

The
use
of the
Hollerith
Type 555
Electronic
Calculator
for
scientific
computation

By J. E. HAILSTONE
Computer Group
Atomic Energy
Research Establishment
Harwell

**Published, with permission of A.E.R.E., Harwell,
by the Hollerith Group of Companies:**

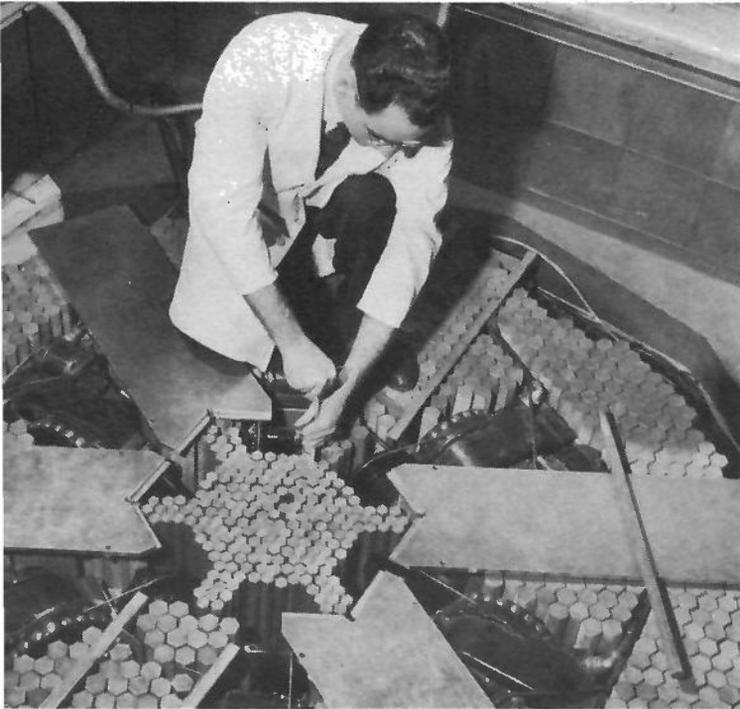
**The British Tabulating Machine Company Limited
Hollerith Machines (South Africa) (Proprietary) Limited
Hollerith (India) (Private) Limited
Hollerith (Central Africa) (Private) Limited
Hollerith (Australia) Pty. Limited**

**The
use
of the
Hollerith Type 555
Electronic
Calculator
for
scientific
computation**

**By J. E. HAILSTONE
Computer Group
Atomic Energy
Research Establishment
Harwell**

Contents

	Page
Introduction	5
General Description of the Machine	7
Programming and Plugging	11
Reviewing and explaining a number of Typical Applications	12
The Use of Rational Approximations for the Elementary Functions	15
Monte Carlo Calculations in Nuclear Reactor Design	17
Data Processing	23
Nuclear Structure Theory - a Combinatorial Problem	27
Heat Flow Equations	31
Evaluations of Determinants and the Solution of Simultaneous Equations	35
General Comments	37
Legend	38
Appendix 'A' - $\left \text{Log}_e x \right $ Routine	
Appendix 'B' - Monte Carlo Example	



The reactor ZEUS (Zero Energy Uranium System) at Harwell. ZEUS has a highly enriched uranium core surrounded by a natural uranium breeding blanket.



The Cockcroft - Walton high voltage generator with the injector of the proton - linear accelerator in the centre - background.

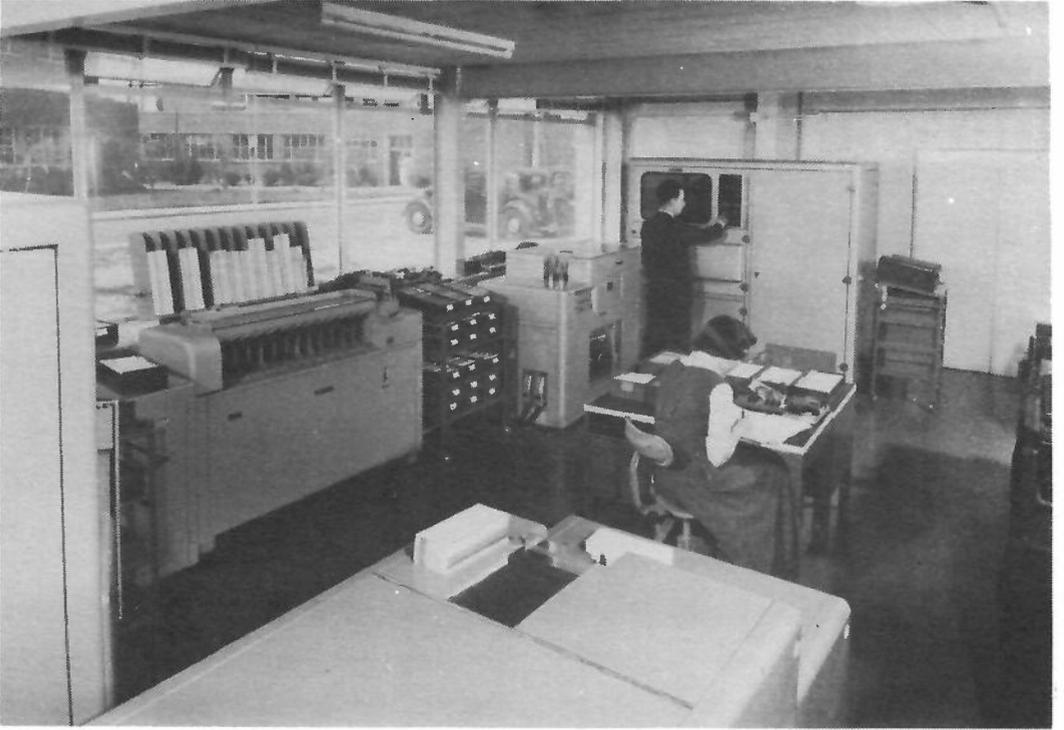
HERMES - the heavy element radio - active material - electromagnetic separator which began operation in January 1957.

Introduction

(i) The Hollerith Type 555 Electronic Calculator is manufactured by The British Tabulating Machine Co. Ltd., and is a development of their Types 542 and 550 Electronic Calculators. The "555" was designed primarily for commercial and accounting work, but it has now proved to be also a machine which can be used usefully and efficiently in scientific computation.

The following account is intended to show, by means of some examples, the powers of the Type 555 Electronic Calculator in various types of scientific work. These examples have been chosen from problems dealt with by the Computer Group at A.E.R.E. Harwell, which has a Ferranti Mercury electronic digital computer and a Hollerith punched card machine installation including two Type 555 Electronic Calculators. The latter installation which is complementary to, and not in competition with, the Mercury computer is used in general for two types of work; Monte Carlo calculations in connection with reactor studies, and the processing of experimental data.

(ii) Although the Type 555 Electronic Calculator has been chosen primarily to carry out this work at Harwell, various other problems have been dealt with and this account includes a selection of such problems. It is intended to show what order of complexity and size of problem can be conveniently handled within the capacity of the machine, and to give information about the time required to program such problems.



The installation of two Hollerith Type 555 Electronic Calculators at A.E.R.E., Harwell



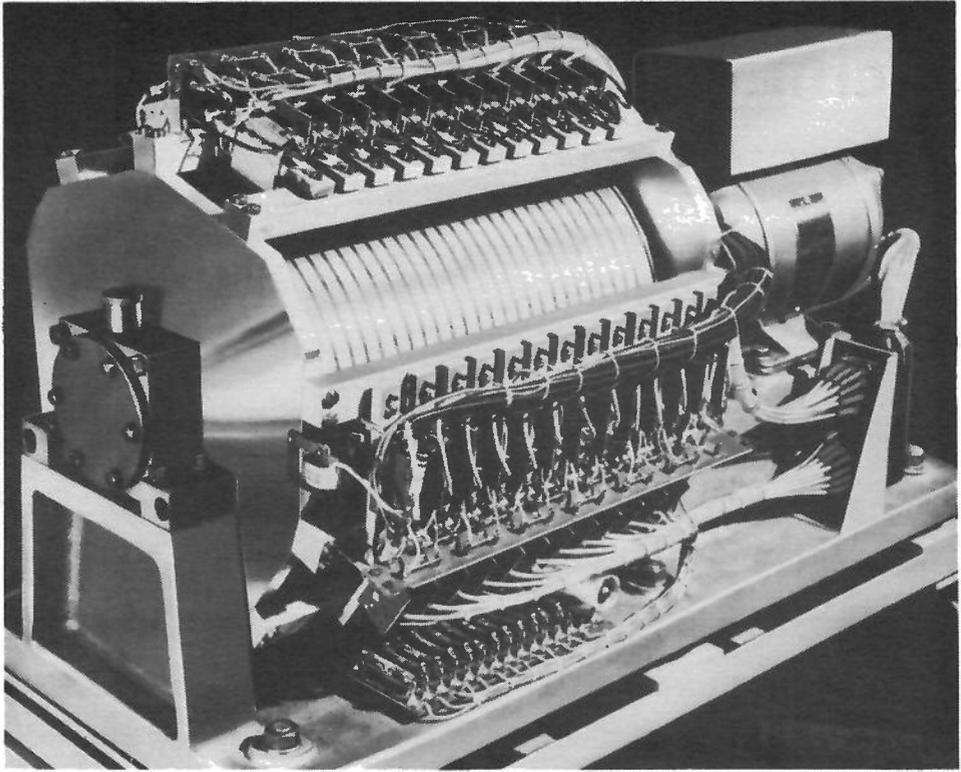
General description of the machine

The Type 555 is an Electronic Calculator with card input and output, controlled by means of manually plugged control panels; the machine reads information from a punched card, performs a sequence of arithmetical or logical operations on this information (as defined by a program), and punches the results into the same or following cards. The program comprises a maximum of 150 steps but any group of these can be repeated, either a fixed number of times or until some desired result is achieved. The machine operates at a maximum speed of 6,000 card passages per hour. At this speed of input a fixed time of 600 milliseconds is allowed for reading in the numerical data and carrying out the calculation regardless of the complexity of the program. Should a calculation on a job call for more than this standard time allowance, then card feeding will be held up until the calculation is completed. Experience here ranges down to the feeding of one card every 50 minutes.

The machine uses decimal arithmetic with "words" or numbers of 10 digits; the internal representation is in binary coded decimal, each digit being represented by four binary digits which in all operations are dealt with in parallel. There is a magnetic drum store for 105 words of numerical data arranged in five channels each of 21 words. In a program step a number is withdrawn along either or both of two highways, put through functional units (e.g. adder, divider) according to the operation to be performed by that step, and the result returned to store. All operations take place in multiples of a word time or cycle ($13/14$ ths milliseconds) during which, for example, the sum of two numbers can be formed and stored. Some operations, including multiplication, although constituting only one program step, occupy a varying number of word times depending on the size of the numbers involved.

