1st ed U2780
3rd ed (partial) O0025

FERRANTI MARK I
PROGRAMMING MANUAL

2nd. Edition
edited R.A. BROOKER, OCT. 1952

Xeroxed July '76 by S.H. LAVINGTON
MANCHESTER UNIVERSITY

PROGRAMMERS' HANDBOOK
(2nd Edition)

FOR THE

MANCHESTER ELECTRONIC COMPUTER

MARK II.

Table of Contents.

Preface.

This handbook is written mainly for those who expect to prepare problems for the MK II machine. Although almost complete in itself it is intended to be supplemented by some closer contact with the machine and by instruction from persons already familar with its use.

Much material has beeh taken over unaltered, or only slightly modified from the 1st Edition which was written by Dr. A.M.Turing. In addition some of the results of the first year's experience of programming for the MK II have been incorporated in the later chapters.

Miss C.M.Popplewell and N.E.Hoskin of the staff of this Laboratory, and A.E.Glennie of the Armament Research Establishment, Fort Halstead, Sevenoaks, Kent, were responsible for chapters 3, 7, and 6 respectively. Some of the examples of chapter 2 were originally designed as exercises for the Summer School in Programme Design held at the University Mathematical Laboratory, Cambridge (England), in September 1951.

At all stages in the preparation of this edition Dr. A.M. Turing offered valuable criticism and advice.

R. A. Brooker,
August 1952.

OCTOBER 1952.

Alterations and Additions to the Second Edition of the Programmer's Handbook.

It is our aim that the 2nd Edition of the Handbook should contain the latest information on any subject. In order to achieve this it will be necessary from time to time to make certain alterations and additions. These will take the form of loose leaves which will be issued to persons already possessing copies of the Handbook. The loose leaves will bear page numbers and where they replace existing pages the new page will also carry a serial number in the top right hand corner. Thus, (1) denotes the first reissue of a page. When such replacements are received the original page should be destroyed.

It is further proposed to make the Handbook self-contained by issuing lists of library routines and the names of their authors as appendices to appropriate chapters. Thus e.g., a list of routines available for calculating the elementary functions will ultimately be issued as an appendix to chapter 6; a list on print routines will appear as an appendix to chapter 4. These lists will contain the necessary and sufficient information about each routine to enable it to be used.

R. A. BROOKER,
Computing Machine Laboratory,
The University, Manchester, 13.

Programmer's Handbook

for

Manchester Electronic Computer Mark II.
(Second Edition)

Errata 1.

Chapter 1.	page	1.1.	line	18	For "formidable complicated" read "forminably complicated".
	"	1.4.	"	9	For "in a block" read"in a book"
	"	1.14	"	7	For "M = O" read "M' = O".
	"	1.16	"	14	For "AY££££££" read "RY££££££".
	"	1.18	"	6	For "@CAK" read "@C:K"
	"	1.19	"	18	For "[B7]' = [B7]" read "[HU]' = [B7]"
			"	22	For "B1 = HUIO" read "B1 = HUPO"
			"	26	For "[@E]' = @E/O" read "[B7]' = @E/O"
	"	1.22	"	28	For "(The named tube must....) read "(assuming that the named tube is....)
	4	1.24	"	15	For "(see page 21)" read "(see page 1.21).
Chapter 2.	page	2.4.			Substitute new page
	"	2.5	line	7	For "|b| < 2^{-6} and o < ac < 2^{-14}" read "|b| < 2^{-7} and o < ac < 2^{-15}"
			"	20	For "|J|IST/" read "|J|J:T/".
	"	2.6	"		Substitute new page
	"	2.7			" " " " "
	"	2.8	"	43	For "|C|E:/P" read "|C|E:/H"
	"	2.9	"	20	For "self-resulting" read "self-resetting".
	"	2.15	"	1	For "six warning characters" read "eight warning characters"
			"	4	For "seven control transfers" read "nine control transfers".
Chapter 3.	page	3.3	line	28	For "$e^{\frac{1}{2}}$ cos 2πx" read "$e^{\frac{1}{2}}$cos 2πx"
	"	3.8	"	19	For "onvolved" read involved"
Chapter 4.	page	4.14	line	8	For "(see p.)" read "(see p 1.6)
	pps.	4.19 - 4.39			Substitute new pages.
	page	4.22	"	29	For "S6 and S7" read "S4 and S5".
	"	4.28	"	5	For "PIUF" read TIUF"
			"	51	For "TYP" read "YTP"
	"	4.32	"	1	For "Q sequence" read "G sequence" N.B. The cue to B.DEC.INPUT is omitted from the account given on p.4.32. It is //....@/.
Chapter 9.	page	9.5	line	7	For "author's" read "authors".
Appendix.	page	A.4	line	17	For "|/K| D' = S_+" read "|/K|D' = $S_\pm$"
	"	A.5	"	4	For "5 best" read "5 beat"

Chapter 1.

The Logical Design of the Machine and The Instruction Code.

1. General remarks on electronic computers.

Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner. The electronic computer should however obtain its results very much more quickly. The human computer with whom we are comparing it may be imagined as supplied with various computing aids. He should have a desk machine, paper to write his results on, and more paper on which is written a detailed account of how the calculation is to be carried out. These aids have their analogues in the electronic computer. The desk machine is transformed into the *arithmetical unit*, and the paper becomes the "information store" or more briefly the *store*, whether it is paper used for results or paper carrying instructions. There is also a part of the machine called the *control* which corresponds to the computer himself. If his possible behaviour were very accurately represented this would have to be a formidably complicated circuit. However we really only required him to be able to obey the written instructions and those can be made so explicit that the control can be quite simple. There remain two more components of the electronic computer. These are the *input* and *output* mechanisms, by which information is to be transferred from the outside world into the store and conversely. If the analogy of the human computer is to be maintained these parts would correspond to his ears and voice, by means of which he communicates with his employer.

2. Scales of notation.

The information stored on paper by the human computer will mostly consist of sequences of digits drawn from 0, 1, ..., 9. There may also be other symbols such as decimal points, spaces etc., and there may be occasional remarks in English, Greek letters etc. There may in fact be anything from 10 to 100 different symbols used, and there is no particular need to decide in advance how many different symbols will be concerned. With an electronic computer however such a decision has to be made; the number of symbols chosen is ruled very largely by engineering considerations, and with the vast majority of machines the number is two. The reason for this is that the "scale of 2" matches naturally with the "on-off" property of electronic switching circuits. Machines (e.g., ENIAC) have however been made with 10 different symbols. The number for the MK II machine is two, and the symbols used are 0 and 1.

It is not difficult to see that information expressed with decimal digits can be translated into information expressed with 0's and 1's by some suitable conventions, e.g., we could replace 0 by 0000, 1 by 1000, 2 by 0100, 3 by 1100, 4 by 0010, 5 by 1010, 6 by 0110, 7 by 1110, 8 by 0001, and 9 by 1001.

The scheme that is most economical in 0's and 1's however, and the one adopted for the MK II, is to use the pure binary representation of a number in which the digits, 0 or 1, are the coefficients of successive powers of 2. The most natural convention to choose is that by which the value of a 1 in the rth position from the right hand end is 2^{r-1}, so that e.g., 11001 means 25. This is the closest possible analogue of the ordinary Arabic notation for integers. The convention chosen for use with the Mark II machine is different. The value of a 1 in the rth position from the left hand is 2^{r-1}, so that 25 is represented by 10011 instead of 11001. These facts may be described by saying that the machine uses 'the scale of two with the most significant digits at the right hand end'.

Although the scale of two is appropriate for use within an electronic computer it is not so suitable for work on paper, and it is not possible to avoid paper work altogether. Without attempting to to explain the reasons at this stage let us accept that there are occasions when it is desirable to write down on the paper the sequence of symbols stored in some part of the machine. Suppose for instance that the sequence was

10001110111010001001100011100101010101101100100110

The copying of such sequences is slow and very liable to inaccuracy. It is very difficult to 'keep one's place'. It is therefore advisable to represent such a sequence on paper in a different form, not subject to these difficulties. The method chosen is to divide the sequence into blocks of five

10001 11011 10100 01001 10001 11001 01010 10110 11001 00110

and then to replace each block by a single symbol, according to the table below. The above sequence then becomes Z"SLZWRFWN.

0	00000	/	11	11010	J	22	01101	P
1	10000	E	12	00110	N	23	11101	Q
2	01000	@	13	10110	F	24	00011	O
3	11000	A	14	01110	C	25	10011	B
4	00100	:	15	11110	K	26	01011	G
5	10100	S	16	00001	T	27	11011	"
6	01100	I	17	10001	Z	28	00111	M
7	11100	U	18	01001	L	29	10111	X
8	00010	½	19	11001	W	30	01111	V
9	10010	D	20	00101	H	31	11111	£
10	01010	R	21	10101	Y			

These symbols are essentially the teleprinter code, except that the combinations 00000, 01000, 00100, 00010, 11011, 11111 which in true teleprint are represented by

no effect, line feed, space, carriage return, figure shift, letter shift

respectively have here been given the representations /, @, :, ½, ". £ These symbols have been chosen so as to enable the upper case of the typewriter to be used throughout.

The user is strongly recommended to learn the above table. In principle it is possible to do without these aids for the machine itself can do all the conversion processes required. In practice it frequently happens that some single number is required in the scale of 32, and it is found less trouble to do the conversion by hand than to use the machine. To convert a decimal fraction to the scale of 32 multiply repeatedly by 32, subtracting and recording the integral part at each stage. This can be done very quickly with a Brunsviga with transfer. The integral part obtained may then be looked up in the above table. See also appendix.

3. The forms of storage used.

The information store in the MK II machine consists of the magnetic store and the electronic store. The information in the magnetic store is of considerable volume viz. 655360 binary digits: in other words it corresponds to paper on which is written 655360 digits each of which might be either 0 or 1. But this information is not particularly readily available. It is (to maintain the analogy) as if it were written in a book. In order to find any required piece of information it is necessary to open the book at the appropriate page. The electronic store has a considerably smaller capacity viz. 10240 digits but this information is much more readily available and is to be compared rather to a number of sheets of paper exposed to the light on a table, so that any particular word or symbol becomes visible as soon as the eye focusses on it.

The information in the magnetic store consists of magnetised areas of nickel on the cylindrical surface of a rotating wheel. Each digit stored is represented by one magnetised area. These 655360 areas are arranged in 256 tracks of 2560 digits each. The centres of the areas forming one track lie in a plane perpendicular to the axis of the wheel (and therefore on a circle). The 256 planes are equidistant; the 2560 digit areas on one circle are not however equidistant. The information in one track is further subdivided into two equal parts, which may be described as the left page and the right page if we continue to follow the simile of the book.

The electronic store.

The information in the electronic store consists of 8 tubes of 1280 binary digits each. A tube thus contains the same amount of information as a half-track or page of the magnetic store. It may als be described as an electronic page. A tube of information is divided into 64 lines of 20 digits each. These digits are stored as charges on the inner surface of the front of a cathode ray tube, the digits of one line forming a straight horizontal segment. (A line is the basic unit of information processed by the machine: lines represent both the numbers and the instructions which tell the machine what to do with the numbers).

The 64 lines of a page are to be regarded as arranged in 2 columns of 32 thus

0	32
1	33
2	34
3	35
.	.
.	.
.	.
.	.
31	63

This is in fact how they are displayed on the monitors of the machine (see photo).

The lines may be numbered consecutively through the 8 tubes. They could be numbered 0, 1, ..., 511. However, for reasons which the reader is asked to accept at the moment, the labelling obtained by transforming to the scale of 32, i.e., to the teleprinter code, is used. Thus the lines are known as //, E/, @/, ..., £K. In their geometrical arrangement on the tubes they are as below

Tube 0		Tube 1			Tube 7	
//	/E	/@	/A		/C	/K
E/	EE	E@	EA		EC	EK
@E	@E	@@	@A	. . . .	@C	@K
.	.	.	.		.	.
.	.	.	.		.	.
.	.	.	.		.	.
.	.	.	.		.	.
£/	£E	£@	£A		£C	£K

Each tube is formed of two columns and the second character describing a line gives the column in which it is to be found.

Associated with each line is a line pair or long line consisting of that line together with the next line. e.g., The lines R/ and J/ form a pair and so do BK and GK. These long lines will be referred to by R/ and BK, i.e., the name of the first line of the pair. There is one exception to this rule: the line pair associated with the last line

Information represented by monitor display.

```
     (65th line)
|/| / / / / | / | / / / £ |E|
    £ E / : | E | / / / /
    W E / : | @ | / / £ /
    H E / : | A | / / / /
    H E / : | : | / £ / /
    Q E W O | S | / / / /
    O E L O | I | £ / / /
    / @ Z / | U | / / / /
    / E L R | ½ | / / / /
    ½ E T A | D | / / / /
    J E @ C | R | / / / /
    U E / N | J | E / / /
    /./ / I | N | / / / /
    Y E / : | F | / E / /
    B E L " | C | / / / /
    O E / T | K | / / E /
    P E / : | T | / / / /
    G E W " | Z | / / / E
    " E / O | L | / / / /
    W E / : | W | / / L /
    H E / : | H | / / W /
    Q E W O | Y | / / T /
    M E Z " | P | / / Z /
    B E / O | Q | U / / /
    X E Z " | O | I / / /
    V E / P | B | @ / / /
    / / / L | G | E / / /
    / / / / | " | A / / /
    / / / / | M | £ E / /
    / / / / | X | / V £ £
    / / / / | V | S / / /
    / / / / | £ | / / Y /
```

PHOTOGRAPH OF MONITOR TUBE.

of each page consists of that line together with the first line of the same page (not the next page). Thus the line pair £E consists of £E and //.

Although the information on the magnetic wheel is arranged geometrically so differently from that in the electronic store it may be found convenient to imagine it as if it were similarly arranged. There is very little to interfere with the illusion. The only convenient method for making the content of a track visible is to copy the track on to a pair of tubes, a process which effectively conceals the true arrangement of the digits of the track. This way of thinking permits us to divide up the magnetic information also into lines or line-pairs. There is a good deal of ambiguity about the naming of these lines, for a page of the magnetic store could be copied onto any one of the 8 electronic tubes. We make no attempt to remove this ambiguity and accept that there are 8 alternative names for a line stored on the wheel.

Two further units of the machine must now be described, the control and the arithmetical unit. Roughly speaking the control interprets lines of the electronic store (i.e., the information held on these lines) as instructions and under their direction causes the arithmetical unit to interpret line pairs as numbers and perform certain arithmetical operations on them. These units will be described in detail.

4. The control.

The control of the machine causes instructions to be selected from consecutive lines of the electronic store and decoded. The instruction will, in general, cause the arithmetical unit to select a number (line pair) and perform some arithmetical operation on it. When this operation has been carried out, i.e., when the instruction has been obeyed, the instruction in the next line is selected, decoded, and obeyed. And so on. At the completion of any instruction the control unit contains two quantities, the instruction last obeyed, and the name of the line in which it stands. This latter quantity, called the control number (C), is a 10 binary digit number. Certain kinds of instructions can interrupt the sequential process of selecting instructions and determine instead which instruction is to be selected next. Such instructions are said to cause a transfer of control. Furthermore this transfer of control can be made conditional on the sign (and hence the size) of numbers occuring in the calculation. Thus one of several courses of action can be taken according to how the calculation is proceeding. So far we have mentioned arithmetical instructions and discriminatory instructions. Other kinds of instructions are available which control the input and output units (punched tape reader and teleprinter) and the transfer of large blocks of information from the magnetic to the electronic store and vice-versa.

5. Representation of instructions.

We have already said that the control can interpret the information on a single line as an instruction. It is now necessary to explain in detail how the 20 binary digits of an instruction are made up. It is divided into 4 parts

(i) digits 0-9 give the address of a line in the store

(ii) digits 10,11 and 12 are the B-digits (see below)

(iii) digit 13 is spare.

(iv) digits 14-19 specify the operation (function digits)

The meaning and use of the B-digits will be described later. At this point it will be sufficient to state that they specify one of eight different lines which has to be added to each instruction before it (i.e., the resulting instruction) is obeyed. This could become a nuisance but we avoid any such difficulty by adopting the convention that normally one of these lines (called B0 and specified by 000) is to be zero. For the present we assume that digits 10, 11, 12 and 13 are all zero.

The address digits (1st 2 teleprint characters) give the name of the line involved in the operation which is specified by digits 14-19. The named line may be intended as a destination for the number in the accumulator, or the number standing in the named line may be the operand. The types of instruction may be classified as follows

1. Arithmetical instructions
2. Control transfer instructions
3. B-tube instructions
4. Miscellaneous instructions
5. Magnetic instructions

The magnetic instructions are concerned with transferring blocks of information from the magnetic to the electronic store and vice-versa, and operating the input and output devices - the tape reader and teleprinter. The B-tube instructions will be explained later. The 4th class include instructions which cause the machine to hoot, and to interpret information set up by hand switches on the console.

6. The arithmetical unit.

This unit is similar in its logical design to that of a desk calculating machine. It consists of 2 registers: a 40 (binary) digit register D, and an 80 digit (accumulator) register, A. D is equivalent to the multiplicand register of the desk machine and A is equivalent to the product register. The range of instructions enables numbers to be added to or subtracted from the accumulator or to be multiplied by the number in D and the resulting product added to or subtracted from the accumulator. It is now necessary to explain how the arithmetical unit interprets rows of digits as numbers.

6.1 Representation of numbers.

If we regard a standard number as consisting of 40 binary digits then the product of 2 such numbers will occupy 80 digits, the length of the accumulator. Thus it is possible to regard the row of 40 digits as representing either an integer or a fraction according as to whether the answers are taken from the least or the most significant half of the product. But in whichever way they are regarded the range of instructions enables the arithmetical unit to interpret th m as positive or signed numbers in accordance with the following two conventions. On the plus convention the rows of digits are treated as positive integers (or fractions). On the plus-minus convention the most significant digit is used to represent the sign. The following examples illustrate the two conventions (for convenience rows of 5 digits are used in place of the 40 digits of a line pair).

Plus convention.

$01010_{+} = 10$	$01010_{+f} = 10/32$
$01011_{+} = 26$	$01011_{+f} = 26/32$
$11111_{+} = 31$	$11111_{+f} = 31/32$
$00001_{+} = 16$	$00001_{+f} = 16/32$

Plus-minus convention.

$01010_{\pm} = 10$	$01010_{\pm f} = 10/32$
$01011_{\pm} = -6$	$01011_{\pm f} = -6/32$
$11111_{\pm} = -1$	$11111_{\pm f} = -1/32$
$00001_{\pm} = -16$	$00001_{\pm f} = -16/32$

These conventions are relevant when adding (or subtracting) line pairs to the least significant 40 digits of A (hereafter denoted by L) and also when multiplying numbers. E.g., (still using 5 digits in place of a line-pair) suppose that A contains the row of digits 01011 01100 and the storage location s contains the row 10011. (Hereafter we shall abbreviate such descriptions in the following style: [A] = 01011 01100, [s] = 10011). [s] can be added to the most significant half of A (denoted by M) thus

```
  L       M

01011   01100
        10011
-------------
01011   11111
```

The resulting row of digits is independent of the + or ± interpretation of s.. However when s is added to L there are two possibilities thus

```
(i)          01011   01100
             10011   00000
             -------------
             11001   11100
```

and

```
(ii)         01011   01100
             10011   11111
             -------------
             11001   01100
```

In the second case the augend is extended by 5 copies (i.e., 40, in the actual machine) of its most significant digit before the addition takes place. These two operations will be described by the equations

$$[A]' = [A] + [S]_{+}$$

$$[A]' = [A] + [S]_{\pm}$$

where the dash notation is used to denote the result of an operation. The first example could be described by the equations

$$[A]' = [A] \cdot + \quad 2^{40} \; [S],$$

$$[M]' = [M] + [S] ,$$

or $$[A]' = [A] + [S] ,$$

the signs being dropped because they are irrelevant.

6.2. The Multiplier.

The operation of the multiplier is fairly closely analogous to multiplication on a desk machine. We have already mentioned that there is a multiplicand register (D) which closely corresponds to the lever system of e.g., a Brunsviga. D is capable of being set to 3.2^{39} alternative states. The multiplication will normally be done in two steps, the setting of D and the multiplication proper following afterwards. If an instruction is described as a multiplication it is understood to be one of the latter kind, and corresponds to turning the handle of a Brunsviga. The result of the multiplication is to add to the accumulator (or possibly subtract from it) the product of the number referred to in the multiplication instruction and the number set in the multiplicand. As with a desk machine the operation of multiplication does not alter the multiplicand setting, so that if it is desired to do a further multiplication with the same multiplicand it is not necessary to set it again. Another point of similarity with desk machines is that the product register, i.e., the accumulator, is (as already mentioned) of twice the length of a long store line, i.e., of a multiplier or multiplicand. A point of difference is that there is nothing to correspond to the (not particularly helpful) feature by which multipliers may be accumulated in the counting register on a desk machine. It should also be noticed that with a desk machine multiplicands are always positive, and the sign of the product must be determined through the sign of the multiplicand used i.e., to form the product ab with a desk machine one in effect multiplies $|a|$ and $b(\mathrm{sgn}\ a)$. With the computer it is possible to have either sign in the multiplicand and also in the multiplier. It is also possible to add or subtract the product according to the instructions used.

In explaining the detailed significance of the instructions it is convenient to depart from the principle that the content of a part of the machine is a row of digits rather than a number, and to say that the content of D is an integer satisfying

$$-2^{39} \leq D < 2^{40}$$

One may also put (as a definition) $D_f = 2^{-40}D$. This interpretation of D greatly simplifies the equations describing the instructions.

There are two distinct instructions concerned with setting the multiplicand. Their function digits and defining equations are

001110 (/C) $D' = [s]_+$

011110 (/K) $D' = [s]_\pm$

Thus for instance with the short register machine if $[s] = 01100$, then either s,/ C or s,/ K will make the content D of the multiplicand take the new value $D' = 6$, but if $[s]$ is 01001, then s,/ C will set $D' = 18$ but s,/ K will set $D' = -14$.

There are four distinct instructions for the multiplication proper, to cover the use of either the 'plus' or the 'plus-minus' convention, and also to allow for the product to be either added to the accumulator or subtracted from it. Their function digits and equations are

000010 (/½) $[A]' = [A] - D[S]_+$

010010 (/D) $[A]' = [A] - D[S]_\pm$

000110 (/N) $[A]' = [A] + D[S]_+$

010110 (/F) $[A]' = [A] + D[S]_\pm$

These equations will most often be abbreviated by omitting the brackets, thus

$$/\tfrac{1}{2} \quad A' = A - D\,S_{+}$$

$$/D \quad A' = A - D\,S_{\pm}$$

$$/N \quad A' = A + D\,S_{+}$$

$$/F \quad A' = A + D\,S_{\pm}$$

These equations may also be written in the fractional convention as follows.

$$/C \quad D'_{f} = S_{+f}$$

$$/K \quad D'_{f} = S_{\pm f}$$

$$/\tfrac{1}{2} \quad A'_{f} = A_{f} - D_{f}\ S_{+f}$$

$$/D \quad A'_{f} = A_{f} - D_{f}\ S_{\pm f}$$

$$/N \quad A'_{f} = A_{f} + D_{f}\ S_{+f}$$

$$/F \quad A'_{f} = A_{f} + D_{f}\ S_{\pm f}$$

6.3. The Arithmetical Instructions.

A list of the arithmetical instructions is given on p.1.14 Each instruction is known by the pair of teleprint characters given by digits 10-19 where digits 10, 11, 12, and 13 are 0. Thus the instruction with function digits 011100 will be denoted by /U and that with 100100 by T:. The square bracket notation, used above to denote "the content of ", is dropped. In every case s, the address referred to, denotes a 40 digit line or line pair.

The example below should clarify any points which are still doubtful in the reader's mind. The long lines are assumed to be of length 5 digits as in previous examples, and the following initial conditions are assumed

$[NC] = 01010$	$[NC]_{+} = [NC]_{\pm} = 10,$		$[NC]_{+f} = [NC]_{\pm f} = 10/32$	
$[CC] = 11011$	$[CC]_{+} = 27$	$[CC]_{\pm} = -5$	$[CC]_{+f} = \frac{27}{32}$	$[CC]_{\pm f} = \frac{-5}{32}$
$[TC] = 00001$	$[TC]_{+} = 16$	$[TC]_{\pm} = -16$	$[TC]_{+f} = \frac{1}{2}$	$[TC]_{\pm f} = -\frac{1}{2}.$

The results of a number of consecutive instructions are shown below

	A	A_+	$A_\pm$	A_+	$A_\pm$	D	D
NCT/	01010 00000	10	10	$\frac{10}{1024}$	$\frac{10}{1024}$		
CC/K						-5	$\frac{-5}{32}$
TC/N	01011 10111	+954	-70	$\frac{954}{1024}$	$\frac{-70}{1024}$		
//T:	00000 00000	0	0	0	0		
CC/C						27	$\frac{27}{32}$
NC/½	01001 10101	+754	-270	$\frac{754}{1024}$	$\frac{-270}{1024}$		
NC/½	00100 11010	see following note					

As a result of the last operation the accumulator is said to have exceeded capacity. In this case the defining equation is more correctly written thus

$$A' = \left\{ A - D\,S_+ \right\}_{0}^{79} ,$$

where the notation means 'the eighty least significant digits of the number inside the curly brackets'. It is one of the tasks of the programmer to prevent, by e.g., the use of scale factors, the loss of information in this way.

A closely related problem is the following. When dealing with fractions it is most convenient to take products from M (by /A or /E); with integers, from L (with TA, /S, or /U). In the first case it is necessary to provide for the occurence of products too small to have much significance in M, and in the second case for products too large to remain in L.

The Arithmetical Instructions

Code	Name	Function
010000	/E	$S' = M$ (i.e., replace [S] by most significant 40 digits of A).
011000	/A	$S' = M, \quad M' = 0$
010100	/S	$S' = L$
001100	/I	$L' = M, \quad M' = L$
011100	/U	$S' = L, \quad L' = M, \quad M' = 0$
000010	/½	$A' = A - DS_+$
010010	/D	$A' = A - DS_\pm$
011010	/J	$M' = M + S$
000110	/N	$A' = A + DS_+$
010110	/F	$A' = A + DS_\pm$
001110	/C	$D' = S_+$
011110	/K	$D' = S_\pm$
100000	T/	$A' = S_+$
111000	TA	$S' = L, A' = 0$
100100	T:	$A' = 0$ (clears accumulator).
101100	TI	$A' = A + S_+$
100010	T½	$A' = S_\pm$
100110	TN	$A' = A - S_\pm$
110110	TF	$A' = -S_\pm$
101110	TC	$A' = A + S_\pm$
111110	TK	$A' = 2S_\pm$

In addition to the above functions, which should be self explanatory, there are a few auxiliary instructions which are concerned with the arithmetical unit. They are

001000	/@	most significant digit	
001010	/R	sideways adder	
011001	/W	random number	
110010	TD	$A' = A \vee S_\pm$	Logical functions.
101010	TR	$A' = A \;\&\; S_\pm$	
111010	TJ	$A' = A \not\div S_\pm$	

They will be described separately.

Most Significant Digit.

In certain kinds of numerical work, e.g., the calculation of logarithms, quotients, etc., it is desirable to be able to standardise numbers, i.e., to express them in the form $2^n\alpha$ where $\frac{1}{2} \leq \alpha < 1$. This is made possible at high speed by the function /@. This function determines the position of the most significant digit of $[S]_+$ and adds the digital position number, $\mu(s)$, to M, e.g., if $[S]_+ =$ / X / R Y N / / then $\mu(S) = 28$. If $[S]_+ = 0$ then $\mu(S) = 63$.

