 in the above) and any determinants which have more than one entry of A values against them are irregular (exp. $3n$).

It is found, in practice, that to make extracts from the series for each block of 10,000 determinants takes approximately 40 hours' work.

R. J. PORTER

266 Pickering Road
Hull, England

¹L. E. DICKSON, *History of the Theory of Numbers*, v. 3, Washington 1927, New York, 1934, Chap. 5.

AUTOMATIC COMPUTING MACHINERY

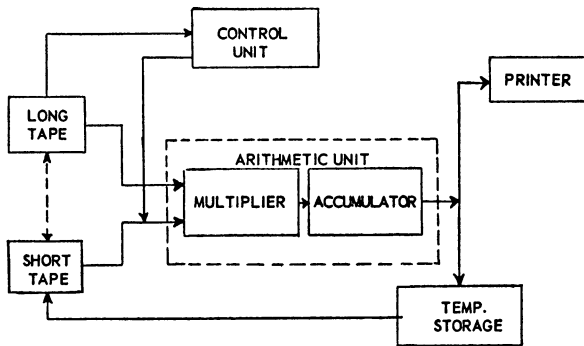
Edited by the Staff of the Machine Development Laboratory of the National Bureau of Standards. Correspondence regarding the Section should be directed to Dr. E. W. CANNON, 415 South Building, National Bureau of Standards, Washington 25, D. C.

TECHNICAL DEVELOPMENTS

THE USAF-FAIRCHILD SPECIALIZED DIGITAL COMPUTER

The USAF-Fairchild Computer (or "SPEC," for Special Purpose Electronic Computer) was designed and constructed at the NEPA Project¹ in Oak Ridge, Tennessee, to obtain solutions for a specific type of problems. These problems consisted of large systems of linear, algebraic equations involving up to 300 variables.

The SPEC, shown in the frontispiece, is an electronic, fixed-decimal machine, with all numbers lying between -1 and $+1$, and has four-digit capacity, although it will be explained later that eight digits are carried in the accumulator. By specializing the purpose of the computer, construction time, development of special components and circuitry were minimized. It was also possible to simplify the problem insertion so that little skill is required of the operating staff.



UNIT BLOCK DIAGRAM OF COMPUTER

FIG. 1.

The NEPA Computer employs the Gauss-Seidel iteration method for the solution of simultaneous, linear equations. By this method, a sequence of approximate solutions is obtained by successive iterations such that the limits of these solutions are the correct solutions of the equations.

When the problems have been arranged for solution, each of the equations in the set is solved for one of the variables as a function of a constant term and the remaining variables. In this manner, if the order of the set is N , there will be N equations of the following type:

$$(1) \quad X_i = \sum_{j=1}^{j=N} A_{ij}X_j + C_i \text{ where } A_{ii} = 0.$$

The method requires the summing of products and the addition of a constant term C_i . Values are assumed for each variable and substituted successively in the equations where at the conclusion of the evaluation of the right-hand member of each equation the new value for X_i replaces the old.

In the SPEC, two synchronous, magnetic tapes provide the required storage. These are shown in the block diagram of Fig. 1. A short, endless

tape stores the unknowns. As this tape revolves, the unknowns are read in order. Values for the unknowns, altered by successive iterations, will be repeated during succeeding tape revolutions. All the coefficients and constants are stored on the long tape, with accompanying instructions to the control unit. An instruction indicates to the computer which multiplier is to be selected for the number being read from the long tape and also signals the conclusion of each equation, at which time the new result is printed.

A common sprocket drive provides simultaneous reading of the two tapes. The synchronized drive permits arrangement of the coefficients on the long tape and the unknowns on the short tape in such a manner that a coefficient and the unknown by which it is to be multiplied are read by the two inputs simultaneously when they are needed during the course of the computation.

The computer is, basically, a summing device for a series of products. As shown in Fig. 1, the arithmetic unit consists of a multiplier coupled to an accumulator. The numbers received at the two inputs to the multiplier are multiplied and then added in the accumulator to the results obtained from previous similar operations. The control unit determines the sign of the operation to be performed by the accumulator, the product being either added to or subtracted from the total in the accumulator as the computation requires.

The multiplier is a diode matrix incorporating in its connections all the products of two decimal digits between 0 and 9. A single digit from each input number is coupled to the matrix, and the output of the matrix represents the product of those digits selected at the input. A complete product of two four-digit numbers requires sixteen operations, with the partial products being added, as obtained, into appropriate locations in the accumulator. The complete product requires approximately 400 microseconds.

The accumulator capacity is eight decimal digits. The eight-decimal-digit capacity prevents the formation of an excessive error due to repeated roundoffs in the summation of a series of products. Roundoff of the result is accomplished only at the conclusion of a series, prior to the transfer of a result out of the accumulator.

Each decimal digit is represented by a coded group of four binary digits. All operations in the accumulator are binary operations, using the "excess-three" code. The accumulator consists of 32 parallel-fed binary adder circuits. Sufficient inputs are available to handle the number and the required "excess-three" correction factor simultaneously.

At the conclusion of each series of products, and upon command from the control unit, the resulting total is rounded off and transferred both to a printer circuit, which provides a typed copy of the results for the operator, and to the temporary-storage register. This transfer occurs at the end of a sequence of operations, and the short tape is not then in the proper position for recording the new X over the old X . The temporary-storage register holds this X value until the point on the short tape is reached at which this value is to be recorded. The recording will normally occur during the solution of the next equation, and the computer will both record and use the X in the temporary-storage register during the solution of that equation. The recording location on the short tape is determined by the set of instructions placed on the long tape by the operator prior to the solution of the problem.

A total of 2206 vacuum tubes are used in the computer. Over half of these tubes are high-vacuum diodes, types 6AL5, which were used in preference to crystal diodes for matrix circuits.

Design and construction of the SPEC started in July 1949, and the first problem was solved on February 1, 1950. It was in continuous operation at the NEPA Project from the spring of 1950 until February 1951 and is being installed at the Oak Ridge National Laboratory. The computer will be called the "ORACLE" which is derived from Oak Ridge Automatic Computer for Linear Equations.

During the operating period, the computer was in an operating condition for approximately 85% of the total available time, with 15% of the time devoted to testing and servicing. Most machine errors were detected during the solution of problems, although test problems were used for periodic checks to assure proper operation of the machine. Check circuits, with visual and audible indications, are employed to detect improper operation of the machine in the most critical places, but no complete checking system is employed.

Throughout the period of operation, the computer was exceedingly useful as a means of obtaining solutions to systems of simultaneous, linear equations up to the limit of its useful capacity. In addition, problems involving matrix products, Fourier analysis, numerical integration, and matrix inversion were undertaken with considerable savings in time and effort.

The author wishes to acknowledge the outstanding contribution of Mr. V. G. LEWIS and his staff of technicians at the NEPA facility in Oak Ridge in the construction and assembly of the computer. Special recognition is due Mr. L. C. OAKES for his valuable contributions in testing, servicing and operating the machine.

J. J. STONE

Oak Ridge National Laboratory
Oak Ridge, Tenn.

¹ Nuclear Energy for the Propulsion of Aircraft Project, conducted by Fairchild Engine and Airplane Corporation under special contract with the United States Air Force.

DISCUSSIONS

FLOATING OPERATIONS ON THE EDSAC

Summary. The difficulties which arise when programming calculations for large automatic calculating machines which have a fixed decimal point are discussed. This leads to a consideration of the possibility of using floating decimal arithmetic for certain kinds of calculations. A method by which floating decimal arithmetic can be carried out with any fixed decimal-point machine is outlined and the scheme adopted for use with the EDSAC is described in detail.

This scheme is based on a special kind of subroutine which we call *interpretive*. This enables the programmer to use a new order code of his own choice. The 'orders' of a programme drawn up with such a code are selected under the action of the interpretive subroutine and interpreted in terms of sequences of orders which perform the required operations. With the EDSAC

a single address 'order' code has been adopted, similar to the ordinary order code, in which the arithmetical 'orders' are interpreted in terms of floating decimal arithmetic. Special 'orders' for simplifying counting operations and the modification of other 'orders' are provided. The 'order' code is described in detail and an example is given of a programme drawn up using this code. Other topics discussed are the use of auxiliary subroutines with the interpretive subroutine, a method of facilitating programme assembly, and techniques for using the input tape as a form of auxiliary store. Finally the times of operation of the individual 'orders' of the code are given, together with an estimate of the factor by which the calculation time is increased as a result of using floating decimal arithmetic for an entire calculation.