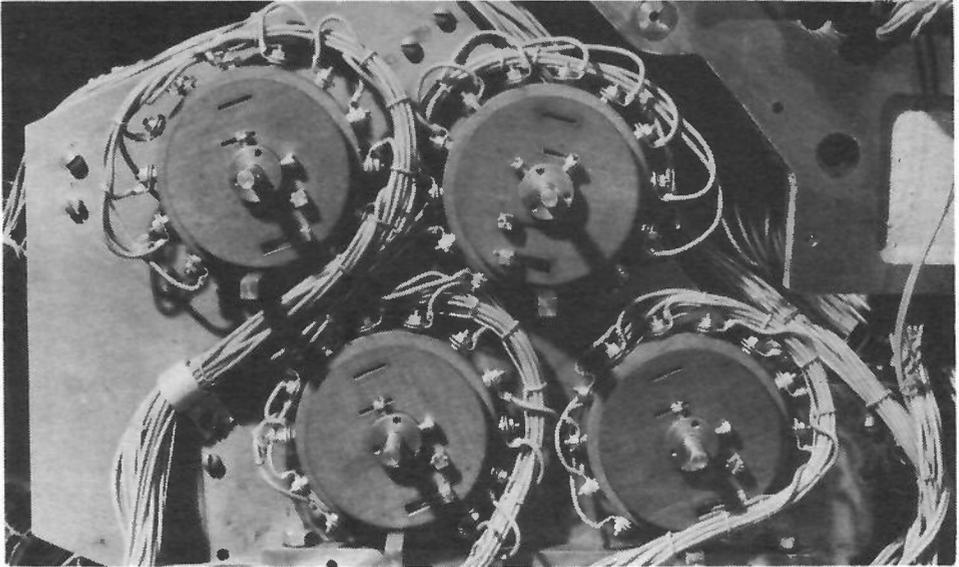
Among features which make the Type 555 Electronic Calculator a powerful and efficient tool for scientific computation are:-

- (1) Fast input and output.
- (2) High arithmetic speeds.
- (3) Fairly large storage capacity.
- (4) A number of program steps much larger than usual in punched card calculators.



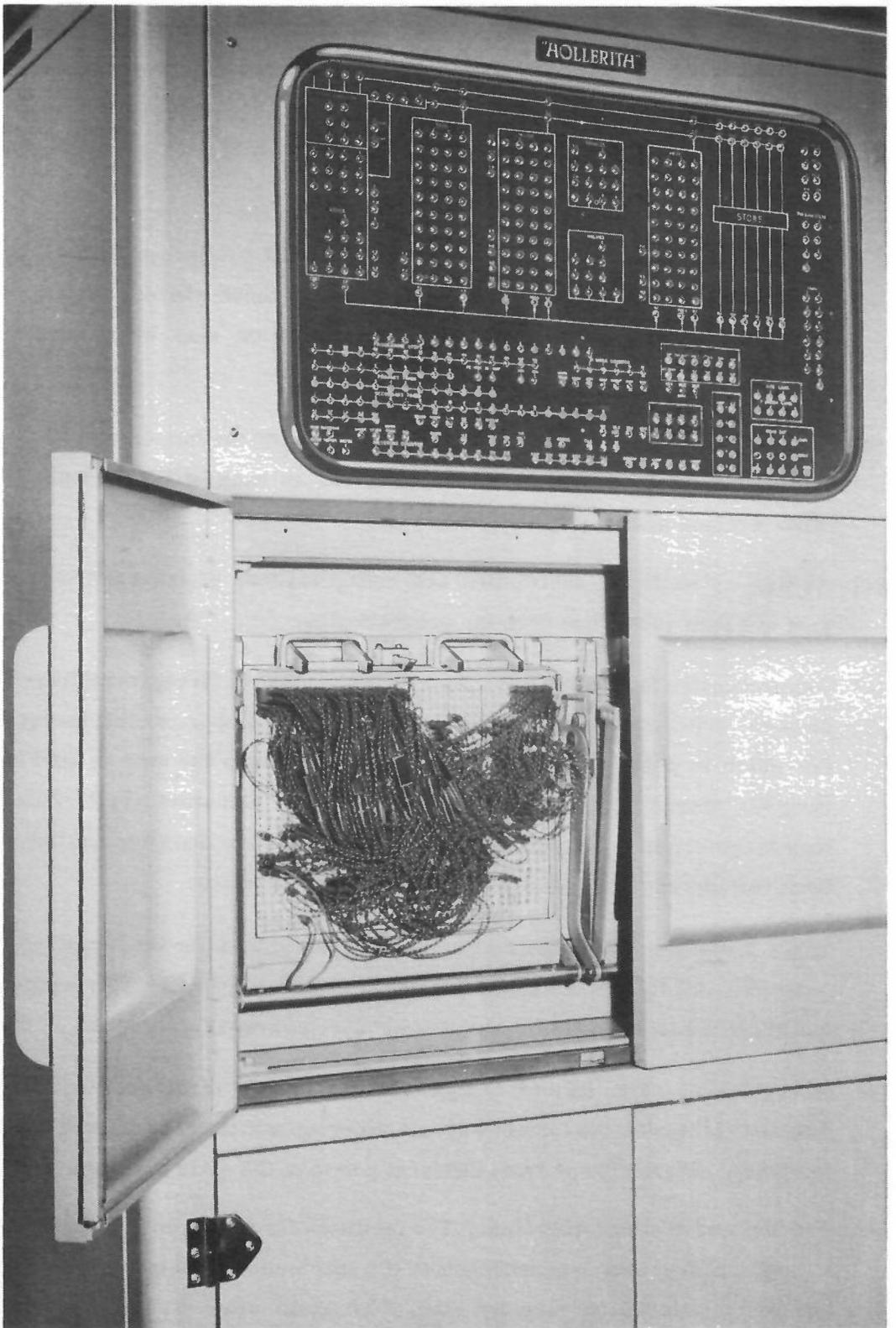
The key to the calculating capacity of the 555 lies in this magnetic drum store, which revolves at 3,000 r.p.m.

These emitters are used for automatic punching of designations identifying specific conditions in the punched card



To summarise, the important features are:-

- (a) Parallel input and output. Reading available from all 80 columns of a card and punching into 60 columns of a card together with emitted information at a maximum speed of 6,000 cards per hour.
- (b) Three Address system. The basic addition function (Source one plus or minus Source two to Destination) can be carried out in one basic time unit = a word time - so that the machine uses a three-address system of instructions, although more than one Destination may be addressed at any time.
- (c) Three registers, of 10 decimal digits each, provide the immediate access store, but in 1 word time all 5 words one in each of the 5 channels also become available.
- (d) As well as addition, subtraction and multiplication, division and square root are built in and use only one program step.
- (e) Comprehensive facilities for testing a number. T.C.O. (true, complement or zero) units may be set up to record whether a number passing through the adder is positive, non negative, negative or zero and may be used to vary the course of the program. These enable elaborate iterative and logical operations to be performed. "Jumping" instructions are available from calculated quantities and also from card designation.
- (f) Table Look-up. A numerical table can be stored on the drum and consulted during the calculation using a "table look-up" programming facility in conjunction with "repeat" and "special repeat" functions.
- (g) Sub-routines. The logical design enables one to treat sections of a program in much the same way as sub-routines in a stored program computer, entering these from different points in the main program.
- (h) Proving and testing programs. The facilities for testing and monitoring a program are very comprehensive; the machine can be taken manually through a calculation step by step, the state after each step being shown on a display panel.



Programming and plugging

The general logical design of the machine is straightforward, and apart from a few special operations the instruction code is quickly acquired. The limit of 150 program steps and some slight restrictions on other facilities is countered by the fact that there is scope for considerable ingenuity in programming, especially when the program as first written exceeds capacity. This has frequently happened at Harwell, but it can be said that some means has always been found to overcome this apparent difficulty.

The actual plugging of a program by means of the control panel might seem at first sight to involve difficulties because of the closely packed plugs; plugging, however, can be reduced to an almost faultless operation if it is carefully planned at the outset. A simple code and routine is used, which enables even a stranger to a program to follow a calculation when checking through all its stages of plugging and operation; this greatly assists in fault-finding.

The design of the control panels is such that the number of times a particular function may be used is variable; there is an upper limit to the total number of functions of all kinds which may be called, but experience so far shows that this limit is unlikely to be reached and it is therefore not a factor which has to be taken into account by the programmer.

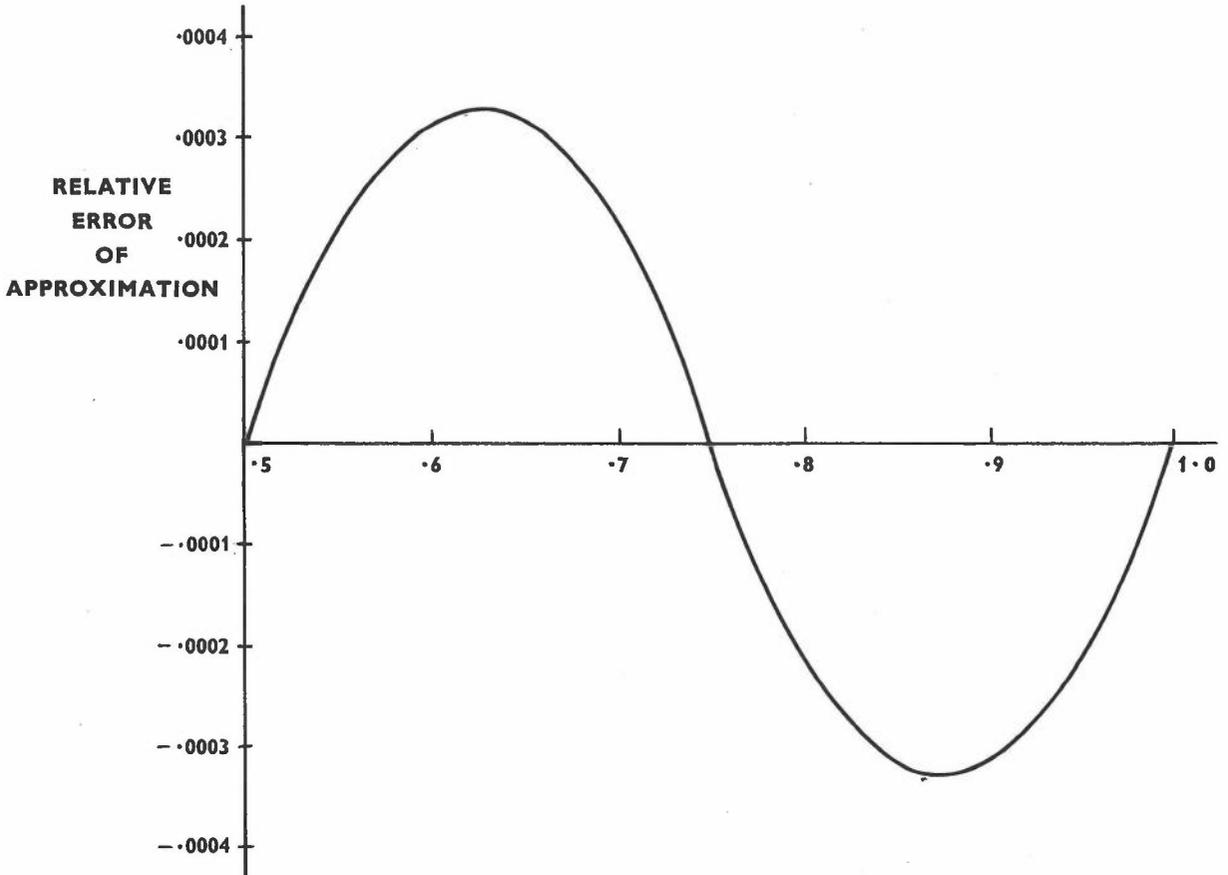
**Reviewing
and
explaining
a number
of
typical
applications**

Examples of current problems have been chosen to show the use of the various features of the Type 555 Electronic Calculator: these have been presented to the Harwell Computing Group by scientists who have a practical need for large scale calculations. Some of these applications have become standard routines on which already hundreds of hours of machine time have been used.

- 1** The use of rational approximations for the elementary functions.
- 2** Monte Carlo calculations in nuclear reactor design.
- 3** Data processing.
- 4** Nuclear Structure Theory - a Combinatorial Problem.
- 5** Heat Flow Equations.
- 6** Evaluation of determinants and the solution of simultaneous linear equations.