Sideways Adder.

This function determines t (s), the number of 1's in [s] (it has no relation to the interpretation of [s] as a number), and adds it to M. E.g., if $[s]_+ =$ / X / R Y N / /, then t (s) = 11.

Random Number Generator.

The function /W puts random digits into the twenty least significant digits of the accumulator. (The randomness is derived from a resistance noise generator). The following problem would be suitable for the use of this facility.

A man in New York starts walking from a street intersection, and at each street intersection decides in which direction to walk by twice tossing a coin (each of the four directions is chosen equally often). It is required to find the probability that before walking twenty blocks he will have succeded in returning to his starting point. For this purpose New York is to be assumed to be an infinite rectangular lattice of streets and avenues.

The logical instructions s, T D; s, T J; s, T R

These are operations on rows of digits and have no relation to the interpretation of these rows as numbers. The symbols V, &, and $\not\equiv$ satisfy the equations

$$\begin{array}{lll}
0 \vee 0 = 0 & 0 \,\&\, 0 = 0 & 0 \not\equiv 0 = 0 \\
0 \vee 1 = 1 & 0 \,\&\, 1 = 0 & 0 \not\equiv 1 = 1 \\
1 \vee 0 = 1 & 1 \,\&\, 0 = 0 & 1 \not\equiv 0 = 1 \\
1 \vee 1 = 1 & 1 \,\&\, 1 = 1 & 1 \not\equiv 1 = 0
\end{array}$$

However the instructions perform these operations on rows of digits thus (01110) V (10010) = (11110).

The main application of these operations occurs where several different pieces of information are packed into one line. For this purpose the operation & is of the most use. It can be used for breaking up a line into its various significant parts. The V operation may be used for combining parts together. Another use is to round off a number by forcing a 1 into its least significant digit position. The following table gives some examples. [A] = RYRYRYRY and [s] are the rows concerned in the operation and the result is placed in A.

Operation	[s]	[A]'
s, T D	ABCD	JXCXRYRY
	ABCB	JXCX££££
	E///	JYRYRYRY
	//££	RY££££££
s, T J	ABCD	DN:MRYRY
	ABCB	DN:NYRYR
	//££	RYYRYRYR
	££//	YRRYRYRY
s, T R	ABCD	@ZRE////
	££//	RY//////
	ABCB	@ZRZRYRY
	//££	//RYRYRY

7. <u>The control transfer instructions.</u>

The codes and defining equations are as follows. C denotes the 10 binary digit control number (see p.26). If, as a result of one of the following operations, C' = 25 (say), then control will start selecting instructions from lines 26, 27, 28, etc., until another transfer of control is encountered. $\{\ \}_0^9$ denotes the 10 least significant digits of the <u>number</u> inside the brackets. This should not be confused with the square bracket notation which refers to a <u>storage location</u>.

001101	/P	$C' = \{s_+\}_0^9$
011101	/Q	$C' = \{C_+ + s_+ + 1\}_0^9$
000101	/H	$C' = \{s_+\}_0^9$ if $[A]_+ \geqslant 0$, otherwise $C' = C + 1$
000111	/M	$C' = \{C_+ + S_+ + 1\}_0^9$ if $[A]_\pm \geqslant 0$; otherwise $C' = C + 1$

The essential feature of control transfers on the MK II (which also distinguishes it from other machines in this respect) is that 2 lines are involved. Thus, e.g., to transfer control unconditionally to CE the control transfer number (one less than the name of the line to which it is required to jump), FE, is first made available in some line s. The instruction s,/P then effects the required transfer. /H is similar to /P but the transfer is only operative if the most significant digit of the accumulator is 0, otherwise the next instruction is selected in the usual way. The instructions /Q and /M cause *relative* transfers of control. They are best remembered as follows. If $\{s\}_0^9$ = j, then s, /Q will cause control to skip j instructions. If j = -1 then s, /Q is obeyed repeatedly and we have a *dynamic stop*. If j = -2 control returns to the previous instruction. And so on. /M is the relative transfer corresponding to the absolute transfer /H.

8. The B-tube and the associated instructions.

The machine is intended to take advantage of the repetitive nature of calculations. Thus if we wish to instruct the machine to add up 100 numbers we prefer to specify just one addition instruction and some how make it refer to each of the 100 numbers in turn, stopping this process at the end of the list. The *B-tube* is primarily a device to enable this to be done more readily.

The B-tube consists of 8 lines each of 20 digits (referred to as B0, B1, B2, ..., B7) and the control unit is arranged so that the content of one of these lines - that specified by the B digits of the instruction - is added to the (*presumptive*) instruction before it is obeyed. The resulting instruction is called the *actual instruction*. Certain instructions are exempt from this rule and in these cases the actual instruction is identical with the presumptive instruction. These - *B exceptional* - instructions have function codes

100011	TO
110011	TB
101011	TG
111011	T"
100111	TM
110111	TX
101111	TV
111111	T£

Examples of the formation of actual from presumptive instructions.

presumptive instruction	named B-line	content of named B-line	actual instruction
/ C E K	B 1	@ / / /	@ C E K
/ C @ K	B 2	@ / / /	@ C @ K
: C A K	B 3	V £ / /	@ C A K
: C £ C	B 7	V £ / /	@ C / K
: C U K	B 7	V £ £ £	@ C U K
/ / / /	B 0	@ C / K	@ C / K

The actual instructions in all these examples are effectively identical because the digits 10, 11, 12, and 13 have no further significance once the actual instruction has been formed.

Setting and altering the numbers in the B-lines is the purpose of the instructions with the following function codes and defining equations.

B-normal	100001	TT	B' = s
	110001	TZ	s' = B
	101001	TL	B' = B - s
B-exceptional	100011	TO	B' = s
	110011	TB	s' = B
	101011	TG	B' = B - s

In these equations B and B' denote, as usual, the initial and final contents of the B-line named by the B digits of the actual instruction (in the case of the B-exceptional instruction this is identical with the presumptive instruction). Likewise s, s' refer to the contents of the short store lines whose addresses are to be found in the actual instruction and the equations applied are specified by the function digits of the actual instruction. Finally there are two conditional switching instructions s,/T (direct) and s, /O (relative) similar in every respect to /H and /M except that they discriminate on the sign of the last B-line altered. For this purpose the operation B → S counts as an alteration. These instructions are intended for use in counting operations.

The B-tube and the associated instructions have three purposes. The primary purpose is the modification of presumptive instructions to give actual instructions. Secondly for the counting operations which are usually associated with the systematic modification of B additives, and lastly as a 'shunting station' for 20 digit lines.

The B-exceptional instructions.

The exceptional instructions were made necessary by the fact that when one is setting a B line one does not usually wish the instruction involved to be modified by what is already in the B line in question. However when using the B line as a shunting station, particularly when transferring from a B line to a position specified by the content of a further B line, it is necessary that the instruction involved should bring two different B-lines into action, one containing information as to the destination of the content of the other. These cases can be covered by the devices revealed in the following examples.

Examples illustrating the effect of the B instructions

Presumptive instruction	relevant B-line and its content	actual instruction	Effect
H U Q O	B7	=presumptive	[B 7]' = [H U]
H U Q B	B7	=presumptive	[H U]' = [B 7]
L U Z Z	B1 = @/I/	H U Q Z	[H U]' = [B 7]
L U Z T	B1 = @/I/	H U Q T	[B 7]' = [H U]
P U P T	B6 = V£//	H U Q T	[B 7]' = [H U]
/ / E /	B1 = HUPO	H U Q O	[B 7]' = [H U]
/ / / /	BO = HUQT	H U Q T	[B 7]' = [H U]
/ / / /	BO = HUQO	H U Q O	[B 7]' = [H U]
/ / Q Z	B7 = @E//	@ E Q Z	[@ E]' = @ E / /
/ / Q /	B7 = @E/O	@ E Q O	[@ E]' = @ E / O

Several applications of the B-tube are given in Examples 2 (next chapter). The first example illustrates a typical counting process and deserves some remarks. It will be seen that the use of the B-tube for the modification of instructions fits very well with its use counting repetitions of an operation. The two are combined in this sequence of instructions. It is not necessary to copy the partial sums out into the store, as the whole process of altering the B-tube and testing is done without interferring with the accumulator.

9. Miscellaneous instructions

1) Dummy stops. It is possible by setting certain switches to arrange that the machine stops when certain instructions are reached. The instructions in question are /L and /G. Like the /I and T: instructions they are addressless, i.e., the first two characters are irrelevant. There are two on-off switches on the console labelled "/L" and "/G". If, e.g., the "/L" switch is in the "on" position, then the machine will stop when control encounters a /L instruction; otherwise /L acts as a dummy or time wasting instruction. Similarly for the /G instruction.

Dummy stops are very useful in testing routines. The programmer is recommended to insert them at points where major operations may be considered complete.

2) The hooter. When an instruction with function symbol /V is obeyed an impluse is applied to the diaphragm of a loudspeaker. By doing this repeatedly and rythmically a steady note, rich in harmonics, can be produced. This is used to enable the operator to be called to attend to the machine in some way. The simplest case is where the whole of a job is completed and it is required to clear the electronic stores and start something different. All that is then required is to repeat a cycle of instructions including a hoot, e.g.,

```
FS   NS/V
CS   FS/P
```

In this case every second instruction will put a pulse into the speaker. These pulses will occur at intervals of 8 beats i.e., 1.92 ms giving a frequency of 521 cycles (An octave above middle C). Or one could use the loop of three instructions

```
O@     /V
B@   Q@/V
G@   B@/P
```

which gives a slightly louder hoot a fifth lower in frequency. Single pulses applied to the loudspeaker are distinctly audible as something between a tap, a click, and a thump. This fact can be turned to good account. By putting hoot instructions into programmes at suitable points one is enabled to 'listen in' to the progress of the routine.

Some indication of what is going on is given by the rhythm of the clicks that are heard.

3) The hand switches. One can set up a row of twenty digits on twenty switches. This row 'H' can effect the behaviour of the machine through instructions with function symbol // or /Z. The former of these will be discussed under magnetic transfers. The latter causes the number set up to be copied into the named line ([s]' = H): in this way small pieces of information can be put into the machine by hand. Suppose for example that we have a routine for calculating some function of a four character line, and suppose that the calculation takes five minutes.. It would then be reasonable to put the argument in through the switches. This would be particularly so if the argument used depends partly on the judgment of the experimenter and partly on the values recently obtained, e.g., if one were trying to find a zero of the function but one was not wishing to repeat the process often enough to mechanise it fully. Again if one were playing chess against the machine this would be the natural way of registering one's own moves.

10. The Magnetic Instructions.

The Magnetic Wheel. The organisation of the magnetic storage into tracks and pages has already been described. To recapitulate there are 256 tracks each consisting of a left hand page and a right hand page. Each page contains the same amount of information as a tube of the electronic store. The two pages of e.g., track 35 will be denoted by 35L and 35R. They may also be referred to as left and right half tracks.

In order to be able to make use of the information stored in the wheel, arrangements are made to enable one to transfer either a half-track or a complete track from the magnetic to the electronic store. The process by which this is done is described as a reading transfer. Likewise in order to store information previously held in the electronic store arrangements are made for transfers in the opposite direction, from the electronic store to a track. These are called writing transfers. In addition there are check transfers by means of which the content of a part of the magnetic store is compared with a part of the electronic store.

These operations are initiated by the instruction s,/ : with the aid of a magnetic instruction which stands in the short line s. Thus the instruction V E / : specifies that $\left[V\ E\right]_0^{19}$ is to be interpreted as a magnetic instruction. A magnetic instruction specifies a half-track in the magnetic store and a tube or pair of tubes in the electronic store and the kind of transfer required. In the case of a checking operation, if the check succeeds, i.e., if there is no discrepancy in the blocks of information compared, then control is advanced to s + 3; otherwise to s + 1. If the transfer involves a whole track of information then the specified pair of tubes on the electronic store must be S0 and S1, S2 and S3, S4 and S5, or S6 and S7. S1 will be called the partner tube of S0, S3 that of S2, and so on. The detailed coding of a magnetic instruction will now be described.

Digits 0 - 9 (1st 2 characters) specify a track of the magnetic wheel.

Digits 16 - 19 (4th character; digit 15 is spare) specify a tube in the electronic store.

Digits 10 - 14 (3rd character) specify the manner of transfer. The following table defines the interpretation of the third character.

/ The content of the left half of the named track is transferred to the named tube.

E The content of the right half of the named track is transferred to the named tube.

@ The content of the complete track is transferred to the electronic store, that of the left half to the named tube and that of the right half to the partner tube assuming that (the named tube must is ~~be~~ one of the following S0, S2, S4, S6).

A As for @, but the left half track is related to the partner of the named tube and vice-versa.

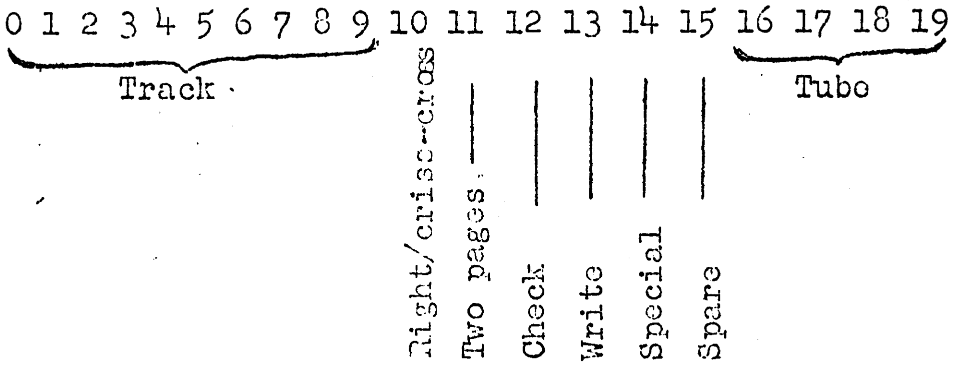
:
S
I
U } Checking transfers used to compare the two, supposedly identical, blocks of information resulting from the operations specified by /, E, @, and A respectively.

½
D   Writing transfers corresponding to the reading transfers
R   specified by /, E, @, and A.
J

N
F   Checking transfers used to compare the results of the
C   operations specified by ½, D, R, and J.
K

The interpretation of all 20 digits can be summarised by the following figure.

Magnetic transfers.



Magnetic instructions whose 3rd character is one of the letters T, Z, ..., £ do not correspond to magnetic transfers, but are used in connection with the input and output devices which are explained in the next section. The following examples of magnetic instructions conclude this section.

/ / / /   Means left half of track 0 to be read into S0

I / E /   "   right half of track 6 to be read into S0

A E @ /   "   left half of track 35 to be read into S0 and the right half into S1

E E A /   "   left half of track 33 to be read into S1 and the right half into S0.

The corresponding checking instructions are / / : /, I / S /, A E I /, and E E U /. Thus A E I / means that the content of 35L is compared with that of S0 and the content of 35R with that of S1.

/ @ ½ I   Means S3 to be written on left half of track 64

½ E D :   "   S2 to be written on 40R

U / R :   "   S2 to be written on 7L and S3 on 7R

U / J :   "   S2 to be written on 7R and S3 on 7L

The corresponding checking instructions are / @ N I, ½ E F :, U / C :, and U / K :.

Notes.

(i) Digit 15 is irrelevant and, in the case of two-page transfers, digit 16 is irrelevant.

(ii) A 1 in digit position 10 has essentially different meanings according as the transfer involves one page or two. In the case of a one-page transfer it means that the right half of the track is involved: for a two-page transfer it indicates that the pairing of the pages is to be criss-cross, i.e., the left half of track to right page of pair and vice-versa.

(iii) Times of operation: writing transfers take about 90 ms and reading and checking each takes about 35 ms.

(iv) The (ordinary) instruction with function letters // causes the row H (see page1.21) to be interpreted as a magnetic instruction. The use of this instruction for starting the machine is described in the section dealing with the input routine.

Input and output equipment.

Information can be fed into the machine from teleprint tape and out of it onto further teleprint tape and onto a teleprinter. The teleprint tape is the five-hole type and is read by a photo-electric tape reader. The teleprinter is a Creed page printer, modified slightly to enable it to print 32 distinct characters. The operation of this equipment is initiated by the special magnetic functions whose 3rd character is T, Z, ..., £.

The tape-reader

The tape reader is operated when the special magnetic function O is obeyed. The character in the reading head is then superimposed ('or') on the 5 most significant digits of the accumulator, and the tape moves forward so that the next character is in the reading head. The latter process takes a certain time (about 5 ms) and arrangements are made to prevent another reading process from occurring until the next character is in position. The operation is most usually done with the accumulator clear so that $2^{75}T$ is placed in the accumulator, where T is the numerical (integral) equivalent of the character read.

The teleprinter and punch

The teleprinter and/or punch are operated when the special magnetic function T is obeyed. Its effect is to punch the character given in the five most significant digits of the accumulator, i.e., $\{A\}_{75}^{79}$. This character will be set up on the printer, and the character previously set up will be printed unless the printer is in the figure-shift position. In this case the corresponding figure is obtained. These figures are given in the table below:

/	E	@	A	:	S	I	U	½	D	P	M	F	
0	1	2	3	4	5	6	7	8	9	+	-	.	

No guarantee can be given concerning what will happen when other characters are printed on figure shift. There may be only a smudge.

A normal teleprinter responds differently to the stunt characters /, @, :, ½, ", £ producing respectively no effect, line feed, space, carriage return, figure shift, and letter shift. The printer associated with the machine prints these characters, but arrangements are also made to do the stunt operations. These are provided by other special magnetic functions: see table below. No effect is produced on the punch in these cases. A three position switch on the console enables the printer and/or punch to be used for output.

Once one of the input or output magnetic functions has been initiated, the machine will continue to obey all other instructions, including the special function B (see below), until a further input or output instruction is encountered when the machine is held up until the first is complete.

The special magnetic function B, superimposes the character set up for punching on digits 35 to 39 of the accumulator.

The following table summarises the special magnetic functions:

3rd character	Effect	Time.
T	Print and/or punch {A}$^{79}_{75}$	140 ms.
Z	Space the teleprinter	140 ms.
L	Carriage return	300 ms.
W	Line feed	140 ms.
H	Figure shift	140 ms.
Y	Letter shift	140 ms.
P	------	
Q	------	
O	Input character from tape	5 ms.
B	Check	5 beats
G	------	
"	------	
M	------	
X	------	
V	------	
£	no effect	4 beats

Notes.

1. It has been found that up to 140 ms. of calculation can be arranged between 2 consecutive output instructions without retarding the operation of the printer

2. In a similar way the figure of 5ms. applies to the input instruction.

3. In general an instruction calling for a carriage return should be followed by one calling for line feed. It is recommended however that a carriage return should be effected by 3 instructions calling for carriage return, line feed, and carriage return, in this order. This eliminates the possibility of carriage bounce and ensures a perfectly even margin throughout the length of the page.

11. The time occupied by various operations.

The machine is synchronised by an oscillator with a frequency of 100 kc/s. One cycle (occupying 10 μ s) of this oscillator may be called a digit period. These digit periods determine the most fundamental rhythm of the machine, but there is another almost equally important rhythm, on which time is divided into beats of 24 digit periods (240 μ s). The number of beats for each instruction are given below. The times for the magnetic instructions have already been mentioned.

Arithmetical and logical instructions other than multiplications.	5 beats (1.2 ms)
Multiplication instructions (i.e., /$\frac{1}{2}$, /D, /N, & /F)	9 beats (2.16 ms)
All the rest of the instructions.	4 beats (0.96 ms)

Chapter 2.

Coding Examples.

Introduction.

The examples given in the next few pages are intended to show some applications of the instruction code. They are divided into 5 groups which, broadly speaking, illustrate

1. The arithmetical and control transfer instructions
2. The B-tube, the most significant digit instruction and the random number generator
3. The magnetic instructions
4. The special magnetic instructions and their use in input and output routines
5. Some of the methods of calculation to be described in chapter 6.

Conventions.

There are certain conventions relating to the use of the electronic store which will be explained in detail in the next chapter. At the moment we need only remark that it is assumed, unless otherwise stated, that certain information is permanently available in S2. This consists of the powers of 2 and certain instructions whose business does not concern us yet. The 40 powers of 2 are so arranged that they occupy only the 41 lines /: to ½S inclusive (see fig. 2.1). Thus e.g., the number standing in the line pair U: and ½: is 2^3 and that in I: and U: is 2^{23}. The lines and line pairs ½S, DS, and RS are also useful.

In the solutions to the examples the sequences of instructions and any necessary working space have been arranged to occupy the first few lines of S0. They could of course stand anywhere in the store but this scheme matches the ideas to be explained in the next chapter. The entry line of each sequence is indicated. A single dot in any character position indicates that the character is irrelevant. Unless otherwise stated any numbers referred to, e.g., $x = [/\ C]_{\pm}$, stand in long lines.

The Powers of 2.

```
└ :|/ / / / |/ |/ / / / |S|
   |E / / / |E |/ / / @ |
   |/ / / / |@ |/ / / / |
   |@ / / / |A |/ / / : |
   |/ / / / |: |/ / / / |
   |: / / / |S |/ / / ½ |
   |/ / / / |I |/ / / / |
   |½ / / / |U |/ / / T |
   |/ / / / |½ |/ / / / |
   |T / / / |D |£ £ £ £ |
   |/ / / / |R |£ £ £ £ |
   |/ E / / |J |/ / / / |
   |/ / / / |N |        |
   |/ @ / / |F |        |
   |/ / / / |C |        |
   |/ : / / |K |        |
   |/ / / / |T |        |
   |/ ½ / / |Z |        |
   |/ / / / |L |        |
   |/ T / / |W |        |
   |/ / / / |H |        |
   |/ / E / |Y |        |
   |/ / / / |P |        |
   |/ / @ / |Q |        |
   |/ / / / |O |        |
   |/ / : / |B |        |
   |/ / / / |G |        |
   |/ / ½ / |" |        |
   |/ / / / |M |        |
   |/ / T / |X |        |
   |/ / / / |V |        |
   |/ / / E |£ |        |
```

Fig. 2.1

Coding hints.

A number of detached and rather trivial tricks and precepts, which are nevertheless considered worth while having in writing, have been collected together at the end of this chapter. They are intended to be studied in conjunction with the examples.

Do coding directly in teleprint code.

It is never too soon to learn the meaning of the 64 functions. The way to do so is to start coding in teleprint code straight away. Keep a list of the meanings always at hand, and refer to it as much as you wish: you will find that after a week very few references are necessary. You will not yet know all the codes, but you will know a working selection. Likewise you will eventually get to know the teleprint equivalents (p 1.3) but this is likely to be slower, chiefly because it is less essential to know them. Although the lines are given names which are in teleprint code, and which also correspond to numbers, for many applications, it is not necessary to know anything concerning the relation of these labellings, or even to have very much to do with the numbers at all. The names of the lines are just used as labels. Later it will be desirable to know the teleprint equivalents of the single characters by heart, but it is never necessary to know the equivalents for pairs of characters.

Examples 1

(1) Given $x = [/\ C]_{\pm f}$ and $y = [@\ C]_{\pm f}$, place $(x + y)$ in : C and $(2x - y)$ in I C.

```
      |/|          |
 ---> |E|/ C T ½|   x
      |@|@ C T C|   x + y
      |A|: C T A|   plant x + y
      |:|/ C T K|   2 x
      |S|@ C T N|   2 x - y
      |I|I C T A|   plant 2 x - y
```

(2) Replace $[/\ C]_{\pm f}$ by its cube

```
      |/|          |
 ---> |E|I S T /|   clear accumulator and set round-off
      |@|/ C / K| ]
      |A|/ C / F| ] x^2
      |:|/ C / A|   plant x^2
      |S|/ C / F|   x^3
      |I|/ C / A|   plant x^3
```

The rounding-off procedures are explained in Hints, 5,b, & 5,c.

(3) Multiply $[@\ C]_{\pm}$ by π^2

```
         |/|
  ——→    |E|  .  .  T  :|   clear accumulator
         |@|  @  C  /  K| ]                 2
         |A|  T  /  /  N| ]  x.  π /16
         |:|  @  C  /  A|
         |S|  U  ;  T  /|   round-off
         |I|  @  C  /  K|
         |U|  D  :  /  N|   x π 2
         |½|  @  C  T  A|   transfer

         |T||T  T  W  N|       2
         |Z||V  H  Q  W|  =  π /16
```

(4) Given z = x + iy, where x = $[/\ C]_{\pm f}$ and y = $[@\ C]_{\pm f}$, place the real and imaginary parts of Z^2 in : C and I C respectively, and $|Z|^2$ in ½ C.

```
         |/|
  ——→    |E|  I  S  T  /|   clear accumulator: round-off.
         |@|  @  C  /  K| ]
         |A|  @  C  /  D| ]  - y²
         |:|  ½  C  /  E|    - y² to ½ C
         |S|  /  C  /  K| ]
         |I|  /  C  /  F| ]  x² - y²
         |U|  :  C  /  A| ]  plant x² - y²
         |½|  :  C  T  ½| ]
         |D|  ½  C  T  N| ]  x² + y²
         |R|  ½  C  T  N| ]
         |J|  ½  C  T  A|    plant |Z|²
         |N|  @  C  /  F| ]
         |F|  @  C  /  F| ]  2xy,
         |C|  I  S  T  I| ]  round-off
         |K|  I  C  /  A|    plant 2xy.
```

(5) Place $\left|[@\ C]_{\pm}\right|$ in / C.

```
         |/|  A  /  .  .|
  ——→    |E|  @  C  T  ½|   [@ C]± to accumulator
    ┌────|@|  /  /  /  H|   test sign
    │    |A|  @  C  T  F|   change sign if -ve
    └───→|:|  /  C  T  A|   replace
```

or alternatively, using a relative transfer of control instruction (see Hints, 10).

```
         |/|
  ——→    |E|  @  C  T  ½|
    ┌────|@|  E  :  /  M|   skip one instruction if +ve
    │    |A|  @  C  T  F|
    └───→|:|  /  C  T  A|
```

(6) $[/\ C]_{\pm f}$ is either ¼ or ⅛. Whichever it is, replace it by the other.

```
         |/|
  ——→    |E|  @  S  T  ½|
         |@|  :  S  T  C|   ⅜
         |A|  /  C  T  N|   ⅜ - [/ C]±
         |:|  /  C  T  A|   replace
         |S|
```

(7) Repeately double $[/ C]_+$ until it becomes ⩾ ¼

```
 - ->|/|/ C / E|]double
--+->|E|/ C / J|]
  '..|@|D S / H| test sign
```

N.B. The sequence must be entered with the accumulator clear.

(8) a, b, and c are given in / C, @ C, and : C respectively, and it is known that $|b| < 2^{-6}$ and $o < ac < 2^{-14}$. Form 2^{20} abc without undue loss of accuracy, and place it in / C.

```
--> |/ |    T : |  clear accumulator
    |E |/ C / K|]
    |@ |: C / F|]  ac (double length product)
    |A |/ C / U|]  to /C and :C
    |: |: C / U|]
    |S |X : / C|]
    |I |: C / F|]  2^14 ac
    |U |. . / I|]
    |½ |/ C / N|]
    |D |I S T I|   round-off
    |R |/ C / A|   2^14 ac to /C
    |J |I S T /|   set round-off
    |N |F : / C|]
    |F |@ C / F|]  2^6 b to @C
    |C |@ C T A|]
    |K |@ C / K|]  2^20 abc
    |T |/ C / F|]
    |Z |I S T I|   round-off
    |L |/ C / A|   result to /C
```

<u>Examples 2.</u>

These examples illustrate some applications of the B-tube.