Introduction. One of the most tedious tasks arising in the programming of calculations for solution by fixed decimal-point machinery is to arrange the calculation so that all quantities concerned remain within the limits of the machine and yet are expressed to an accuracy sufficient to ensure the desired precision in the results. Our experience with the EDSAC has shown that if this task can, in some way, be relegated to the machine then the time taken to prepare a programme after a problem has been understood is considerably reduced.

The problem does not arise with machines designed to operate directly with numbers expressed in the floating radix form. Numbers in this form are represented by $a \cdot r^p$. The first machine of this kind was the Bell Telephone Laboratories Relay Computer Model V⁽¹⁾. This is a decimal machine (that is, $r = 10$) in which $1 > |a| > 0.1$, $19 \geq p \geq -19$ and a is expressed to an accuracy of seven significant figures. Since this was completed all important relay machines have been equipped with similar facilities. No electronic machine of this kind has yet been built but we would remark that in our opinion an electronic machine provided with a floating point arithmetical unit would be a powerful computing instrument even if it had a relatively slow store, a magnetic drum, for example. It would be particularly suitable for a laboratory which had to solve quickly a wide variety of problems as they are presented.

For a fixed decimal-point machine the usual arrangement is to associate scale factors with some or all of the quantities occurring in the calculation. These scale factors are of two kinds. First certain quantities can be associated with *fixed scale factors* which remain unaltered throughout the calculation. Their value must be chosen so that the quantities concerned do not exceed capacity and yet can be represented at all times with sufficient accuracy. This is the problem of providing 'elbow room.' It may not be possible to satisfy both requirements by the use of a single fixed scale factor and it is then necessary to introduce an *adjustable scale factor*, that is, a scale factor which is altered during the course of the calculation. It may be adjusted either continually or occasionally in accordance with some criterion or it may take a preassigned set of values. All these things the coder has to think about if the calculation is to be arranged to use the least possible machine time. The difficulties arising will vary from one calculation to another and may be trivial, moderately complex, or really hard. The individual nature of calculations and the necessity for a mathematical understanding of them, however, makes a general solution to the problem on these lines unattractive to attempt. The only way in which the difficulties may be avoided is to associate every number occurring in the calculation with its own ad-

justable scale factor. The scale factors can be stored most economically if they are powers of 2 or of 10. In the latter case the above scheme virtually amounts to programming a floating decimal point. This is the solution that we feel should be adopted for many kinds of calculation.

The floating decimal form of representation of numbers is especially useful when it is required to evaluate algebraic expressions. For example, to code the evaluation $(au^2 + bu + c)/(du + e)$ with a fixed decimal-point machine can be very troublesome if it is required to maintain accuracy over the entire range of numbers. If the reader has any doubts about this he is advised to try it. If floating decimal arithmetic is used, then the coding requires little thought and is quite direct.

It is the purpose of this paper to describe a method by which floating decimal arithmetic can be carried out on fixed decimal-point machines, and in particular to describe the scheme adopted for the EDSAC.

General considerations. The following remarks are fairly general and apply to almost any machine although certain features mentioned, for example, short and long locations, are made with the EDSAC in mind.

The most convenient way to programme operations on numbers expressed in the floating decimal form is by means of a special type of subroutine which we call *interpretive*. This enables words similar to those representing ordinary machine orders to be interpreted in terms of floating decimal operations. Such words will be referred to as 'orders' (with quotation marks). Each 'order' resembles an ordinary machine order but is never obeyed as such by the control circuits of the machine. Instead, 'orders' are selected in a definite sequence by the action of the interpretive subroutine and interpreted in accordance with a preassigned 'order' code, by means of sequences of orders which form part of the interpretive subroutine. The selective action of the interpretive subroutine will be referred to as the 'control' (in quotation marks). The advantage of such interpretive subroutines is that the 'order' code may be chosen to suit the convenience of the programmer. There need be no relation between the form of the 'order' code and the ordinary code of the machine. However, we have become so familiar with the ordinary EDSAC order code that a similar single address 'order' code has been adopted for the interpretive subroutines. Interpretive subroutines have also been placed in the EDSAC library for carrying out arithmetical operations on complex numbers and on double-precision numbers.

When representing floating decimal numbers in a fixed decimal-point machine it is most economical to pack both the numerical part and the exponent into a single storage location. This means, of course, that the numerical part of the number has fewer binary digits to represent it than would normally be available. However, they are all significant whereas in fixed decimal-point working digits are thrown away to provide 'elbow room.' In both cases accuracy may be lost owing to cancellation of leading digits at one end of a number and to rounding-off errors at the other.

Two long and two short storage locations are set aside to form a kind of 'arithmetical unit.' One long location holds the numerical part of a number and one short location holds the exponent. Together they form the *floating decimal accumulator*. In a similar fashion the other long location and the other short location form the *floating decimal register*.

An arithmetical 'order' first causes a subroutine to select and unpack the operand and place the numerical part and the exponent into a further pair of storage locations—the 'multiplier register.' In the case of an add 'order' the operand is then added to the number held in the floating decimal accumulator. This is done as follows. Let a_0 , a_a , and a_s denote the numerical parts of the operand, augend, and sum respectively, and let p_0 , p_a , and p_s similarly denote their exponents. Then we have

$$\begin{aligned} a_s 10^{p_s} &= [a_0 + a_a 10^{-(p_0-p_a)}] 10^{p_0} & p_0 \geq p_a \\ &= [a_0 10^{-(p_a-p_0)} + a_a] 10^{p_a} & p_0 < p_a. \end{aligned}$$

The *subtract* 'order' works in a similar fashion. A *register* 'order' places the operand in the floating decimal register. Other 'orders' enable the product of the operand with a number held in the floating decimal register to be added to or subtracted from the number held in the floating decimal accumulator. The *transfer* order causes the number held in the floating decimal accumulator to be converted to standard form, packed, and finally transferred to the store; the floating decimal accumulator is then 'cleared' by replacing the number held in it by zero, that is, by the special number 010^{-63} . The *input* 'order' causes the two parts of a number to be read from the input tape, packed, and transferred to the store. The *output* 'order' causes the numerical part and the exponent of the number held in the floating decimal accumulator to be printed on the same line in two adjacent columns on the page of the teleprinter.

The use of two separate storage locations for the floating decimal accumulator allows the range and accuracy of numbers held therein to be greater than those held in a single storage location elsewhere. This enables products to be accumulated without loss of accuracy due to intermediate rounding-off errors.

The scheme adopted for the EDSAC. The interpretive subroutine that has been developed to carry out floating decimal arithmetic with the EDSAC will now be described in more detail. Throughout this and the following section references to specific features or conventions used with the EDSAC have been avoided as far as possible but the following terms are used at least once: *initial orders*, *control combinations*, *preset parameters* (not to be confused with the *current parameter* defined below), *short* and *long* locations and the *E* order. Detailed descriptions of all these features can be found in reference 2.

The following abbreviations will be adopted:

- $F(A)$ denotes the floating decimal number held in the floating decimal accumulator
- $F(R)$ denotes the floating decimal number held in the floating decimal register
- $S(mD)$ denotes the long storage location having address m
- $S(mF)$ denotes the short storage location having address m
- $F(mD)$ denotes the floating decimal number held in $S(mD)$
- $F(mF)$ denotes the floating decimal number held in $S(mF)$
- $C(mD)$ denotes the ordinary number held in $S(mD)$
- $C(mF)$ denotes the ordinary number held in $S(mF)$

Number representation. The method described above of packing the numerical part a and the exponent p of a floating decimal number $a \cdot 10^p$ in a

single storage location has been adopted. With the EDSAC it amounts to representing the floating decimal number by the ordinary (fractional) number $p2^{-6} + a2^{-7}$, where a is expressed with an accuracy of 28 binary digits and lies in the range ($1 \geq |a| \geq 0.1$) and p is an integer such that $63 \geq p \geq -63$. Zero has the special representation 010^{-63} . When numbers are transferred from the floating decimal accumulator to the store they are automatically transferred to the standard representation.

The 'order' code. Since the 'orders' are similar in form to the ordinary machine orders of the EDSAC they will be presented in the conventional form adopted for the latter. The 'orders' of the code fall into two classes, *arithmetical* and *organizational*. The arithmetical 'orders' can refer either to short or to long storage locations according as the 'order' is terminated with the code letter F or D. This is the usual EDSAC convention. If, however, these 'orders' are terminated by $\pi\Delta$ instead of D or Δ instead of F then a number, known as the *current parameter*, held in a certain storage location, will be added to the address of the order before it is obeyed. Among the organizational 'orders' are two—the P and the F orders—which enable the current parameter to be set to some initial value and subsequently adjusted after each cycle of a repetitive operation. In this way an arithmetical 'order' can refer to different numbers during different cycles of the calculation. These orders are described in more detail below. The scheme is similar in some respects to the way in which orders are modified in the Manchester University Electronic Computer Mk. 2. In this machine a built-in facility enables a number held in one of 8 registers—called the B registers—to be added to the address of an order immediately before that order is obeyed.