ERROR CURVE (APPROXIMATION - FUNCTION) / (FUNCTION)

	0	1	2	3	4	5	6	7	8	9	Δ
I·00	0·0 00 000	01 000	01 998	02 996	03 992	04 988	05 982	06 976	07 968	08 960	990
·01	0·0 09 950	10 940	11 929	12 916	13 903	14 889	15 873	16 857	17 840	18 822	981
·02	0·0 19 803	20 783	21 761	22 739	23 717	24 693	25 668	26 642	27 615	28 587	972
·03	0·0 29 559	30 529	31 499	32 467	33 435	34 401	35 367	36 332	37 296	38 259	962
·04	0·0 39 221	40 182	41 142	42 101	43 059	44 017	44 973	45 929	46 884	47 837	953
I·05	0·0 48 790	49 742	50 693	51 643	52 592	53 541	54 488	55 435	56 380	57 325	944
·06	0·0 58 269	59 212	60 154	61 095	62 035	62 975	63 913	64 851	65 788	66 724	935
·07	0·0 67 659	68 593	69 526	70 458	71 390	72 321	73 250	74 179	75 107	76 035	926
·08	0·0 76 961	77 887	78 811	79 735	80 658	81 580	82 501	83 422	84 341	85 260	918
·09	0·0 86 178	87 095	88 011	88 926	89 841	90 754	91 667	92 579	93 490	94 401	909



I·40	0·3 36 472	37 186	37 900	38 613	39 325	40 037	40 749	41 460	42 170	42 880	710
·41	0·3 43 590	44 299	45 007	45 715	46 423	47 130	47 836	48 542	49 247	49 952	705
·42	0·3 50 657	51 361	52 064	52 767	53 470	54 172	54 873	55 574	56 275	56 975	699
·43	0·3 57 674	58 374	59 072	59 770	60 468	61 165	61 861	62 558	63 253	63 948	695
·44	0·3 64 643	65 337	66 031	66 724	67 417	68 109	68 801	69 492	70 183	70 874	690
I·45	0·3 71 564	72 253	72 942	73 630	74 318	75 006	75 693	76 380	77 066	77 751	685
·46	0·3 78 436	79 121	79 805	80 489	81 172	81 855	82 538	83 219	83 901	84 582	680
·47	0·3 85 262	85 942	86 622	87 301	87 980	88 658	89 336	90 013	90 690	91 366	676
·48	0·3 92 042	92 718	93 393	94 067	94 741	95 415	96 088	96 761	97 433	98 105	671
·49	0·3 98 776	99 447	00 118	00 788	01 457	02 126	02 795	03 463	04 131	04 798	667
I·50	0·4 05 465	06 132	06 798	07 463	08 128	08 793	09 457	10 121	10 784	11 447	663
n	1	2	3	4	5	6	7				
$\ln 10^n$	2·302 585	4·605 170	6·907 755	9·210 340	11·512 925	13·815 511	16·118 096				



The use of rational approximations for the elementary functions

It is almost always necessary in mathematical work to use one of the elementary functions, e.g. $\log_e x$, $\sin 2\pi x$ etc.; it is therefore important to have available routines which will evaluate these functions since the storage of tables is seldom desirable or even possible. Much work has been done in the last few years on finding approximations to the elementary functions using rational functions (i.e. ratios of polynomials which are sometimes obtained by means of continued fractions). There are a number of rational approximations which can be used⁽¹⁾, and when a specified accuracy is required it is imperative that an efficient routine is chosen for the Type 555 Calculator which uses only the minimum number of program steps consistent with the required accuracy. The fact that the division and square root routines are built in is, of course, instrumental in making the machine a powerful computing tool, for without these being available immediately and occupying only one program step much of the work which can be done would be impossible if sub-routines for these had to be programmed.

Experience with the Type 555 shows that between 20 and 30 steps are needed to evaluate such functions with maximum relative errors in the range 0.001 to 0.0001. For example, one such routine for evaluating $|\log_e x|$, $0 < x \leq 1$, takes 28 steps for a maximum relative error of 0.0004.

The rational function taken is:

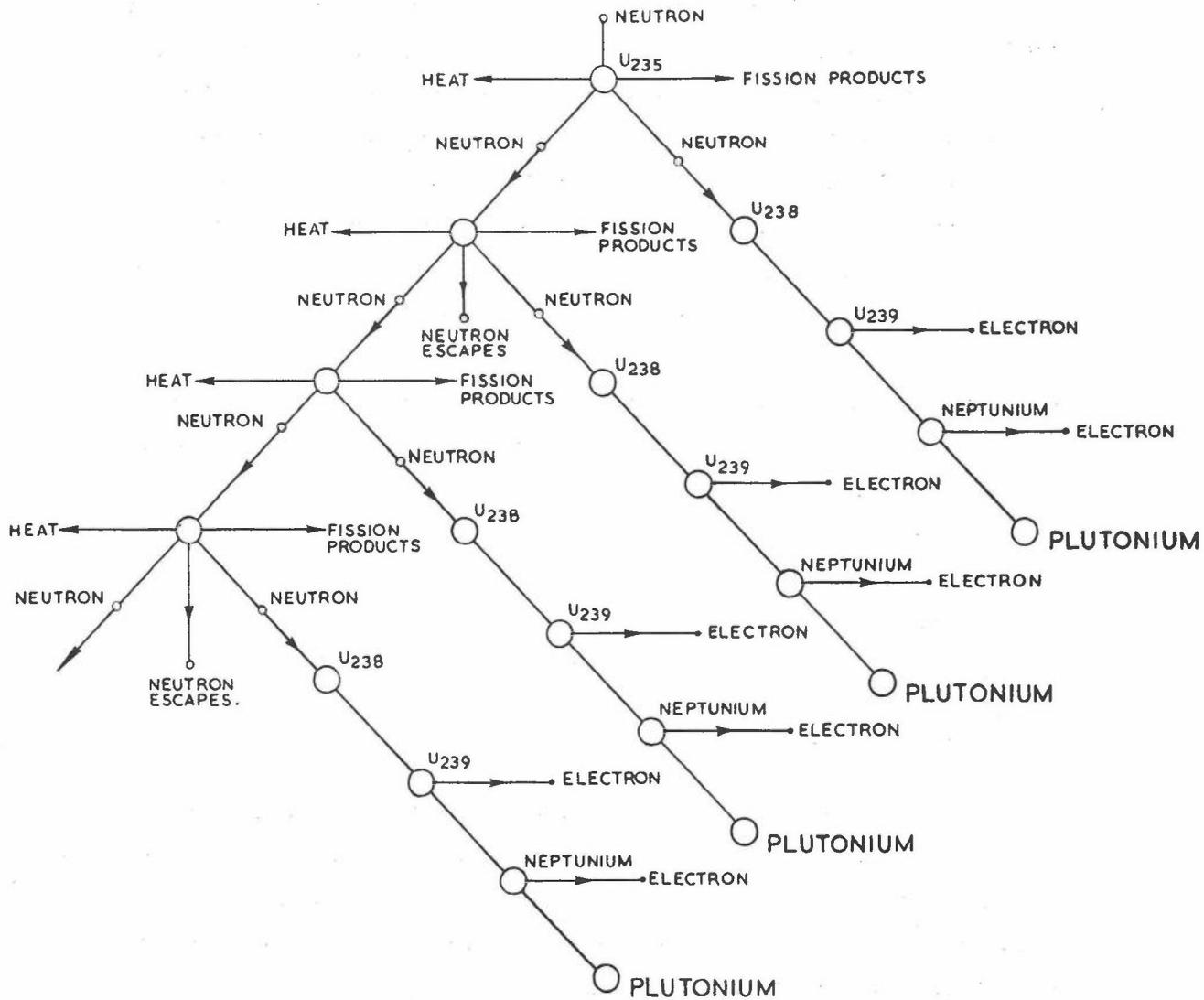
$$y = |\log_e x| = (1 - x) \frac{1.06535 + .31774x}{.38308 + x}, \quad \frac{1}{2} \leq x \leq 1.$$

x which may initially be 9 decimals, is scaled first by powers of 10 to obtain significant digits in the first five decimal places, and then by powers of 2 to bring it within the range $\frac{1}{2} \leq x \leq 1$. This scaling and the subsequent correction by adding multiples of $\log_e 2$ and $\log_e 10$ takes 18 of the 28 steps, the remaining 10 steps being used to evaluate y above.

Where two functions e.g. $|\log_e x|$ and $\sin 2\pi x$, are required in the same program it is possible to obtain rational functions which differ only in the numerical coefficients, thus allowing the same routine to be entered to evaluate both functions. (The $|\log_e x|$ routine is given in Appendix A).

(1) Rational Approximations of Functions. B. Carlson and M. Goldstein. Report LA-193 Los Alamos Scientific Laboratory. University of California.