(1) Place in L the <u>check sum</u>, i.e., the sum modulo 2^{40}, of the line pairs, of page 4.

```
         |/|V E / /|  count number
  ------>|E|@ / T :|  clear accumulator
         |@|/ / Q O|  set B7 to 62
  .----->|A|/ ½ Q I|  partial sum
  |      |:|A : Q G|  adjust counter
  '------|S|E / / T|  test for last cycle (see Hints 7)
```

(2) Place in / C the scalar product of the two vectors whose elements are in (i) / N, @ N, ... ,L N and (ii) / F, @ F, ..., L F inclusive.

```
         |/|L / / /|
  ------>|E|I S T /|   set round-off
         |@|/ / Q O|   set B7
  .----->|A|/ N U K|]  add product
  |      |:|/ F U F|]  to partial sum
  |      |S|A : Q G|   adjust counter
  '------|I|A : / T|   test for last cycle
         |U|/ C / A|   transfer result.
```

(3) Given $x = [D]_{\pm f}$ and $a_o, a_1, \dots, a_n$ in locations / C, @ C, : C, etc., respectively, where $2n = [T\ I_s]_+$, place $f(x) = a_o + a_1 x + a_2 x^2 + \dots + a_n x^n$ in V K.

```
      | / | A / . . |  control number
----> | E | I S T / |  set round-off
      | @ | V K / A |  clear V K
      | A | T I Q O |  set counter
----> | : | V K / F |
      | S | / C U J |  p_{r+1} = p_r x + a_{n-(r+1)}.
      | I | V K / A |
      | U | A : Q G |  adjust counter
      | ½ | / / / T |
```

The polynomial, is built up by the recurrence relation

$$p_0 = a_n$$

$$p_{r+1} = p_r\, x + a_{n-(r+1)}$$

$$p_n = f(x).$$

(4) It is required to multiply the 80 digit number in A by 2^n, where n is such that $x = 2^n \left[A\right]_0^{79}\,{}_{\pm f}$ lies in one of the ranges $\frac{1}{2} > x \geqslant +\frac{1}{4},\ -\frac{1}{4} \geqslant x \geqslant -\frac{1}{2}$. Record the number n in B7 thus: $\left[B7\right]_{\pm} = -2n$. Code this operation: (i) using the fewest possible orders; (ii) so that it requires the least possible time.

(i)

```
      | / |         |
      | E |         |  working space
      | @ |         |
      | A |         |
----> | : | / / / S |
      | S | @ / / E |
      | I | / : Q O |  clear B7
      | U | A : / M |  test sign of A
----> | ½ | U / T : |  clear A
      | D | K / / P |
----> | R | E : T N |
      | J | ½ / / H |  case A > 0
      | N | . . / P |  case A = 0
      | F | / / / / |
      | C | / / / O |  ≡ -¼
      | K | Z / / / |  control number
----> | T | A : Q G |  records each doubling
      | Z | F / / J |  restores accumulator
----> | L | @ / / J |
      | W | / / T I |  double
      | H | @ / / E |
      | Y | / / / S |
      | P | : S / J |  +¼
      | Q | D / / H |  test range
      | O | F / / J |  -¼
```

Note:- Using this method x lies in one of the ranges

$$\tfrac{1}{2} > x \geqslant \tfrac{1}{4}, \qquad -\tfrac{1}{4} > x \geqslant -\tfrac{1}{2}.$$

In the method that follows x lies in one of the ranges

$$\tfrac{1}{2} > x \geqslant \tfrac{1}{4}, \qquad -\tfrac{1}{4} \geqslant x > -\tfrac{1}{2}.$$

and therefore originally $-\frac{1}{4} \leqslant \left[A\right]_{\pm f} \leqslant \frac{1}{4}$

```
(ii)  /|           |
      E|           |   working space
      @|           |
      A|           |
      :|T  @  /  / |   = 80
      S|N  @  /  / |   = 76
      I|½  E  /  / |   = 40
      U|E  :  T  N |
      ½|M  /  /  M |   test for zero
      D|      /  P |   case L = 0, M = 0
      R|/  /  /  S |   deposit L
      J|@  /  /  E |   deposit M
      N|E  E  /  H |   test sign
      F|D  S  T  J |
      C|E  :  T  C |   change sign if -ve
      K|/  :  Q  O |   set B7 = 0
      T|½  /  /  E |   deposit |M|
      Z|D  S  /  J |
      L|O  /  /  H |   test M = 0
      W|:  /  Q  G |   subtract 80 from B7
      H|½  /  /  S |   deposit |L| in place of |M|
      Y|/  /  T  / |
      P|@  /  T  A |   substitute L for M
      Q|/  /  T  A |   clear //
      O|Q  /  T  : |
      B|½  /  /  @ |
      G|½  /  /  @ |   2μ to ½/
      "|½  /  /  A |
      M|S  /  P  O |
      X|½  /  P  G |   76 - 2μ
      V|½  /  P  B |
      £|½  /  Q  G |   subtract 76 - 2μ from B7
   E  /|E  :  I  C |
      E|I  /  P  G |   subtract 40 from B6
      @|E  :  /  O |   test 36 - 2μ
      A|E  :  /  Q |   case 19 ≤ μ ≤ 39
      :|/  :  I  C |   case μ < 19
      S|@  /  /  F |
      I|@  /  T  A |   form result
      U|@  /  /  J |
      ½|/  /  /  N |
```

Note: Lines :/ and S/ are used as working space so that as it stands the sequence of instructions is not self-resetting, i.e., it cannot be used again unless the instructions in these lines are restored. There is no need for this however if a fresh copy is brought down from the magnetic store every time it is to be used.

(5) Fill S4 with random digits.

```
(i)   /|V  E  /  / |
      E|/  /  Q  O |   set B7
      @|E  /  /  W |
      A|E  ½  Q  A |   build up 40 random
      :|:  :  /  W |   digits
      S|/  ½  Q  C |
      I|/  ½  Q  A |   plant random line
      U|A  :  Q  G |
      ½|@  /  /  T |   test for 32 repetitions
      D|           |
      R|           |
      J|           |
```

(ii)

```
| / | A B C D |  working space
| E | E F G H |
| @ | / / / / |
| A | / / / / |
| : | V E / / |
| S | I / / / |
| I | : / Q O |  set B7
| U | / / / C |  square
| ½ | / / / F |
| D | / / / U |
| R | @ / / U |  select middle
| J | E / T / |  40 digits
| N | / / / S |
| F | / ½ Q A |  plant long line
| C | A : Q G |  test for 32 repetitions
| K | S / / T |
```

Examples 3.

(1) Transfer the content of S6 to half track 25R: check by means of the appropriate write-like checking instruction.

```
| / | B / D N |
| E | B / F N |
| @ | / / / : |  write
| A | E / / : |  check
| : | S / / Q |  failure
| S | X £ / / |
| I |         |
```

(2) Repeat (1) but check by bringing down 25R to S7 and then comparing S6 and S7 line for line.

```
| / | B / D N |
| E | B / E C |
| @ | / / / : |  S6 to 25R
| A | E / / : |  25R to S7
| : | S / T : |
| S | N / Q O |  set B7
| I | / N Q C |  form difference of
| U | / C Q N |  corresponding lines of S6 to S7.
| ½ | A : Q G |
| D | : / / T |
| R | A : / M |
| J | E : / P |  test if [S6] - [S7] = 0
| N | V E / / |
| F | E : T N |
| C | E : / P |
```

(3) It is required to write the content of S4 onto the half track specified in the first 11 digits of line n of half track 34L, where $n = [B6]_{+}$. All magnetic transfers to be checked by the method given in example (1).

```
   / @ E / @
   E @ E : @
   @ £ £ E /
   A / / ½ ½
-> : / / / :
   S E / / :    brings down
   I U / / Q    "directory"
   U X £ £ £
   ½ / @ P /
   D @ / T R    Form and plant the write
   R A / T C    and write-like check magnetic
   J / / / S    instructions in // and E/
   N B : T C    respectively
   F E / T A
   C / / / :
   K E / / :    writes [S4] to
   T U / / Q    named track
   Z
   L
```

Note: this sequence is not self-resu~~lt~~ing.

Examples 4.

These examples illustrate the use of the special magnetic instructions (s.m.i.) for input-output operations.

(1) If $[/\ C]_{\pm} < 0$, then print a minus sign; otherwise space the teleprinter. Also replace $[/\ C]_{\pm}$ by its modulus.

```
   / I / T /    used as s.m.i.
-> E / C T ½
   @ / / / M    test sign
   A . . T :
   : E / / J
   S / / / :    print "-"
   I / C T F    change sign
   U A : / Q    skip 2 instructions
   ½ / / Z /    space (s.m.i.)
   D ½ / / :    space teleprinter
   R / C T A    plant modulus
```

(2) Print, on a new line, the positive number $[M]_{+}$ in decimal form, rounded-off to 9 decimal places.

```
   /                working space
   E
   @  R / / /
   A  / / / /       = 10
   :  I Z / /       = 5.10^-10, round-off constant
   S  / / / /
   I  . . L .       carriage return (s.m.i.)
   U  . . W .       line feed (s.m.i.)
   ½  . . H .       fig. shift (s.m.i.)
→  D  I / / :       carriage return.
   R  U / / :       line feed
   J  ½ / / :       figure shift
   N  U : Q O       set digit count
   F  : / / J       round-off
   C  / / / A
→  K  C / T :
   T  / / / C       multiply by 10
   Z  @ / / N
   L  / / / A       digit (integral part) to D
   W  / / / C
   H  / / T A       fractional part to //
   Y  V : / N       shift digit to top end of
   P  . . / I       accumulator
   Q  K / / :       print digit
   O  E : Q G       adjust counter
   B  K / / T       test for last repetition
```

(3) Read a positive integer n punched in decimal form and terminated by the symbol T and add it to L.

```
   /  . . O /       input (s.m.i.)
→  E  . . / A       clear top half of accumulator
→  @  / / / :       read digit from input tape
   A  E : / M       test for the symbol T (-ve)
   :  . . / P       exit from sequence
→  S  C / / A       shift new character to least
   I  J : / C       significant 5 digits of M
   U  C / / N
   ½  C / / U       multiply partial sum by 10
   D  N / / C       and add new digit
   R  C / / N
   J  E : / P       read next digit
   N  R / / /
   F  / / / /
   C
   K                working space
```

(4) Read from the input tape and place in M, a fraction punched in decimal form and followed by sign. The routine is to be arranged so that any number of digits can be punched-even though only about 11 are significant. e.g., - $\frac{1}{7}$ is to be punched as

142857142857-.

N.B., the symbol + is identical with P and the symbol - with M.

```
/ . . T :
E H / Q O     set B7
@ W / / :     input character
A W / / H     test for sign symbol
: W / P O     if the sign is one of the first 11
S @ / P B     symbols read, then the input instruction
I O / / A     (in @/) is replaced by a dummy for the
U / / / A     remainder of the digit cycle.  The sign is
½ / / / C     stored in O/.
D J : / N
R / / / U
J / / / C     digit cycle: 11 repetitions
N H / / N
F E : Q G
C E : / T
K / / T A
T O / / J     test whether sign symbol has
Z O / / H     already been read
L U : / Q
W I / O £
H R / / /     = 10
Y / / / /
P N V @ V     = 2^36. 10^-11
Q £ Y C Y
O Q / T :
B W / / :     search tape for sign symbol
G O / / H
" @ S / J
M A : / M     test for "-" sign
X / / T F     change sign in the case of a
V / / T A     "+" sign
£ / / / K
/ D : Q O
E @ E T :     multiply by 2^40 10^-11 by
@ P / / N     accumulating 16
A P / / ½     multiplications by 2^36 10^-11
: E : Q G
S E E / T
```

Note: this sequence is not self-resetting.

Examples 5.

These examples illustrate some of the methods of calculation described in chapter 6.

(1) Replace $[M]_{\pm f}$ by $(\exp [M]_{\pm f})$ -1. Use the method based on the recurrence relation

$$z_{n-1} = z_n + \frac{z_n^2}{2^n}, \quad z_{40} = x, \quad z_1 = e^x - 1.$$

/		working space
E		
@	strobe	
A		
:		
S	Z_n	
I	: / / A	plant x
U	E : / J	set strobe
½	@ / / A	
D	: / / K	Z_n^2
R	: / / F	
J	/ / / A	
N	I S T /	set round-off
F	@ / / K	$Z_n + \frac{Z_n^2}{2^n}$
C	/ / / F	
K	: / / J	
T	: / / A	
Z	@ / / J	shift strobe
L	@ / / J	
W	H / / H	test for end of process
H	U / T :	
Y	: / / J	

(2) Replace $[M]_{\pm 1}$ by $[M]_{\pm 1}/[D]_{\pm 1}$. Use the repetive process

$$a_{n+1\circ}= a_n(1-C_n) \qquad a_n \longrightarrow \frac{a_o}{C_o+1}$$

$$C_{n+1} = -C_n^2 \qquad C_n \longrightarrow 0.$$

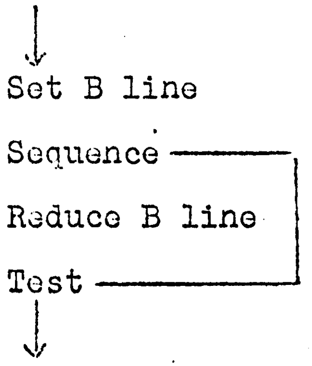
/		working space
E		
@		
A		
:	/ / / A	plant dividend
S	L / T :	
I	E : / N	divisor to accumulator
U	@ / / S	
½	S : / M	test sign of divisor
D	@ / T F	make divisor +ve and change sign of dividend
R	@ / T A	
J	/ / T F	
N	/ / T A	
F	/ / T K	repeatedly double divisor and dividene dividend until divisor ⩾ ½
C	/ / T A	
K	@ / T K	
T	@ / T A	
Z	@ / T ½	
L	M / / H	
W	I S T /	set round-off
H	@ / / K	repetitive cycle
Y	/ / / J	$a_{n+1} = a_n - a_n C_n$
P	/ / / D	
Q	/ / / A	
O	@ / / D	$C_{n+1} = -C_n^2$
B	@ / / E	
G	E : / M	test for end of process
"	S / / P	
M	N / T :	
X	/ / / J	quotient to M

(3) Replace $[M]_{\pm f}$ by its square root. The method to be used is the non-restoring form of the digit-by-digit process described in chapter 6.

```
       / / / / /     y_n
       E / / / /
       @ / / / /
       A / / / O     negative strobe
       : / / / /     a = [M]±f
       S / / / /
       I : / / A     plants a initially
       U I / T :
       ½ : / / J
       D / / / K     forms a-y_n^2
       R / / / D
       J Q / / H     tests sign of residual
       N @ / T ½     negative correction
       F / / T C     enters digit                  digit
       C / / T A                                   cycle
       K @ / / K     shifts strobe
       T I S / N
       Z I S T I     tests for end of process
EXIT   L       / H   plants strobe
       W @ / / A
       H U / / P
       Y N / / /
       P @ / T F     +ve correction
       Q Y / / P
```

Coding Hints.

1. Counting procedure. One of the commonest operations is a sequence of instructions to be repeated a given number of times. The counting process may be done in the B tube e.g., in B7. The most usual scheme is as follows.



The number of times that the sequence is obeyed is greater by one than the number set in the B-line. It is frequently desirable to subtract something other than 1. This may be because the number of repetition is the result of a computation, and is given with some factor applied, e.g., a power of two. Alternatively it may be desired to save a line by setting the B-line with some quantity already available, e.g., to count 12 one might set GC/J into the B-tube, and subtract ///E, the former being supposed an instruction which is used in any case, the latter available in PERM.

2. Ommission of counting. If the operation to be repeated contains rather few instructions e.g., three, and is crucial for the speed of the whole process, then it may be best to omit the instructions concerned with counting and to repeat the instructions of the process in question the requisite number of times. Sometimes the number of repetitions may be the result of calculation, but even then the omission of counting method may still be applied, the number of repetitions being controlled through a control transfer entering the sequence of repeated operations at the appropriate point.

3. Discrimination by control transfer. When two cases have to be given quite different treatment, involving different sequences of instructions it is natural to choose the relevant sequence by a test instruction (i.e., conditional transfer /T, /H, /M, or /O). When there is a large number the best method is to manufacture a control transfer number and use a directory. A good example of this is in the input programme where

the ~~six~~ eight warning characters each have to be given a seperate treatment. This is dealt with as follows. If α is the character in question α/// is set to B6, and the instruction //IP given. The lines // to £/ contain the $\overset{\text{nine}}{\text{seven}}$ control transfer numbers appropriate to the thirty two possible values of α and the sequence required is immediately entered.

4. Changing sign in the accumulator. The instruction DSTJ has the effect $A_{\pm}' = -1 - A_{\pm}$ which for most purposes is as good as $A_{\pm}' = -A_{\pm}$ which can only be achieved in two instructions.

5. Clearing the accumulator. The beginner is liable either to leave things in the accumulator to get mixed up with the next calculation or else to put in accumulator clearing instructions which could easily have been avoided. In fact it is very seldom necessary to give a special instruction for clearing the accumulator, if the points below are born in mind.

(a) Instruction TA clears the accumulator as well as transferring L to store. If both halves of the accumulator are required to be stored one can use /U twice and the accumulator will be cleared.

(b) If an expression of form a + bc is required and the accumulator is not clear the term a should be put into the accumulator first. This applies if the final value required will be in L, and rounding off multiplications with results taken form M.

(c) Alternatively when doing multiplications with results taken form M it is not necessary to clear the whole of the accumulator in advance, but only M. The maximum error will be 1 in either case: the mean square error will be one third with clearing but only one sixth without. If the results are taken out with /A then M remains clear for another multiplication.

6. Electronic space economy measures.

The economising of instructions in order to reduce the space occupied in the magnetic store is seldom worth while. There are however occasions when it is worth while to economise them to save space in the electronic store. This is nearly always in order to get the instructions either into one page or into two pages (see next chapter). To do so makes the routine tidier and usually has time-economy effects. We have mentioned two or three devices for keeping the number of instructions down, but these will mostly be learnt by experience.

7. Duplication of use of lines. The chief economy measure available other than reducing the number of instructions is the use of a line for more than one purpose. One or two forms of this have already been mentioned. It is usual for instance to use the 1st 2 characters of addressless instructions as control transfer numbers. Another case was mentioned under the heading of counting. No attempt can be made to list all such devices, but there are a number associated with control transfer numbers. These are sufficiently numerous that it should nearly always be possible to avoid using any lines specially for control transfer numbers. To make a point of doing so in cases where it is not strictly necessary is however strongly to be discouraged, as liable to lead to a most wasteful use of programmers time. The methods already known to be available are mentioned below.

8. Sandwiching. If a (short) line is sandwiched between two sequences of instructions, the instruction which uses that short line can also be used as a control transfer number for the beginning of the later sequence of instructions. A long line can only be used in this way if it ends with a pair of characters representing a harmless instruction. Lines ending with T£, Z£,.., ££ are almost the only suitable ones.

9. Positioning of dummy stops. If a dummy stop or other addressless instruction be placed immediately before a junction, (i.e., an instruction to which a control transfer is made, but which is not the beginning of a straight sequence) then this addressless instruction may be used as a control transfer number in the usual way, and the control transfer instruction may also be used as control transfer number for the transfer to the junction. This position may not however always be suitable from other points of view.

10. Relative control transfers. Relative transfers (e.g., /Q) are more troublesome to use than the direct ones, but provide a second

string. The necessary relative transfer number may sometimes be found in the powers of 2, or if part of the routine itself is being used as special working space, it may be found in the routine itself.

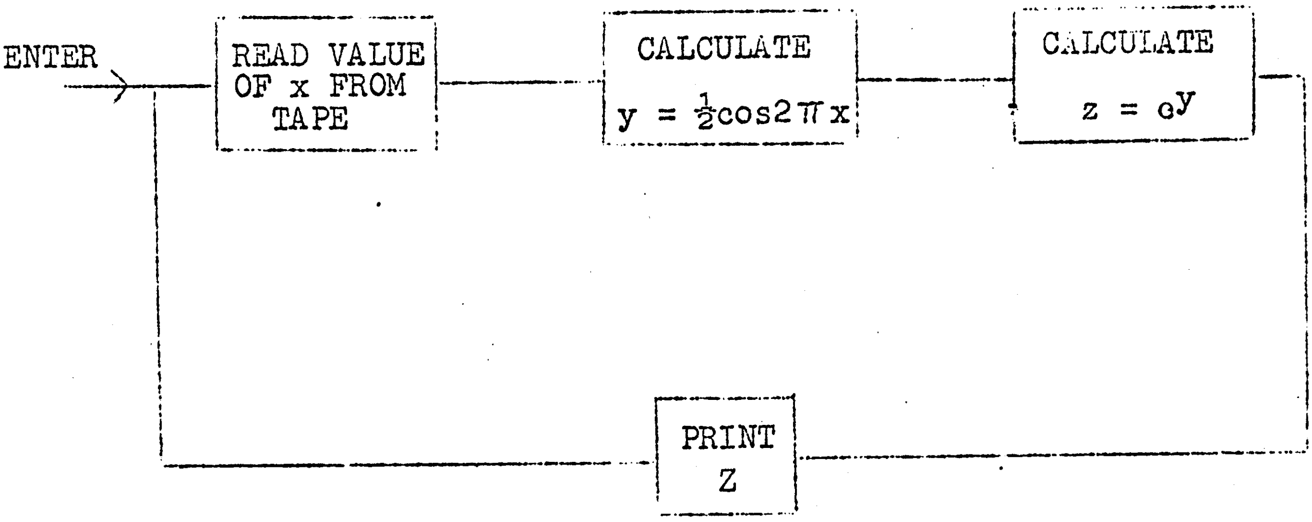
11. Inaccurate numbers. If a line pair is required to be known to more than twenty but less than thirty binary digits, the first two characters of the line pair used may be changed to a control transfer number. Likewise if a number is required to less than ten digits a control transfer number may also be concealed in it.

The preparation of a problem: Programming: Scheme A.

The coding of a complete problem is normally achieved by working up from relatively simple requirements to more complex ones. Thus for instance if it is desired to do a Fourier analysis on the machine it would be as well to arrange first that one can calculate cosines on it. For each process that one wishes the machine to be able to carry out one constructs a routine. This consists mainly of a set of instructions, which when obeyed by the machine, effect the required operations. The term programming is used to describe the process of breaking down the problem into routines: coding to describe the making of the routines.

The following simple example illustrates how a problem is broken down into simple processes. It is required to calculate $e^{\frac{1}{2}\cos 2\pi x}$ for various values of x. The calculation is to be arranged so that the required arguments can be punched on the input tape and that the machine reads these values one at a time, printing corresponding function values on the teleprinter.

A suitable schematic diagram is


ENTER
READ VALUE OF x FROM TAPE
CALCULATE $y = \frac{1}{2}\cos 2\pi x$
CALCULATE $z = e^y$
PRINT Z

The break down of this problem is fairly obvious. In more complex problems it would be by no means so simple or unique. On the basis of the above breakdown one would start to make routines to do the following processes.

1. Calculate cos $2\pi x$ given x in some preassigned location.
2. Calculation of e^x similarly.
3. Read a number punched on tape in decimal form.
4. Print a number, held in a given line in the store, in decimal form.

Of course there are a lot of details about these routines which remain to be filled in, e.g., in which storage location will the cosine routine find the argument x and where will it place the result? What lines, if any, may be altered in the process? On the numerical side it is necessary to give some attention to scale factors. Thus in this case consideration of the permissible range of numbers on the plus-minus fractional convention would indicate that the cosine routine should calculate $\frac{1}{2}$ cos $2\pi x$ rather than cos $2\pi x$.

What is normally required of a routine of this type is that a certain function of the content of one or more lines shall be calculated and stored in a given place, the rest of the content of the store being unaffected by the process, except certain named lines (which should be kept as few as possible) called the working space of the routine. Thus e.g., the specification of the cosine routine might read as follows.

Name of Routine: COSINE/A

Effects: Replaces $[:\,C]$ by $\frac{1}{2}\cos 2\pi\,[/\ C]_{\pm f}$, the line @ C being altered in the process.

It is one of the tasks of the programmer to ensure by means of his 'cementing' instructions that no valuable information is left lying in @ C and : C when COSINE/A is called in.

Library routines.

It is clear thet certain processes are common to all kinds of problems. E.g., most problems will need some kind of print routine Some common processes for which it would be useful to have standard ready-made routines are the following.

Calculation of functions

(a) Algebraic: division, square root, cube root, and nth root,

(b) Elementary: cosine, exponential, logarithm, & arcotangent.

Algebraic processes

Solution of linear simultaneous equations, inversion of a matrix.

Solution of algebraic equations and the evaluation of the latent roots and vectors of a matrix.

Analytical processes

Quadrature, integration of ordinary differential equations, and the solution of integral equations.

A library of ready-made routines covering these and other processes is essential if programming is to become possible for those who cannot devote their whole time to the subject. The use of library routines saves programming time in two ways. In the first place it saves the programmer the task of coding certain processes for himself, and secondly the library routines are error free so that further time is saved at the fault location stage. Furthermore his programme will be more efficient becausd library routines are economical both in their time of operation and storage space.

The master routine of a problem.

Consider the problem suggested above. When the routines for the basic processes have been coded then we must think about connecting them together, i.e., arranging that the instructions of the component routines are obeyed in the correct sequence and that any intermediate bits of arithmetic or shunting of numbers that may be required are carried out. E.g., the result calculated by the cosine routine must be placed in the storage location specified by the exponential routine. The instructions required for these tasks are called the cementing instructions. The totality of cementing instructions constitute the master routine. The master routine is at the same time a routine for calculating $e^{\frac{1}{2}\cos 2\pi x}$ $\cos 2\pi x$ and is said to use the basic routines as sub-routines. The master routine may itself be a sub-routine in a larger problem. And so on. Likewise the sub-routines of any routine may themselves have sub-routines. One eventually comes down to a routine without sub-routines.

There exist at present two distinct systems for organizing sub-routines, known respectively as Schemes A and B. Fundamentally these two systems are similar though they differ considerably in detail. In order to avoid excessive cross-referencing the two schemes and their associated input organizations will each be described completely although this will of necessity involve a certain amount of duplication.

Scheme A.

Scheme A which was first chronologically, will be described here. It depends on the presence in the electronic store of certain fixed material known as PERM, and on the use of a 40 digit number associated with a routine and called the cue. In general it is assumed that the master routine and its sub-routines are kept in the magnetic store, the various parts being brought to stores 0 and/or 1 as required, this being effected by certain orders contained in PERM and called the Routine Changing Sequence or R.C.S. Without at the moment considering the details either of the constitution of the cue or of the operation of the Routine Changing Sequence, both of which are described later, it can be said that from the cue, the R.C.S. extracts the following information about a routine:-

(1) The number of the magnetic track in which it is stored.

(2) The electronic store or stores to which it is to be brought.

(3) The line at which it is to be entered.

It then carries out the appropriate magnetic transfer, applies certain checks and enters the routine at the correct line.

To enter a sub-routine from a master routine, the latter places the cue to the sub-routine in a pre-assigned storage location (VS) and then transfers control to the R.C.S., which calls down the sub-routine from the magnetic store and enters it at the correct place. When the end of the sub-routine is reached, a return is made to the master routine by a second cue known as the link; i.e., the master routine has to provide two cues, one to the sub-routine and the other for return to itself. Library routines intended for use with Scheme A are designed to use as link the quantity held in the least significant half of the accumulator on entry. Such routines start with an instruction which stores away this quantity and then finally emerge to the R.C.S. with it as cue.