The arithmetical 'orders'

- A m D* Add $F(mD)$ to $F(A)$
- B m D* Subtract $F(mD)$ from $F(A)$
- H m D* Replace $F(R)$ by $F(mD)$
- V m D* Add the product $F(R) \cdot F(mD)$ to $F(A)$
- N m D* Subtract the product $F(R) \cdot F(mD)$ from $F(A)$
- D m D* Replace $F(A)$ by $F(A)/F(mD)$
- $\phi m D$ Replace $F(mD)$ by $F(A)$ and $F(A)$ by 0.10^{-63}
- L m D* Similar to $\phi m D$ but in addition the non-standard content of the floating decimal accumulator is printed as negative sign or space; exponent (2 figs.); space; negative sign or space; 8 decimal digit fraction. For example, -98.283742 would be printed as $02 - 98283742$.
- $\Delta m D$ Input a sequence of numbers on the tape terminated by X into the locations mD , $(m-2)D$, etc. Each number is punched in the following way: Characters to represent the exponent; sign; numerical part. In the numerical part the decimal point is understood to be before the first digit punched. Any number of digits may be punched. For example, -98.283742 would be punched as $2 - 98283742$. X is punched before the first number of the sequence and then the sequence is copied on to the data tape in the reverse direction. Numbers read into the machine must have exponents in the range -16 to $+15$, so that it can be represented by a single tape character.

The organizational 'orders'

- M m F* Conditional order: if $F(A) > 0$ transfer 'control' to the 'order' which stands in $S(mF)$; otherwise replace $F(A)$ by $|F(A)|$ and proceed with the next 'order.'
- C m F* Transfer control of the machine to the order which stands in $S(mF)$, that is, return to the machine order code.
- X m F* Transfer 'control' to the X -auxiliary whose entry 'order' stands in $S(mF)$. When the X -auxiliary is completed it will return 'control' to the 'order' immediately following the X 'order.'
- G m F* Transfer 'control' to the 'order' standing in $S(mF)$.
- P m F* }
F m F } The function of these 'orders' is described below

The P and F 'orders.' These orders are provided to facilitate the coding of cycles of orders and of cycles within cycles, etc. Each cycle of 'orders' or 'loop' is started by a P 'order' and terminated by an F 'order.' The two are said to *correspond*. This correspondence is similar to that between left and right handed brackets in a lengthy algebraic formula if the direction left to right in the formula corresponds to the direction in which 'control' is normally advanced. Associated with each loop of 'orders' is a parameter whose value is set, initially, by the P 'order' and is subsequently adjusted by the F 'order.' The current parameter (see above) associated with any particular 'order' of a loop is the parameter set by the P 'order' which the 'control' encountered last. It is now possible to give a formal description of the function of these 'orders.'

- P m F* Record a new parameter—the current parameter of the following 'orders'—and set it to the value $-m2^{-15}$
- F m F* If the current parameter is $< -m2^{-15}$, increase its value by $m2^{-15}$, return 'control' to the 'order' immediately following the corresponding P 'order'; otherwise proceed with the following 'order' after restoring the current parameters to the value it had before the corresponding P 'order.'

The above description should be sufficient to enable a programmer to use the P and F orders in a programme. We now give a detailed description of the means by which the effects of these 'orders' are achieved.

The number of parameters has been limited to 3. This restriction, although not essential, is reasonable as problems involving loops within loops more than three 'deep' are likely to occur very rarely.

The parameters are stored together with the addresses of the locations of the corresponding P 'orders,' in 3 long storage locations $S(hD)$, $S(h+2)D$, and $S(h+4)D$. At any stage in the programme $S(hD)$ contains the current parameter, $S(h+2)D$ the previous parameter, and $S(h+4)D$ the previous parameter but one.

A P 'order,' for example, $P m F$ in $S(nF)$ causes $C(h+4)D$ to be replaced by $C(h+2)D$, $C(h+2)D$ to be replaced by $C(hD)$, and $C(hD)$ to be replaced by the number $-m2^{-15} + n2^{-32}$. When the corresponding F 'order,' $F q F$ is encountered the following operations take place. The size of the current parameter (the number $-m2^{-15}$ in the short storage location $S(h+1)F$) is tested. If this number is less than $-q \cdot 2^{-15}$ it is increased by the amount $q \cdot 2^{-15}$ and 'control' is transferred to the 'order' which stands in

$S(n + 1)F$, where $n \cdot 2^{-15}$ is the number in the short location $S(hF)$. Otherwise $C(hD)$ is replaced by $C(h + 2)D$, $C(h + 2)D$ is replaced by $C(h + 4)D$ and then 'control' proceeds with the next 'order.'

Example. The action of the 'orders' and the context in which they are meant to be used may be more clearly understood from a study of the following programme for the calculation of a root of an algebraic equation by the Newton-Raphson iterative process.

Location of data: let $f(x) \equiv \sum_{r=0}^{r=n} a_r x^{n-r} = 0$ be the equation to be solved.

It is assumed that the coefficients a_n, a_{n-1}, \dots, a_0 are in $S(100D), S(98D), \dots, S(100 - 2n)D$. The initial approximation x_0 stands in $S(6D)$ and all subsequent approximations are placed there. A small quantity δ , used in the convergence criterion, stands in $S(8D)$.

Formula used: the iterative formula

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

is used. The iteration is arranged to terminate when $|x_n - x_{n-1}| < \delta$. $f(x)$ is calculated from the recurrence relations

$$q_0 = a_0; \quad q_{r+1} = q_r \cdot x + a_{r+1}; \quad q_n = f(x)$$

and $f'(x)$ is calculated from the similar recurrence relations

$$q_0' = 0; \quad q_{r+1}' = q_r' \cdot x + q_r; \quad q_n' = f'(x).$$

The orders of the programme are listed below. For those readers who are not familiar with the EDSAC conventions it may be stated that the code letter θ provides for a system of numbering relative to the order immediately following a control combination G K. Thus, in the example below, the 'order' M 18 θ refers to the 'order' B 8 D.

	G		K	
0	ϕ		D	'clear' floating decimal accumulator
1	ϕ	10	D	'clear'; $S(10D)$
2	H	6	D	place x_n in the floating decimal register
3	P	$2n+2$	F	
4	ϕ	12	D	
5	V	10	D	} $p_{r+1}' = xp_r' + p_r$ } cycle $n + 1$ times
6	A	12	D	
7	ϕ	10	D	
8	V	12	D	} $p_{r+1} = xp_r + a_{r+1}$ }
9	A	$102 \pi\Delta$		
10	F	2	F	
11	D	10	D	} $f(x_n)/f'(x_n)$ to $S(10D)$
12	ϕ	10	D	
13	B	10	D	} $x_{n+1} = x_n - f(x_n)/f'(x_n)$
14	A	6	D	
15	ϕ	6	D	
16	A	10	D	} form modulus of $(x_{n+1} - x_n)$
17	M	18	θ	
18	B	8	D	subtract δ
19	M	1	θ	discriminate
20	ϕ		D	'clear' floating decimal accumulator
21	A	6	D	} print the root x
22	L	6	D	

Auxiliary subroutines. To facilitate the coding of entire calculations in floating decimal arithmetic it is useful to extend the range of action of the interpretive subroutines by the addition of auxiliary subroutines for carrying out common numerical processes. The auxiliary subroutines, henceforth referred to simply as *auxiliaries*, are of two kinds. The first kind consists entirely of ordinary orders and will be referred to as *C auxiliaries*. Auxiliaries of the second kind consist largely of 'orders' and hence themselves use the interpretive subroutine. They will be referred to as *X auxiliaries*.

C auxiliaries. These are called in by a *C* 'order' which transfers control to the entry order of the auxiliary. This then carries out the appropriate calculation by means of machine orders. When this has been completed control is transferred back to a point in the interpretive subroutine. This causes the 'control' to resume the obeying of 'orders' starting at the 'order' immediately following the *C* 'order.' Thus the *C* auxiliary can be considered as being an extension of the 'order' code and used as such.

A typical *C* auxiliary is one which replaces $F(A)$ by its square root. The way in which this is done using ordinary orders is as follows. Let p_a and n_a denote the exponent and numerical part of $F(A)$. Two cases arise: If p_a is even, $p_s = p_a/2$ and $n_s = \sqrt{n_a}$; if p_a is odd, then $p_s = (p_a + 1)/2$ and $n_s = \sqrt{n_a/10}$. In both cases the arithmetical operations can be done most simply by using the ordinary techniques of fixed decimal-point working.