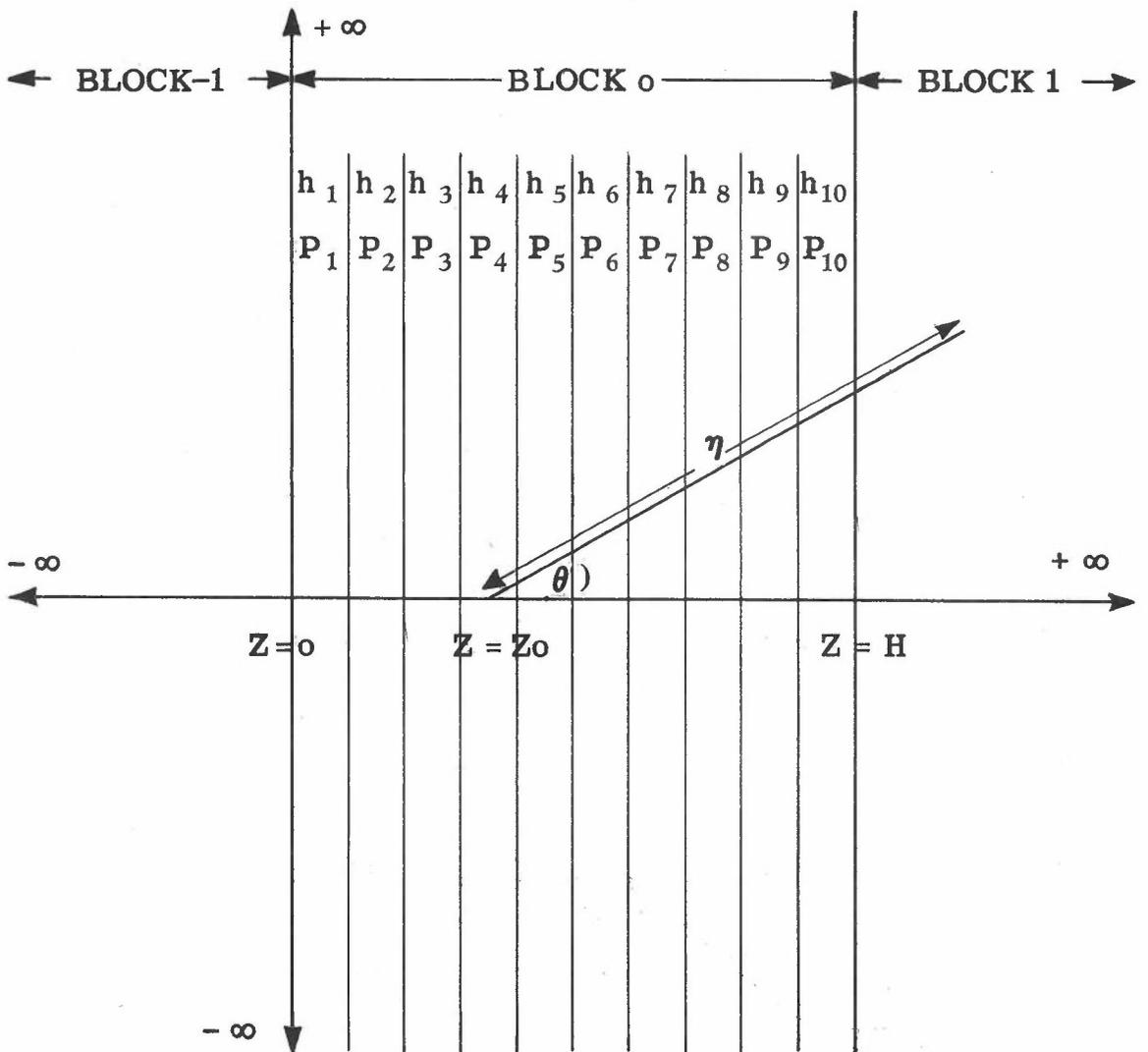
DIAGRAMMATIC ILLUSTRATION OF CHAIN REACTION IN A NUCLEAR REACTOR



2

Monte Carlo calculations in nuclear reactor design

The active agents in a nuclear reactor are the neutrons, and the essential purpose of all reactor calculations is to find out something about the neutron population. This population is discrete, that is, it is made up of individual neutrons whose behaviour is governed by known physical laws which involve the nuclear properties of the materials of the reactor. The number of neutrons is so great (10^{10} /c.c. is a low value for this) that no accuracy is lost if the population is assumed to be continuous, and an analytical mathematical treatment of the problem developed as is done in hydrodynamics; a vast amount of reactor calculation has been, and will continue to be, done on this basis. However, problems have arisen in which the analytical treatment leads to mathematical difficulties insuperable even with the most powerful computing machinery, and in such cases, the only method of attack is one which takes into account the discrete nature of the neutron population. The laws describing this neutrons' behaviour are all expressions of probability: for example, the probability of travelling the distance x before making a collision with a nucleus is proportional to e^{-px} where p depends on the medium. This collision can result in absorption or scattering of the neutron, or fission of the nucleus, with given relative probabilities. One can therefore study the behaviour of the system by computing theoretical life histories of sample batches of neutrons, using "random" numbers having suitable distributions to decide the course of each history; this is the mathematical equivalent of throwing dice or spinning a roulette wheel. Standard statistical theory can then be used to infer from the results of the sample calculation the properties of the population as a whole. This type of approach, called the Monte Carlo method, from an obvious analogy, has the great advantage of side-stepping all formal mathematical difficulties, but the disadvantage of giving a result subject to statistical error. This error can be estimated, and can be reduced by increasing the sample size (apart from using more refined techniques), but for high accuracy a very large sample may be needed. It is common practice to use samples of a few thousand neutrons from which one may usually expect to obtain final results correct to about 1%. This result is



a generalisation, however, and is not necessarily reliable for particular applications. The basic operation in this process is the tracking of a neutron from one collision to the next; the 150 program steps of the Type 555 will often accommodate this even in systems which seem complicated, so that a collision is accounted for in a single card passage. (1) (2)

The following is a simple problem of reactor design, in which subject the application of Monte Carlo methods has received considerable impetus. A sample of neutrons is to be tracked through a system consisting of a series of plane parallel-sided cells. A specified number of these cells, which may be of different materials and widths, make up one block, and the system consists of an infinite series of such blocks. Each neutron having a given starting position, $Z = Z_0$, and a random direction θ , may travel a certain distance, η , before collision with a nucleus. This distance depends upon the material through which the neutron passes and its magnitude is the result of sampling from a known probability distribution. A neutron having a collision will be absorbed or scattered with given probabilities, $(1-P_i)$, P_i , according to the material in which the collision occurs. The proportion of neutrons absorbed in each cell of a block is required, together with the neutron collision distribution for a block, each neutron being tracked until finally absorbed.

The program written for the Type 555 deals with a maximum of 10 cells per block, so that any system of 10 or less cells can be studied. For the purpose of calculation only the distribution and proportion of neutrons absorbed across any one block is of interest. Therefore, neutrons having collisions outside block 0 are considered as having their collisions in block 0.

The following information is given:

- (a) H , block width.
- (b) h_i , i^{th} cell width, $i = 1$ to 10.
- (c) P_i , probability of scattering in the i^{th} cell, $i = 1$ to 10.
- (d) $Z = Z_0$, a starting position for each neutron in the sample.

The card sequence:-

Master Card A contains h_i , P_i , $i = 1$ to 6.

Master Card B contains h_i , P_i , $i = 7$ to 10.

Master Card C contains H , constants for generating random numbers, and constants for evaluating $\log_e x$.

Detail Card D contains $Z = Z_0$, different for each of say 20,000 sample neutrons.

Summary Card E1 contains the number of neutrons absorbed in each of the 10 cells.

Summary Cards E2 to E11 contain the distribution of collisions across the i^{th} cell, $i = 1$ to 10.

The Master Cards supply the constant information to be placed in the drum store, all calculation is performed on card D as below, and the Summary Cards cause the transfer of information from the working section to the output locations of the drum:

- (a) Generate ⁽¹⁾ 3 random numbers ξ_1, ξ_2, ξ_3
- (b) $\mu = \cos \theta = 1 - 2\xi_1$, i.e. $\cos \theta$ is uniformly distributed between $-1 \leq \cos \theta \leq +1$.
- (c) $\eta = \left| \log_e \xi_2 \right|$ i.e. the distance travelled to the next collision is distributed as the negative exponential e^{-x} . The $\log_e \xi_2$ is calculated from the rational approximation given above. ⁽¹⁾
- (d) $Z' = Z + \mu\eta$ i.e. the horizontal distance to the next collision.
- (e) $Z'' = Z' \pmod{H}$ i.e. the positive fractional part of Z'/H which scales Z' so that it lies in block 0.
- (f) Repeatedly subtract $h_1, h_2, h_3, h_4 \dots h_i$ from Z'' until the sum is negative at the subtraction of h_i .

$$\text{i.e. } Z'' - \sum_{r=1}^i h_r < 0$$
- (g) Put $i = S$. S now indicates that the collision occurred in the S^{th} cell. In order to obtain a distribution of collisions, each cell is divided into 10 equal intervals.
- (h) Find the intergral part of $(Z'' - \sum_{r=1}^{s-1} h_r) \times \frac{10}{h_s} = j$

The position of collision in cell S is the j^{th} interval.

- (i) Score 1 in the appropriate counter for the S^{th} cell and the j^{th} interval. For 10 cells and 10 intervals per cell there are thus 100 counters.
- (j) Test:
 If $\xi_3 < P_S$, repeat the calculation from (a).
 If $\xi_3 \geq P_S$, the neutron has been absorbed and the tracking is complete.
- (k) Score 1 in one of ten counters, for the last collision in cell S, to obtain the number absorbed in each of the ten cells.

Since a sample neutron may have a random number of collisions before being absorbed, it is not possible to give an exact calculator speed, but, where the average number of collisions per neutron was ten, the rate was approximately 1,000 cards per hour.

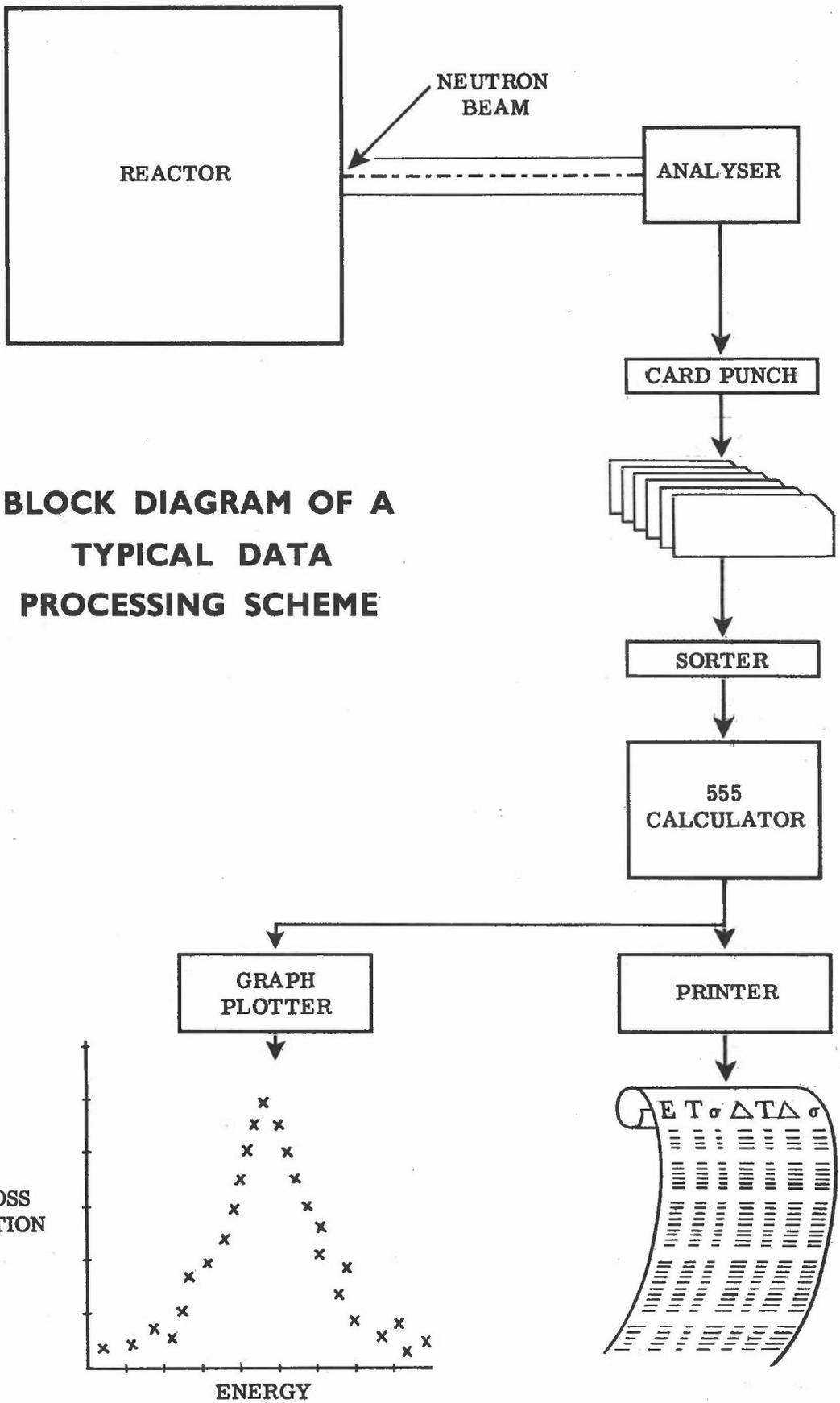
It should be noted that apart from the initial preparation of the starting values, no card handling is necessary. The results are punched in eleven cards which have only to be listed to provide the record of the run. No exact figures are available for the time required to prepare this program, but it was certainly less than 10 days from its conception.

The program takes 130 steps and it is of interest to note how much the program depends upon the "Repeat" and "Special Repeat" functions. Clearly, in the sequence of calculations above there is a good deal of searching through the drum store, not only to locate the h_S and P_S values, but also to find the correct counter to store the number of collisions.

In the space available it is not possible to give a detailed program for each problem treated, but for this calculation the complete program is given in Appendix B, together with an explanation of certain abbreviations of the function codes which have been found convenient.

References:

- (1) Symposium on Monte Carlo Methods. Wiley & Sons, New York, London.
- (2) Symposium on Monte Carlo Methods. Journal of Royal Statistical Society, Series B, Vol. XVI, No. 1, 1954.



3 Data processing

There is an increasing demand from physicists and chemists who require to summarise and transform results from large scale scientific experiments. The automatic output of data on to punched cards or tapes, which are then processed by a computer or calculator, saves great manual effort and also allows results to be inspected soon after an experiment has been completed. In many cases the calculation required is only of moderate size, but it may have to be repeated a great many times. It is therefore important to have a computer which can readily cater for a steady stream of such data.