True and False Cues.

Since it is not desirable that each routine should always be stored in the same magnetic track, provision has been made for changing the track in which a routine is stored without altering its cue. This leads to the use of two types of cues which are treated identically by both master and sub-routines but differently by the R.C.S. The first type of cue - the true cue refers directly to the number of the track in which a routine is kept. The second type, the false cue refers instead to a line on a special cue bearing track. Instead of allocating a fixed track to a routine with a false cue, it is allocated a fixed line in the directory which is held in one or more cue bearing tracks. This line which can be set as required, contains the number of the actual track in which the routine is stored. We shall now describe in detail the composition of both types of cue.

The first 20 digits of both true and false cues are divided as follows:-

(1) Digits 0-9. These provide a control transfer number. i.e., one less than the number of the line at which the routine starts.

(2) Digits 10-19. These are the check characters and are used to check the track selection mechanism. They are given by the value of $\left\{1025\left[\left[/\ E\right]_{+} - \left[/\ A\right]_{+}\right]\right\}_{10}^{19}$ when the routine is first entered. The short lines / E and / A are known as the principal lines of the routine and for two page routines, i.e. those occupying S0 and S1 the check characters may be immediately determined. For details of the formation of these characters in the case of one page routines see see below (p. 3.7)

(3) Digits 20-39. Here two cases arise according to the value of digit 39:-

(i) If digit 39 is 0 then the cue is a true cue and these last 10 digits give the magnetic transfer necessary to bring the routine to Store 0 and/or Store 1.

(ii) If digit 39 is 1 then the cue is a false cue and the last

10 digits are divided into digits 20-29 which give the line in the directory containing the required magnetic transfer, and digits 30-39 which, with digit 39 replaced by 0 give the number of the track containing this directory.

The Routine Changing Sequence.

The permanent material PERM, which is assumed to be in the electronic store is as shown in figure 3.1 . The lines /: to ½S contain powers of 2 which may either be considered to range from $2^0 - 2^{19}$ or from $2^{20} - 2^{39}$. Lines DS and RS contain -1 and the R.C.S., is in lines NS - XS of store 4 and UK - WK of store 7. When PERM is first brought from the magnetic to the electronic store the lines VS and £S contain TEUN@/EZ which is the false cue of the routine WRITE (see p.3.16). These lines will however be altered every time a routine is called in as the cue is always deposited in this location before entering the R.C.S. With the cue in the lines V.S. and £S and with the link (if any) in the least significant half of the accumulator, the R.C.S., is entered by the instruction NS/P which transfers the control to line CS. The first step after entering the R.C.S., is to deposit the link temporarily in VK and £K, leaving the accumulator empty, and free for other purposes. This instruction VKTA is followed by JS/L which is a dummy stop. This stop will be encountered whenever routines are changed by the R.C.S., and enables one, when checking routines on the machine to speed through sub-routines which are known to be correct. The next two instructions differentiate between true and false cues. In the case of a true cue the next instruction to be obeyed is £S/: which obeys the magnetic instruction contained in the cue. The next five instructions form the quantity $\left\{1025\left([/\ E] - [/\ A]\right)\right\}_0^{19}$ whose value depends on the previous magnetic transfer, and which is unlikely to be correct if the wrong transfer has been made. The four subsequent instructions are concerned with verifying whether the value obtained, agrees with that given by digits 10-19 of the cue. If the two agree the link is replaced in the least significant half of the accumulator and the new routine is entered by the control transfer VS/P using the first 10 digits of the cue. If the check characters do not agree, control is transferred to line NS and a loop stop containing a hoot occurs.

In the case of a false cue, instead of proceeding with the magnetic transfer £S/: control is transferred to line ½K of store 7. Four instructions now form, from the last 10 digits of the cue, a new magnetic transfer to bring the left half of the appropriate cue - bearing track to store 0. This transfer is carried out and the next six instructions select the line specified by digits 20-29 of the cue and copy the content of this line into £S. The R.C.S., is now re-entered just after the dummy stop i.e., at line TS. £S now, however, contains a new line, not ending in 1 and consequently [V S] is treated as a true cue in the manner described above.

Check characters for single page routines.

We are now in a position to consider the value of the check characters for single page routines. In the case of a single page routine occupying either store 0 or store 1 and having a true cue, one principal line will belong to the routine itself and the other will depend on its master routine - i.e., the check characters will depend on the context in which the routine is used. This also applies to single page routines occupying store 0 and having false cues. In the case, however, of a single page routine occupying store 1 and having a false cue, it can be seen from the above description of the R.C.S., that the cue bearing track will be left in Store 0 when the routine is brought to store 1. As there is a convention that line /E of a cue bearing track always contains the number of that track, both principal lines are known and the check characters can be given without qualification. The difficulty of having to work out the check characters of a single page routine for each application can often be avoided by pairing routines in two halves of the same track and providing an alternative cue which brings down both halves.

Routines having more than two pages

There is nothing in this scheme to preclude a routine having more than two pages. If this occurs the first two pages will in general be brought down by the cue and these will contain the cue to the next one or two pages. In this case, which is equivalent to the calling in of an ad-routine in Scheme B (cf p 4.2), no link is needed when passing through the R.C.S. If the routine has a false cue however, a line of the directory will be needed for each part brought down by a separate transfer. In the simple case of entering a subroutine from a master routine, the sub-routine usually starts with the instruction VSTA which places the link in VS, and ends with the instruction NS/P which transfers control to the R.C.S. If, however, either the sub-routine has more than two pages, or else has a sub-routine of its own, VS will be used for one or more cues before the link is required. In this case the link must be stored where it will not be interfered with by the sub-routines, say in line 'p'. The first instruction of the sub-routine will then b e 'p' T A and at some stage before the final NS/P the instructions 'p' T /, VSTA must be inserted.

Variable Sub-routines.

Sometimes a routine will have a variable or undetermined subroutine, i.e., one which is not decided until after the routine itself has been completed. This occurs for instance when an undetermined function is involved, e.g., in calculating $\int f(x)dx$. In such cases the function is determined by a sub-routine, and the subroutine is given through its cue. These variable sub-routines will be subject to numerous restrictions imposed by the master routine, e.g., certain store lines, used by the master routine must not be used by the sub-routine, and the result must be given in a specified form. The account of the master routine must also specify its own principal lines at the time of entering the sub-routine, in order that the cue of the latter may be determinable. These principal lines will be presumed to be the same as applied on entering the master routine unless otherwise specified.

The Official Account of a Routine.

The official account of a routine is designed to provide the user with all the information he needs to inco rporate the routine in his programme. This information is usually presented in the form shown in Fig. 3.2, and is divided into the following sections:-

1. Name of Routine. This name, which may be punched on the tape for

purpose of identification, is generally about 10 characters long and in any case must not exceed 29. It should give some indication of the purpose of the routine.

2. Purpose. This should contain a short description of the object of the routine.
3. Cues. Where there are alternative cues, they should all be given here.
4. Sub-routines. The names of any sub-routines should be given. If a variable sub-routine is to be used a remark should be made to this effect.
5. Principal Lines. The contents of either /E or /A or both on leaving the routine, should be stated.
6. Tapes. The system of naming tapes will be described later (p.3.14) and the names of all tapes needed to input the routine should be given.
7. Magnetic Storage. For routines with true cues the actual track numbers can be given here, otherwise the word 'variable' should be entered.
8. Electronic Storage. This should give the numbers of the electronic stores in which the routine operates.
9. Stores Altered. All store lines altered in the course of the routine with the exception of those in stores 0 or 1 or in the special working space GK-£K (see p.3.20) should be listed here. Also all B-lines altered apart from B7.
10. Effects. The effects of the routine should be described accurately in so far as they are known or considered to be of interest. These will often consist of equations or inequalities relating the states of the machine immediately before entering and immediately after leaving the routine. Conditions of validity, accuracy of results etc., should be included, also some statement of the time taken. Some account should be given of the method used if this is not obvious and unusual tricks should be pointed out.

The INPUT Routine.

The first four tracks of the magnetic store contain routines which are considered vital to the successful operation of the machine.

As a precaution against their content being accidentally altered by faulty writing transfers, these tracks are isolated, i.e., the wiring is so arranged that writing transfers to them are impossible while reading transfers are unaffected. Track 0 contains the routines INITIAL and ROUGHWRITE (see $\overset{3.15}{\text{p3.18}}$), track 1 contains the two parts of PERM and tracks 2 and 3 contain the INPUT programme. This latter provides the standard method of reading routines and certain other information from tape into the machine. Its effect is to scan the tape in the tape reader, distinguish certain sequences of characters and treat these in particular ways depending on their first or warning character.

The INPUT programme can be entered by setting the hand-switches to A///, pressing the key marked KEC and then switching on the completion signals. The KEC key clears all the electronic stores including the control tube and consequently when the completion signals are switched on the first instruction to be obeyed will be that in line E/. Since this line is also empty this instruction is //// i.e., 'obey H as a magnetic transfer' which brings the left half of track 3 to Store 0. One page of INPUT is now in the electronic store and the control takes the instructions from this routine, starting at line @/. The following instructions cause the second page of INPUT to be brought to Store 1 and then the two pages of PERM to be brought to Stores 2 and 7. Having effected these magnetic transfers the reading of the tape is started. INPUT may also be entered by setting [H] = //// instead of A/// (see INITIAL p 3.18) or else entered in the normal way by a cue, in which latter case PERM is not brought down again.

Warning Characters.

For the purpose of the INPUT programme certain characters i.e., J, K, Z, W, Y, Q, ", X, are treated as warning characters i.e., are used to designate the start of a meaningful sequence. When the tape is placed in the reader and the INPUT routine started, everything on the tape is ignored until the first warning character is reached. A number of subsequent characters are then read and treated in some particular way, the length of the sequence and the nature of its treatment depending on the value of the warning character as described below.

J. The length of the associated meaningful sequence is eleven characters. If the last character has 0 for its last digit, two lines of store 4 are altered, namely the two lines of columns $\frac{1}{2}$ and D whose numbers are given by the 6th character of the sequence. The second, third, fourth and fifth characters are copied into the line in column $\frac{1}{2}$ while the seventh, eighth, ninth and tenth go into the corresponding line in column D. For example, the effect of the sequence JABCDUFGHIJ is to put ABCD in line U$\frac{1}{2}$ and FGHI in line UD. This apparently somewhat involved process is designed to make the punching of material as simple as possible. It is only necessary to imagine a page of material flanked with a column of Js on either side and then punch each row straight across. If the last digit of the last character is a 1 the sequence has no effect. Thus if a mistake in the punching is detected before the second J is punched the line can be cancelled by replacing this J by one of the characters T, Z, ... £. If the sequence is correct the last character can be any of /, E .... K, but J is usually chosen for ease of punching.

K. This character causes information to be copied from the tape into consecutive lines of the electronic store. The second and third characters give the address of the first line to be filled, the fourth gives the number of such lines, while the following characters give the content of these lines, the total length of the sequence being four, plus four times the number of lines. Thus KVS@J@LVA/@/ places J@LV in line VS and A/@/ in line £S. Unlike the warning character J, K does not restrict the input to store 4. Naturally neither the INPUT programme nor PERM must be interferred with, but otherwise any page may be written into. The fourth character must not exceed 17, and therefore to read in a full page of information four such sequences will be necessary.

Z. This is a single character sequence and has the effect of transferring control to the R.C.S., i.e., calling in whichever routine has its cue in VS. The quantity held in HK when Z is read, is treated as the link.

W. This is a single character sequence whose effect is to stop the reading of the tape and enter a loop in Store 0.

Y. This is another single character sequence and is similar in its effects to Z. Control is transferred to the R.C.S. and the routine whose cue is in VS is entered, but in this case the cue to INPUT is used as link instead of [H K].

Q. The second character gives the length of the sequence, reduced by two. The effect is to print (on a new line) and/or punch all the characters of the sequence, including the first two. Its main purpose is to enable titles (or other identity symbols) of input tapes to be recorded on the printer during the input process. For example the effect of the sequence

QKSCOTLANDFOREVER

is to print and/or punch these same symbols.

". This is used to input integers in decimal form. The length of the sequence is the shortest consistent with being at least of length four and ending with P, M, or £. The second and third characters specify an address (line pair) in the store. The remaining characters up to the first occurance of a P, M, or £ are treated as a decimal integer provided that they are all chosen from /, E, @, A, :, S, I, U, ½, or D; otherwise the tape is considered to be incorrectly punched and a hoot occurs. The last symbol if P(lus) or M(inus), specifies the sign of the integer and the effect is to transfer the first 40 digits of the (signed) integer to the line pair specified. If the last character is £, then there is no effect. For example the effect of the sequence

"/½I@SP is $[/\ \tfrac{1}{2}]_{+}$ = ·625

and that of "/½I@SM is $[/\ \tfrac{1}{2}]_{\pm}$ = -625.

The sequence "/½I@S£ has no effect.

X. This is a five character sequence of which the last four characters form an instruction which is obeyed shortly after reading the last character of the sequence. Before doing so, however, the accumulator is filled from the long lines HK and PK, the former filling the least significant half. After the instruction has been obeyed the accumulator is emptied back intothese lines which are consequently referred to as the pseudo-accumulator. (Hence too, the choice of the content of HK as link when using the warning character Z). The orders which lead to the obeying of this instruction are as follows:-

```
V K Q O     Place instruction in B7
H K T / }
P K / J }   Fill accumulator
/ / U /     Obey instruction
H K / U }
P K / U }   Fill pseudo-accumulator.
```

Since the instruction is obeyed in the form of the B-modified order //U/ it is obvious that it cannot itself be B-modified. Also if it is one of the special B instructions TO, TB, TG, etc., it will always refer to the B line with number one less than that specified. Thus XDSQO will set [B6] = ££££. Instructions following an X cannot refer to B7. X cannot be used for multiplications since the setting of the multiplicand is not retained.

The characters J, K, Z, W, Y, Q, ", and X only have the properties of warning characters when they occur at the start of a meaningful sequence. If they occur as any of the characters from the second to the last of such a sequence they have their ordinary significance. Once such a sequence has been read the INPUT programme ignores all characters on the tape until another warning character is reached. If an error occurs in the punching of a tape it is sometimes possible to make a sequence ineffective by punching £ in place of the warning character but if this is done care must be taken that the subsequent characters of the sequence are not themselves liable to be read as warning characters. e.g., if in the sequences XDSTJXVKTA, the first X were replaced by £, then the J would be treated as the next warning character and XVKT would be read into line $A\frac{1}{2}$ etc., possibly throwing out the whole of the subsequent reading of the tape.

Rough Tapes.

When a routine has been coded, it is punched in the form of one or more rough tapes, these being a temporary form of storage, later to be superseded by the corresponding writing tapes when the routine has been tried on the machine and any errors eliminated. One rough tape is punched for each page of the routine, and has the effect of transferring the content of that page from the tape to the electronic Store 4 and thence to the appropriate magnetic track. The punching on a rough tape consists of the following four parts:-

1. Titling sequence.
2. Destination sequence.
3. Punching proper.
4. Final sequence.

1. The Titling Sequence.

This consists of Q n Name of the tape. Where n is the number of characters in the name. As the library routines have names for identification purposes, so do the tapes on which they are stored. In general the name of the tape is the name of the routine to which it refers, followed by one, two etc., according to the number of the page, and ending with the word ROUGH if it is a rough tape. There are, however, exceptions to the above and hence the necessity for entering the names of the tapes in the official account of a routine. e.g., the library routine RECIPROOT occupies one and a half pages, while the other half page contains EXAPP. The rough tapes for the former routine are RECIPROOT ONE ROUGH and RECIPROOT TWO ROUGH while that for the latter is RECIPROOT TWO ROUGH. The titling sequence enables a printed record to be kept of all tapes put in during INPUT

2. The Destination Sequence.

This sequence which is used in connection with the routine ROUGHWRITE (see p. 3.15) determines the magnetic track into which the content of the tape is to be written. It takes the form:-

KAK@///$\frac{1}{2}$ (Magnetic half cue) if the tape corresponds to a left hand electronic page.

or KAK@//E$\frac{1}{2}$ (Magnetic half cue) if it corresponds to a right hand electronic page.

In general the magnetic half cue consists of the last four characters of the cue whether this be true or false. If however the tape corresponds to a page which is not brought down by the cue, but by a subsequent transfer within the routine, then it is this transfer which must be used in the destination sequence. It may happen occasionally that a page of information is used in both left and right hand electronic pages but in this case the column numbers on the programme sheet will indicate whether the left or right hand use is to be considered. Also if there are several possible cues, the one chosen to form part of the destination sequence must be that to which the numbering of columns on the programme sheet is applicable.

3. The Punching Proper.

This is the information which is to be read into the electronic store prior to copying it into the magnetics. It may be punched with either J or K as warning character but if K is used, the third character of each sequence must be ½ or D so that the content of the tape is always read into Store 4.

4. The Final Sequence.

On rough tapes this is always KVS@//////A/YW. //////A/ is the cue to ROUGHWRITE which, as has been mentioned above, is kept in the isolated track 0. This routine forms the correct writing transfer from the destination sequence, if necessary using the directory, and then writes the content of Store 4 onto the appropriate magnetic track. The sequence KVS@//////A/ places the cue in VS and then Y transfers control to the R.C.S., which calls down ROUGHWRITE and enters it with the cue to INPUT as link. After the magnetic writing transfer has been carried out, INPUT is re-entered and the W on the tape is read, causing a loop stop on Store 0 i.e. stopping the tape.

In addition to the four parts of a rough tape as described above it is usual to start all tapes with a number of Cs and end with a number of £s to give a visible means of distinguishing between the starting and finishing ends.

Setting the Directory.

ROUGHWRITE will deal with routines having false cues provided that the directory is correctly set first. This can be done by punching a rough tape for the directory in the following form:-

```
QDDIRECTORY
KAK@///½EE//
Punching proper
KVS@//////A/YW
```

This tape must be read in before any of the other rough tapes.

Writing Tapes.

Once a routine has been checked on the machine it can be punched as a writing tape which is the standard form of storage for library routines. These tapes take the form:-

```
Titling Sequence
Destination Sequence
KPK@  Check sum
Y
Punching proper
Z.
```

The titling sequence is similar to that for the corresponding rough tape but has the five characters ROUGH omitted and consequently n is reduced by 5. The destination sequence is the same as for the rough tape. The check sum which occurs in the sequence following the destination sequence, is the sum mod 2^{40} of all the long lines of the punching proper and is placed in lines PK and QK. This sequence does not appear on a rough tape since it is used as a check by WRITE (see below) but not by ROUGHWRITE. The punching proper is the same as for the rough tape apart from any corrections that may have been necessary. The Z at the end of the tape replaces the final sequence and provided VS has not been altered, has the effect of entering WRITE via the R.C.S., since the cue of this routine is in lines VS and £S of PERM as stored in the magnetics.

WRITE

This routine whose false cue is held in line VS of PERM has the same effects as ROUGHWRITE but incorporates checks on the reading of the tape and the magnetic transfers. When the punching proper has been read into Store 4, the sum of the long lines (Mod 2^{40}) is compared with the content of PK and QK. If these agree, the content of Store 4 is written onto the appropriate magnetic track as found from the destination sequence and this transfer is checked. As a final check, the content of the track is read back to Store 4 and the check sum formed again and compared with the content of PK and

QK. If any of these three tests fails, control enters a loop stop containing a high pitched hoot, while if they are passed it enters a low pitçhed hoot. There exists also another version of WRITE called WRITE/A which has the same cue and the same effects but re-enters INPUT if all the tests are passed instead of stopping at the low pitched hoot.

Punching Writing Tapes.

Usually writing tapes will be punched automatically from the correct version of a routine as held in the magnetic store. To punch these tapes WRITETAPE/A must be in the magnetics and the directory set correctly. Having done this the following tape is put in the reader:-

```
K A K @ (Destination sequence of tape to be punched).
K V S @ £ E W G : E E Z
Y
Titling sequence of tape to be punched.
Y.
```

The effect of this tape is to bring down and enter WRITETAPE/A which, treating the destination sequence in a manner similar to ROUGHWRITE brings down the appropriate magnetic half track to S4, forms the check sum and punches the following tape:-

```
/ / / / . . . . / / / /
C C C C C C C C
Titling sequence
/ / / /
K A K @ (Destination Sequence)
K P K @ (Check Sum)
Y / / /
Punching proper punched with warning character K.
/ / / / Z / / /
£ £ £ £ £ £ £ £
/ / / / . . . / / / /
```

If the titling sequence is omitted from the steering tape which will then end YY, it will be omitted from the writing tape also.

Compound Tapes.

In preparing a programme involving several pages it is not desirable to have each page on an individual tape, as apart from the danger of omitting one such tape, a great deal of time is wasted in putting the individual tapes into the reader. Hence we make compound tapes onto which are copied all the tapes needed for a programme. In such a case where there is no danger of overlooking part of the tape, the titles may be omitted in the copying. Such a compound tape will in general start with a rough tape to set the directory, followed by the rough tape for WRITE and will then consist of a series of writing tapes. If one writing tape fails on any of the checks in WRITE, the tape can be moved back through the reader by hand to the start of that section and then INPUT reentered by pressing KEC.when another attempt can be made.

Steering Tapes.

A further type of tape which may be mentioned here is the steering tape. This tape is used to start a computation or initiate some machine process. Its exact composition does of course depend on the nature of this process but in general it consists of a number of K sequences, setting parameters, inputting numerical data etc., followed by a sequence KVS@(Cue)Y which after setting the first cue of the programme, enters it via the R.C.S.

INITIAL. This routine is kept in Store 0 together with ROUGHWRITE. Its purpose is to record the start of a computation when working in what may be termed the Formal mode where a printed record is kept of all action taken during the operation of the machine. It is entered by setting [H] = //// and pressing KEC before switching on the completion signals. When entered it causes the printing ++INITIAL-- and then enters INPUT.

Conventions.

The machine has a very great flexibility. Although this has obvious advantages, it has also certain disadvantages which can become serious unless precautions are taken. It is for instance possible to alter the whole content of the electronic and magnetic stores merely by putting an appropriate tape into the input.

Although we may be often glad of this fact it increases the possible damage which can be caused by mistakes. The remedy for this kind of difficulty lies in the introduction of conventions. These are in effect decisions to restrict the freedom of flexibility of the machine in various ways. It is hoped that the loss of flexibility will be fully compensated for by the advantage of the resulting reduction of the uncertainty of the state of the machine. The conventions are mostly not to be regarded as absolute commands or prohibitions, but rather as normal procedure, any deviation from which must be noted in the descriptions of the routine in which they occur.

The whole of what has been said about programming is in fact an elaborate convention regarding the use of the electronic store and the user can ignore it altogether if he considers it desirable. The conventions should not be regarded as pure tyranny, but to know that they have been obeyed in the programmes one is using is a great comfort. Moreover they enable one to reduce considerably the lengths of official accounts of routines, since they allow a great deal to be taken for granted.

Use of Electronic Store.

When Scheme A was originally devised it was feared that only 5 electronic stores might be available. This has fortunately not been the case but it appeared probable at the time. In laying down the conventions for operating the machine and in deciding which 5 stores should be used, the following considerations were taken into account:

(a) The choice of the five pages must be convenient if there were only five available, but must also be convenient if there were six, seven or eight.

(b) The routines must be restricted to relatively few pages so as not to interfere with other forms of storage.

(c) Similarly used pages should preferably be partnered. This applies particularly to systematic working space and to the space used for routines.

(d) Systematic working space should if possible consist of consecutive lines.

(e) The powers of two must be consecutive with the space used for routines.

(f) The first few instructions after operating KEC are taken from column /.

As a result it was decided to take the 5 stores numbered 0, 1, 2, 4 and 7, and to allocate their use as follows:-

Stores 0 and 1 would normally contain routines, i.e., instructions and auxiliary fixed numbers etc. The transfers used in the R.C.S., would normally be to one or both of these stores.

Store 2 would contain PERM i.e., the powers of 2 and the R.C.S. If it was desired to extend the powers of 2 for a particular routine $2^0 - 2^{19}$ could be in lines O@ - £A of store 1 i.e., continuous with the powers of 2 in PERM.

Store 4 would be used for systematic working space i.e., could be used for tables or other systematically stored material. Usually Input would be into this store.

Store 7 would contain the parts of the R.C.S. relating to false cues, the remainder of the page being used for unsystematic working space i.e., in general unrelated long lines.

The other pages if and when they became available could be used as follows:-

Store 3 could be used either to extend PERM or as further systematic or unsystematic working space, or for a combination of these.

Store 5 would be available as further systematic working space being partnered by store 4.

Store 6 would be available for either systematic or unsystematic working space.

In practice all 8 stores are available but in general their use is as outlined above. The lines GK, MK, VK are generally used as special short term working space i.e., to contain quantities which are no longer of interest once the routine is finished. Lines MK and VK are in any case used in the routine changing sequence. It will be seen that the presence of these three lines and the part of PERM on Store 7 makes this page useless for systematic working space. If further short term working space is required one may use those lines of Stores 0 and 1 which contain instructions which will not be obeyed again before they are wiped out by a magnetic transfer.

4. Replacability conventions.

It is essential for the possibility of programming at all that t the properties of the machine should hardly ever be changed. There are however certain features of the machine which can be reasonably be described as 'disadvantageous'. It is desirable to leave open the possibility that these features might at some time be removed. This suggests the convention that *no disadvantageous feature of the machine should be used in a library routine*. This requires some further definition. It is understood in connection with each such feature that it is known what modification improving the machine is contemplated. The routines must work whether such a modification has been introduced or not. The features at present recognised as disadvantageous are mentioned below.

a) Certain functions are not considered of particular value and rated as 'foul'. These are /B, /", /X, /£, TE, TS, TU, TH, TY, TP, TQ, TM, TX, TV. The modifications envisaged are the changing of these to other operations, so that the convention amounts to the avoidance of their use.

b) The exceptional nature of the line pairs at the end of a page (see p. 1.6) is disadvantageous. The modification would consist in bringing these into line with the other line-pairs. The use of such line-pairs must be avoided.

Conventions when using Scheme A.

When working with Scheme A the following conventions are adopted:-

(i) It is assumed that PERM is in position in Stores 2 and 7 i.e., that the powers of 2 and the R.C.S. are available for use. Routines are assumed to be entered by their cue via the R.C.S., the link being held in the least significant half of the accumulator.

(ii) Instructions are in general confined to Stores 0 and 1 and in order to avoid confusion in the forming of the check characters lines /E and /A are never used as working space.

(iii) In the official account of a routine, mention is made of any lines that are altered, except for those in Stores 0 or 1 and the special working space GK - £K. Similarly, mention is made of any B lines that are altered apart from B7 which is itself altered by the R.C.S.

(iv) Where possible the higher numbered B lines are to be used. If B0 is used it must be clear on leaving the routine.

(v) Disadvantageous features should not be used.

(vi) Magnetic tracks 0 - 3 are isolated i.e., it is impossible to change their content. (see p.3.10).

Track 32 is special working space i.e., it can be altered in the course of a routine without being mentioned in the official account.

Track 33L contains the directory.

Strategical Considerations.

When this chapter has been read and understood the reader will be in a position to appreciate the remarks under the above heading at the end of the next chapter (see p.4.14). Since these remarks apply equally to the two Schemes they are not reproduced here.

PERM & The Routine Changing Sequence.