C auxiliaries are almost as fast and as economical in storage space as the corresponding routines for fixed decimal-point working. The extra orders required to handle the exponent are largely offset by the orders that would otherwise be required to cater for a large range of the numerical part.

X auxiliaries. These are called in by the special 'order' $X m F$, where m is the location of the entry order of the auxiliary. When the *X* auxiliary has completed its part of the calculation 'control' is returned to the 'order' immediately following the *X* 'order' which called the auxiliary into use. This cannot be done by the same method as is used for the *C* auxiliary because the *X* auxiliary itself uses the interpretive subroutine. Instead the *X* 'order' causes the address of the location of the next 'order' to be stored in a certain location for future reference. This record is called the *link*. At the end of the auxiliary is a *C* 'order' which directs control to a set of orders within the interpretive subroutine which use the link to return 'control' to the main programme.

X auxiliaries can use other *X* auxiliaries. To enable this to be done the *X* 'order' causes a list of links to be kept. At any point in the programme the link at the head of the list refers to the 'order' to which 'control' must be returned after the current subroutine has been finished with. The number of links in the list measures the depth to which 'control' has passed within the auxiliaries. A reference number which is adjusted every time a link is added to or removed from the list records this depth. This enables the link at the head of the list to be selected by those orders of the interpretive subroutine which are called into use when the *X* auxiliary has been completed.

The Directory. Reference to the auxiliaries can be facilitated, if desired, by using the following scheme. The auxiliaries are first enumerated in the order in which they are to be read from the input tape into the store. Thus auxiliary no. 1 is the first auxiliary read from the tape, auxiliary no. 2 is the

second auxiliary read from the tape, and so on. When the auxiliaries have been ordered in this way the m -th auxiliary can be called in by the 'order' $X m L$ or $C m L$ (depending on whether it is an X or a C auxiliary).

This is achieved by means of a table of switching orders—called a *directory*—stored in consecutive storage locations beginning with $S(hF)$. Each entry directs 'control' (or control) to the first 'order' (or order) of one of the auxiliaries. If, for example, the m -th auxiliary is an X auxiliary whose first 'order' stands in $S(nF)$, then the m -th entry in the directory, that is, the entry standing in $S(h + m)F$, is the 'order' $g n F$. Thus to call in the auxiliary 'control' is first transferred, by the 'order' $X h + m F$, to the 'order' $G n F$ which in turn transfers 'control' to the first 'order' of the auxiliary. Similarly, if the m -th auxiliary is a C auxiliary, the m -th entry is the ordinary order $E n F$, and the auxiliary is called in by the 'order' $C h + m F$.

The directory is assembled by the initial orders as the auxiliaries are read from the tape into the store. At the same time the address of the locations, $S(hF)$ of the first entry in the table is placed in the preset parameter location L so that $X m + h F$ and $C m + h F$ can be punched as $X m L$ and $C m L$ respectively.

The advantage of the scheme is that the master routine can be drawn up in the final form once the set of auxiliaries have been ordered. It is not necessary for the coder to keep a record of the locations in the store of individual auxiliaries. In effect, the scheme shifts the burden of 'book-keeping' from the programmer to the machine. Similar principles are used by the assembly subroutines (for programme assembly) which already exist in the EDSAC library of subroutines (see reference 2).

The use of the input tape as a form of auxiliary store. The EDSAC is not provided with an auxiliary store so that for many problems the shortage of storage space is a real difficulty. This difficulty may be partly overcome by using the paper tape input medium as a form of auxiliary store. There are two ways in which this may be done.

The first way, used when the storage requirements grossly exceed those available, is to carry out the calculation in stages, using the input tape to store numbers and 'orders' not required throughout the calculation. This method will be referred to as *piecewise control* of the calculation.

In the second method the 'orders' of the master routine are placed on the tape and read into the store one at a time, each being obeyed immediately after it has been read. This method is used when the master routine consists of a lengthy sequence of 'orders' not many of which are repeated. If none of these are repeated the scheme takes no longer than putting the complete master routine into the store and then entering it. The principal advantage is that no storage space need be allocated to the master routine. A further advantage is that the progress of the calculation is apparent from the progress of the tape. This method of control will be referred to as *input control*. In the Manchester University Computer Group, tapes drawn up on these lines are referred to as *job-steering tapes*.

Piecewise control. To facilitate this mode of working the Δ 'order' and the initial orders can be used as follows.

The Δ 'order' enables sequences of numbers of any length to be read from the tape into the store when required, overwriting, if necessary, information no longer wanted.

If the initial orders are retained intact during the course of the calculation they can be recalled into use by a *C* 'order'. In this way 'orders' (or orders) can be read from the input tape into the store in the usual way at any time during the calculation. The reading of 'orders' can be halted and control transferred back to the interpretive subroutine by a suitable control combination.

The above scheme together with the use of the Δ 'order' enables quite complicated problems to be tackled.

Input control. In this scheme a special *C* auxiliary—the *input control auxiliary*—is used. This causes 'orders' to be read from the tape and interpreted immediately after they are read. Normally they will be of an arithmetical character or will call in auxiliary subroutines. The *C* auxiliary will continue to read and obey such 'orders' until this mode of 'control' is terminated by a suitable *C* 'order' on the tape itself. This directs control to a point within the interpretive subroutine and in this way causes 'control' to be returned to the 'order' immediately following the *C* 'order' which called the input control auxiliary into use.

Times of operations of the 'orders.' The times of execution of the individual 'orders' are as follows:

'orders'	Times of execution
<i>A, B</i>	90 ms.
<i>V, N</i>	105 ms.
<i>H, C, G, X, M, P, F</i>	50 ms.
<i>D</i>	140 ms.
ϕ	80 ms.

The times of operation of the *L* and Δ 'orders' are largely determined by the speed of the input and output units. The teleprinter can print six characters per second and the tape-reader can read about 25 characters per second. These rates allow for the time taken for binary to decimal conversion and vice versa.

For comparison it should be stated that each ordinary order of the EDSAC takes $1\frac{1}{2}$ ms. with the exception of the multiplication orders, *V* and *N*, which take 6 ms. each.

From a direct comparison it would seem that the floating 'orders,' other than those used for reading and writing, are about 60 times as slow as the machine orders and hence that a programme using the interpretive subroutine would be slower by the same factor. This is not altogether true because in such a programme fewer 'orders' are needed than would otherwise be necessary as there are no scale factors to deal with and the techniques for counting and for the modification of 'orders' have been streamlined. Moreover, the time taken by the *C* auxiliaries is about the same as that taken by the corresponding subroutine in fixed decimal-point working.

These factors vary from problem to problem but our experience has shown that the reduction in speed varies from about 20 to 1 to about 4 to 1. The reduction of the time taken to code a problem has to be experienced to be believed!

R. A. BROOKER
D. J. WHEELER

University Mathematical Laboratory
Cambridge, England

The authors wish to thank Dr. M. V. WILKES for his encouragement and advice in preparing this paper.

¹ F. ALT, "A Bell Telephone Laboratories 'Computing Machine—1 and II,'" *MTAC*, v. 3, 1948, p. 1-13 and 69-84.

² M. V. WILKES, D. J. WHEELER, & S. GILL, *The preparation of programmes for an electronic digital computer, with special reference to the EDSAC and the use of a library of subroutines*. Addison-Wesley Press, Inc., Cambridge, Mass., 1951.

BIBLIOGRAPHY OF CODING PROCEDURE

The National Bureau of Standards is forming a central library of notes, reports and technical publications concerned with programming and coding for electronic digital computers.

The collection of material has in view the possibility of increasing the efficiency in the use of high speed digital computers. The material will include for example coding manuals, computing routines, supervisory routines and codes for generating other codes. Those interested in research and instruction in coding practices are invited to avail themselves of this material.

Computation laboratories are invited to submit material to this library. A list of such acquisitions together with a short descriptive review will appear in the future issues of *MTAC*; they will be numbered serially for reference. Material should be sent to the attention of J. H. WEGSTEIN, Computation Laboratory, National Bureau of Standards, Washington 25, D. C.

Some material of this kind has already been included or reviewed in *MTAC*. This bibliography begins with references to eleven such items. The last four items are new.