In the design of nuclear reactors the nuclear properties of the materials to be used are of great importance; these properties are the fission, absorption and scattering cross sections of the fuel, the moderators and coolants, which in general vary also for the different energies of the free neutrons. These cross sections are simply the probability of any one of the three events occurring when a free neutron of a particular energy has a collision with a nucleus. There are several methods for determining these cross sections; the neutron velocity selector or "chopper", the neutron crystal spectrometer, the Van de Graaff electrostatic generator, the linear accelerator, and the reactor itself. In almost all methods the behaviour over wide ranges of energy must be observed, so that for each material examined a large amount of data may be obtained, usually in the form of counts of neutrons as indirect functions of energy. The experimental physicist using these machines cannot make use of this information as it is obtained, because of its volume and because it is not in the form he requires, i.e. cross sections as a function of energy.

The Type 555 is an efficient machine for this kind of work, and an illustration, dealing with cross section data from the linear accelerator, as carried out on the machine, is discussed. Although the calculation is small it will be seen that the amount of input and output data is considerable.

Information obtained from 1,000 sources, gives rise to 1,000 numbers n_i . Several runs of the experiment are made giving input data of, say r sets, each of 1,000 numbers $j n_i$, $i = 1$ to 1,000. $j = 1 \dots r$.

It is required to calculate:

$$(1) N_i = 1 n_i + 2 n_i + 3 n_i + \dots r n_i. \quad \text{Output } N_i.$$

$$(2) A_i = \sum_{S=0}^{i-1} N_S.$$

$$(3) \delta N_i = \frac{A_i \times N_i}{T \times C}. \quad \text{Where } T \text{ and } C \text{ are given constants.}$$

$$(4) N_i' = N_i + \alpha \delta N_i. \quad \text{Where } \alpha \text{ is a given constant.}$$

$$(5) B_i' = A_i + R K C_i. \quad \text{Where } A, R, K \text{ are given constants, and } C_i \text{ is given for each } i.$$

$$(6) N_i'' = N_i' - B_i'. \quad \text{Output } N_i''$$

$$(7) T_i = \frac{N_i''}{M S_i}. \quad \text{Output } T_i$$

M is a given constant. S_i is given for each i .

$$(8) \frac{1}{n} \left| \log_e T_i \right| = \sigma_i \quad \text{Output } \sigma_i. \quad \frac{1}{n} \text{ given constant.}$$

(This is the value of the cross section, which is what the physicist is searching for).

$$(9) \delta T_i = \sqrt{N_i' / N_i''}.$$

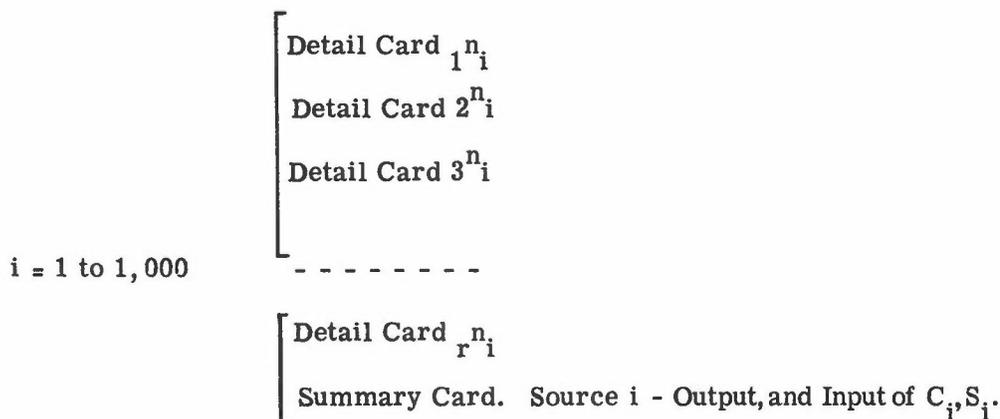
$$(10) T_i \pm \delta T_i. \quad \text{Output.}$$

The calculation in (8) of the $\left| \log_e T_i \right|$ makes use of the rational approximation discussed above in (1) and, in order to deal with a very wide

range of the numbers which can arise during the calculation, double length arithmetic has been used in steps (3), (5), (7) and (8).

The card sequence is as follows:

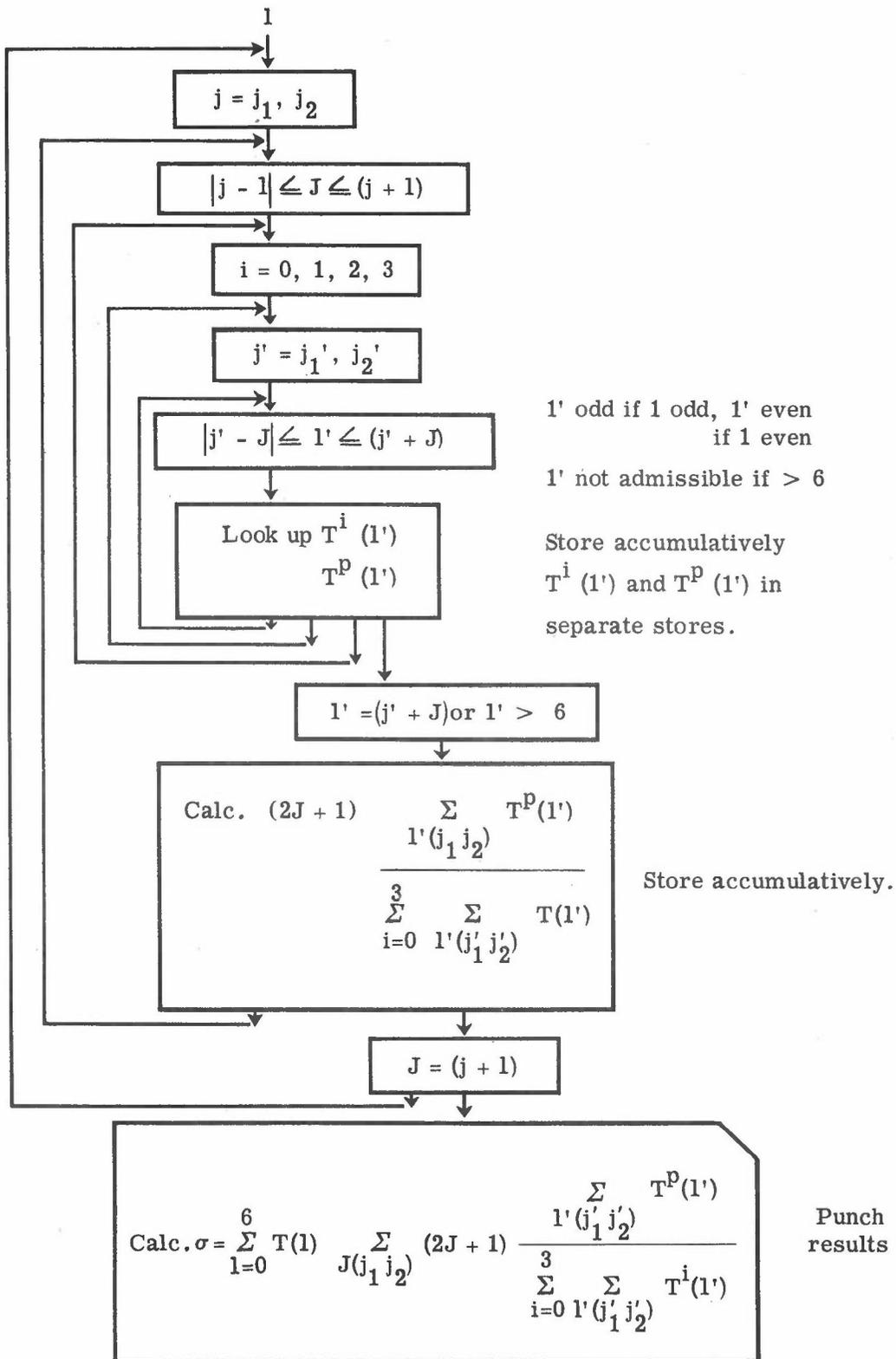
Master Card - containing constants T, C, R, K, etc., and constants for calculating $\log_e T_i$.



The time taken to calculate 4 sets of 1,000 such numbers followed by 1,000 summary cards is 50 minutes; i.e. the calculator is running at full speed. The program, and the job itself, were dealt with entirely by junior staff.

There are other similar processes at Harwell and the demand for schemes such as that described above is increasing as the potentialities of automatic recording and processing are realised.

BLOCK DIAGRAM OF COMBINATORIAL PROBLEM



l' odd if 1 odd, l' even if 1 even
 l' not admissible if > 6

Store accumulatively $T^i(l')$ and $T^P(l')$ in separate stores.

Store accumulatively.

Punch results

4 Nuclear structure theory —a combinatorial problem

The following calculation is again one of nuclear physics and is unusual because very little arithmetic is required. For each stage of the problem numerous decisions must be taken, following a set of given rules, to determine which of a group of numbers on the drum store is to be operated upon. The "Table Look Up" feature, one of the facilities of the Type 555, is at the heart of the problem, and the following notes explain its essential points and the program instructions.

A table is in channels 1 and 2 having 42 entries assumed to be at equal intervals of the argument, e.g. $F(E)$ for $.01 \leq E \leq .42$.

As the result of a calculation of E we have, say E_0 , and we wish to find $F(E_0)$. With the value E_0 in the product register P , the following routine is entered.

Program Step

- (1) Set $T.C.O.1 > 0$ by the transfer of a positive number and reset the channel selector.
- (2) From the product register (P) on path A.
Emit 21, on the path B (Complement if $T.C.O.1 > 0$).
To the Product register.
Set $T.C.O.1$. and move channel selector.
Repeat Program Step if $T.C.O.1 > 0$.
Word time, instantaneous.
- (3) From P on path A.
Emit 1 position 1 on path B. Complement.
To P .
Set $T.C.O.1$.
Special Repeat step if $T.C.O.1 \geq 0$.
Word time 19.
- (4) From Channel 1, path A, if Channel Selector = 2
2 path A, if Channel Selector = 3
To Multiplicand register.
Word time, instantaneous

N.B. There is an important difference between the "repeat" and "special repeat" functions. "Repeat" will repeat a program step, under the control of a T.C.O., either consistently at a specified word time, or, "instantaneously" to progress through successive word time in sequence. "Special repeat" must have a specific starting word time and will then use, in sequence, subsequent word times. Both functions are controlled by numbers passing through the adder and when a control condition no longer exists, that is $T.C.O.1 \leq 0$, in (2), the "repeat" automatically causes one more operation before passing on to the next program step. Program step (2) illustrates this, 21 being subtracted until the result in the product is zero or negative; the complement function is then cut off and the further operation restores the fractional part of 21 to the product register. The channel selector can then indicate which channel holds the number, the fractional part selects the appropriate word time by means of program step (3). Program step (4) then performs the required operation on the selected number

This routine can be used also for placing constants on the drum and also when convenient for selecting numbers for output.

Example:

We are given a function $T(l)$ of l and two values, $j = j_1$ and j_2 .

l	$T(l)$
0	-
1	-
2	-
3	-
4	-
5	-
6	-

together with a set of tables $T^i(l')$ of l' , where $i = 0, 1, 2, 3$ and for each i , j' has two values j_1', j_2' .

	i=0	i=1	i=2	i=3
l'	$T^0(l')$	$T^1(l')$	$T^2(l')$	$T^3(l')$
0	-	-	-	-
1	-	-	-	-
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-

For l and $l' > 6$, $T(l)$ and $T^i(l') = 0$. $i = 0, 1, 2, 3$.