:		S	C		K
/ / / /	/	/ / / /		/	
E / / /	E	/ / / @		E	
/ / / /	@	/ / / /		@	
@ / / /	A	/ / / :		A	
/ / / /	:	/ / / /		:	
: / / /	S	/ / / ½		S	
/ / / /	I	/ / / /		I	
½ / / /	U	/ / / T		U	£ £ / /
/ / / /	½	/ / / /		½	H : / C
T / / /	D	£ £ £ £		D	V S / F
/ / / /	R	£ £ £ £		R	L : / J
/ E / /	J	/ / / /		J	M K / A
/ / / /	N	F S / V		N	X K / :
/ @ / /	F	K S / P		F	£ S T /
/ / / /	C	V K T A		C	U K T R
/ : / /	K	J S / L		K	M K / S
/ / / /	T	V S T F		T	M K Q O
/ ½ / /	Z	C K / H		Z	/ / Q T
/ / / /	L	£ S / :		L	£ S Q B
/ T / /	W	/ E T /		W	F S / P
/ / / /	H	/ A T N		H	
/ / E /	Y	M K / S		Y	Pseudo-
/ / / /	P	Y : / C		P	accumul-ator
/ / @ /	Q	M K / N		Q	
/ / / /	O	V S T J		O	
/ / : /	B	R S T R		B	
/ / / /	G	Y : T N		G	
/ / ½ /	"	K S / H		"	Special
/ / / /	M	V K T /		M	Working
/ / T /	X	V S / P		X	Space
/ / / /	V	Cue		V	
/ / / E	£			£	

The routine changing sequence is entered by the instruction N S / P.

Fig. 3.1.

MANCHESTER UNIVERSITY COMPUTING MACHINE LABORATORY.

Programme Sheet 1.

Name of Routine. COSFAST.

Date. 8.10.51.

Purpose. To calculate cosines.

Cues.

££..U///

Sub-routines.

——————

Principle Lines.

$[/\,E]_{0}^{19}$ = V K / F

Tapes.

COSFAST ONE.

Magnetic Storage.

7 L

Electronic Storage

S 0

Stores Altered.

/C, EC, :C, SC.

Effects.

Initially $[/\,C]_{\pm\frac{1}{2}} = \alpha$.

Then $[:\,C]'_{\pm\frac{1}{2}} = \frac{1}{2}\cos 2\pi x + \Theta\,(14 \times 2^{-40})$.

Method. (1). $x = \frac{\pi 2 \alpha 2}{8}$

(2). $u = 1 - \cos\sqrt{2x}$. (by power series).

(3). $v = 2u - u^2 = \frac{1}{2}\,(1-\cos 2\sqrt{2x}\,)$.

(4). $\frac{1}{2}\cos 2\pi\alpha = (2v-1)^2 - \frac{1}{2}$.

Note.

If $[/\,C] = 0$ $[:\,C]'$ = M££££££K.

if $[/\,C] = \pm\frac{1}{2}$ $[:\,C]'$ = ///////T.

Time. Approx .05 sec.

Fig. 3.2

Chapter 4.

Programming (cont): Scheme B.

The size of a routine.

In this scheme for handling routines, it is a necessary requirement that each routine should not occupy more than one track of the magnetic store, i.e., each routine shall contain 128 instructions or less. However mathematical processes do not arbitrarily limit themselves to any particular size and if the direct coding of the process results in more than 128 instructions then one of the following two plans must be adopted. The process could be further broken down into yet simpler processes and the whole coded as a master routine with one (or more) sub-routine The other plan is to arrange the process as two ad-routines (see next section).

Organisation of routines: the routine changing sequence/B.

It has been explained how a complete programme is organised by splitting it up into bits called routines. In the next few sections we describe a scheme for arranging that the various routines are used in the correct sequence.

All the routines of a programme are stored in the magnetic store, each routine occupying, at the most, one track. Each routine is transferred from the magnetic store to the electronic storage S0, S1 when required, overwriting the previous routine which called it in. When the routine has been read into S0, S1 in this way then control is transferred to its entry instruction (which is not necessarily the first). Thus only one routine stands in the electronic store at any stage of the programme and, if not a self contained programme, must contain the instructions necessary to replace itself by another routine. It is intended that certain material shall be permanently available in S2. This includes the powers of 2 and the routine changing sequence/B (see below). All the rest of the store with the exception of the first few lines of S3 is available as working space for the routines.

Sub- and ad-routines.

In this section we discuss the possible circumstances under which it may be necessary to replace one routine by another. There are two important cases.

1. At some point in the sequence of instructions of a routine A it may be necessary to break off and carry out the instructions of another routine B. After obeying the instructions of B, i.e., after routine B has carried out its task, it is required to resume the obeying of instructions of routine A at the point immediately following that at which it was left. This relationship is described by saying that routine A uses routine B as a sub-routine.

2. In this case it is not required to return to routine A after completing routine B. Under this circumstance routine A is said to use routine B as an ad-routine (adjacent routine).

The essential feature of the first case is that certain information specifying routine A, i.e., the location of A within the magnetic store, and the point at which it was left must be recorded and preserved until routine B has completed its task, when this piece of information - called the link - is made use of to return control to A . If routine B itself uses a sub-routine then 2 links must be preserved until they are needed. And so on. In case 2 no link is involved in the change of routine.

Levels of organisation.

We shall speak of routines being organised on separate levels. Thus the ad-routines constituting the master programme define the zeroth level. If one of these routines uses sub-routines, then these define the first level. If these sub-routines in turn use other sub-routines, then the latter will lie on the second level, and so on. It is very seldom that a problem spreads over more than 4 or 5 levels. The accompanying block diagram (fig 4) of a problem illustrates the concepts already introduced and in addition two associated concepts - those of open and closed routines - which are explained below.

The problem has been split up into 10 routines numbered 0 - 9. There are 4 levels. Routine 0 calls in routine 3 as a sub- (closed) routine. A similar relationship holds between 2 & 6, 2 & 7, and between 8 & 9. Routine 1 calls in routine 4 as a sub- (open) routine

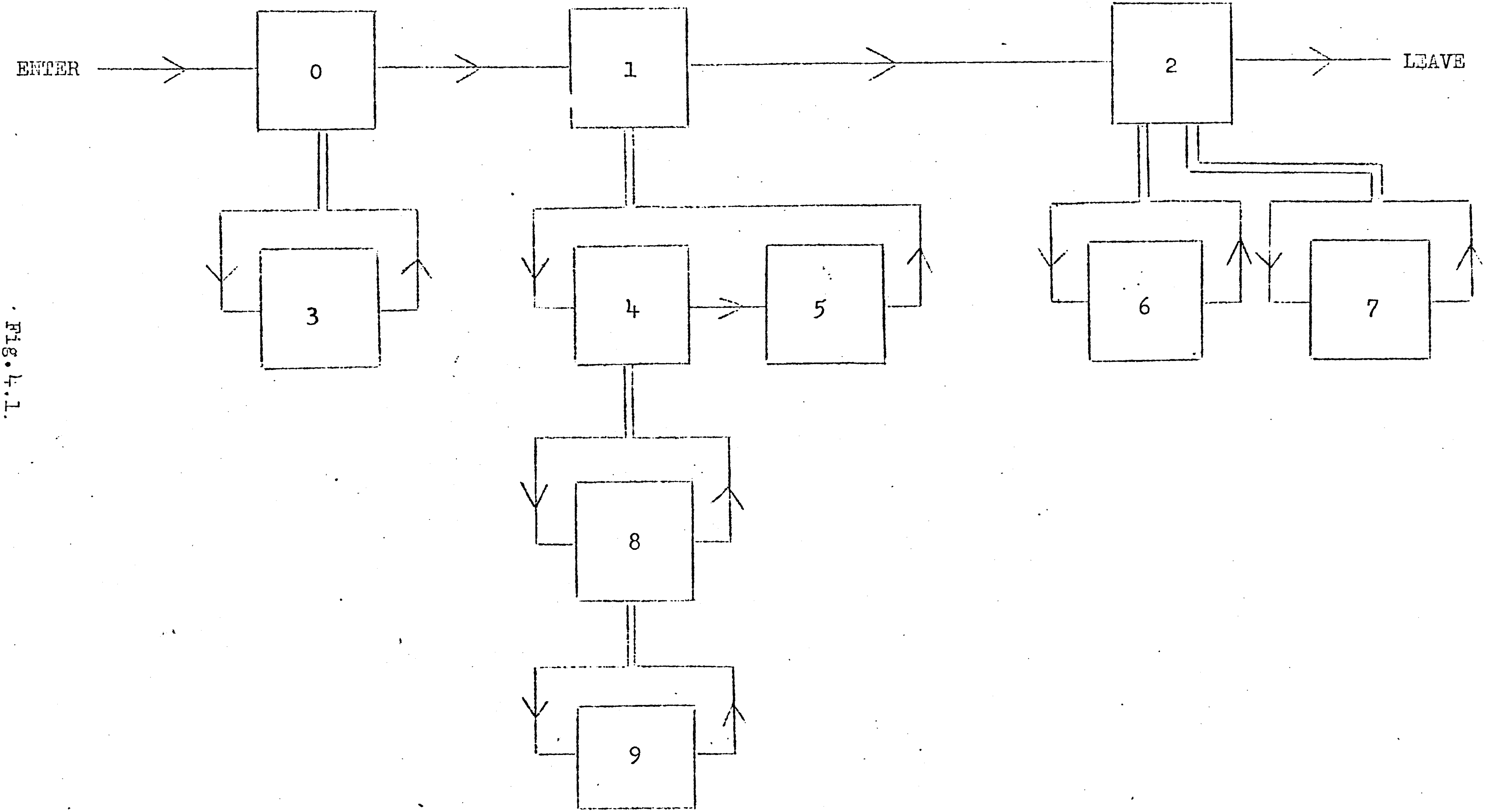


Fig.4.1.

routine 4 calls in routine 5 as an ad- (closed) routine, and routine 0 calls in routine 1 as an ad- (open) routine.

Open and closed routines.

We now define more clearly the terms *open* and *closed* which have been used tentatively above. An *open* routine is designed so that when its task is completed it calls in another routine *on the same level*. A *closed* routine is designed so when it finishes control reverts *to* a routine on *the previous level* at a point immediately following that at which it was *last* left. It should be emphasised at this point that the opening or closure of a routine is a property of the routine i.e., it depends on the design of the routine, whilst the sub- or ad- prefixes describe only how the routine is used. Thus it is possible to use either open or closed routine as both sub- or ad- routines in different contexts, as in the above example.

Cue of a routine.

With each routine of the programme is associated a line pair of information, called its *cue*, made up as follows.

(a) digits 0 - 9. This is the control transfer number, i.e., 1 less than the name of the line in which the routine starts.

(b) digits 10 - 19. These are irrelevant.

(c) digits 20 - 39. This is the magnetic instruction which would bring the routine to electronic pages 0, 1.

Example.

(a) A / / / F / / /

Routine held on track 13. Starts with an order in :/.

A given track of instructions may have several entry points so that several routines may be associated with the same track, i.e., have identical magnetic half cues.

The Cue Directory.

When the problem has been split up into convenient steps the corresponding routines are enumerated 0, 1, 2, etc., i.e., each distinct routine is given a number. Routines corresponding to distinct points of the same page of instructions count as distinct routines for this purpose. An ordered list of cues, called the *cue directory* is then prepared. This list is held on a fixed track of the magnetic store (34 at the time of writing) and is brought down to S0, 1, for reference purposes when a routine is changed. The list must be drawn up so that the cue of routine 0 stands in lines / /, E /, the cue of routine 1 stands in lines @ /, A /. And so on.

The routine changing sequence/B

The routine changing sequence of instructions (denoted henceforth by R.C.S/B) occupies lines NS to £S inclusive of S2 and, together with the powers of 2, is intended to remain permanently available in S2 throughout the duration of the problem. The first few lines of S3 are also permanently used by R.C.S/B.

The purpose of R.C.S/B is to enable routines to be changed without having a lot of preparatory bother in the routine that is being left. A scheme which necessitates the fewest possible instructions - 3 lines - in the main routine has been adopted. This is possible because the task of recording and preserving a list of links is performed by R.C.S/B. When a sub-routine at any level is called in the link back to the original (i.e., the calling in) routine on the previous level is added to the list. The links in this list thus constitute a chain of control extending back to the zeroth (or master) level. The length of the list measures the level of the current routine. The level j is recorded in B1 thus: [B 1] = 2j+2, I /. (As a consequence of this B1 can only be used within a routine if this parameter is temporarily stored elsewhere and replaced before using R.C.S/B to change a routine). The list of links is kept in the line pairs / I, @ I, : I, etc., that to the master level from the 1st level standing in / I, that to the 1st level from the second level standing in @ I and so on. B7 is altered by R.C.S/B.

In the routine being left the instructions necessary to call in the nth routine as a sub-routine and as an ad-routine are respectively:

```
s    |  s, Q 0          (s = 2 characters)
s+1  | G S / P         (2n = 2 characters)
s+2  | 2n, / V
```

```
s    |  s, Q 0
s+1  | Q S / P
s+2  | 2n, / V
```

Consider the first group. The effect of the instruction s, Q O in location s is to substitute itself into B7. This will form part of the link. The next instruction G S / P transfers control to a point within R.C.S/B. The subsequent events are, roughly, as follows. The content of B7 is changed to s+2// and this - the control number for the return - is used to replace the control number in the cue of the current routine. By means of certain shunting operations the 3rd. line of the group, 2n, / V is placed in B7. The directory is then called down from 34 to S0, S1 (A useful variation of R.C.S/B is one which provides for the directory or a suitably large fraction of it, to be permanently available in the electronic store on page 3 or 4 say). A B7 modified instruction is then used to select the cue of the new routine from the line pair 2n of S0. This cue is added to the list immediately below the cue (now the link) of the previous routine. The value of j given by B1 is then increased by 2. The magnetic half of the new cue is then obeyed as a magnetic instruction, bringing down the new routine to S0, S1. At this point a dummy stop order / G is encountered. This is a suitable point to use this facility if any visual observation of the monitors is contemplated. When this stop has been passed the control number part of the cue is used to transfer control to the routine .

The behaviour of R.C.S/B when used to call in an ad-routine differs from the above in that the cue of the new routine is written over the cue of the routine being left, and parameter j is not altered.

Design of closed routines.

The essential feature of a closed routine is that when its task is completed it causes control to revert to the previous level. R.C.S/B enables this to be done simply by inserting the instruction

N S / P at the operational end of the routine. The operation of R.C.S/B under this circumstance is roughly as follows. The parameter j given by [B1] is decreased by 2. The previous cue in the list, the link to the previous routine, is used in the usual way to call down and re-enter the main routine. This will be at the line s+3, immediately following the instructions by which it was left.

Role of the accumulator in routine changing.

A further feature of R.C.S/B is that [A] and [D] are not altered when a routine is changed. Thus, e.g., a routine for finding $\sqrt{x}$ may be coded on the assumption that x is held in the accumulator.

R.C.S/B.

```
                             S| N | " S / L |  control number
                                | F | E E / / |  magnetic instruction: 33L to SO
Entry for ad-routine  --->      | C | Z S Q G |  [B7]' =  s+2, / /
                                | K | A : Z G |  adjusts  B1
                                | T | S : / Q |
                                | Z | V £ P O |
                                | L | M £ £ £ |  constants
Entry for sub-routine --->      | W | Z S Q G |  [B7]' =  s+2, / /
                                | H | V S Z Z |  plants link
                                | Y | / / Q T |  [B7]' =  2n, / V
                                | P | @ I E A |  plants M
                                | Q | F S / : |  calls down directory
                                | O | / / U F |  selects cue of new routine
                                | B | / I E A |  adds cue to list
                                | G | L S Z G |  adjusts [B1]
                                | " | V S E J |  restores accumulator
Entry from closed     --->      | M | A : Z G |  adjusts [B1]
      routine.                  | X | £ S E : |  calls down new routine
                                | V |     / G |  dummy stop
                                | £ | V S E P |  enters new routine
```

Notes.

1. The nth routine is called in as a sub-routine by the group of instructions:

```
s   |  s, Q O
s+1 | G S / P
s+2 | 2n, / V
```

and as an ad-routine by

```
s   |  s, Q O
s+1 | Q S / P
s+2 | 2n, / V
```

2. If the programme involves the j th level (master level $\equiv$ j = 0), then line pairs /I, @I, . . . , (2j)I are used by R.C.S/B. Line pair (2j+2)I is used as working space.

Example.

We describe in detail the programming of the problem suggested at the beginning of Chapter 3, namely, to calculate and print the function $e^{\frac{1}{2}\cos 2\pi x}$ for various values of x punched on the input tape. To make the problem more precise (and easier) we shall suppose that $\frac{1}{4} > |x| > \frac{1}{12}$. The reason for this limitation will become clear later. From the library of routines we select routines for input and printing processes and for the calculation of the exponential functions and of the cosine function. These routines together with a master routine (yet to be made) are enumerated and allocated magnetic storage and cues as in the following table.

0	5 L	Master	£ £ / / S / / /
1	5 R	PRINT/A	C E / / S / E /
2	6 L	INPUT/B	/ / / / I / / /
3	6 R	EXPONENTIAL/B	@ @ / / I / E @
4	7 L	COSINE/A	/ / / / U / / /

It is a convention that the master routine shall be the zeroth routine. The first half of the cue of each routine (the control number) is given in the official account of the routine. Abbreviated accounts of the above routines are given below.

Name of Routine. PRINT/A.

Purpose. To print, in fractional form, the number held in A.

Control Numbers.

££// E///	± convention.
CE// TE//	+ convention.

Magnetic Storage.

Variable.

Electronic Storage.

S 0.

Stores Altered.

None.

Effects.

1. When the routine is called in it causes the 80 binary digit number held in A to be printed as a fraction on the + or ± convention according to the cues used.

2. If the fraction is negative, a - sign is printed, otherwise a space is left before the number.

3. The style of printing is specified by a 40 digit line (the digit layout constant) which must be sent to D before the routine is called in. This digit layout constant is interpreted as follows:

Starting at the most significant end the digits of the number are examined in turn. If the rth digit (counting the most significant as the 1st) is 0, then the (r + 1)th character printed is the next decimal in the fraction; if a 1, thenthe (r + 1)th character printed is a space - provided that such 1's are isolated. Two consecutive 1's cause the last figure printed to be followed by 2 spaces after which the routine is left and control returned to R.C.S/B.

Examples.

(i) //A:½T/E causes the number to be printed as 5 blocks of 5 digits each (each block being separated by a single space).

(ii) /////A@@ causes printing in the style of 3 blocks, each of 4 digits. In this case the 1st 5 characters are irrelevant.

4. If the lesser control number is used in the cue, then a carriage return and line feed occurs before printing.

Name of Routine. INPUT/B.

Purpose. To read from the tape into the accumulator one signed fraction.

Control Number.

/ / / /

Magnetic Storage. Electronic Storage.

Variable. S 0.

Stores Altered.

None. (i.e., other than A, D, B7, or lines within the routine itself).

Effects.

1. Let x denote the fraction punched in decimal form and followed by sign, e.g., $-\frac{1}{3}$ is punched as 333333333333333-, where it is understood that the decimal point lies immediately before the first digit punched and that any number of digits may be punched - although only the first 23 will be treated. The effect of the routine can be described by the equation

$$[A]'_{\pm f} = x$$

2. The accumulator must be clear when INPUT/B is called in.

3. Accuracy: maximum error is 5.2^{-80}

4. Speed: about 50 characters per sec.

Name of Routine. EXPONENTIAL/B.

Purpose. A slow routine for calculating accurately the exponential function over a li ited range of arguments.

Control Number

@ @ / /.

Magnetic Storage Electronic Storage.

Variable. S 1.

Stores Altered.

$$[@C]\,[AC]\,[:C]\,[SC]$$

Effects.

(1) $[:\ C]' = \exp\,[L] - 1$
where the quantities are expressed in plus - minus fractional convention.

(2) Accuracy: maximum error = 2^{-37}.

(3) Time: approx. 0.5 seconds.

(4) Method: the repetitive sequence used is

$$z_{n-1} = z_n + z_n^2 / 2^n. \quad \text{(See Example 5.1).}$$

initially $z_{40} = x$: finally $z_2 = (\exp.\ x) - 1$.

(5) If $-\tfrac{1}{2} \leqslant [L]$ log 1.5 the result is within range; if log $1.5 \leqslant [L]$ log 1.5625 the result is given modulo 1.

Name of Routine. COSINE/A.

Purpose. To calculate cosines.

Control Number

////

Magnetic Storage. Variable.

Electronic Storage. S 0.

Stores Altered.

/C, EC, :C, SC.

Effects

Initially $[/\ C]_{\pm} = \alpha$.

Then $[:\ C]'_{\pm} = \tfrac{1}{2}\cos 2\pi\alpha + O\ (14 \times 2^{-40})$.

Method.

(1) $x = \dfrac{\pi^2 \alpha^2}{8}$

(2) $u = 1 - \cos\sqrt{2x}$. (by power series).

(3) $v = 2u - u^2 = \tfrac{1}{2}\ (1 - \cos 2\sqrt{2x}\ )$.

(4) $\tfrac{1}{2}\cos 2\pi\alpha = (2v-1)^2 - \tfrac{1}{2}$

Note.

If $[/\ C] = 0$, then $[:\ C]' =$ M££££££K.

if $[/\ C] = \pm\tfrac{1}{2}$, then $[:\ C]' =$ ///////T

The next task is to code the master routine. One form of this routine is given below with suitable annotations. These, together with the help of the information given in the above accounts, should be self-explanatory.

/	T :	clear accumulator (see note 2 of INPUT/B)
E	E / Q O	
@	G S / P	calls in INPUT/B: reads x from tape
A	: / / V	
:	/ C / A	
S	S / Q O	
I	G S / P	calls in COSINE/A: $\left[: C\right]'_{\pm f} = \frac{1}{2}\cos 2\pi x$
U	½ / / V	
½	: C T /	
D	D / Q O	
R	G S / P	calls in EXPONENTIAL/A: $\left[: C\right]'_{\pm f} = e^{\frac{1}{2}\cos 2\pi x}_{-1}$
J	I / / V	
N	: C / J	prepare to print function value.
F	Z / / C	set digit layout constant
C	C / Q O	
K	G S / P	calls in PRINT/A: prints value
T	@ / / V	
Z	D S / P	return to read next number
L	/ O / E	digit layout constant

Conventions.

Tho machine has a very great flexibility. Although this has obvious advantages, it has also certain disadvantages which can become serious unless precautions are taken. It is for instance possible to alter the whole content of the electronic and magnetic stores merely by putting an appropriate tape into the input. Although we may be often glad of this fact it increases the possible damage which can be caused by mistakes. The remedy for this kind of difficulty lies in the introduction of conventions. These are in effect decisions to restrict the freedom of flexibility of the machine in various ways. It is hoped that the loss of flexibility will be fully compensated for by the advantage of the resulting reduction of the uncertainty of the state of the machine. The conventions are mostly not to be regarded as absolute commands or prohibitions, but rather as normal procedure, any deviation from which must be noted in the descriptions of the routine in which they occur.

The whole of what has been said about programming is in fact an elaborate convention regarding the use of the electronic store and the user can ignore it altogether if he considers it desirable. The conventions should not be regarded as pure tyranny, but to know that they have been obeyed in the programmes one is using is a great comfort. Moreover they enable one to reduce considerably the lengths of official accounts of routines, since they allow a great deal to be taken for granted.

1. Further conventions regarding the use of the electronic store.

A routine shall, as far as possible, confine any special working space to S0 and S1, overwriting some of its own instructions if necessary (they will automatically be reset when the routine is recalled from the magnetic store). Thus, e.g., routines intended to calculate a function of a single argument are best designed so that the argument is held in the accumulator when the routine is entered and replaced by the function value when quitted, any incidental working space, i.e., shunting stations and dustbins, being confined to the routine itself. If two arguments are involved, as e.g., would be the case in a routine designed to calculate $^x/_y$, then the second one can be held in the D register. If the routine involves parameters, e.g., the number of terms to be used in a series, then these should be placed in B lines.

If in spite of these devices it should be necessary to use other lines of the store for special working space, then the last few lines of column K may be used.

The routine changing conventions which have been described amount to the use of S2 for the strictly permanent information - the powers of 2 and the routine changing sequence - and the first few lines of S3 for the list of links. It is intended that the remaining lines of S3 shall also be restricted to more or less permanent information. The following important uses of these lines are envisaged.

Electronic cue directory.

A trivial modification of the routine changing sequence enables the programmer to arrange for some or all of the directory of routine cues to be held permanently in S3, thus eliminating the necessity for bringing it down from the magnetic store each time a routine is changed. This is a useful time economy measure.

The modification consists of replacing the instructions in lines QS and OS by FST£ and ..UF respectively, where the dots denote a line pair in S3, the zeroth entry of the directory.

Frequently used routines.

A further time economy measure is to use the remaining lines of

S3 for one or more frequently used routines. Of course if the routine is specifically designed for S0, then some instructions may have to be altered, either because they are not self-resetting or because certain control numbers would be wrong.

The routine changing sequence can be used to call in such routines held either in S3 or else where in the electronic store, simply by using a dummy special magnetic instruction (..£.) as the magnetic half cue of the routine. As a time economy measure this schème is mostly suitably used with an electronic directory.

2. B-tube conventions.

The conventions concerning the use of the B-tube are

(1) At the end of a routine B0 = //// unless otherwise stated. Indeed any alteration of B0 require special mention in the official accounts.

(2) Where a choice of B lines is available the higher numbered lines are to be preferred.

(3) A lterations of B lines other than B7 must be mentioned.

3. Conventions regarding the use of magnetic storage.

So far these amount to the use of 34L for the DIRECTORY and 98 for the two pages of permanent material - PERM - which it is intended to store in S2 and S3. More will be said in this connection in the next section.

4. Replacability conventions.

It is essential for the possibility of programming at all that the properties of the machine should hardly ever be changed. There are however certain features of the machine which can be reasonably be described as 'disadvantageous'. It is desirable to leave open the possibility that these features might at some time be removed. This suggest the convention that no disadvantageous feature of the machine should be used in a library routine. This requires some further definition. It is understood in connection with each such feature that it is known what modification improving the machine is contemplated. The routines must work whether such a modification has been introduced or not. The features at present recognised as disadvantageous are mentioned below.

a) Certain functions are not considered of particular value and rated as 'foul'. These are /B, /", /X, /£, TE, TS, TU, TH, TY, TP, TQ, TM, TX, TV. The modifications envisaged are the changing of these to other operations, so that the convention amounts to the avoidance of their use.

b) The exceptional nature of the line pairs at the end of a page (see p. 1.6) is disadvantageous. The modification would consist in bringing these into line with the other line-pairs. The use of such line-pairs must be avoided in library routines.

Strategical considerations.