1. S. LUBKIN, "Decimal point location in computing machines," *MTAC*, v. 3, p. 44-50.
2. HERMAN H. GOLDSTINE & JOHN VON NEUMANN, *Planning and Coding of Problems for an Electronic Computing Instrument*. Institute for Advanced Study, Princeton, New Jersey.
Part II, v. 1, 1947, 69 p., 21.6 × 27.9 cm. [*MTAC*, v. 3, p. 54-56]
Part II, v. 3, 1948, iii + 23 p., 21.6 × 27.9 cm. [*MTAC*, v. 3, p. 541-542]
Part II, v. 2, 1948, 68 p., 7 figs., 21.6 × 27.9 cm. [*MTAC*, v. 4, p. 44-46]
3. FLORENCE KOONS & S. LUBKIN, "Conversion of numbers from decimal to binary form in the EDVAC," [*MTAC*, v. 3, p. 427-431]
4. 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. [*MTAC*, v. 4, p. 116]
5. ANON., *Description and Use of the ENIAC Converter Code*. Ballistic Research Laboratories, *Technical Note* no. 141, Aberdeen Proving Ground, Maryland, November 1949, 23 pages, mimeographed. [*MTAC*, v. 4, p. 170-171]
6. R. W. HAMMING, "Error detecting and error correcting codes," *Bell System Technical Jn.*, v. 29, 1950, p. 147-160. [*MTAC*, v. 5, p. 40-41]
7. M. V. WILKES, "Automatic computing," *Nature*, v. 166, December 2, 1950, p. 942-944. [*MTAC*, v. 5, p. 171]
8. S. GILL, "The diagnosis of mistakes in programmes on the EDSAC," *R. Soc. London, Proc.*, v. 206A, 1951, p. 538-554. [*MTAC*, v. 6, p. 49-50]

9. M. V. WILKES, D. J. WHEELER & S. GILL, *The Preparation of Programs for an Electronic Digital Computer*. Addison-Wesley Press, Inc., Cambridge, Mass., 1951, 167 p., 22.9 × 15.2 cm. Price \$6.00 [*MTAC*, v. 6, p. 51]
10. R. A. BROOKER, S. GILL & D. J. WHEELER, "The adventures of a blunder," [*MTAC*, v. 6, p. 112-113]
11. DAVID P. PERRY, "Minimum access programming," [*MTAC*, v. 6, p. 172-182]
12. UNIVERSITY OF ILLINOIS RESEARCH BOARD, *University of Illinois Electronic Digital Computer Library Routines*. This volume is an expandable library of the routines which have been prepared for the Illinois machine and at present contains perhaps fifty routines. It contains among many other such routines as the Wheeler Leapfrog Test, the solution of simultaneous linear equations, solution of a system of first order differential equations, post mortem, integration routine, address search routine, eigenvalues and eigenvectors of a symmetric matrix, and square root.
13. ECKERT-MAUCHLY COMPUTER CORPORATION, *Programming Manual for the UNIVAC System*. This describes the UNIVAC and how to code for it. The manual includes chapters on: flow charts, an illustrative problem (Denominated Payroll routines), floating decimal and double precision routines, elementary mathematical problems, input and output operations, editing, problem preparation.
14. REMINGTON-RAND CORPORATION, *Roster of UNIVAC Problems*.
15. J. D. GOODELL, "The foundations of computing machinery," *Jn. of Computing Systems*, v. 1, 1952, p. 1-13.

J. T.

BIBLIOGRAPHY Z

1000. MARTIN GARDNER, "Logic machines", *Scientific American*, v. 186, March 1952, p. 68-73.

This is an historical account of the development of these machines beginning with the period of GEORGE BOOLE, the English scientist who first conceived the idea of symbolic logic as a system designed for the more efficient handling of problems. The paper discusses one of the devices of LULL, a 13th century Spanish mystic, who probably was the first to construct a machine to handle logical problems. The device consisted of a set of wheels having a common center. Around the edge of the disks were letters representing various ideas. When the wheels were rotated, various combinations of ideas would appear for further investigation. This primitive machine prompted CHARLES STANHOPE in the 18th century to invent the Stanhope demonstrator for solving syllogisms. Later WILLIAM STANLEY JEVONS developed his "abecedarium," a method of applying Boole's approach. In the case of a syllogism all possible combinations of the three terms *A*, *B* and *C* are listed (the negative of these terms being represented by *a*, *b*, and *c*). In solving a problem all classes inconsistent with the premises are crossed out—leaving only the true combinations. These are then examined to determine the relationship between the terms in an effort to weed out the false conclusions. Jevons went on to construct his more rapid "logical piano" on the same principle.

Since 1900 many devices for solving syllogisms have been invented. With the publication of an article by CLAUDE E. SHANNON of M.I.T. entitled, "A symbolic analysis of relay and switching circuits," it was proved that circuits could be constructed in series and in parallel making use of the binary system to indicate the two truth values which correspond to basic Boolean relations. The Kalin-Burkhart Calculator, a small low-cost device, is the first in history to use electrical circuits for solving problems in symbolic logic. It is similar in principle to the Jevons piano but excels in that it can handle a greater number of problems far more intricate in complexity than those solved on Jevons' machine.

At present these machines have limited application; probably their chief value lies in the fields of business procedures and of electronic computers. In the latter field, especially, decisions involving considerations of logic are often necessary in setting up problems to be solved on the machine. It has even been suggested that with built-in probability values the machine could after considering all possibilities make important logical decisions. The question of whether or not such a machine could develop a creative imagination has been the subject of much fantastic speculation.

EDITH T. NORRIS

NBSMDL

1001. JOHN D. GOODELL, "The foundations of computing machinery," *The Journal of Computing Systems*, v. 1, June 1952, p. 1-13. TENNY LODE, "The realization of a universal decision element," *The Journal of Computing Systems*, v. 1, June 1952, p. 14-22.

These articles are, according to their authors, the beginning of a series discussing the logical design of computing machinery in terms of Boolean algebra or, more generally, symbolic logic. They treat mainly the two-valued functions of one and of two two-valued variables, and particularly the recognition and realization of complete subsets of the two-valued functions. Functions of many arguments, variable functions and many-valued logics are very briefly discussed.

R. D. ELBOURN

NBSEC

1002. HARVARD UNIVERSITY, COMPUTATION LABORATORY, *Annals*, v. 27: *Synthesis of Electronic Computing and Control Circuits*. Cambridge, Mass., Harvard University Press, 1951, 278 p., 20 × 27 cm. Price \$8.00.

The bewildering complexity of electronic computing and control circuits and the profusion of means available for achieving the same functional performance make it extremely desirable to have 1) a concise and unique notation for writing the function of a circuit so that functionally identical requirements or circuits can readily be recognized, 2) a simple translation of this unique notation into a non-unique one corresponding more closely to the actual circuits so that one has a facile shorthand for manipulating a functional requirement into various possible realizations and 3) a method for finding a most economical realization of a required function.

It is well known that Boolean algebra is useful for 1) and 2), but this book is the most extensive treatment of these applications which has appeared.

The method of "minimizing charts" introduced in chapter 5 is a significant contribution to the very formidable problem of most economical synthesis. At the present state of the theory, minimizing charts and tables included in the book provide a rather complete solution for circuits with one output and four or five input variables. For six or seven variables the charts become quite unwieldy. Some useful methods are given for circuits with multiple outputs and circuits containing signal delay or storage devices, but there remains the necessity for much arbitrary decision in their synthesis, and the economy problem is wide open.

The second half of the book covers systems for binary coding decimal digits and the design of a variety of adders and multipliers mostly of the coded decimal type.

The Boolean algebra of this book uses only the logical product $X Y$ and negation, written $X' = 1 - X$, because the arithmetic of these operations coincides with ordinary arithmetic. While this is a complete system, the reviewer would have included the logical sum, in the belief that the difficulty of remembering $1 + 1 = 1$ is outweighed by having symmetrical notations for both types of rectifier circuits and by having the powerful duality principle to manipulate expressions; however this is merely personal preference.

Although the efficient designing of electronic computing and control circuits has only begun to be systematized, anyone who wishes to use the best known techniques and possibly to contribute some new ones must certainly consult this volume.

R. D. ELBOURN

NBSEC

1003. OFFICE OF NAVAL RESEARCH, *Digital Computer Newsletter*, v. 4, July 1952, 9 p.

The contents are as follows:

1. Whirlwind I
2. The SEAC
3. Naval Proving Ground Calculators
4. Moore School Automatic Computer (MSAC)
5. The Logistics Computer
6. Aberdeen Proving Ground Computers
 - The ENIAC
 - The EDVAC
 - The ORDVAC
7. The Circle Computer
8. Electronic Computer Corporation (ELECOM—100)
9. The CADAC (CRD 102)
10. The ACE Pilot Model, England
11. The Ferranti Computer at Manchester University, England

Data Processing and Conversion Equipment

1. Digital to Analogue Converters
2. Oscillograph Trace Reader
3. Flying Typewriter

1004. INSTITUTE OF RADIO ENGINEERS, "Standards on Electronic Computers: Definitions of Terms, 1950." Institute of Radio Engineers, *Proc.*, v. 39, no. 3, 1951, p. 271-277.