It is required to calculate the following sum and to record the individual sums for $l = 0$ to 6 separately.

$$\sigma = \sum_{l=0}^6 T(l) \sum_{J(j_1, j_2)} (2J + 1) \left[\frac{\sum_{l'(j'_1, j'_2)} T^P(l')}{\sum_{i=0}^3 \sum_{l'(j'_1, j'_2)} T^i(l')} \right]$$

For a given l , J takes integral values given by

$$|j - l| \leq J \leq (j + l). \quad j \text{ takes the values } j_1, j_2 \text{ in turn}$$

For a J , l' is given by

$$|j' - J| \leq l' \leq (j' + J). \quad j' \text{ takes the values } j'_1, j'_2 \text{ in turn for each } i.$$

An l' is only admissible when it is of the same parity as the l from which it is derived, i.e. l' can only be even if l is even, or odd if l is odd. Otherwise it makes no contribution to the sum.

The suffix P takes a particular value of i , e.g. $P = 2$, and this is specified at the start.

To illustrate the layout of the program the block diagram shows more clearly the operations involved, in particular the number of loops which are necessary.

5 Heat flow equations

Many of the problems of mathematical physics involve the solution of differential equations. Analytical difficulties are however often encountered, and it is frequently necessary to obtain numerical solutions for which various numerical methods have been devised. An approximate method is to replace the differential equation by a finite difference equation which for many cases leads to an adequate approximate solution. Such a situation arises in heat flow equations, and the following account is concerned with the solution of the finite difference equations which approximate to the differential equation.

In this type of problem, an initial state j is given for a series of positions i , $i = 1, 2 \dots n$, and it is required to compute values for each position at state $j + 1$. where the relationship between the j^{th} and $(j + 1)^{\text{th}}$ state is given by a finite difference equation of the form.

$$A_i^{j+1} = \alpha A_i^j + \beta A_{i-1}^j \quad \alpha, \beta \text{ given,}$$

together with boundary conditions which deal with the cases $i = 0$ and $i = n$.

The example deals with the following equation and boundary conditions:

$$V_i^{j+1} = a V_i^j + b(V_{i-1}^j + V_{i+1}^j). \quad a, b, \text{ given.}$$

$i = 1, 2 \dots n.$

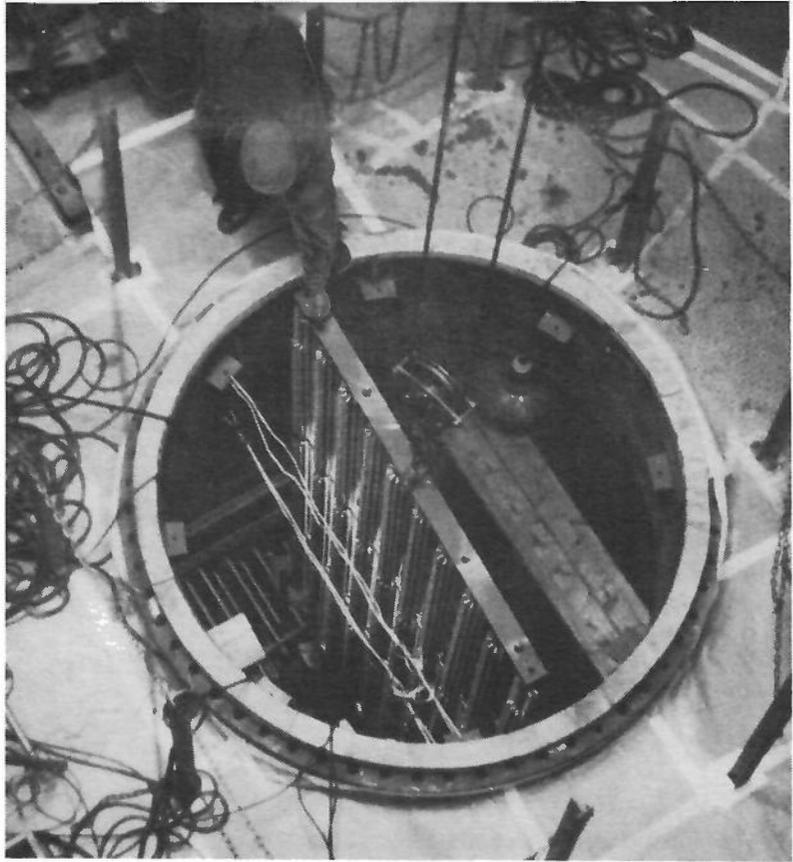
Boundary Conditions:

$$V_{n+1}^j = \alpha V_{n-1}^j + \beta V_n^j. \quad \alpha, \beta \text{ given.}$$

$$V_0^j = k_j. \quad k_j \text{ given.}$$

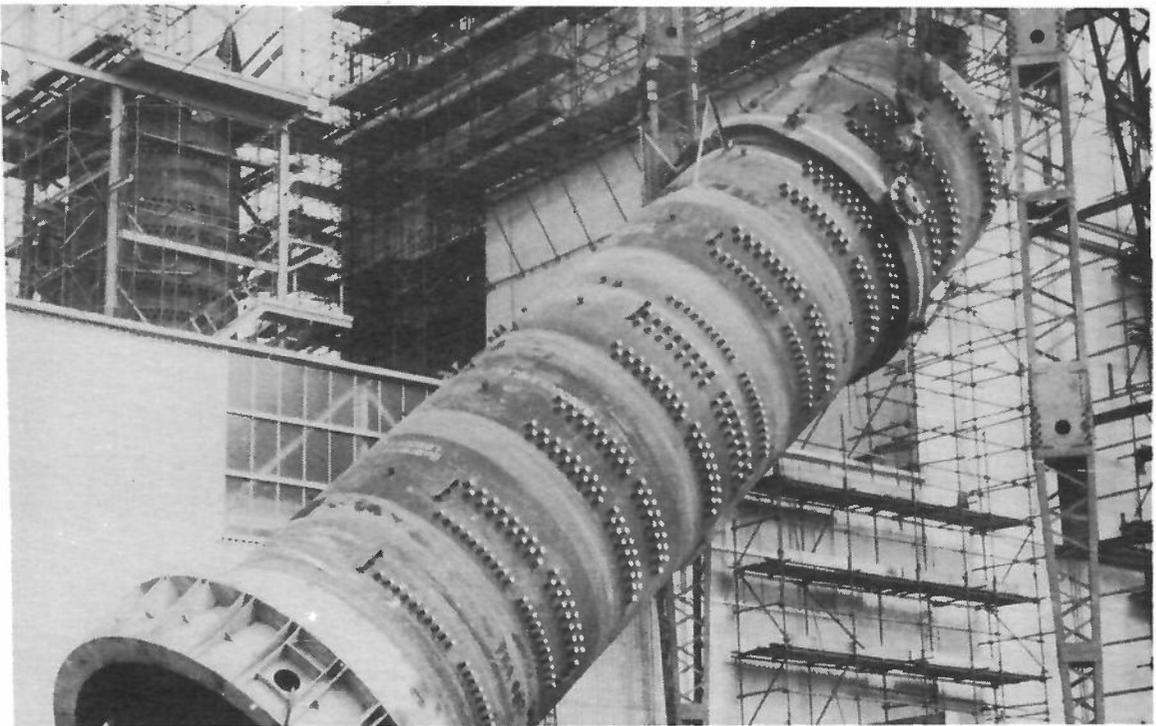
Starting Values:

$$V_i^0 = a_i. \quad \begin{matrix} i = 1, 2 \dots n \\ j = 0 \end{matrix}$$



Calder Hall Heat Exchanger. Looking down into the Heat Exchanger shell, where banks of boiler tubes are being placed in position.

One of Calder Hall's sixteen Heat Exchanger Shells being lifted into position on its plinth. Each shell is approximately 80 feet high.



A program has been written for the Type 555 which deals with $n \leq 10$; 63 steps are required in the calculation of all 10 positions of the j^{th} state. Because one card cannot contain all values for the j^{th} state, a second card to be punched is inserted. The card sequence is simple, the constants and starting values are read in from two master cards and all subsequent values are punched on the blank cards. The speed is such that 1,000 values of V_i^j ($i = 1$ to 10, $j = 1$ to 100) can be obtained in approximately 3 minutes, further illustrating the advantage of the Type 555 for the output of a large amount of data. The programming time required only a few hours, the possible value of n having been decided previously. In common with the Monte Carlo and Combinatorial problems, the ability to select numbers from the drum store according to given rules is vital; without it the solution of many problems would not be possible on the Type 555 and it is this ability which brings the machine into the borderline class between electronic calculators and general purpose digital computers.



Plugging up a Control Panel for one of the jobs described in this booklet.

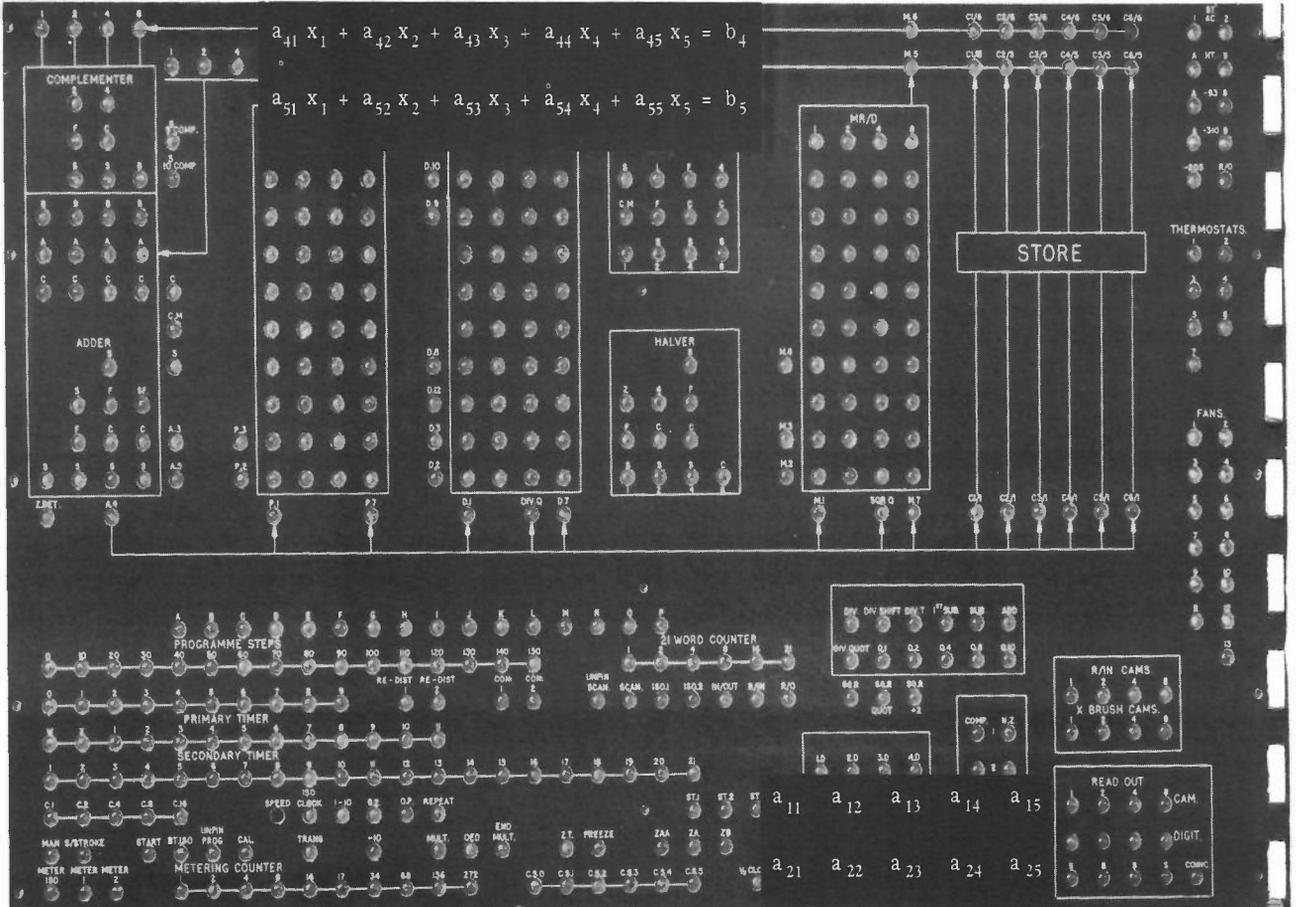
$$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + a_{14} x_4 + a_{15} x_5 = b_1$$

$$a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + a_{24} x_4 + a_{25} x_5 = b_2$$

$$a_{31} x_1 + a_{32} x_2 + a_{33} x_3 + a_{34} x_4 + a_{35} x_5 = b_3$$

$$a_{41} x_1 + a_{42} x_2 + a_{43} x_3 + a_{44} x_4 + a_{45} x_5 = b_4$$

$$a_{51} x_1 + a_{52} x_2 + a_{53} x_3 + a_{54} x_4 + a_{55} x_5 = b_5$$



a 31 a 32 a 33 a 34 a 35

a 41 a 42 a 43 a 44 a 45

a 51 a 52 a 53 a 54 a 55

6

Evaluation of determinants and the solution of simultaneous equations

Since the Type 555 is restricted to 150 program steps, and a relatively small numerical store for this kind of work, matrix operations would not, at first sight, appear to be convenient, but it is of interest to consider what can be done in this field.