When the programming example given above has been understood then the reader will be better able to appreciate the following remarks which deal mainly with the higher strategy of programming but include hints on the coding of individual routines.

i) Make a plan. This rather baffling piece of advise is often offered in identical words to the beginner in chess. Likewise the writer of a short story is advised to 'think of a plot' or an inventor to 'have an idea'. These things are not the kind that we try to make rules about. In this case however some assistance can be given, by describing the decisions that go to make up the plan.

a) If it is a genuine numerical computation that is involved (rather than e.g., the solution of a puzzle) one must decide what mathematical formulae are to be used. For example if one were calculating the Bessel function J_0 (x) one would have, amongst others, the alternatives of using the power series in x, various other power series with other origins, interpolation from a table, various definite integrals, integration of the differential equation by small arcs, and asymptotic formulae. It may be necessary to give some small consideration to a number of the alternative methods.

b) Some idea should be formed as to the supply and demand of the economic factors involved. A balance must always be struck between the following incompatible desires

To carry the process through as fast as possible

To use as little storage space as possible

To finish the programming as quickly as possible

We may express this by saying that machine time, storage space, and programmer's time all cost something. The plan should take this into account to some extent, though a true optimum cannot be achieved except by chance, since programmer's time is involved, and thus a determination of the optimum would defeat its own end. The state of the market for these economic factors will vary greatly from problem to problem. For instance there will be an enormous proportion of problems where there is no question of using the whole storage capacity of the machine, so that space is almost free. With other types of problem one can easily use ten million digits of storage and still not be satisfied. The s ace shortage applies mainly to working space rather than to space occupied by the routine. Since these usually have to be written down by someone this in itself has a limiting effect. Speed will usually be a factor worth consideration though there are many 'fiddling' jobs where it is almost irrelevant. For instance the calculation of tabular values for functions which are to be stored in the machine and later used for interpolation, would u usually be in this class. Programmers time will usually be the main factor in special jobs, but is relatively unimportant in fundamental (library) routines which are used in most jobs. Accuracy may compete with machine time e.g., over such questions as the number of terms to be taken in a series, and with space over the question as to whether 20 or 40 digits of a number should be stored.

c) The available storage space must be apportioned to various duties. This will apply both to magnetic and electronic storage. The magnetic storage will probably be mainly either working space or unused. It should be possible to estimate the space occupied by instructions to within say two tracks, for a large part will probably be previously constructed programme, occupying a known number of tracks. The quantities to be held in the working space should if possible be arranged in packets which it is convenient to use all at once, and which can be packed into a track or a half-track or quarter-track. For instance when multiplying matrices it might be convenient to partition the matrices into four rowed or eight rowed square matrices and keep each either in a track or a quarter-track. The apportionment of the electronic store is partly ruled by the conventions we have introduced, but there is still a good deal of freedom, e.g., pages 4, 5, 6, and possibly 7 can be used for systematic working space and may be used for various different purposes that require systematic working space.

The beginner will do well to ask for advice concerning plans. Bad plans lead to programmes being thrown away.

d) If questions of time are at all critical the plan should include a little detailed programming, i.e., the writing down of a few instructions. It should be fairly evident which operations are likely to consume most of the time, and often these will consist of a small number of instructions repeated again and again. In these cases the few instructions in question should be written down so as to give an estimate of the time, and help decide whether the plan as a whole is satisfactory. In this connection see Hints, 2.

e) If one cannot think of any way, good or bad, for doing a job, it is a good thing to try and think how one would do it oneself with pencil and paper. If one can think of such a method it can usually be translated into a method which could be applied to the machine.

Coding the indibidual routines.

(a) As with programming a whole problem a plan is needed for a routine. A convenient aid in this is a block schematic diagram, This consists of a number of operations described in English (or any private notation that the programmer prefers) and joined by arrows. Two arrows may leave a point where a test occurs, or more if a variable control transfer number is used. Notes may also be made showing what is tested, or how many times a loop is to be traversed. The operations appearing as blocks may be replaced by actual instructions. It is usually not worth while at first to write down more than the last two characters of the (presumptive) instruction, i.e., the B line and function parts. These are quite enough to remind one of what was the purpose of the instruction.

One may then write the instructions into a page, deciding at the same time what are to be the addresses involved. Some of the finer points of this have been described in the Hints in chapter 2.

(b) Manoeuvring space. It is seldom that one writes down a page of instructions for the first time without having forgotten a few vital instructions. It is therefore considered desirable to aim, not at pages which are chock-full, but say ones which are about five-sixths full. The extra space is best left between sequences of consecutive instructions, so that one sequence may be extended without interfering with another. The space can also be filled with numbers if desired, when the discovery of mistakes calls for retrenchment. In this connection see also sandwiching. Another useful way of using reserve space is to put in a number of dummy stops or of time wasting instructions, or hoots. The latter provide 'rhythm clicks' which are very informative concerning the progress of the routine.

(c) Alternative entry. It is often necessary to have a number of routines differing in certain minor particulars. One would like to use essentially the same instructions for all of them. The most convenient method seem to be to use one assembly of instructions with various points of entry. The cues of these routines will then differ only in their first ten digits.

(d) Space economy measures. We have already explained that the economising of instructions in order to reduce the space occupied in the magnetic store is seldom worth while. There are however occasions when it is worth while to economise them to save space in the electronic store. This is nearly always in order to get the instructions either into one page or into pages. To do so makes the routine tidier, and usually has time-economy effects. For instance a one-page routine may be combined with another one-page (master) routine which uses it again and again without losing time over magnetic transfers. In these cases the two routines are able to be in the electronic store together, one on one page and one on the other. This effect can also be achieved by the device of borrowing part of the systematic working space (if available) for part of the master routine. If a routine occupies more than two pages, then (unless the same device is used) it must be coded as 2 ad-routines and thus involve magnetic transfers, and consequent loss of time. To some extent then the considerations mentioned under manoeuvring space may be overruled, though they generally apply for routines of three or more pages. Some possible economy measures have been described in the Hints of chapter 2.

The Input Organisation for Scheme B

In this chapter we describe how the routines and numerical information which constitute the problem are fed into the machine from the outside world.

The Routine B.INPUT.

This is a sequence of instructions held permanently in the magnetic store on track 96. This track is isolated by arranging thecircuitry so that information can be read from it to the electronic store but not vice versa. This sequence of instructions when read into S0 and S1 causes the tape reader to scan the tape and assembles the symbols punched thereon to form instructions which are then transferred elsewhere in the electronic or magnetic store.

In addition to B.INPUT the material known as B.PERM - the powers of 2 and R.C.S/B - is isolated on track 97L (actually the whole of track 97 is isolated).

B.INPUT is read from the magnetic store to S0 and S1 as follows. On the console there is a key denoted by K.E.C. If this is operated, then the whole of the electronic store is cleared including the arithmetical unit, the B tube, and the control lines. Operation of a further switch (C.S) causes the machine to start obeying instructions at the line E/ (since C = //). Because the electronic store has been cleared the first instruction obeyed is ////. The function of this instruction is to obey the handswitches (see p. 1.21) as a magnetic instruction. These should be set to /A@/ or /A@E. The content of track 96 will then be read to S0 and S1. The routine is arranged to start at line @/ (because now C = E/) and from this point B.INPUT takes over causing the tape reader to scan the tape until one of the characters K, T, Z, L, W, H, Y, Q, or G is encountered. These characters are called warning characters and they designate the start of a meaningful sequence. The length of the sequence and its subsequent treatment depends on which warning character is used.

The routine can also be entered by cue (GE///A@/) in the course of computation. The treatment of certain warning characters differs slightly according as to whether the routine is entered by cue or by hand with H set to /A@/ or /A@E.

Warning character K

This permits the writing of instructions into any consecutive sequence of lines. The second and third characters specify the first line and the fourth character specifies the number of lines. Thus, e.g., the effect of KZSAVKTAVS/E:TCC is to put VKTA into ZS, VS/E into LS, and :TCC into WS. If the fourth character is /, then the number of lines involved is 32. This means that a complete column of instructions can be punched as a single meaningful sequence. Thus to input a complete page of instructions into (say) S4 it is only necessary to punch them as two meaningful sequences each of 32 instructions

K/½/ (1st column of 32 instructions),
K/D/ (2nd " " " " ")

If the page is not complete but (say) the last instruction would stand in line PD if read into S4, then the second sequence would take the form K/DQ (2nd column of 23 instructions).

The set of meaningful sequences which cause a page of material to be read into S4 or S5, or in the case of two pages, into S4. and S5, is referred to as the punching proper.

It need hardly be mentioned that extravagent effects are to be expected if this warning character is used to write into lines containing B.INPUT itself, that is, into columns /, E, @, or A.

Warning character Z

The meaningful sequence consists of only one character, the warning character itself. It is equivalent to the dummy stop /G. Thus if the "/G" switch is in the "on" position, then the reading action of B.INPUT is halted when this warning character is encountered. Operation of the single prepulse key causes reading to be resumed.

The following warning characters are associated with three character sequences.

Warning character T

The effect of the sequence Tab is to transfer control to the instruction standing in line ab + 1.

Warning characters H and Y

If the routine was originally entered by hand then the first effect is to transfer the content of track 34 - the DIRECTORY - to S6 and S7. In the case of entry by cue this effect is omitted.

The subsequent effect of the sequence $\left\{\begin{matrix}\text{Hab}\\ \text{Yab}\end{matrix}\right\}$ is to $\left\{\begin{matrix}\text{read to}\\ \text{write the material in}\end{matrix}\right\}$ S4 and/or S5 $\left\{\begin{matrix}\text{the material on}\\ \text{on}\end{matrix}\right\}$ the half track or track specified in line $\{/N + ab + 1\}_0^9$ of the electronic store. The manner of transfer is specified as follows. The twenty digits of the entry are first collated (the & operation) with ££AA and then added to ///½. The resulting twenty digits form the magnetic reading instruction which is ultimately obeyed in the case of the warning character H; otherwise //½/ is added to form a writing instruction. In both cases a corresponding checking instruction is formed and obeyed. In case the check fails a line feed is signalled to the teleprinter and the reading or writing operation repeated until the check succeeds. Signalling a line feed is merely a convenient device for drawing the operator's attention to the fact that the magnetic tracks involved may be faulty (see chapter 9).

It will be clear from the above that magnetic transfers to or from the stores 4 or 5 can be arranged with suitable H or Y sequences. In particular if ab takes one of the values // through VA, then the short line /N + ab + 1 corresponds to the magnetic half of entry no. ab of the cue DIRECTORY if this latter is standing in S6 and S7. In these circumstances if the material constituting a routine has been read into S4 and/or S5 by means of the warning character K, then it can be transferred to its appropriate magnetic address simply by following the punching proper with the sequence Yab, where ab is the number of the routine in the DIRECTORY.

Warning characters L and W

These are similar to H and Y except that the characters ab which follow them refer to the line /N + ab. The reason for introducing such an apparently trivial variation is to facilitate the writing of pages of material other than routines, that is, pages which may only be associated with single line entries in the DIRECTORY, not line-pairs. With these warning characters ab is the name of the actual line in which the magnetic address stands rather than that of the one before.

The characters which follow L, W, H, or Y are not restricted to the values // through VA. Other values of ab cause lines in other pages of the magnetic store to be interpreted as magnetic addresses. Thus we have in general

b	=	H	corresponds	to		S0
	=	P	"	"	"	S1
	=	O	"	"	"	S2
	=	G	"	"	"	S3
	=	M	"	"	"	S4
	=	V	"	"	"	S5
	=	/	"	"	"	S6
	=	@	"	"	"	S7.

In particular the content of three lines within B.INPUT itself, [K @] = EA@:, [T @] = @A@:, and [Z @] = @E@N provide the following important uses of L, W, H, or Y.

(i) LKP or HCP causes the content of track 97 - B.PERM - to be transferred to S4 and S5.

(ii) WTP or YKP causes the content of S4 and S5 to be transferred to track 98.

(iii) WZP or YTP causes the content of S4 and S5 to be transferred to the DIRECTORY track 34.

A dummy stop instruction (/L) occurs at the end of the sequence of instructions in B,INPUT which deals with the warning characters L, W, H, and Y. This enables the results of the transfer to be examined before more tape is read, a facility which is often very useful.

Warning characters Q and G

If the routine was originally entered by hand, then the first effect is to transfer to S2 and S3 the content of track 97 - B.PERM - or the content of 98 - a modified PERM which is selected by the user in a manner described below - according as to whether H was set to /A@/ or /A@E. If the routine was entered by cue then this effect will be omitted.

Before Q and G can be treated it is necessary not only that R.C.S/B (or some suitable modification of it) should be present in column S, but that a number in B1 should specify a level of operation and, in the case of G, that the cue to B.INPUT should stand on the level so defined, that is, at the head of the list of links. For this reason it is arranged that when B.INPUT is entered by hand, and not otherwise, then //I/ is set in B1; and that the cue to B.INPUT, GE/'//A@/, is stored in the first two lines of the right half of the (isolated) track 97. Thus if the content of track 97 is transferred to S2 and S3, then this cue will be found to stand in the line-pair /I, the zeroth level as defined by the content of B1.

The subsequent effect of the sequence $\left\{\begin{array}{l}\text{Qab}\\ \text{Gab}\end{array}\right.$ is to call in the routine whose number in the DIRECTORY is ab as an $\left\{\begin{array}{l}\text{ad}\\ \text{sub}\end{array}\right.$ - routine of B.INPUT. If control is subsequently returned to B.INPUT from a closed sub-routine, then it continues to search the tape for warning characters. Note that re-entering B.INPUT in this way amounts to a cue entry and thereafter the preliminary effects in the case of the warning characters L, W, H, Y, Q, and G are omitted.

The treatment of these characters is of course dependent on B.PERM, or some suitable modification of it, but otherwise B.INPUT makes no use of any lines of PERM.

The Assembly of a Problem Tape.

B.Input has been designed to streamline the making of a tape for a problem drawn up on the lines described in the last chapter. The basic idea is to write routines, read from the tape into S4 and/or S5, by means of the warning character K, into the magnetic addresses specified by the corresponding entries in the cue DIRECTORY, and finally to start the computation by calling in the master routine with a Q or G sequence. Moreover it will be seen that the properties of the warning characters H and Y provide a convenient means of altering the standard B.PERM if this is desirable (some of the possibilities have already been indicated on p.4.12 others will be given). In such cases the modified version - called the WORKING PERM - is stored on track 98. These are the most general cases and for them the tape consists of four parts and is read with B.INPUT entered by hand with H set to /A@E. If however the standard B.PERM is used, then the first part, the PERM altering sequences, can be omitted and H should be set to /A@/.

The four parts are as follows:-

(i) The PERM altering sequences

These take the form	HCP	source sequence
	K........	punching proper
	YKP	destination sequence.

The effects of HCP and YKP have already been described. Their purpose will now be clear: HCP transfers the standard B.PERM to S4 and S5 for alteration by K sequences (the punching proper); YKP then transfers the modified version to track 98.

(ii) The DIRECTORY setting sequences

These take the form	K..........	punching proper
	YTP	destination sequence

The punching proper must put the list of cues in their appropriate lines in S4 and/or S5: the destination sequence, as explained above, then copies this material on to the DIRECTORY track 34.

Once this section of the tape has been read the letters ab which follow L, W, H, or Y can, sensibly, refer to the DIRECTORY. Hitherto they have referred to certain preset lines in B.INPUT itself.

(iii) The routines of the problem

These take the form	K........	punching proper
	Y(n)	destination sequence

In the case of a one page routine the punching proper must write the routine into the corresponding lines of S4 or S5 according as to whether it is destined for S0 or S1: in the case of a two page routine the material must be written into S4 and S5.

(iv) Starting sequence

This will usually take the form Q(n) where n (=2 characters) refers to the cue of the master routine.

A sequence of the type G(n) enables one to call in a routine, e.g., a routine for reading numbers punched in decimal form on the tape, as a sub-routine of B,INPUT. When the (closed) sub-routine has completed its task, then B,INPUT is recalled and continues to search the tape for warning characters.

Alternatively a sequence of instructions written into the electronic store can be entered directly by means of the sequence Tab which directs control to the instruction standing in line ab + 1. This facility is very useful when it is required to temporarily interrupt the action of B.INPUT and transfer control to a separate sequence of instructions which have been written into the electronic store from the tape for some special purpose, e.g., to calculate a constant used in a routine, and which are then no longer required. Such a sequence should be terminated by the instruction N@/P which will return control to the reading cycle. This device is called an interlude.

Correcting sequences

In addition to the above parts to a tape it may be necessary to introduce corrections to certain routines from time to time during the period when the programme is being got to work - which may be a matter of days or even weeks. The warning characters H and L provide a convenient means whereby any routine which is already held in the magnetic store can be read down to S4 and/or S5, a ltered, and then transferred back to the magnetic store. A typical correcting sequence takes the form

```
H(n)
K......
Y(n),
```

where n is the number of the routine. The PERM altering sequences afford an illustration of this device. All the correcting sequences can be punched on a single length of tape which can be put through the reader after the main tape has been read. The starting sequence should then come at the end of the tape bearing the corrections.

Example.

As an example of the foregoing directions the details are given here of the assembly of a tape for the problem described in the previous chapter. The sections of the tape corresponding to the DIRECTORY and the master routine must be punched by the programmer. The library routines already exist in the form of bits of punched tape bearing the punching proper, varying in length up to two feet depending on the size of the routine.

(i) The standard B.PERM will be used: section (i) will thus be omitted.

(ii) The cue DIRECTORY. The table of p.4.7, reproduced below, gives the numbers and cues of the routines.

Routine	Number	Mag.Address	Cue
Master	//	5L	££//S///
PRINT/A	@/	5R	CE//S/E/
INPUT/B	:/	6L	////I///
EXPONENTIAL/B	I/	6R	@@//I/E@
COSINE/A	½/	7L	////U///

This section of the tape thus consists of the following sequences

```
K/½R££//S///CE//S/E/////I///@@//I/E@////U///  punching proper
YTP                                           dest. seq.
```

(iii) The Master Routine. This is given on p.4.11. There are 19 instructions. This section of the tape is punched as follows:

```
K/½W//T:E/QOGS/P....................DS/P/O/E   punching proper
Y//                                            dest. seq.
```

In between meaningful sequences it is permissible to leave any number of spaces (or any characters other than warning characters). However there is no point in making tapes any longer than they need to be.

When the above tapes and those of the library tapes are available they are all copied on to one long tape, following the punching proper of each library tape by the appropriate destination sequence. Thus, e.g., the punching proper of EXPONENTIAL/B would be followed by YI/. Finally at the end of the tape the starting sequence Q// is punched. The effect of this is to call in the master routine as an ad-routine of B.INPUT.

It is customary to leave a few inches of blank tape between the individual routines and to terminate the tape with a few £'s so as to indicate visibly which end is which. This last device is usually applied to the individual routine tapes in the library.

If the /L stop is switched on when the tape is read, then the reading will be halted immediately after L, W, H, or Y sequences are treated. By studying the monitors the operator can then see what has just been transferred to or from S4 and/or S5 (extensive scrutinies of the monitors however are not, in general, approved of and more will be said about this in chapter 8). Operation of the single prepulse key will cause B.INPUT to resume reading tape. If the /G stop is switched on, then the machine will come to a halt in the routine changing sequence during the treatment of the starting sequence Q//. At this point the master routine should be standing in S0 and if the single prepulse key is operated, then the programme will be initiated.

Example illustrating the use of a WORKING PERM

We shall now explain how some of the devices described on pages 4.12 and 4.13 can be applied to speed up the operation of the programme by eliminating many of the magnetic transfers. (In this particular programme there would be no point in doing so for 99% of the time consumed is input and output time, but the devices illustrated can often usefully be applied elsewhere).

The necessary changes in the programme amount to making both the directory of cues and the master routine part of the material held permanently in S3. Thus the cue directory can stand in lines

TI to BI inclusive and the master routine in lines /U to LU inclusive. The latter will now be associated with the cue £I....£.. Line QS of R.C.S/B has to be altered to ..T£ and, if the routines are to have the same directory numbers, line OS should be altered to TIUF. Finally certain alterations to the master routine itself are necessary because many of the instructions depend on the names of the lines in which they stand. The final version of the master routine is given below together with the rest of the material on S3.

```
|I||       |/|£ I T :|U|
   |       |E|E U Q O|
   |       |@|G S / P|
   |       |A|: / / V|
   |       |:|/ C / A|
   |       |S|S U Q O|
   |       |I|G S / P|
   |       |U|½ / / V|
   |       |½|: C T /|
   |       |D|D U Q O|
   |       |R|G S / P|
   |       |J|I / / V|
   |       |N|: C / J|
   |       |F|Z U / C|
   |       |C|C U Q O|
   |       |K|G S / P|
   |£ I / /|T|@ / / V|
   |/ / £ /|Z|/ U / P|
   |C E / /|L|/ O / E|
   |S / E /|W|       |
   |/ / / /|H|       |
   |I / / /|Y|       |
   |@ @ / /|P|       |
   |I / E @|Q|       |
   |/ / / /|O|       |
   |U / / /|B|       |
   |       |G|       |
   |       |"|       |
   |       |M|       |
   |       |X|       |
   |       |V|       |
   |       |£|       |
```

We can now write down the PERM altering sequences. They are

```
HCP
KQD@//T£TIUF
KTRR£I////£/CE//S/E/////I///@@//I/E@////U///
K/JW£IT:EUQO......../O/E
YKP
```

This material will still be standing in S4 and S5 after it has been copied on to track ~~98~~ 15 so that by simply following the PERM altering sequences with TYP this material can also be written on to track 34 to serve as the DIRECTORY.

The second part of the tape thus consists simply of the sequence YTP.

As before these tape sections and those of the library routines are all copied on to one long tape, but this time the destination sequences of the library routines are different, e.g., that which follows EXPONENTIAL/B is YP@ (because /N + P@ = PC, which is the store line in which the relevant cue is standing during input). The starting sequence however remains Q// (this is because the 2 characters which follow Q or G refer to the DIRECTORY as used by R.C.S/B and not as it stands in S6 and S7). The programme is run with H set initially to /A@E.

Further example of a WORKING PERM

Quite extensive alterations to the basic scheme of routine changing are made possible by the facility for modifying the standard B.PERM. Our example relates to an actual problem where, with the usual scheme, the electronic storage available for working space, was inconveniently small. To overcome this it was necessary to confine the permanent material to a single page and each routine to one page. Fortunately it was possible to dispense with the powers of 2 (other than those required by R.C.S/B) and so the list of links could stand in lines /S, ES, etc.. In no case did this list extend more than three deep. Other lines of S2 were used as special working space and to store various constants required by the programme. The duties of the electronic pages were thus allocated as follows.

S0, S1	systematic working space
S2	PERM as described above
S3	instructions
S4, S5	systematic working space
S6, S7	" " " " " ".

The routines were all single page routines and when called in were read down to S3. Their cues were thus of the form ....../I or ......EI, and for input purposes were punched for S5. The DIRECTORY likewise occupied only a single page and was read to S3 to be consulted. The alterations which were necessary to B.PERM are listed below.

```
    line pair /I altered to GE///A@/ (the cue to B.INPUT)
         line FS    "    "  @E/I
           "  HS    "    "  V:ZZ
           "  PS    "    "  @SEA
           "  OS    "    "  /IUF
           "  BS    "    "  /SEA
           "  "S    "    "  V:EJ
           "  XS    "    "  £:E:
           "  £S    "    "  V:EP
```

Setting the cue to B.INPUT at the head of the list of links is a necessary alteration if the warning character G is to be treated with the aid of the WORKING PERM. It will be recalled that normally this cue will be found standing in line pair /I if the content of track 97 is transferred to S2 and S3.

The corresponding PERM alteration sequences are

```
HCP
K/D@GE///A@/
KFDE@E/I
KHDEV:ZZ
KPDE@SEA
KOD@/IUF/SEA
K"DEV:EJ
KXDE£:E:
K£DEV:EP
YKP
```

The DIRECTORY sequences are punched in the usual way for S4. The remainder of the tape also takes the usual form with the exception of course that the routines are punched for S5.

Use of 'electronic' routines.

In problems where the requirments for working space are small it may be desirable for time economy reasons to keep as many routines as possible in the electronic store. Such might be the case for inst,nce when forward integrating a small set of ordinary differential equations which involve the calculation of several functions. The device used for calling in such 'electronic' routines - a dummy magnetic half cue - has already been used in the example treated earlier where the master routine was held permanently in S3. It remains however to describe how such routines, since they are associated with a dummy magnetic half cue, are read down to the electronic store in the first place. In the example quoted there was no difficulty because the routine in question formed part of the WORKING PERM which was read down to S2 and S3 when the starting sequence was read. In other cases however a special device must be resorted to if it is desired to 'set' the electronic store from the tape before entering the programme. The device used amounts to giving each 'electronic' routine two cues. One will be the cue in the ordinary sense, that is, will be used for calling in the routine once it is located in the electronic store: the other will take the form "S//(magnetic instruction). The effect of such a cue when referred to by a G sequence is to obey the magnetic instruction and then to return control to B.INPUT as from a closed sub-routine - thus causing more tape to be read. In this way stores 4, 5, 6, and 7 can be set. If it is further desired to replace B.INPUT itself by an 'electronic' routine - in which case an electronic DIRECTORY in the WORKING PERM, or elsewhere, must be available - then the same device can be used provided that the cue of the routine which replaces B.INPUT has the same control number as that of B.INPUT, i.e., GE. With this restriction the final setting of S0 and S1 can be achieved directly from the tape by means of only two sequences. First a Q sequence referring to the true cue of the replacing routine followed by a G sequence referring to the special cue. It is left as an exercise for the reader to show that these have the effect of replacing B.INPUT by the routine in question and entering the latter at "E. Generally speaking the use of such special devices is not to be encouraged in general purpose programmes but in this case they have only amounted to unorthodox uses of the standard properties of R.C.S/B.

Input of Numerical (Decimal) Data.

For this purpose a special closed routine B. DECIMAL INPUT is available. If the data is to be transferred to the magnetic store, then this routine may conveniently be used as a sub-routine of

Cue to B.dec input is //----@/

B.INPUT. When called in from the tape by a Q sequence B.DEC.INPUT continues to read the tape until one of the warning characters F, W, C, L, or G is encountered.

The warning characters F and W permit the input of a sequence of decimal numbers into consecutive line-pairs, F being used for fractions and W for whole numbers. The next two characters on the tape specify the first line-pair and the fourth character gives the number of line-pairs; then follow the numbers punched in decimal form each followed by sign. The character £ punched after any number in place of the sign causes that number to be omitted from thesequence. It may thus be used to cancel mistakes. The details will be clear from the following examples.

The effect of

F/½U 25+025-250+002500-25£75+750+075-

is to put 0.25 into /½, -0.025 into @½, 0.25 into :½, -0.0025 in I½, 0.75 into ½½ and R½, and -0.075 into N½.

If F were replaced by W then the effect would be to put 25 in /½. -25 in @½. 250 in :½. -2500 in I½. 75 in ½½. 750 in R½, and-75 in N½.

In the case of fractions not more than 11 digits should be punched, otherwise the number placed in the store will be incorrect: 11 digits is the limit of accuracy.

The warning character C is the closure, that is, it causes control to revert to the routine which called in B.DEC. INPUT.

The other two warning characters, L and G, are the equivalents of the dummy stops /L and /G respectively. Thus if the "/L" stop is "on" when L is encountered then the reading action is halted: it is resumed on operation of the single prepulse key.

Library routine tapes.

For scheme B these consist solely of the punching proper: one page routines being punched for S4 or S5 according as to whether they are ultimately destined for odd or even electronic page; two page routines being punched for S4 and S5.

Classified lists of library routines will be issued as supplements to this Handbook at future dates. As far as possible routines will be made for use with scheme A and scheme B.

B.INPUT, ISOLATED ON TRACK 96.