This is the first report of the IRE Subcommittee on Definitions of Electronic Computer Terms. R. SERRELL was chairman of this subcommittee, J. W. FORRESTER headed the Electronic Computers Committee, and J. G. BRAINERD was chairman of the Standards Committee. There is a total of approximately 135 terms for which definitions are given. About 30% of these terms are arithmetical or logical, over one-half relate specifically to digital devices, and approximately 20 terms concern analog devices. (It might be of interest to note that a new IRE subcommittee is at present reviewing these definitions.)

H. D. HUSKEY

Wayne University
Detroit, Michigan

1005. IBM Scientific Computation Forum, *Proc.* edited by H. R. J. GROSCH, 1948, 126 pages. 22 × 28.5 cm.

This volume refers to one of a series of meetings arranged from time to time by the IBM Corp. (cf. the following reviews). The contents are as follows:

1. "Evaluation of higher order differences on the Type 602 Calculating Punch" by FRANK M. VERZUH
2. "Differencing on the Type 405 Accounting Machine" by GERTRUDE BLANCH
3. "The use of optimum interval mathematical tables" by H. R. J. GROSCH
4. "Punched card techniques for the solution of simultaneous equations and other matrix operations" by WILLIAM D. BELL
5. "Two numerical methods of integration using predetermined factors" by LELAND W. SPRINKLE
6. "Integration of second order linear differential equations on the Type 602 Calculating Punch" by N. ARNE LINDBERGER
7. "Integration of the differential equation $\frac{d^2P}{dr^2} = P \cdot F(r)$ using the Type 601 Multiplying Punch" by PAUL HERGET
8. "Some elementary machine problems in the sampling work of the census" by A. ROSS ECKLER
9. "IBM applications in industrial statistics" by CUTHBERT C. HURD
10. "Some engineering applications of IBM equipment at the General Electric Company" by FRANK J. MAGINNISS
11. "Planning engineering calculations for IBM equipment" by BEN FERBER
12. "A survey of the IBM project at Beech Aircraft Corporation" by JOHN KINTAS
13. "Aerodynamic lattice calculations using punched cards" by HANS KRAFT
14. "Dynamics of elliptical galaxies" by JACK BELZER, GEORGE GAMOW and GEOFFREY KELLER
15. "Application of punched cards in physical chemistry" by GILBERT W. KING

16. "Application of punched card methods to the computation of thermodynamic properties of gases from spectra" by LYDIA G. SAVEDOFF, Jack Belzer and H. L. JOHNSTON
17. "Calculation of the equilibrium composition of systems of many constituents" by STUART R. BRINKLEY, JR. and ROBERT W. SMITH, JR.
18. "Punched card calculating and printing methods in the Nautical Almanac Office" by FREDERICK H. HOLLANDER
19. "Programming and using the Type 603-405 Combination Machine in the solution of differential equations" by GEORGE S. FENN
20. "Applications of punched card equipment at the Naval Proving Ground" by CLINTON C. BRAMBLE
21. "Use of the IBM relay calculators for technical calculations at Aberdeen Proving Ground" by JOSEPH H. LEVIN
22. "Simultaneous linear equations" by FRANCIS J. MURRAY
23. "Computation of shock wave refraction on the selective sequence electronic calculator" by HARRY POLACHEK
24. "Computation of statistical fields for atoms and ions" by L. H. THOMAS

At the meeting reported here, most of the papers described details of the use of IBM machines for various problems in numerical mathematics, which are adequately characterized by their titles. A few of the papers present surveys of the computing work done in certain organizations on IBM machines. The paper of Fenn introduces the 603-405 combination (cf. the following review). Murray's paper refers to his special-purpose machine for solving linear equations. Grosch's article is of more special interest to table-makers.

F. L. ALT

NBSCL

1006. IBM Seminar on Scientific Computation, *Proc.* edited by CUTHBERT C. HURD, November 1949, 109 pages. 22 × 28.5 cm.

The contents of this volume are as follows:

1. "The dynamics of nuclear fission" by DAVID L. HILL
2. "Monte Carlo calculations" by WILLIAM W. WOODBURY
3. "Modification of the Monte Carlo method" by HERMAN KAHN
4. "Analyzing exponential decay curves" by ALSTON S. HOUSEHOLDER
5. "On the distribution of Kolmogorov's statistic for finite sample size" by Z. W. BIRNBAUM
6. "The IBM Card-Programmed Electronic Calculator" by CUTHBERT C. HURD
7. "Stochastic methods in quantum mechanics" by GILBERT W. KING
8. "Calculation of resonance energies" by GEORGE E. KIMBALL
9. "Cam design calculations on the Card-Programmed Electronic Calculator" by E. A. BARBER
10. "Calculation of the equilibrium composition of homogeneous multi-component systems" by STUART R. BRINKLEY, JR. and ROBERT W. SMITH, JR.
11. "Eigenvalue problems related to the Laplace operator" by DONALD A. FLANDERS and GEORGE H. SHORTLEY

12. "Numerical solution of partial differential equations of parabolic type" by L. H. THOMAS
13. "Solutions of the wave equation" by PAUL HERGET
14. "Sampling methods applied to differential and difference equations" by JOHN H. CURTISS

The conference about which this volume reports was on a higher scientific level than other similar conferences arranged by the IBM Company and requires more detailed review. Not all the papers in this volume are primarily of interest to the numerical analyst. For example, Kahn's paper is an exposition of Monte Carlo methods as applied to definite integrals and integral equations, with emphasis on importance sampling, but without reference to machine methods or to any of the deeper questions of numerical analysis. The main purpose of Birnbaum's paper is "to point out that in view of the development of high-speed sequence computing equipment, the tabulation of the exact probability distribution of Kolmogorov's statistic for finite N has become practically feasible, and to propose that such a tabulation should be carried out." The article by G. W. King is principally concerned with the formulation of certain quantum mechanical problems in terms of finite-difference equations, and thence in terms of random walks; some particular numerical results are given as an indication of the rate of convergence of the Monte Carlo method when used for the solution of such problems. Similarly, Kimball's paper is principally interesting for the mathematical formulation of a physical problem rather than for the method of solution.

Other papers contain more substantial contributions to our understanding of machine methods of computation. The article by Hill outlines the theory of the liquid-drop model of nuclear fission and gives details of the numerical techniques used on the SSEC for certain prototype computations in this field. The paper by Woodbury describes shielding computations by a Monte Carlo method, run on the "405-603 Combination," the famous machine improvised by Woodbury and Toben by combining two standard IBM machines. This combination was the predecessor of the IBM Card-Programmed Calculator. The approximation of a given function by a linear combination of exponentials is the subject of Householder's paper. Several methods—graphical, least squares, iterations based on Poisson-distributed errors—are compared. C. C. Hurd gives a functional description of the Card-Programmed Calculator. This is the so-called "Model I" which was in process of construction at the time of the presentation of this paper. Barber's paper describes the evaluation of certain polynomials. Brinkley and Smith report on progress in their long-standing problem which consists mathematically of the solution of numerous systems of nonlinear algebraic equations. They describe a special method applicable in certain degenerate cases and mention briefly the use of the Card-Programmed Calculator. The paper of Flanders and Shortley is a discussion of several iterative methods for the computation of wave functions; particularly methods employing matrix operators which are obtainable by forming polynomials of simpler operators. Herget gives a sketchy description of the numerical integration of a system of second order ordinary differential equations on an IBM 602-A calculator.

The most important papers in this volume, from the standpoint of numerical analysis, are those of L. H. Thomas and J. H. Curtiss. The former is an

exhaustive treatment of questions of approximating parabolic differential equations by finite-difference methods, the errors in the solution caused by such approximation, stability and rate of convergence. While the discussion is purposely nonrigorous, it is fundamental and complete. An unpublished but frequently quoted method due to J. VON NEUMANN is used.

The paper by J. H. Curtiss, which occupies almost one-fourth of the volume, is the only one written with the care and rigor customary in mathematical publications, while the rest of the volume is in the nature of an exchange of information on techniques, with emphasis on timeliness, effectiveness in getting results, and "know-how" rather than on scientific requirements. The paper begins with a historical survey of the field, which is shown to be much older than its currently fashionable name of "Monte Carlo Method." There follows an exposition which brings together results, some of which had previously been widely scattered and relatively inaccessible in the literature on mathematical physics and theory of probabilities. Significant new results are presented on the standard error of the solution, the number of samples (random walks) necessary to achieve a given accuracy, the mean length of a walk and thus the expected amount of computing effort necessary for a given problem, and on importance sampling.