Another inherent difficulty when the equations are ill-conditioned, is that of scaling; the Type 555 is not easily programmed to deal with floating point arithmetic, because the shifting operations are not simple. However, this can be overcome to some extent by careful attention to the basic logic of the machine and by ingenuity in programming.

For operating on large size matrices a large fast general-purpose digital computer is essential, but, for matrices of a small order r say, $3 \leq r \leq 12$, the Type 555 can be used to solve a set of equations and evaluate the determinant; the inversion of a matrix and the evaluation of its latent roots seem to be unsuitable because of the difficulty of storing all the elements of the matrix, together with the subsidiary matrices which are needed.

From the many methods for the solution of simultaneous linear equations, a choice has been made of the Choleski method. So far only symmetric matrices have been considered, and the following discussion is exclusively concerned with them. The Choleski⁽¹⁾ method is based on the fundamental theorem that

$A = LL'$ A is a square symmetric matrix.

L is a lower triangular matrix.

L' is the transpose of L .

Writing a matrix equation

$Ax = b$ (where x is an unknown vector and y a known vector)

we have

$$LL'x = b.$$

Put

$$L'x = y,$$

then

$$Ly = b,$$

$L'x = y$, and by back substitution x may be found.

There are two reasons for considering this method to be suitable for the Type 555. First, it is not necessary to keep all elements of L' in the drum store, so that matrices of order ≤ 12 may be considered; second, the method of solution allows one element of L' to be computed for each input of an element of A , thus allowing the cards to be used as a temporary store of L' .

A program has been written for the Type 555 based on this, and experience will show how successful it is in dealing with ill-conditioned sets.

The general scheme adopted is as follows:

- (a) The upper triangle of the symmetric matrix A is fed in, in row order, together with the vector b , with one element a_{ij} per card.
- (b) L'_{ij} is punched on card of element a_{ij} , the vector b being treated as an extra column of A .
- (c) If the determinant only is required a special final card is inserted on which is punched $D^{\frac{1}{2}}$,
- (d) To obtain the solution the cards are put through the machine a second time in reverse order to complete the back substitution.
- (e) Special cards are punched containing the vector x .

Reference:

- (1) Practical Methods for the solution of linear equations and the inversion of matrices. L. Fox. Journal of Royal Statistical Society, Series B. Vol. XII, No. 1. 1950.

General comments

Although the Hollerith Type 555 Electronic Calculator cannot be considered in the same class as a stored program computer it is nevertheless a powerful machine for many problems. The facilities of a computer are available; although a program has to be written within 150 steps, these steps are however very powerful, and in some cases can be set up to operate as sub-routines.

The basic circuitry of the Type 555 is similar to that which has been successfully employed in the Hollerith Type 542 and 550 Calculators. Many hundreds of these machines are in operation throughout the United Kingdom and overseas; the "555" has the same proven reliability as these machines, from which it has been developed. An incidental point is that no special installation facilities are needed apart from some means of dispersing the heat generated by the 8KW. which is the normal running load.

The problems described form only a selection from those already programmed, and as time goes by it is confidently expected that the Type 555 Electronic Calculator will be employed on further new projects.

Legend

P	:	Product
M/D	:	Multiplicand
M/R	:	Multiplier
INST.	:	Instantaneous
COMP.	:	Complement
N.R.G.	:	Non Ring
W. (1)	:	Word time 1
C.S.	:	Channel Selector
ZA_{x}^{y}	:	Zeroise digits x to y on Highway A
ZB_{u}^{v}	:	Zeroise digits u to v on Highway B
LSR. 1.	:	Long Skip Start Number 1
LSP. 1.	:	Long Skip Stop Number 1
LSR. 2.	:	Long Skip Start Number 2
LSP. 2.	:	Long Skip Stop Number 2
SSR	:	Short Skip Start
SSP	:	Short Skip Stop
E. (x). P. (y).	:	Emit number x in position y on to Highway B.
(Stop) Skip if "P"	:	(Stop) or Start a skip if present card is designated "P".
(Stop) Skip unless "P"	:	(Stop) or Start a Skip if present card is not designated "P".
Set TC0. (a)	:	Sets unit (a) to take up the state of the number which last passed through the adder on which Set TC0 (a) was called. States of (a), ≥ 0 , > 0 , $= 0$, < 0 .
as (16)	:	Programme step is same as step (16). Instruction wires for step (16) are used for all subsequent steps, e.g. 19, 22, which do not have to be separately instructed.

**TYPE 555
PROGRAM SHEET**

X IN THE MULTIPLICAND X - 7 DECIMALS

LOG_eX

PROG. STEP	FROM		TO	MPY. DIV. S.Q.	COL. SHIFT	ZEROISE	WORD	CYC	REMARKS
	A	B							
1	4/		P				1		Count (d) for p of 10 ^P
2	M/D	Comp M/D					Set TC02	Inst	Set TC02 = 0
3		M/D					Set TC03	Inst	Test X = 0
4	M/D			M/D ⁹ P ⁹	ZA ₁ ⁷		SSR (TC03 = 0) Set TC02 Repeat (TC02 = 0) LSP1 (TC01 < 0)	Inst	Shift and count until X ≥ .1
5	M/D	Comp E5 P ⁹					Set TC01	Inst	Test for X' > .5
6	M/R	E1 P1	M/R				X 2(TC01 < 0)	Inst	Add 1 to count for S of 2 ^S Double X'(X' < .5)
7	M/R	Comp E1 P1	M/R				LSR1 (TC01 < 0)	Inst	Loop
8		M/R	3/					15	Store S of 2 ^S
9	P		3/		ZP ZA ₂ ^{end}			19	Store P of 10 ^P
10		4/	M/R					2	A ₂
11				M/D ⁴	ZM/D			Inst	Shift x to 5 dec.
12	M/D		3/					4	Store x
13				Mult					
14					P5	ZP			
15	4/	P	M/D			ZP		3	A ₁ + A ₂ ^x
16	4/	3/	M/R					4	A ₃ + x
17				Div			Div Shift 5. 10 cycles		
18		Comp 3/	M/R			ZP		4	- x
19	M/R	E1 P6	M/R					Inst	1 - x
20				Mult					
21					P4	ZP			
22	4/		M/D					5	log _e 2
23		3/	M/R					15	S of 2 ^S
24				Mult					
25	4/		M/D					6	log _e 10
26		3/	M/R					19	P of 10 ^P
27				Mult			SSP (TC03 = 0)		
28	4/		P				SSR (TC03 > 0) SSP	7	If X = 0 put log _e X = log _e 5 x 10 ⁶

NOTE: Program Step 4 - X' (i.e. X shifted such that X ≥ .1) is left in positions 3-9

**TYPE 555
PROGRAM SHEET**

MONTE CARLO EXAMPLE

PROG STEP	FROM		TO	MPY. DIV. S.Q.	COL. SHIFT	ZEROISE		WORD	CYC.	ASS. CYC.
	A	B								
1		E9 P1	M/R 5/				LSP1.Skip unless 'C'	12		
2		E1 P1					Set TC01	Inst.		
3	M/R	E1 P1 Comp	M/R				Set TC01 Sp. Repeat (TC01 ≥ 0)	20		
4	1/		M/D					Inst		
5		E7 P1	M/R				Set TC01.	Inst		
6	M/R	E1 P1 Comp	M/R				Set TC01. Repeat (TC01 ≥ 0)	Inst		
7	M/D		1/					Inst		
8	5/	E1 P1 Comp	M/R 5/				Set TC03	12		
9						M/R	LSR1 (TC03 ≥ 0)	Inst		
10		E9 P1 Comp	M/R					Inst		
11	M/R	E1 P1	4/ M/R				Set TC01. Sp. Repeat (TC01 < 0) Stop skip if 'A' Stop skip if 'B'	11		
12							As 2			
13	5/		1/				Set TC01. Sp. Repeat (TC01 ≥ 0) W.1. unless des B) W.7 if des B) LSP2 (TC05 < 0) Stop skip if 'D'			
14	1/		M/R				Skip unless 'D'	1		
15	1/		M/D			P		2		
16				Mult NRG						
17	P		4/ M/D			P		10		
18	1/		M/R					1		
19							As 16			
20	P		5/ M/D			P		7		
21							As 18			
22							As 16			
23	P		1/					2		
24					P6	P				
25		P Comp	M/D					Inst		
26				X2 NRG						
27	M/D	E1 P5	4/					9		
28	5/		M/D					7		
29					M/D3	M/D				
30	1/		P M/R					3		
31	M/D	M/D Comp					Set TC01. Set TC02.	Inst		
32	M/D						Set TC03.	Inst		
33		M/D			P9 M/D9	Z ₁ ⁷ B	SSR (TC03 = 0) Set TC02 Repeat (TC02 = 0)	Inst		
34					M/D4	M/D	Set TC02. Reset C.S.			
35		M/D		X2	M/R9	Z ₁ ⁶ B	Set TC02. Move C.S. Repeat (TC02 = 0)			
36			5/				C.S. = 4 E1 P1 C.S. = 5 E2 P1 C.S. = 0 E3 P1	1		
37	M/D		M/R					Inst		
38		P	5/	- 2		Z ₂ ^{end} B P		3		
39	1/		M/D	- 2				4		
40				- 2				Inst		
41	M/R		5/					6		
42				Mult						
43					P.5	P				
44	1/	P	M/D			P		5		
45	1/	5/	M/R					6		
46				Div.			Div. Shift 5 10 cycles			
47		5/ Comp	M/R			P		6		
48	M/R	E1 P6	M/R					Inst		
49							As 42			
50					P.4	P				
51	1/		M/D					7		
52	5/		M/R					1		
53							As 42			
54	1/		M/D					8		
55	5/		M/R					3		
56				Mult			SSP			
57	1/		(TC03=0) P					9		
58	P		M/R			P		Inst		
59	4/		M/D				Set TC04	9		
60		M/D Comp (TC04 < 0)	M/D					Inst		
61					M/R1	M/R				
62							As 42			
63							As 50			
64	1/		M/D					11		
65	5/	P Comp (TC04 < 0)	P				Set TC02. Set TC05.	13		
66	P	M/D Comp (TC05 ≥ 0)	P				Set TC02 Repeat (TC02 ≥ 0) Repeat (TC05 ≥ 0) Repeat (TC02 < 0) Repeat (TC05 < 0)	Inst		
67				(TC05 ≥ 0) X2						
68	P	M/D Comp (TC05 < 0)	P				Set TC01	Inst		
69	P		5/					13		
70	P	1/ Comp	P			Z ₆ ^{end} B	Set TC01. Sp.Repeat (TC01 ≥ 0)	12		
71		E16	M/R				Set TC01	Inst		
72	M/R	E1 P1 Comp	M/R				Set TC01. Repeat (TC01 ≥ 0)	Inst		
73	P	1/	P			Z ₆ ^{end} B		Inst		
74							As 73			
75		E15	M/R				Set TC01	Inst		
76							As 72			