@1I1 → B1
EA@1 → B6
HSW → B7

Take in add 1 to ch if → ch.
char → A1

ch → □111 in @1
ch → B7

```
 / |M £ £ £|/|/ / / J|E        @|E N P /|/|V / / G|A
   |£ £ £ £|E|@ / / U|          |L @ T R|E|F @ / P|
   |W @ Z O|@|D / / :|          |½ @ U Q|@|       |
   |K @ P O|A|@ / / A|          |Q @ T C|A|X E    |
   |: / / Z|:|@ / / N|          |P @ T C|:|X E    |
   |: / Q O|S|D / / :|          |@ / / S|S|V E    |
   |£ @ Q G|I|@ / / A|          |Y @ T C|I|V E    |
   |" @ / O|U|@ / / N|          |A / T A|U|       |
   |T @ P B|½|D / / :|          |@ / / :|½|L E    |
   |E E O :|D|@ / / A|          |A / / :|D|       |
   |D / / :|R|@ / / N|          |X @ / :|R|       |
   |V @ / J|J|D / / :|          |/ / / Q|J|L E    |
   |N @ / H|N|@ / / A|          |½ / / L|N|       |
   |@ / / A|F|A / Q O|          |N @ / P|F|       |
   |O @ / C|C|E / P G|          |A / / P|C|       |
   |@ / / N|K|£ £ P Z|          |E A @ :|K|       |
   |@ / / A|T|V @ T N|          |@ A @ :|T|L /    |
   |@ / Q O|Z|D / / H|          |@ E @ N|Z|L /    |
   |/ A U P|L|N @ / P|          |£ £ A A|L|£ @    |
   |D / / :|W|H @ P G|          |@ / I /|W|L /    |
   |V @ / C|H|G E P B|          |/ / / @|H|L /    |
   |@ / / A|Y|T @ / :|          |/ / : /|Y|L /    |
   |@ / / N|P|@ / Q B|          |/ / / ½|P|L /    |
   |D / / :|Q|@ / P O|          |/ / ½ /|Q|½ /    |
   |@ / / A|O|O E Q O|          |/ E / /|O|L /    |
   |B @ / C|B|£ : I P|          |/ / / /|B|½ /    |
   |@ / / N|G|@ A @ :|          |/ / E /|G|½ /    |
   |@ / / A|"|/ / T /|          |E / / /|"|L /    |
   |V @ / C|M|T @ T A|          |/ / / /|M|½ /    |
   |A / P O|X|N @ / P|          |E / W /|X|½ /    |
   |T @ U P|V|" @ P G|          |/ / / /|V|½ /    |
   |D / / :|£|Z @ / :|          |/ / / E|£|½ /    |
```

B.PERM, ISOLATED ON TRACK 97.

```
: |/ / / / |/|/ / / / |S      I|G E / / |/|      S|
  |E / / / |E|/ / / @ |        |/ A @ / |E|       |
  |/ / / / |@|/ / / / |        |        |@|       |
  |@ / / / |A|/ / / : |        |        |A|       |
  |/ / / / |:|/ / / / |        |        |:|       |
  |: / / / |S|/ / / ½ |        |        |S|       |
  |/ / / / |I|/ / / / |        |        |I|       |
  |½ / / / |U|/ / / T |        |        |U|       |
  |/ / / / |½|/ / / / |        |        |½|       |
  |T / / / |D|£ £ £ £ |        |        |D|       |
  |/ / / / |R|£ £ £ £ |        |        |R|       |
  |/ E / / |J|/ / / / |        |        |J|       |
  |/ / / / |N|" S / L |        |        |N|       |
  |/ @ / / |F|@ E / / |        |        |F|       |
  |/ / / / |C|Z S Q G |        |        |C|       |
  |/ : / / |K|A : Z G |        |        |K|       |
  |/ / / / |T|S : / Q |        |        |T|       |
  |/ ½ / / |Z|V £ P O |        |        |Z|       |
  |/ / / / |L|M £ £ £ |        |        |L|       |
  |/ T / / |W|Z S Q G |        |        |W|       |
  |/ / / / |H|V S Z Z |        |        |H|       |
  |/ / E / |Y|/ / Q T |        |        |Y|       |
  |/ / / / |P|@ I E A |        |        |P|       |
  |/ / @ / |Q|F S / : |        |        |Q|       |
  |/ / / / |O|/ / U F |        |        |O|       |
  |/ / : / |B|/ I E A |        |        |B|       |
  |/ / / / |G|L S Z G |        |        |G|       |
  |/ / ½ / |"|V S E J |        |        |"|       |
  |/ / / / |M|A : Z G |        |        |M|       |
  |/ / T / |X|£ S E : |        |        |X|       |
  |/ / / / |V|    / G |        |        |V|       |
  |/ / / E |£|V S E P |        |        |£|       |
```

B.INPUT (page 0).

```
 |/| /|M £ £ £|  -4
     E|£ £ £ £|  -1
     @|W @ Z O|  set B1
     A|K @ P O|} tests H: if
     :|: / / Z|  H = /A@/, then
     S|: / Q O|  [T @]' = EA@:;
     I|£ @ Q G|  otherwise line
     U|" @ / O|  T@ is not
     ½|T @ P B|  altered.
  →  D|E E O :|} reading cycle: test
     R|D / / :|  if character read
     J|V @ / J|  (α) is ≯ K
     N|N @ / H|
     F|@ / / A|} case α ≯ K
     C|O @ / C|  α/// is formed
     K|@ / / N|  and transferred
     T|@ / / A|  to B7
     Z|@ / Q O|
     L|/ A U P|  multiway switch:uses
      --------      directory 1
  →  W|D / / :|}
-K,L,H|V @ / C|
W,H, Y|@ / / A|  reads 2nd and
Y,Q, P|@ / / N|  3rd characters,
& G  Q|D / / :|  α and β ; forms
     O|@ / / A|  αβ// and
     B|B @ / C|  transfers this to
     G|@ / / N|  B6; uses directory
     "|@ / / A|  no.2 for multiway
     M|V @ / C|  switch
     X|A / P O|
     V|T @ U P|
      --------
  K→ £|D / / :|  read 4th character →
                                   H,Y
```

```
 |E| /|/ / / J|  set counter
     E|@ / / U|
  →  @|D / / :|}
     A|@ / / A|
     :|@ / / N|
     S|D / / :|  Reads & assembles
     I|@ / / A|  4 characters of
     U|@ / / N|  an instruction
     ½|D / / :|
     D|@ / / A|
     R|@ / / N|
     J|D / / :|
     N|@ / / A|  instruction trans-
     F|A / Q O|  ferred to B7
     C|E / P G|  adjust address
     K|£ £ P Z|  transfer to store
     T|V @ T N|  adjust counter
  ←  Z|D / / H|  test for end of seq.
     L|N @ / P|  return to reading
      --------                     cycle
     W|H @ P G|  ← Q & G entry
     H|G E P B|
     Y|T @ / :|  prelim mag. transfers
                   PERM to S2 & S3
     P|@ / Q B|} [B 6]' = [B 7]
     Q|@ / P O|
     O|O E Q O|} calls in
     B|£ : I P|  routine
     G|(@K A/ @ :)|  useful constant
     "|/ / T /|} set suppresion of
     M|T @ T A|  prelim.mag transfers
     X|N @ / P|  return to reading
      --------                     cycle
     V|" @ P G|  ←— L,W
     £|Z @ / :|  prelim.mag.transfer:
                 DIRECTORY to S6 &
                 S7.
```

B.INPUT (page 1)

```
     |@| / | E  N  P  / |  select DIRECTORY entry
         E | L  @  T  R |)
         @ | ½  @  U  Q | Forms and plants
         A | Q  @  T  C | magnetic instru.
         : | P  @  T  C | together with the
         S | @  /  /  S | corresponding
         I | Y  @  T  C |  check instruction
         U | A  /  T  A |)
      -> ½ | @  /  /  : |  magnetic transfer
         D | A  /  /  : |  check
         R | X  @  /  : | signals line feed
         J | /  /  /  Q | in case of failure.
      -> N | ½  /  /  L |  dummy stop
         F | N  @  /  P |
  T->    C | A  /  /  P |  transfers control
         K | £C A  @  : |  track 97 to S4 & S5
         T | @K A  @  : |    "   98  "  "  "  "
         Z | @  E  @  N |    "   34  "  "  "  "
         L | £  £  A  A |)
         W | @  /  I  / |
         H | /  /  /  @ |
         Y | /  /  :  / |
         P | /  /  /  ½ |
         Q | /  /  ½  / |
         O | /  E  /  / | constants
         B | /  /  /  / |
         G | /  /  E  / |
         " | E  /  /  / |
         M | /  /  /  / |
         X | E  /  W  / |
         V | /  /  /  / |
         £ | /  /  /  E |)
```

```
     |A| / | V  /  /  G | <-Z: dummy stop;
         E | F  @  /  P |   return to reading
         @ |            |              cycle
         A | X  E       |
         : | X  E       |
         S | V  E       |
         I | V  E       |
         U |            |
         ½ | L  E       | directory no.2
         D |            |
         R |            |
         J | L  E       |
         N |            |
         F |            |
         C |            |
         K |            |
         T | L  /       |)
         Z | L  /       |
         L | £  @       |
         W | L  /       |
         H | L  /       |
         Y | L  /       |
         P | L  /       |
         Q | ½  /       | directory no.1
         O | L  /       |
         B | ½  /       |
         G | ½  /       |
         " | L  /       |
         M | ½  /       |
         X | ½  /       |
         V | ½  /       |
         £ | ½  /       |)
```

B.DECIMAL INPUT (Page 0)

```
 /   /  / / T £   dummy
     E  / / T £   dummy
     @  E / O :   reading cycle:
     A  @ / / :   after a new
     :  / / / A   character has
     S  / / / C   been read control
     I  O @ / N   is transferred by
     U  / / / A   multiway switch
     ½  / / Q O   using α th entry
     D  / A U P   in directory.
W -> R  Y @ Q O   [R E]' = dummy instr.
     J  G @ / Q   in case of whole
F -> W  D @ Q O    number; otherwise
  -> F  R E Q B   YE/P
     C  @ / / :
     K  V @ / C   inputs 2nd and
     T  / / / A   3rd characters,
     Z  / / / N   α & β: forms
     L  @ / / :   αβ// and
     W  " @ / C   transfers this
     H  / / / A   to B6.
     Y  / / / N
     P  / / / A
     Q  E / P O
     O  @ / / :   read 4th
     B  P @ / J   character to
  -> G  / / / U   set counter
     "  K @ T A   plants counter
  -> M  V E Q O   [B7]' = ///J
     X  " / T :   clear A
  -> V  / / / A   part of digit
     £  / / / C   cycle
```

```
 E   /  O @ / N   digit cycle:
     E  / / / U   builds up
     @  X @ / C   decimal integer
     A  / / / N
     :  £ @ Q G
     S  @ / / :   read digit
     I  ½ E / H   test for +,-,or £
     U  V @ / J
  <- ½  X / / H   test for £
     D  C @ / E
     R  Y E / P   Jump in case of fraction
     J  S E / J <-return from line :@
     N  A / / M   test for + or -
     F  P @ T J   change sign in
     C  G @ T C   case of -.
     K  / / P A   transfer to store
     T  Y @ P G   adjust address
     Z  K @ T /   adjust counter
     L  V @ T N
     W  U @ / H   test for end of seq.
     H  @ / / P   return to reading
     Y  Q E / /                   cycle
  -> P  / / / N   complete building
     Q  £ @ Q G   up of decimal
  -> O  / / T A   integer
     B  R E / T
     G  Z @ / C   multiply 2^36/10^11
     "  / / / N
     M  : / Q O   and again
  -> X  / / / E   by 2^4
     V  / / / J
     £  £ @ Q G                   multiply by 2^40/10^11
```

B.DECIMAL INPUT (page 1)

```
     @  / | S @ / T
        E | W @ / J   round-off
        @ | / / / U
        A | C @ / J   r.set terminal charact.r
        : | B E / P   switch to sign test
  L→    S | M E / L   dummy stop
        I | @ / / P
  G→    U | G / / G   dummy stop .
        ½ | @ / / P
        D | Y E / P
        R |
        Y |
        N |
        F |
        C | working
        K | space
        T |
        Z | N V @ V }  2^36/10^11
        L | £ Y £ Y }
        W | ½ / / / }
        H | / / / / }
        Y | V £ £ £ }
        P | £ £ £ £ }
        Q | £ £ £ £ }- useful constants
        O | / E / / }
        B | / / / / }
        G | E / / / }
        " | / / / / }
        M | / / E / }
        X | R / / / } ]10
        V | / / / / } ]
        £ | / / / E }
```

```
     A  / | E /
        E | E /
        @ | E /
        A | E /
        : | E /
        S | E /
        I | E /
        U | E /
        ½ | E /
        D | E /
        R | E /
        J | E /
        N | E /
  ✓     F | J /
  ✓     C | " S
        K | E /
        T | E /
        Z | E /
  ✓     L | : @
  ✓     W | D /
        H | E /
        Y | E /
        P | E /
        Q | E /
        O | E /
        B | E /
  ✓     G | I @
        " | E /
        M | E /
        X | E /
        V | E /
        £ | E /
```

DIRECTORY USED FOR TREATING EACH NEW CHARACTER ENCOUNTERED BY READING CYCLE.

Chapter 5.

Aids to Coding.

Introduction.

At this stage of development of the technique of programming a principle objective is to make coding easier, i.e., to reduce the time taken to translate the mathematical formulation of a problem into the language of the machine. In this chapter this problem is discussed on the following lines.

Certain types of problem are best coded by using a special 'instruction' code. Thus e.g., in a problem involving arithmetic with complex numbers it would be economical in coding time if, by some means, we could code an operation on a complex number x + i y stored (say) in consecutive line pairs thus, $x = [S]_{\pm f}$ $y = [S+2]_{\pm f}$, by using a single instruction which need only specify S and the type of operation. In other words to code the operations as if only the real parts of the numbers were involved. This could be done by making use of an interpretive type of routine which, given the 'instructions' relating to the real parts, would automatically arrange the calculation to embrace both the real and imaginary parts. If the only operations were complex additions and subtractions no great saving would have been effected but with complex multiplications and divisions it would begin to be worthwhile.

This problem can be approached in two ways.

Translation Routines.

In this scheme the instructions punched on the tape are not those which are ultimately placed in the store. Instead a group of characters on the tape, punched in accordance with a preassigned code are translated into sequence of instructions which, when obeyed inside the machine, effect the required operations. In effect the input routine translates the symbols on the tape into a programme in the language of the machine. One such input routine which translates into machine language from mathematical symbols is being developed. With such routines the resulting programme of machine instructions is usually as economical in machine time as if it had been coded directly. However such a scheme may not always be possible and an alternative approach is illustrated by FLOATCODE, an interpretive routine for carrying out operations on numbers expressed in the floating binary form a.2^p.

Interpretive (master) routines.

These routines are suitable when a limited number of sequences of instructions are used very frequently. In such a scheme the problem (or part of a problem) is coded, using a special code, as a sequence of 'instructions' each of which specifies a subsequence of ordinary machine instructions. The 'programme' is placed in the store in the usual way. However the 'instructions' are not intended to be obeyed by the control circuits of the machine as such but instead they are selected under the 'control' of the interpretive routine which 'decodes' them and transfers control to the specified sequence. At the end of each sequence control is returned to the 'control' of the interpretive routine which causes the next 'instruction' to be selected and interpreted in the same way. The term 'control' (in quotation marks) will be used to denote the process of selection and decoding of the 'instructions'.

A general scheme for designing an interpretive routine will now be described. The basis of the scheme is a sequence of instructions (henceforth referred to as INTERCODE) held in S3 throughout the 'programme' which effects the selection and 'decoding' of the 'instructions' and a routine changing sequence similar in principle to R.C.S/B. An 'instruction' is a 4 character word standing in a single line of the store and represented by s, X, where s (1st 3 characters) is the (generalised) address and X (function character) specifies a particular sequence of instructions. It is intended that the sequences of instructions will be held in S4, S5. Each 'instruction' can thus be regarded as calling in a subsequence and also specifying a parameter to be used by that subsequence. Provision is made for at least 13 'instructions' (corresponding to function symbols X = W to £ inclusive) of which, however, the interpretation of three has been fixed as follows:

2 n,X   Call in nth routine as an ad-routine.

2 n,V   Call in nth routine as a sub-routine

n,£ $C' = n + 1$, i.e., transfers control (of the machine) to $n + 1$.

Decoding.

Each 'instruction' is decoded as follows. The address digits and the function digits are separated and stored in KI and ZI thus:-

$$[K\ I]_{0}^{14} = [\text{'instruction'}]_{0}^{14}$$

$$[K\ I]_{15}^{19} = 0$$

$$[Z\ I]_{0}^{4} = [\text{'instruction'}]_{15}^{19}$$

$$[Z\ I]_{5}^{19} = 0$$

Control is then switched to a line specified by the function character X, thus: $C' = \{[T\ S + Z\ I]\}_{0}^{9}$. The sequence switched to in this way must be terminated by an instruction which returns control to line :U (see note 2, p.5.14)

To enter and leave INTERCODE.

The routine changing sequence enables routines of ordinary instructions or interpreted 'instructions' to be used in the same programme with equal facility (see below). Furthermore within any routine ordinary instructions can be interspersed with floating 'instructions' and vice-versa. This section describes the method by which (within a routine) INTERCODE can be entered to interpret 'instructions' and left to obey instructions.

To enter INTERCODE.

This is done most usually with two instructions, thus

```
s      s, Q O
s+1    E I / P
s+2    -------
s+3    -------
 .       .
 .       .
```

Thereafter the 'control' of INTERCODE takes over and the contents of lines s+2, s+3, etc., will be interpreted as 'instructions' until a transfer of 'control' or control is encountered.

Alternatively INTERCODE can be entered for a "single shot" thus

```
s      s, £ 0
s+1    E I / P
s+2    -------    This line interpreted by INTERCODE.
s+3    -------    Control returned to this point.
```

In this case only the content of line s+2 will be interpreted, after which, unless this line is a transfer of 'control' or control, INTERCODE is left to return control to s+3.

To leave INTERCODE.

This can be done with the 'instruction' n, / £ which transfers control to n + 1. Thus 'instruction' s, / £ standing in s will cause the following lines to be obeyed as ordinary instructions.

Re-entering INTERCODE.

If 'control' has been interrupted at some point in a sequence of 'instructions' by a £ - 'instruction' to direct control to a sequence of instructions, then a subsequent instruction which returns control to line : U will cause 'control' to be returned to the sequence of 'instructions' at the point immediately following that at which it was left. This method is illustrated in example 2 given below under the description of FLOATCODE. In effect the £ - 'instruction' can be used to call in a subsequence of ordinary instructions from a main sequence of 'instructions'. Such a subsequence of instructions only needs to be terminated by an instruction which returns control to line :U (see note 2 p.5.14)

The Design of Closed Routines for use with INTERCODE.

Closed routines drawn up for use with R.C.S/B can only be used with INTERCODE if line NS contains certain information, thus $[N\ S]_0^9$ = / U. In practice this may easily be arranged because INTERCODE is intended to be used with other sequences of instructions held more or less permanently in the electronic store. Such is the case e.g., with FLOATCODE, described below, where S2 contains the powers of 2 and other constants. Moreover if the powers of 2 are available, i.e., lines /: to JS of PERM, then certain constants held in lines WI to QI inclusive are no longer necessary and may be omitted if desired, provided that their reference instructions are changed accordingly.

If however / U ... is not available in line NS then closed routines must be terminated by the instruction A U / P. Alternatively closed routines can be terminated by the 'instruction' / U / F if appropriate.

It is possible to use routines which consist wholly or partly of 'instructions'. To facilitate this digits 10-19 of the cue are used to describe whether the routine is entered in an interpretive or non-interpretive manner. If the digits are zero (i.e., if the 3rd and 4th characters are //), then the routine is entered by control in the usual way. If, however, they are 0 £, then the routine is entered by 'control', i.e., if n is the 'control' number, then n+1 is 'obeyed' as an 'instruction'. (N.B. There is one important exception to this rule: if n is ££, then the 3rd. and 4th. characters must be Q£). One important feature of R.C.S/B is not preserved in the routine changing sequence in INTERCODE: the role of the accumulator and register. Thus routines whose effects depend on this feature cannot be used.

Routine Changing 'Instructions'.

The use of B1 and B7 is governed by the same restrictions as when using R.C.S/B. If the cue directory has been assembled, then the nth-routine can be called in as a sub-routine or an ad-routine from a sequence of 'instructions', by the single 'instruction' 2n,V or 2n,X respectively. From a sequence of ordinary instructions however it is first necessary to enter INTERCODE, for a single 'instruction', and follow the entry instructions with the appropriate 'instruction'. The following table summarises the methods used.

To call in the nth - routine as an	From a sequence of 'instructions' use the 'instruction'.	From a sequence of instructions use the group.
ad-routine	2n, / X	s s, Q 0 s+1 E I / P s+2 2n, / X
sub-routine	2n, / V	s s, £ 0 * s+1 E I / P s+2 2n, / V or s s, Q 0 s+1 E I / P s+2 2n, / V

* In this case when the sub-routine has completed its task, then control (not 'control') is returned to line s+3.

Description of FLOATCODE.

1. Introduction.

FLOATCODE is an interpretive routine designed to facilitate the coding of arithmetical operations on numbers expressed, inside the machine, in the floating binary form a.2^p, where a is a 29 digit signed binary fraction such that $\frac{1}{2} > |a| \geq \frac{1}{4}$ and p an integer such that $256 > p \geq -256$.

FLOATCODE is intended to be permanently available in electronic stores 2, 3, 4, and 5 throughout the duration of the programme (or sub-programme). It consists of the powers of 2, i.e., lines / : to J S of PERM, INTERCODE held in S3, and certain sequences of instructions in S4 and S5. These instructions effect the scaling operations on the numerical parts of the numbers involved in the calculation. They include, e.g., the alignment of binary points of numbers with differing exponents and standardisation of the results of additions and subtractions.

FLOATCODE enables the coding to be carried out using a one-address 'instruction' code of 13 'instructions'. Each 'instruction' is similar in structure to an ordinary machine instruction and refers to an 'arithmetical unit' and a 'control' which consists of certain storage locations in S2 and S3. The correspondence between 'instructions' and these units is very similar to that between ordinary machine instructions and the accumulator, multiplicand register, and the control unit of the machine.

The 'arithmetical unit' comprises a floating binary 'accumulator' (A) and a floating binary multiplier register (R). The 'control' is a sequence of instructions in FLOATCODE which cause 'instruction' of the programme' to be selected and analysed in consecutive sequence until a transfer of 'control' is encountered. At any stage in the programme the 'control' number (t) is the address of the 'instruction' being interpreted or 'obeyed'. When each 'instruction' has been 'obeyed' unity is added to t (t' = t + 1) and the next 'instruction' selected.

2. Representation of numbers.

(i) Inside the 'Store', i.e., SO 1, 6, or 7.

A line-pair is used to represent the number as follows:-

$$\left\{s_+\right\}^{39}_{30\pm} = 2p \qquad 256 > p \geq -256.$$

$$\left\{s_+\right\}^{29}_{0\pm} = a \qquad \tfrac{1}{2} > |a| \geq \tfrac{1}{4}$$

That is, the ten most significant digits represent the exponent and the remainder represent the fractional part. F (s) will be used to denote the floating binary number represented in this way.

(ii) Inside the 'arithmetical unit'.

In what follows A is used to denote the floating binary accumulator, not the ordinary accumulator. There is no occasion to refer to this latter unit so that no confusion will arise.

The floating binary number in A will be denoted by F (A) = $a.2^p$.
" " " " " " " " R " " " " " F (R) = $b.2^q$.

F (A) is represented thus:-

(long line) $[F\ S]_{\pm} = a \qquad \tfrac{1}{2} > |a| > 0$

(short line) $[K\ S]_{\pm} = 2p \qquad 2^{19} > p \geq -2^{-19}$, And F (R) similarly thus:-

$$[T\ S]_{\pm} = b \qquad \tfrac{1}{2} > |b| > \tfrac{1}{4}$$

$$[L\ S]_{\pm} = 2q \qquad 256 > q \geqslant -256.$$

Whereas for a number in the 'store' the exponent is restricted to the range ($256 > p \geqslant -256$), in the floating binary accumulator p can lie anywhere in the range $2^{19} > p \geqslant -2^{-19}$, and the numerical part need not be in standard form. When, however, a number is transferred from A to the 'store' the numerical part is first standardised and the exponent adjusted accordingly. If the correct exponent lies outside the range $256 > p \geqslant -256$ action is taken as follows: if $p > 256$ then the interpretive (master) routine goes into a small closed loop (dynamic stop); otherwise (i.e., if $p < -256$) it is merely replaced by -256.

(iii) <u>On the input tape.</u>

A routine is being made for reading numbers in standard form from the tape. Such numbers should be punched in floating decimal form, i.e., in the form $a.10^p$ where e.g., $10 > a \geqslant 1$ and $76 > p \geqslant -76$.

3. <u>The 'instruction code'.</u>

Each 'instruction' is similar in structure to an ordinary machine instruction both inside the machine and on the tape and programme sheets. We shall write the general 'instruction' in the form n, b, X., where n (2 characters) is the address of an operand, destination, etc., b (1 character) specifies the b line in the usual way (digits 13 and 14 are spare) and X (function character) specifies the particular operation. X can be any one of the letters W, H, Y, P, Q, O, B, G, ", M, V, £. 'Instructions' with other function characters will initiate curious effects. The 13 'instructions' together with their times of operation are as follows:

60ms.	n, b, W	F'(A) = F (A) + F (n).
60ms	n, b, H	F'(A) = F (A) - F (n)
80ms.	n, b, Y	F'(n) = F (A), F'(A) = "zero"
25ms.	/, /, P	dummy 'instruction', i.e., no effect
80ms.	n, b, Q	F'(A) = F (A) + F (R). F (n)
80ms.	n, b, O	F'(A) = F (A) - F (R). F (n)
150ms.	n, b, B	F'(R) = 1/F (n)
30ms.	n, b, G	F'(R) = F (n)
25ms.	n, /, "	t' = n + 1, i.e., transfer 'control' to n + 1
25ms.	n, /, M	If F (A) > 0, then t' = n + 1; otherwise F'(A) = -F (A)
80ms.	2n, /, X	Call in nth-routine as an ad-routine
80ms.	2n, / V	Call in nth-routine as a sub-routine
25ms.	n, /, £	C' = n + 1, i.e., transfer control to n + 1.

Notes on 'instruction' code.

(i) Wherever / appears in the representation, this signifies that the character (or characters) is irrelevant.

(ii) "Zero" is the very small number, $x.2^{-256}$, where x is the numerical part (in standard form) of the number transferred.

(iii) B-lines. The number (< 1024) in the specified B-line will be added to the address referred to in the 'instruction' before it is 'obeyed'. Only B2 - B6 inclusive are available however. An example of the use of the B tube with FLOATCODE is given below in example 2.

(iv) The dummy 'instruction'. / / / P.

For the purpose of numerical checking it may be necessary to print F (A) at suitable points in the programme. This can be done by inserting the 'instruction' 2n, / V at the required points, where n, refers to a closed routine which prints F (A). When this printing is no longer required this 'instruction' can be changed to the dummy 'instruction' so that useful results can be obtained at once without any rearrangement of 'instructions'. The resulting programme will not however be as fast as if the dummies were removed entirely because the dummy 'instruction' takes about 25ms.

(v) The modulus 'instruction', n, / M.

In addition to its use as an ordinary conditional transfer of 'control' this 'instruction' may be used to find $|F\ (A)|$ as follows. If s, / M standing in s, is 'obeyed', then 'control' is transferred to s + 1 with $F'(A) = |F\ (A)|$.

4. Examples illustrating the use of the 'instruction' code.

1. Given $z = x + iy$, where $x = F(/C)$ and $y = F(@C)$, place the real and imaginary parts of z^2 in :C and IC respectively n and $|z|^2$ in ½C.