The editorial work is excellent throughout the volume, except for the discussions at the end of the papers, where misprints like the following occur: p. 27, l. 6, for "Prof. Katz and Dr. Dunning" read "[M.] KAC and [M. D.] DONSKER"; p. 78, l. 22, for "Dr. Foster" read "[G. E.] FORSYTHE."

F. L. ALT

NBSCL

1007. IBM Industrial Computation Seminar, edited by CUTHBERT C. HURD, September 1950, 103 p. 22 × 28.5 cm.

Participants in this seminar discussed fundamental computational methods used by research engineers and scientists in a wide variety of research problems. Particular attention was drawn to computational techniques developed in the fields of chemistry and petroleum. The contents of this volume are as follows:

1. "The role of the punched card in scientific computation" by WALLACE J. ECKERT
2. "Machine calculation of the plate-by-plate composition of a multicomponent distillation column" by ASCHER OPLER and ROBERT C. HEITZ
3. "Continuous distillation design calculations with the IBM Card-Programmed Electronic Calculator" by ARTHUR ROSE, THEODORE J. WILLIAMS and WILLIAM S. DYE, III
4. "Application of automatic computing methods to infrared spectroscopy" by GILBERT W. KING
5. "Correlation and regression analysis" by E. L. WELKER
6. "Pile-driving impact" by EDWARD A. SMITH
7. "Punched card mathematical tables on standard IBM equipment" by ELEANOR KRAWITZ
8. "The solution of simultaneous linear equations using the IBM Card-Programmed Electronic Calculator" by JUSTUS CHANCELLOR, JOHN W. SHELDON and G. LISTON TATUM

9. "Two applications of the IBM Card-Programmed Electronic Calculator: The Gauss-Seidel method of solution of simultaneous linear equations; Approximating the roots of a polynomial equation" by I. C. LIGGETT
10. "Matrix by vector multiplication on the IBM Type 602-A Calculating Punch" by ELEANOR KRAWITZ
11. "Numerical solution of two simultaneous second-order differential equations" by WALTER H. JOHNSON
12. "Numerical evaluation of integrals of the form $\int_a^b f(x)g(x)dx$ " by JOHN W. SHELDON
13. "The use of orthogonal polynomials in curve fitting and regression analysis" by JACK SHERMAN
14. "General-purpose ten-digit arithmetic on the IBM Card-Programmed Electronic Calculator" by STUART R. BRINKLEY, JR., G. L. WAGNER and R. W. SMITH, JR.
15. "Remarks on distillation calculations" by JOHN W. DONNELL
16. Some applications of the Monte Carlo method:
 - "Matrix inversion on the IBM Accounting Machine" by Ascher Opler
 - "Remarks on finding roots of, and inverting, a matrix" by GILBERT W. KING
 - "Remarks on the Monte Carlo method" by CUTHBERT C. HURD
17. "Plotting punched card data using the IBM Type 405 Accounting Machine" by PAUL T. NIMS
18. "A method for evaluating determinants and inverting matrices with arbitrary polynomial elements by IBM punched card methods" by L. E. GROSH, JR. and E. UDSIN

The article by Chancellor, Sheldon and Tatum describes a method for solving simultaneous linear algebraic equations up to and including order twenty-one using the Crout method. With this procedure, a 20×20 matrix can be inverted in two hours. The prescribed method assumes a model I CPC equipped with one (941) auxiliary storage. The article by Brinkley, Wagner and Smith, Jr., describes a general purpose ten-digit arithmetic set-up for the model I CPC. In addition to the elementary arithmetic operations, square root and certain combinations, one can automatically select alternative computational routines and also select input data.

R. K. ANDERSON

NBSCL

1008. IBM, Industrial Computation Seminar, *Proc.* edited by Cuthbert C. HURD, August 1951, 148 p. 28.5×22 cm.

The participants of this seminar were research engineers and scientists representing computing facilities which employ IBM card-programmed electronic calculators. The contents of this volume are as follows:

1. "Application of the IBM Card-Programmed Electronic Calculator to engineering procedures at the Glenn L. Martin Company" by W. B. KOCH
2. "Reduction of six-component wind tunnel data using the IBM Card-Programmed Electronic Calculator, Model II" by M. L. LESSER
3. "IBM Card-Programmed Electronic Calculator operations using a Type 402-417BB and 604-2" by H. E. TILLITT, M. KENYON and B. OLDFIELD

4. "The Combomat" by J. D. MADDEN
5. "The IBM Type 604 Electronic Calculating Punch as a miniature electronic calculator" by P. T. NIMS
6. "General-purpose floating point control panels for the IBM Card-Programmed Electronic Calculator" by N. A. PATTON, K. BERGER and L. R. TURNER
7. "Catapult takeoff analysis" by J. R. LOWE
8. "Computation of loan amortization schedules on the IBM Card-Programmed Electronic Calculator" by C. H. GUSHEE
9. "Techniques for handling graphical data in the IBM Card-Programmed Electronic Calculator" by W. D. BELL
10. "Calculation of the flow properties in an arbitrary two-dimensional cascade" by J. T. HORNER
11. "Automatic calculation of the roots of complex polynomial equations using the IBM Card-Programmed Electronic Calculator" by J. GALLISHAW, JR.
12. "A recursion relation for computing least square polynomials over moving-arcs" by GEORGE R. TRIMBLE, JR.
13. "Numerical solution of second-order non-linear simultaneous differential equations" by HENRY S. WOLANSKI
14. "Matrix inversion and solution of simultaneous linear algebraic equations with the IBM Type 604 Electronic Calculating Punch" by GEORGE W. PETRIE, III
15. "The determination of eigenvectors and eigenvalues of symmetric matrices" by EVERETT C. YOWELL
16. "An application of the IBM Card-Programmed Electronic Calculator to analysis of airplane maneuvering horizontal tail loads" by LOGAN T. WATERMAN
17. "Fifth-order aberration in an optical system" by RUTH K. ANDERSON
18. "Theory of plastic vibrations of helicopter fuselages" by PETER F. LEONE
19. "Machine procedure for computation of elastic vibrations of helicopter fuselages" by WILLIAM P. HEISING

Nims describes how a 604 Electronic Calculating Punch may be used as a small sequence-controller calculator. The 604 can perform a different arithmetic operation on each card. In addition it is possible for the machine to shift, look up entries in a table of functions, calculate square roots and stop when certain check conditions do not hold.

Patton, Berger and Turner have designed general-purpose floating point control panels for the CPC which give the coder a great deal of flexibility. In addition to the basic arithmetic operations, one has three independent program fields and four numerical fields (per card), automatic extraction of integral roots from second to seventh, eight kinds of conditional operations and machine stop with automatic list cycles on all improper operations (division by zero, overflow).

Gallishaw's technique for finding complex roots of a polynomial is an iterative procedure employing synthetic division and Newton's method.

Using standard trial values, it takes from twenty to twenty-five minutes to find all the roots of a complex 7th degree equation. If any of the roots are approximately known in advance, they can be used instead of the standard trial values, thus cutting down the solution time. The process uses eight-digit arithmetic throughout.

Petrie has devised a method for matrix inversion and solution of simultaneous linear equations using a 604 Electronic Calculating Punch, a tabulator and two reproducers.

The procedure consists of a continuous flow of cards through these four machines, the cycle being repeated N times to invert an N th order matrix with approximately N^2 cards each time. The size of the matrix is unlimited, and a check operation is included.

R. K. ANDERSON

NBSCL

1009. IBM *Technical News Letter* No. 3, Applied Science Department, December 1951, 106 pages, 22 × 28 cm.

The contents are as follows:

1. "Mass spectrometer calculations on the IBM Type 602-A Calculating Punch" by W. H. KING, JR. and WILLIAM PRIESTLY, JR.
2. "Computations of inverse matrices by means of IBM machines" by JACK SHERMAN
3. "Linear equations and matrix inversion" by ERIC V. HANKAM
4. "Matrix and vector algebra" by Eric V. Hankam
5. "Notes on the IBM Type 604 Electronic Calculating Punch and Type 602-A Calculating Punch" by Eric V. Hankam
6. "Stop controls for the IBM Type 604 Electronic Calculating Punch," by BRUSE MONCREIFF
7. "An eight-digit general-purpose control panel" by WILLIAM P. HEISING
8. "Interpolation on the IBM Card-Programmed Electronic Calculator" by STUART R. BRINKLEY, JR. and G. L. WAGNER.

The article by King and Priestly describes in detail the control panels used in performing a matrix multiplication of the 20th order and in subsequent normalization steps. The general-purpose board described by Heising (for use with either the IBM Type 604 Electronic Calculator or the IBM Card-Programmed Calculator) calculates, in addition to the basic arithmetic operations, \sqrt{A} , $\exp(\pm A)$, $\log(1 \pm A)$, $\sin A$, $\cos A$, $\sinh A$, $\cosh A$, $\arcsin A$, $\arctan A$, $\operatorname{arcsinh} A$, and $\operatorname{arctan} A$.