TYPE 555
PROGRAM SHEET

MONTE CARLO EXAMPLE

PROG STEP	FROM		TO	MPY. DIV. S.Q.	COL. SHIFT	ZEROISE	WORD	CYC	ASS. CYC
	A	B							
77	4/	E9 P1	M/D					Inst	
78	1/		M/R					Inst	
79	M/D		5/					8	
80		M/R	5/			Z ₁ ⁶ B		10	
81	M/R		M/R		P.9	Z ₆ ^{end} A M/D			
82				Div.			1 cycle		
83	5/	E1 P1 Comp	M/R					8	
84	M/D		5/			P		9	
85		E5 P1	M/D					Inst	
86				Mult					
87	5/		M/R					9	
88					M/R9				
89				÷ 2					
90		M/R				Z ₂ ^{end} B	Set TC05	Inst	
91					M/R1	M/R			
92	M/R	E1 P1	M/R					Inst	
93	P	M/R	M/R				Set TC01. Reset C.S.	Inst	
94	M/R	E21 Comp (TC01 ≥ 0)	M/R				Set TC01. Move C.S. Repeat (TC01 ≥ 0)	Inst	
95		E1 P1					Set TC01	Inst	
96	M/R	E1 P1 Comp	M/R				Set TC01. Sp. Repeat.(TC01 ≥ 0)	19	
97	2/ CS=2 3/ CS=3 4/ CS=4	E1 P1 TC05 > 0 E1 P6 TC05 = 0	2/ CS=2 3/ CS=3 4/ CS=4 P					Inst	
98	5/		M/D					10	
99	4/		M/R					10	
100					M/R5 M/D5	M/R			
101	M/R	M/D Comp					Set TC05	Inst	
102							LSR2 (TC05 > 0)	Inst	
103							As 83		
104							As 88		
105							As 89		
106							As 90		
107							As 95		
108							As 91		
109	M/R	E1 P1 Comp	M/R				Set TC01. Repeat (TC01 ≥ 0)	Inst	
110	5/	E1 P1 TC05 > 0 E1 P6 TC05 = 0	5/					Inst	
111							S.S.R.Stop Skip if 'E'	Inst	
112	5/		M/R			P		14	
113							As 85		
114							As 86		
115	P		5/			P	LSP2 (TC03 < 0)	21	
116	5/	E1 P1	M/R 5/				Set TC01. Reset C.S.	21	
117							As 94		
118							As 95		
119							As 96		
120	2/ CS=2 3/ CS=3 4/ CS=4		M/D					Inst	
121							As 95		
122	5/		M/R					2	
123	M/R	E1 P1 Comp	M/R				Set TC01. Sp.Repeat (TC01 ≥ 0)	4	
124	M/D		5/					Inst	
125	5/	E1 P1	5/					2	
126	P	E1 P1	P					Inst	
127	P	E5 P1					Set TC03	Inst	
128							LSR2 (TC03 < 0) S.S.R. Stop Skip if 'F'	Inst	
129	1/		M/D					2	
130	M/D		5/				Stop Skip unless 'C' Stop Skip unless 'D'	21	
131							S.S.R.	Inst	

TYPE 555 STORAGE CHART

WORD	CHANNEL 5	CHANNEL 4	CHANNEL 3	CHANNEL 2	CHANNEL 1
1	r/2 S of 2 ^S			h ₁ 0 1	10 figs P
2	K			2 3	P ₀ ¹⁰ figs
3	P of 10 ^P			4 5	0012345670 α
4		h ₁₀		6 7	.31774 A ₂
5			h ₆	8 9	1.06535 A ₁
6				h ₂	.38308 A ₃
7	ξ ₂				.69315 log _e 2
8	S				2.30259 log _e 10
9	r	μ4D			log _e 5 x 10 ⁸
10	P ₁	ξ ₁	h ₇		
11		-8		h ₃	H 1 +5D
12	transfer (count)	-7			1 + 4D P ₁ h ₁
13	j 4D	-6			P ₂ h ₂
14	R 0,1,...9	-5			P ₃ h ₃
15		-4	h ₈		P ₄ h ₄
16	No. absorbed h ₁ h ₂	-3		h ₄	P ₅ h ₅
17	h ₃ h ₄	-2			P ₆ h ₆
18	h ₅ h ₆	-1			P ₇ h ₇
19	h ₇ h ₈	0			P ₈ h ₈
20	h ₉ h ₁₀	1	h ₉		P ₉ h ₉
21	5R			h ₅	P ₁₀ h ₁₀

ACKNOWLEDGEMENTS

Photographs on pages 4 and 32 are published by kind permission of the United Kingdom Atomic Energy Authority.

The table published on page 14 is reprinted from Chambers Six-Figure Mathematical Tables.

The publisher and author gratefully acknowledge the assistance of personnel at A.E.R.E., Harwell in the compilation of this publication.

THE BRITISH TABULATING MACHINE CO. LTD

Head Office: 17 PARK LANE LONDON W.1 Telephone: HYDe Park 8155

DISTRICT SALES OFFICES

LONDON:

North London, Kent, Surrey, Chilterns City } Victoria House, Southampton Row, London, W.C.1. (*Holborn 7866*)
City Wall House, 84/90 Chiswell Street, E.C.1. (*Monarch 1596/7/8*)

West End } 33 Sackville Street, W.1. (*Regent 2021*)
BIRMINGHAM: } Wolverley House, 18 Digbeth, Birmingham, 5. (*Midland 1812*)

West Midlands } King Edward House, New Street, Birmingham, 2. (*Midland 1812*)

Wolverhampton and Stafford } 12 Waterloo Road, Wolverhampton. (23440 and 23449)

BRISTOL

Warriner's Building, Warriner's Wharf, Bathurst Basin, Bristol, 1. (26395/6)

CARDIFF

40 Windsor Place, Cardiff. (27260)

COVENTRY

28-34 Whitefriars Street, Coventry. (40261/2)

EDINBURGH

6a Rutland Square, Edinburgh, 1. (*Fountain-bridge 6201/2*)

GLASGOW

13/17 Renfield Street, Glasgow, C.2. (*City 6626*)

HITCHIN:

South Midlands and East Anglia } 2 Walsworth Road, Hitchin, Herts. (*Hitchin 2241/2*)

LEEDS

Provincial House, 26 Albion Street, Leeds, 1. (32381)

LIVERPOOL

164 Sefton House, Exchange Street, Liverpool, 2. (*Central 5561*)

MANCHESTER:

North, South } Century Insurance Building, St. Peter's Square, Manchester. (*Central 3524/5*)

NEWCASTLE UPON TYNE

Neville Hall, Westgate Road, Newcastle upon Tyne, 1. (29786/7)

NOTTINGHAM

Westminster Buildings, Theatre Square, Nottingham. (46421/2)

SHEFFIELD

54-56 Fargate, Sheffield, 1. (27186/7/8/9)

SOUTHAMPTON

7 Cumberland Place, Southampton. (24659)

GOVERNMENT SECTION

Victoria House, Southampton Row, London, W.C.1. (*Holborn 7866*)

AGENTS FOR NORTHERN IRELAND

Calculating and Statistical Service: 7 Bedford Street, Belfast. (24635)

HOLLERITH COMPUTER CENTRE

36 Hertford Street, London, W.1. (*Grosvenor 1765*)

EDUCATIONAL AND TRAINING CENTRES

Moor Hall, Cookham, Berks. (*Bourne End 810/1*)

Bradenham Manor, High Wycombe, Bucks. (*Nap Hill 2191/2/3*)

FIELD ENGINEERING HEADQUARTERS

Oxen Road, Luton, Bedfordshire. (*Luton 6530/1*)

SPECIAL SERVICES HOLLERITH DIVIDEND SERVICE BUREAU: Victoria House, Southampton Row, W.C.1. (*Holborn 7866*). LECTURES AND FILMS: 70 North Row, W.1. (*Hyde Park 1881*). SERVICE BUREAUX: Aberdeen, Birmingham, Glasgow, Leeds, Liverpool, London, Manchester, Molesey (*Surrey*), Newcastle upon Tyne, Nottingham and Southampton.

WORKS

Alderley Edge, Cheshire. Castlereagh, near Belfast. Letchworth Garden City, Hertfordshire; Stevenage, Hertfordshire. Stockport, Cheshire.

RESEARCH AND DESIGN

Letchworth Garden City, Hertfordshire. Stevenage, Hertfordshire.

OVERSEA BRANCHES

Alexandria, Auckland, Cairo, Colombo, Karachi, Port of Spain, Rangoon and Wellington

ALSO SERVICE REPRESENTATIVES IN

Abadan, Accra, Chittagong, Hong Kong, Kuwait, Lahore, Masjid-i-Sulaiman (Iran), Mauritius, Rawalpindi, Singapore and Valletta

OVERSEA SERVICE BUREAUX

Alexandria, Cairo, Karachi

OVERSEA WORKS

Alexandria, Egypt

OVERSEA COMPANIES

INDIA: **Hollerith (India) (Private) Ltd.**, Bombay BRANCHES at Bangalore, Calcutta, Lucknow, Madras and New Delhi

SERVICE REPRESENTATIVES in Ajmer, Asansol, Baroda, Chittaranjan, Digboi (Assam), Giridih (Bihar), Jamshedpur, Jealgora (Bihar), Jubbulpore, Kanpur, Meerut, Poona, Satara, Secunderabad and Sindri (Bihar)

SERVICE BUREAUX in Bombay, Calcutta, Madras and New Delhi

WORKS: Bombay

SOUTH AFRICA: **Hollerith Machines (South Africa) (Proprietary) Ltd.**, Johannesburg

BRANCHES at Cape Town, Durban, Johannesburg, Nairobi, Port Elizabeth and Pretoria

SERVICE REPRESENTATIVES in Bloemfontein, Dar-es-Salaam, East London, Kimberley, Vereeniging and Windhoek

SERVICE BUREAUX in Cape Town, Durban, Johannesburg, Nairobi, Port Elizabeth and Pretoria

WORKS: Johannesburg

CENTRAL AFRICA: **Hollerith (Central Africa) (Private) Limited**, Salisbury

SERVICE REPRESENTATIVES: in Bancroft, Bulawayo, Chingola, Kitwe, Luanshya, Mufulira, Ndola and Que Que

SERVICE BUREAU in Salisbury

AUSTRALIA: **Hollerith (Australia) Pty. Ltd.**, Sydney

BRANCHES at Adelaide, Brisbane, Canberra, Melbourne, Perth and Sydney

SERVICE BUREAUX in Adelaide, Brisbane, Melbourne, Perth and Sydney

WORKS: Melbourne

MEXICO: **British Tabulating Machines (Mexico) S.A. de C.V.**, Edificio Jena 604, Lieja, 8, Mexico, 6, D.F.

OVERSEA AGENT

REPUBLIC OF IRELAND: **Calculating and Statistical Service**, Hollerith House, 15 Harcourt Street, Dublin (55411/2/3)