R. K. ANDERSON

NBSCL

NEWS

Association for Computing Machinery.—The University of Toronto Computation Centre acted as host to the Association for the fall meeting held September 8 and 9, 1952. The University had on display for its guests the three computers located at the Computation Centre, namely, FERUT, a newly installed large-scale computer built by Ferranti Limited which was undergoing tests at the time; UTEC, the model computer built at the University;

and BERTIE, a noughts and crosses machine built by Rogers Majestic Electronic Laboratories. The program for the meeting was as follows:

Sept. 8, 1952 10:00 a.m. to 12 noon

General Session	S. B. WILLIAMS, President, ACM, Chairman
Welcoming address	S. SMITH, President, Univ. of Toronto
Pure and Applied programming	M. V. WILKES, Director, University Mathematical Laboratory, Cambridge, England
MANIAC	N. METROPOLIS, E. F. KLEIN, W. ORVEDAHL, J. R. RICHARDSON, H. B. DEMUTH, J. B. JACKSON, Los Alamos Scientific Laboratory
Chebyshev polynomials in the solution of large scale linear systems	C. LANCZOS, NBSINA

Sept. 8, 1952, 2:00 p.m. to 5:00 p.m.

Programming	C. C. HURD, IBM, New York, Chairman
Interpretative sub-routines	J. M. BENNETT, D. G. PRINZ, M. L. WOODS, Ferranti, Ltd., Manchester, England
Machine aids to coding	E. J. ISAAC, NRL
Computer aids to code checking	I. C. DIEHM, NBSCL
Input scaling and output scaling for a binary calculator with decimal output	E. F. CODD, H. L. HERRICK, IBM
Compiling routines	R. K. RIDGWAY, Eckert-Mauchly Division of Remington RAND, Inc., Philadelphia
Function table techniques in programming	R. F. SHAW, Electronic Computer Corp., New York

Sept. 8, 1952, 2:00 p.m. to 5:00 p.m.

Computing machines	K. F. TUPPER, Dean, Faculty of Applied Science and Engineering, Univ. of Toronto, Chairman
The logical design of the Oak Ridge digital computer	C. L. PERRY, Oak Ridge National Laboratory
The Oak Ridge automatic computer	J. C. CHU, Argonne National Laboratory
Designing a low cost general-purpose computer	W. E. DOBBINS, Computer Research Corp., Hawthorne, California
The University of Toronto model electronic computer	R. F. JOHNSTON, Computation Centre, Univ. of Toronto
A description of the electronic computer at the Institute for Advanced Study	G. ESTRIN, Institute for Advanced Study, Princeton

Banquet, Sept. 8, 1952, 7:30 p.m.

Guest Speaker	
The St. Lawrence Seaway Project	R. H. SAUNDERS, Chairman, Hydro-Electric Power Commission of Ontario

Sept. 9, 1952, 9:00 a.m. to 12:00 noon

Machine calculations	MINA REES, ONR, Chairman
Errors in iterative solutions of linear systems	A. S. HOUSEHOLDER, Oak Ridge National Laboratory

- The numerical solution of a partial differential equation on the IBM Type 701 Electronic Data Processing Machines
 A numerical solution of the helium wave equation with the SEAC
 Matrix inversion by partitioning
 The use of subroutines on SEAC for numerical integrations of differential equations and for Gaussian quadrature
 Use of continued fractions in high-speed computing
- Sept. 9, 1952, 9:00 a.m. to 12:00 noon
- Storage devices
 The testing of cathode ray tubes for use in the Williams type storage system
 Williams tubes selection program
 Improvement of Williams memory reliability
 Ferro electric materials as storage elements for digital computers and switching systems
 An experimental rapid access memory using diodes and capacitors
- Sept. 9, 1952, 2:00 p.m. to 5:00 p.m.
- Logical design
 Symbolic synthesis of digital computers
 Logical or non-mathematical programmes
 A simplified universal Turing machine
 Simple learning by a digital computer
 An analysis by arithmetical methods of a calculating network with feedback
- Sept. 9, 1952, 2:00 p.m. to 5:00 p.m.
- Engineering
 A high-speed magnetic-core output printer
 Development of computer components and systems
 An electronic analogue machine for computing the roots of algebraic equations of degrees through the eighth
 Analogue calculation of polynomials and their zeros
 Operating efficiencies and characteristics of the computing machines at Aberdeen Proving Ground
 Installation of a large electronic computer
- D. W. LADD, J. W. SHELDON, Applied Science Dept., IBM, New York
 J. H. WEGSTEIN, NBSCL
 M. LOTKIN, R. REMAGE, BRL, Aberdeen Proving Ground
 P. RABINOWITZ, NBSCL
 D. TEICHROEW, NBSINA
 J. A. RAJCHMAN, RCA, Chairman
 A. ROBINSON, Ferranti, Ltd., Manchester England
 J. C. CHU, R. J. KLEIN, Argonne National Laboratory
 R. SCHUMANN, Argonne National Laboratory
 J. R. ANDERSON, BTL, New York
 A. W. HOLT, NBSCL
 W. H. WATSON, Head Dept. of Physics, Univ. of Toronto, Chairman
 I. S. REED, MIT
 C. S. STRACHEY, National Research Development Corp., London, England
 E. F. MOORE, BTL, New Jersey
 A. G. OETTINGER, The Computation Laboratory, Harvard Univ.
 L. C. ROBBINS, Burroughs Adding Machine Company, Philadelphia
 B. V. BOWDEN, Ferranti, Ltd., Manchester, England
 B. M. GORDON, R. N. NICOLA, Laboratory for Electronics, Inc., Boston
 W. S. ELLIOTT, Research Laboratories of Elliott Brothers (London), Ltd., Borehamwood, England
 L. LOFGREN, Research Institute of National Defense, Radio Department, Stockholm, Sweden
 M. G. SCHERBERG, J. F. RIORDAN, Wright Air Development Center, Wright-Patterson Air Force Base, Ohio
 H. SPENCE, Ballistic Research Laboratories, Aberdeen Proving Ground
 L. R. JOHNSON, Hdqs., U. S. Air Force

NBSNAML.—On May 15, 1952, at Washington, D. C., the National Applied Mathematics Laboratories sponsored a meeting of Mathematical Tables in the Light of Electronic Computers. The program for the meeting was as follows:

Work in Progress

J. TODD, NBSCL, Chairman

E. T. GOODWIN, Royal Society
 Mathematical Tables Committee
 J. P. WONG, RAND Corporation
 M. ABRAMOWITZ, NBSCL
 G. BLANCH, NBSINA

(There was also a written communication from M. PICONE, Instituto Nazionale per le Applicazione del Calcolo, Rome.)

Current Needs

F. L. ALT, NBSCL, Chairman

Physics
 Aerodynamics

G. BREIT, Sloane Physics Dept., Yale Univ.
 C. W. JONES, Dept. of Math., Liverpool Univ.

Military applications

L. S. DEDERICK, Ballistic Research Laboratories, Aberdeen Proving Ground

Statistics

C. EISENHART and D. TEICHROEW, NBSSEL

Is the preparation of a single volume 7-place table of $J_n(x)$ of integral order up to 120, at interval .01, desirable?

R. C. ARCHIBALD, Brown Univ.

Are tables of Bessel functions of the second kind and of orders higher than 20 desirable?

R. C. ARCHIBALD, Brown Univ.

Long Term Policy

P. M. MORSE, MIT, Chairman

Round table discussion of which the principal speakers were:

C. C. HURD, IBM Corporation
 D. H. LEHMER, NBSINA
 E. T. GOODWIN, Royal Society
 Mathematical Tables committee
 J. H. CURTISS, NBSNAML

Editorial Matters

D. H. LEHMER, NBSINA, Chairman

Index of Statistical Tables

D. L. WALLACE, Dept. of Mathematics, Princeton Univ.

Index of Mathematical Tables

C. W. JONES, Dept of Mathematics, Liverpool, University

Editing of Mathematical Tables

G. M. CLEMENCE, Naval Observatory, Washington, D. C. and D. H. SADLER, His Majesty's Nautical Almanac Office

The preparation of a bibliography of calculating machines developed from lists of Aiken, Travis and an Australian physicist, from Patent Office records, and from other sources.

R. C. ARCHIBALD, Brown Univ.

An organization to assemble for MTAC tabular material news from the very numerous research centers of this and other countries

R. C. ARCHIBALD, Brown Univ.