Solution. The first 'instruction' is assumed to stand in line / / in order to fix our ideas.

```
/ | / | / / Q O |
  | E | E I / P |
  | @ | ½ C / Y |  'clear' A
  | A | @ C / G |
  | : | @ C / O |  -y^2 to ½C
  | S | ½ C / Y |
  | I | / C / G |
  | U | / C / Q |
  | ½ | ½ C / W |  x^2 - y^2 to :C
  | D | : C / Y |
  | R | : C / W |
  | J | ½ C / H |
  | N | ½ C / H |  x^2 + y^2 to ½C
  | F | ½ C / Y |
  | C | @ C / Q |
  | K | @ C / Q |  2xy to IC
  | T | I C / Y |
  | Z | Z / / £ |  returns control to L/ (i.e., leave FLOATCODE).
```

2. Place in / C the scalar product of the two vectors whose elements are in (i) / N, @ N, . . ., L N and (ii), / F, @ F, . . ., L F inclusive.

Solution. As in the previous example the 1st instruction is assumed to stand in / /. The sequence is entered at S /, assuming A is 'clear'.

```
    | / | A : Y G |
    | E | : / / T |  counting subsequence (see p. 5.14).
    | @ | I I / P |
    | A | H / / / |  = 20
    | : | S / / / |  control number
--> | S | A / Y O |  set B5 = 20
    | I | I / Q O |
    | U | E I / P |  enter FLOATCODE
    | ½ | / N S G |
    | D | / F S Q |  form product
    | R | £ £ / £ |  switch to counting subsequence.
    | J | / C / Y |  transfer to store
    | N | N / / £ |  leave FLOATCODE to obey the following ordinary instr.
    | F |   ...   |
    | C |   ...   |
    | K |   ...   |
```

Times of operation of the 'instructions'.

The times of operation of the 'instructions' other than division are as follows:

P, ", M, and £	25 ms.
G	30 ms.
W, and H	60 ms.
Y, Q, O, X, and V	80 ms.

Thus it appears that a programme drawn up in the 'instruction' code will be slower by a factor of about 60 than would otherwise be necessary. That this is not a realistic estimate will be seen from the following remarks.

Firstly those routines which consist largely of ordinary instructions (i.e., which 'work behind the scenes') will be little if any, slower than the corresponding routines designed for use in a fixed point context. This applies particularly to the division sequence of FLOATCODE and the routine for finding $\log_e$ F(A), where the numbers involved are already standardised. Input and output also will take the same time in both cases.

Secondly, because there are no scale factors to worry about, a programme which is based on the 'instruction' code uses less instruction lines than would otherwise be necessary. In other respects too, the 'instruction' code is more streamlined than the machine code, e.g., the discrimination and switching 'instructions'.

5. To use FLOATCODE

The four pages of instructions which make up FLOATCODE are given on the next few pages, in some cases with explanatory notes. The library tape for FLOATCODE contains all four pages and is punched in two parts as follows

Part 1

```
K/½/......
K/D/......        punching proper for material
K/R/......        destined for S2 and S3
K/J/......
YKP               destination sequence
```

Part 2

K/½/......	
K/D/......	punching proper for material destined for S4 and S5
K/R/......	
K/J/......	

The first part, if read by B.INPUT with H set to /A@E, causes the first two pages to be written on track 98 for subsequent use as a WORKING PERM: part 2 is intended to be treated as an ordinary two page routine and followed by a destination sequence which refers to a DIRECTORY setting in the usual way. The tape should thus be copied on to the problem tape after the DIRECTORY setting sequences, part 1 taking the place of the PERM alteration sequences.

The operations for starting a programme drawn up for FLOATCODE are similar to those described on p.4.23 for the case of R.C.S/B. Thus it is first necessary to bring down the four pages of the interpretive routine to S2, S3, S4, and S5; secondly to set B1 and the cue to B.INPUT on the level so defined; and finally, by entering the routine changing sequence which FLOATCODE incorporates, to call in the master routine. Moreover if the master routine is called in as an sub-routine of B.INPUT the actual instructions which effect these operations are identical in both cases. Thus once S4 and S5 have been set, the programme can be entered directly from the tape by a G sequence in the usual way. The starting sequences thus take the form

Lab	sets S4 and S5,
G(n)	sets S2 and S3; calls in nth routine as a sub-routine of B.INPUT.

The characters ab refer to the DIRECTORY entry for the material in part 2; they are thus identical with the corresponding characters of the destination sequence which follows part 2.

6. Auxiliary Routines.

A number of routines designed for use with FLOATCODE are available. They greatly increase its general utility. The following table summarises all the relevant information about the auxiliary routines. The library tapes are punched for use with B.INPUT.

Name of Routine. FC/DECPRINT:A

Purpose. To print F(A) in floating decimal form. For use with FLOATCODE, closed routine.

Cues. :///..//
½///..// (see note 3)

Magnetic Storage Variable Electronic Storage S0

Stores Altered None. May alter Q digit (B-sign)

Effects Prints F(A) in decimal floating point form, e.g., ¼ is printed as +2.5000 0000 -01

The layout and number of figures are fixed.

The cue with the lower control number gives C.R. and L.F. before printing.

Notes. 1. The two pages of the routine must be stored in the same track.

2. The cue must bring the left half track only to S.0.

3. FC/DECPRINT:A is allotted two magnetic entries in the DIRECTORY: one is the magnetic half cue for entry to the routine; the other is used merely for reading in. They take the form ..// and ..@/ respectively. The punching proper corresponds to S4 and S5.

4. The routine may be used if F(A) is not standardised, but in this case there may be a loss of accuracy.

Name of Routine. FC/DECINPUT.

Purpose. Decimal input for use with FLOATCODE, closed routine.

Cue. GJ//..@½

Magnetic Storage Variable. Electronic Storage. S4 and S5.

Stores Altered WI, HI, B5, B6, S4, and S5.

Effects Reads a sequence of numbers in floating decimal form $x = a \times 10^p$ $(1.0 \leq |a| < 10, |p| < 76)$, converts them into floating binary form $x = b \times 2^q$ $(\frac{1}{4} \leq |b| < \frac{1}{2}. |q| < 256)$, packs each into a single long line in the form suitable for FLOATCODE and puts them into successive long lines in the store, the first address being specified by B5 on entry. The number should be punched on the tape in the form a, + or -, p, + or - omitting the decimal point in a. Characters other than 0 - 9 +, - have the following effects:-

£ indicates error. Causes the part of the number already read to be ignored. The tape is then read for the corrected number, starting a fresh with the new a.

Y causes the next two characters to be taken as the address of the first number in a new sequence, i.e., $Y\alpha\beta$ sets $B5 = \alpha\beta//$ and returns to read further numbers.

" causes the routine to be replaced on S4 and S5 by FLOATCODE C and D and then enters R.C.S. by the instruction NS/P. The magnetic instruction concerned stands in line HD. It is I/@D which assumes that FLOATCODE pt II is stored on track 6. This instruction can however be altered by following the punching proper with a sequence of the form KHDE..@D. The punching proper corresponds to S4 and S5.

Any other character is assumed to be a punching error, and the machine comes to a dynamic stop in line $M\frac{1}{2}$.

Numbers can be placed anywhere in the store (including S0 and S1) except on S2, S4, S5, and the lines WI, HI, and the R.C.S.

Notes. 1. WI and HI, which are working space in FLOATCODE, have the final values HD/: and NS/P respectively.

2. If it is desired to punch the numbers on a separate tape this should start with a Y-sequence. Blank Tape before this will have no effect.

NAME	EFFECT	TIME OF OPERATION	MAGNETIC STORAGE	ELECTRONIC STORAGE	CUES	OTHER ROUTINES USED
FC:SQUAREROOT	$F'(A) = \sqrt{F(A)}$	0.1 sec.	Variable	S0	££//....	------
FC:NATURALOG	$F'(A) = \log_e F(A)$	0.75 sec.	Variable	S0	££//....	------
FC:ARCOT	$F'(A) = \cot^{-1} F(A)$	2.5 sec.	Variable	S0	££Q£....	FC:SQUAREROOT

Notes.

1. FC:ARCOT uses FC:SQUAREROOT as a sub-routine. The 'instruction', . . / V, which calls it in depends on the enumeration of the routine in the problem. This 'instruction' can be set during input by punching KJ½E../V at the head of the routine when copying it on a problem tape.
2. In the case of the routine FC:NATURALOG, F(A) must be in standard form. To ensure this first 'clear' A then add in the argument.
3. No lines of the 'store' are altered by any of these routines (the 'store' denotes S0, S6, S7, and B2 - B6).

INTERCODE.

```
                             I  /  £ £ V O   } entry instructions
Entry from sequence  ------->   E  / I Q G   }
of ordinary                     @  S U / P   }
instructions                    A            }
                                :            }
                                S            }
                                I            }
                                U            }
                                ½            } subsequence directory
                                D            }
                                R            }
                                J            }
                                N            }
                                F  G I / /   }  control number for X 'instruction'
                                C  " I / /   }     "      "     "  V      "     "
                                K  / / / /   }     "      "     "  £      "     "
                                T  / / / /   } working space
                                Z  / / / /   }
                                L  @ E / /      magnetic instruction: 34L to SO
                                W  @ / / /   } constants (N.B., if PERM is
                                H  / / / /   } available these can be dispensed
                                Y  / E / /   } with and the entry instruction
                                P  / / / /   } placed here instead of in lines /I,
                                Q  £ £ £ £   } EI, and @I).
                                O  £ £ £ /   }
                                B  M £ £ £   } further constants
                                G  £ / / /   }
Entry for ad-routine  ------>   "  A : Z G   }
  "    " sub-routine ------->   M  L I / :   }
                                X  K I Q O   }
                                V  / / Q /   } Routine changing sequence:
                                £  L U Z A   } similar to instructions QS to £S
                             U  /  B I Z G   } of R.C.S/B
Entry from closed  --------->   E  A : Z G   }
routines (by AU/P)              @  Z U E :   }
                                A  / U / G   }
Entry from  ---------------->   :  T U Z T      select 'control' number from list
subsequence (see  --------->    S  I U / L
note 2).                 |      I  T U E T      test digit 19 of control number *
Entry from the ----------'      U  D S Q G      adjust 'control' number
entry instructions              ½  T U Z Z      replace 'control' number
                                D  / / K C   } select 'instruction' from 'programme'
                                R  / / £ /   }
                                J  O I T R   }
                                N  K I T A   } unpack 'instruction': 1st 3
                                F  J : / N   } characters (address) are placed in
                                C  B I T R   } KI, last character is placed in ZI.
                                K  T I T A   }
                                T  Z I Q O   }
                                Z  T S U P      switch to appropriate sequence
```

* (When entering new routines).

Notes. 1. If the programme involves the j th level, then line pairs IU, HU, .., (2j)U are used by INTERCODE.

2. The subsequence of instructions specified in the directory must be terminated by an instruction which transfers control to :U. In FLOATCODE this is achieved by setting $\left\{I\ I_{+}\right\}_{0}^{9} = AU$ and terminating each subsequence by II/P so that the P - 'instruction' becomes effectively a dummy 'instruction'. Alternatively, if the constants

in lines W I to QI inclusive are dispensed with, then the entry instructions and the control number A U can be placed in these lines. This scheme has the advantage that the subsequence directory can be extended to occupy lines /I, EI and @I.

Programme Sheet 2.

ROUTINE. FLOATCODE/A & FLOATCODE/B.

```
 : | / / / / | / | / / / / | S      I | £ £ V O | / | B I Z G | U
   | E / / / | E | / / / @ |          | / I Q G | E | A : Z G |
   | / / / / | @ | / / / / |          | S U / P | @ | Z U E : |
   | @ / / / | A | / / / : |          | £ U / / | A | / U / G |
   | / / / / | : | / / / / |          | £ U / / | : | T U Z T |
   | : / / / | S | / / / ½ |          | £ D / / | S | I U / L |
   | / / / / | I | / / / / |          | A U / / | I | T U E T |
   | ½ / / / | U | / / / T |          | £ U / / | U | D S Q G |
   | / / / / | ½ | / / / / |          | £ U / / | ½ | T U Z Z |
   | T / / / | D | £ £ £ £ |          | £ U / / | D | / / K C |
   | / / / / | R | £ £ £ £ |          | £ U / / | R | / / £ / |
   | / E / / | J | / / / / |          | @ J / / | J | O I T R |
   | / / / / | N | / U / / |          | / J / / | N | K I T A |
   | / @ / / | F | / / / / |          | G I / / | F | J : / N |
   | / / / / | C | / / / ½ |          | " I / / | C | B I T R |
   | / : / / | K | / T £ £ |          | / / / / | K | T I T A |
   | / / / / | T | / / / / |          | / / I / | T | Z I Q O |
   | / ½ / / | Z | / / / / |          | / / / / | Z | T S U P |
   | / / / / | L | / / / / |          | @ E / / | L |         |
   | / T / / | W | / / / / |          | / / / / | W |         |
   | / / / / | H | / / / / |          | / / / / | H |         |
   | / / E / | Y | / / / / |          | / / / / | Y |         |
   | / / / / | P | / / / / |          | / / / / | P |         |
   | / / @ / | Q | / I / / |          | / / / / | Q |         |
   | / / / / | O | O ½ / / |          | £ £ £ / | O |         |
   | / / : / | B | P ½ / / |          | M £ £ £ | B |         |
   | / / / / | G | / / / / |          | £ / / / | G |         |
   | / / ½ / | " | / / / / |          | A : Z G | " |         |
   | / / / / | M | C ½ / / |          | L I / : | M |         |
   | / / T / | X | C ½ / / |          | K I Q O | X |         |
   | / / / / | V | S J / / |          | / / Q / | V |         |
   | / / / E | £ | D ½ / / |          | L U Z A | £ |         |
```

Programme Sheet 2.

ROUTINE. FLOATCODE/C & FLOATCODE/D.

```
½|K I Q O|/|F S / K|D      R|K I Q O|/|/ / / /|J
  £ D Q G|E|W S T /          V J Q G|E|F S T F
  A ½ Q B|@|F S T A          O R Q B|@|P R / H
  / / / K|A|/ : Q O          K S Q O|A|K I Q O
  Y : / N|:|P S Q G          W R / P|:|£ R Q G
  W : T D|S|P S Q B          / / £ £|S|S U / P
  W S / U|I|O D Q O          A : Q G|I|X J Q O
  Y S / U|U|P S Q G          M R / J|U|Y S Q G
  Z I Q O|½|S : / O          F S / J|½|L S Q B
  S S U P|D|D S U N          F S / E|D|: S T /
  W S T /|R|X D Q G          : S / J|R|A : / Q
  T S T A|J|A : / O          C R / H|J|W S T A
  Y S T /|N|I I / P          W : Q G|N|T S T F
  L S T A|F|/ : U N          D S / O|F|T S T A
  I I / P|C|I S / C          S R Q G|C|W S T F
  T S / K|K|F S / F          E : / O|K|" J / H
  W S / F|T|F S / E          / : Q O|T|W S T K
  I S T I|Z|I S / N          W : Q G|Z|W S T A
  W S / A|L|I D / H          K S Q B|L|W S / K
  L S T /|W|K S Q O          U R T :|W|T S / D
  Y S T C|H|V D Q G          H : / C|H|T S / J
  Y S T A|Y|K S Q B          F S / N|Y|I S T I
  E S U P|P|F S T ½          G D / I|P|T S / A
  W S T F|Q|A : / Q          K S / N|Q|T J T :
  W S T A|O|I E / /          / / T A|O|W S / D
  W S / K|B|F S T K          V R Q O|B|I S T I
  K S Q O|G|E : T D          K S Q B|G|I I / H
  Y S Q G|"|F S T A          I I / P|"|R J / I
  U : / O|M|I I / P          / / / /|M|Q J / P
  P S Q B|X|G V £ £          / / / O|X|I / / /
  Y S Q O|V|V £ £ £          / T £ £|V|/ / T M
  K S Q B|£|/ / / Z          / / ½ /|£|/ / / /
```

ROUTINE. FLOATCODE/C.

```
        ½ | / | K I Q O  } plant selection instruction
            E | £ D Q G  }
            @ | A ½ Q B  }
            A | / / / K  }
            : | Y : / N  }
            S | W : T D  } unpack operand
            I | W S / U  }
            U | Y S / U  }
            ½ | Z I Q O  } multiway switch
            D | S S U P  }
    (G)---> R | W S T /  }
            J | T S T A  } G-'order' : replaces F(R) by F(n)
            N | Y S T /  }
            F | L S T A  }
            C | I I / P    return to 'control'
  (Q,O)---> K | T S / K  }
            T | W S / F  }
            Z | I S T I  }
            L | W S / A  } Q & O - 'orders'
            W | L S T /  }
            H | Y S T C  }
            Y | Y S T A  }
            P | E S U P    2 way switch to Q½ or B½
    (H)---> Q | W S T F  } change sign of operand in the case of an
            O | W S T A  }        H - 'order'
    (W)---> B | W S / K
            G | K S Q O
            " | Y S Q G
      ┌---- M | U : / O    Jump if   p ≥ q
      |     X | P S Q B  | Interchange roles of addend and angend
      |     V | Y S Q O  | and place larger exponent in KS.
      ↓     £ | K S Q B  |
```

ROUTINE. FLOATCODE/C (2nd column).

```
 D  / | F S / K    (See note at foot of previous column)
    E | W S T /
    @ | F S T A
    A | / : Q O
    : | P S Q G
    S | P S Q B
    I | O D Q O    arranges that numerical part of smaller
    U | P S Q G    number is multiplied by the appropriate
    ½ | S : / O    (negative) power of two before combining
    D | D S U N    the numbers.
    R | X D Q G
    J | A : / O
    N | I I / P    return to 'control' if exponents differ by
    F | / : U N            more than 40
    C | I S / C
    K | F S / F    combine numerical parts
    T | F S / E
    Z | I S / N
    L | I D / H    test range of answer
    W | K S Q O
    H | V D Q G    correct exponent
    Y | K S Q B
    P | F S T ½
    Q | A : / Q
    O | I E / /
    B | F S T K
    G | E : T D    Round-off by making last digit odd.
    " | F S T A
    M | I I / P    return to 'control'
    X | G V £ £
    V | V £ £ £
    £ | / / / Z
```

ROUTINE FLOATCODE/D.

```
(Y) → R  / | K I Q O   }
         E | V J Q G   }  Plants transfer order
         @ | O R Q B   }
         A | K S Q O
         : | W R / P
         S | / / £ £      = -1024
       → I | A : Q G   }
         U | M R / J   }
         ½ | F S / J   }  standardise numerical part, adjusting
         D | F S / E   }  the exponent accordingly.
         R | : S / J   }
         J | C R / H   }
         N | W : Q G
         F | D S / O      Dynamic stop if p (corrected) ≯ 256
         C | S R Q G
         K | E : / O   }  If p (corrected) < - 512, then replace
         T | / : Q O   }     p by  - 512
       → Z | W : Q G   }
         L | K S Q B
         W | U R T :
         H | H : / C   }
         Y | F S / N   }  packing sequence
         P | G D / I   }
         Q | K S / N   }
         O |(/ / T A)     planted transfer order
         B | V R Q O   }  set exponent of F(A) = -512
         G | K S Q B   }
         " | I I / P      return to 'control'
         M | / / / /   }  = - ¼
         X | / / / O   }
         V | / T £ £      = - 512
         £ | / / ½ /
```

ROUTINE. FLOATCODE/D (2nd column)

```
          |J| / | / / / / |
(M)  -->    E | F S T F |  } M - 'order'
            @ | P R / H |  }
(")  -->    A | K I Q O |  }
            : | £ R Q G |  } " - 'order'
            S | S U / P |  }
(B)  -->    I | X J Q O |  } calculate exponent of
            U | Y S Q G |  } quotient
            ½ | L S Q B |  }
            D | : S T / |
     ┌----  R | A : / Q |
  ┌--│-->   J | W S T A |  }
  │  │      N | T S T F |  } Make divisor positive, changing
  │  └-->   F | T S T A |  } sign of dividend if necessary .
  │         C | W S T F |  }
  └-----    K | " J / H |  }
            T | W S T K |
  ┌----->   Z | W S T A |  }
  │         L | W S / K |  } Repetitive cycle:
  │         W | T S / D |  }
  │         H | T S / J |  }
  │         Y | I S T I |  }
  │         P | T S / A |  }
  │         Q | T J T : |  }
  │         O | W S / D |  }
  │         B | I S T I |  }
  │         G | I I / H |  }
  │         " | R J / I |  }
  └-----    M | Q J / P |  }
            X || I / / / |    = 6
            V || / / T M |    ≡ - (//TA)
            £ || / / / / |
```

Repetitive cycle:

$$a_{n+1} = a_n (1 - c_n)$$

$$c_{n+1} = -c_n^2$$

$$c_o = 2b-1$$

$$a_o = \tfrac{1}{4}$$

$$c_n \to o \qquad a_n \to \frac{1}{8\ b}$$

$= 6$

$\equiv$ - (//TA)

ROUTINE. FC:SQUAREROOT.

| /	/	CS/K	Calculates exponent of result p^1: $p^1 = \frac{p+1}{2}$ if p is odd; $p^1 = \frac{p+2}{2}$ if p is even.	
	E	SS/N		
	@	WSTA		
	A	IS/N		
	:	FS/K		
	S	C//I		
	I	/:TD		
	U	/:TC		
	½	CSTA		
	D	HSQO	forms a_o	
	R	:S/N		
	J	E:/O		
	N	:S/N		
	F	FS/E		
	C	DSTJ	form C_o	
	K	WS/A	plants C_{n+1}	repetitive cycle: $a_{n+1} = a_n (1 + \frac{C_n}{2})$ $C_{n+1} = C_n^2 (\frac{C_n}{4} + \frac{3}{4})$ $C_o = 1 - a_o$ $a_n \to \sqrt{a_o}$ $C_n \to o$
	T	WS/C	$a_{n+1} = a_n(1+\frac{C_n}{2})$	
	Z	FS/D		
	L	AU/H	Test for end of process	
	W	TI/A		
	H	TI/K		
	Y	IS/½		
	P	FS/J		
	Q	FS/A		
	O	WS/C	$C_{n+1}= C_n^2(\frac{C_n}{4} + \frac{3}{4})$	
	B	:S/N		
	G	MR/J		
	"	TI/A		
	M	TI/N		
	X	TI/A		
	V	TI/N		
	£	S//P	repeat cycle	

ROUTINE. FC:NATURALOG.

/	/	F S T K	
	E	W I T A	✓
	@	W : / C	form $\frac{p}{2^{10}}$ and plant in FS
	A	C S / F	
	:	M / T R	?
	S	H : T N	
	I	F S T A	
	U	L : T F	set stroke
	½	Y I T A	repetitive cycle: calculates $\left[\frac{p}{2^{10}} + \left(\frac{\log_2 a}{2^{10}}\right)\right] = \frac{\log_2 a.2^p}{2^{10}}$ by a digit by digit process, using a shifting negative stroke.
	D	W I / C	
	R	W I / N	
	J	C / / H	
$(x^2 < 2)$ (see p.6.8)	N	I S T I	
$x^{1/2}$	F	W I / A	
	C	H / T :	
	K	I S / C	
	T	Y I / F	
	Z	I S T I	
	L	: / / H	
	W	U / / I	
	H	W / / P	
	Y	W I / N	$2 \cdot \frac{x_r^2}{4} = \frac{x_r^2}{2} = \frac{x_{r+1}}{2}$
	P	I S T I	
	Q	W I / A	
	O	Y I T ½	
	B	F S T C	
	G	F S T A	
	"	J / / P	
	M	/ / / /	
	X	/ / £ £	
	V	½ E Q O	/set exponent of result: p = 10
	£	K S Q B	
E	/	I E / C	$\log_e a.2^p = \log_2 a.2^p \times \log_e 2$
	E	F S / F	
	@	/ / / I	round-off
	A	E : T D	
	:	F S T A	replace
	S	A U / P	close
	I	R J X K	$\log_e 2$
	U	E B S P	
	½	H / / /	

ROUTINE. FC:ARCOT.

```
/ | / | A / / H |   test sign of x; finds modulus
  | E | W I / Y |
  | @ | M / / H |
  | A | S / / " |   set t_o = |x| and record sign of x
  | : | W I / Y |   by planting 2" or -2"
  | S | M / / W |
  | I | Y I / Y |
  | U | Z / / £ |   set counter (by subsequence)
  | ½ | W I / G |
  | D | W I / Q |   repetitive cycle:
  | R | G / / W |
  | J | . . / V |   t_{u+1} = t_n + √(1+t_n^2)
  | N | W I / W |
  | F | W I / Y |
  | C | W / / £ |
  | K | W I / B |   form ± 2"/t"
  | T | Y I / Q |
  | Z | / U / £ |   closure
  | L | B / Y O |
  | W | I I / P |
  | H | O / Q O |
  | Y | D S Y G |   counting subsequence
  | P | I I / T |
  | Q | S U / P |
  | O | U / O £ |
  | B | Z £ £ £ |
  | G | / / / / |   F(G/) = 1
  | " | / ½ : / |
  | M | / / / / |   F(M/) = 2"
  | X | / ½ @ E |
  | V |         |
  | £ |         |
```

Chapter 6.

The calculation of functions of a single variable.

Introduction.

Many problems in computation require routines for the evaluation of functions of one variable. In hand computing function values are generally obtained by interpolation in tables. With an electronic machine this method can still be used but it is often preferable to employ more direet methods of calculating functions. The methods available can be classified as follows:

1. Use of power series
2. Iterative
3. Repetitive and digital methods.

Power Series.

It is often possible to represent a function and its derivatives to the accuracy desired, over a restricted range of the argument, by a polynomial thus

$$f(x) = \sum_{r=0}^{n} a_r x^r \quad (x_o < x < x_1).$$

If the function cannot be represented to the desired accuracy in this way, then it may be possible to divide the range and use separate approximations in each interval.

A polynomial is a very convenient formula for machine calculation. Given the coefficients a_r, the numerical value of f(x) and its derivatives may be calculated by the well known recurrence relations

$$\begin{array}{lll} p_o = a_n & q_o = 0 & \rho_o = 0 \\ p_r = p_{r-1}x + a_{n-r} & q_r = q_{r-1}x + p_{r-1} & \rho_r = \rho_{r-1}x + q_{r-1} \\ p_n = f(x) & q_n = f'(x) & \rho_n = \dfrac{f''(x)}{2!} \end{array}$$

A sequence of instructions for evaluating the first set of relations is given in Chapter 2, Examples 2, 3.

The two principle methods of polynomial approximation are (1) the use of the Taylor series, and (2) the expansion in terms of Tchebychef polynomials.

The Taylor Series.

The simplest method of polynomial approximation is to truncate the Taylor expansion of the function. To minimise the error committed - called the truncation error - the expansion should be made about the mid-point of the range. In the case of a real alternating series the truncation error is less in modulus than the first term omitted.

Tchebychef Polynomials. (reference 1).

The Tchebychef polynomials $T_n(x)$ are defined in the range $(-1 \leqslant x \leqslant 1)$ by

$$T_n(x) = \cos(n \cos^{-1}x).$$

Of all the polynomials of degree n which have maximum modulus unity in this range $T_n(x)$ has the greatest coefficient of x^n. This is 2^n. Thus if the expansion

$$f(x) = \sum_{r=0}^{\infty} a_r T_r(x)$$

is truncated after the nth term, then the error committed is less than $\sum_{n+1}^{\infty} |a_r|$ and it is distributed more or less evenly over the whole range - unlike the case of the Taylor expansion where the error is a maximum at the ends of the range.

Tchebychef polynomials can be used to reduce the number of terms in an approximation based on the Taylor expansion. If by a change of scale and origin the function is expanded in a Maclaurin series in the range (-1, +1), and the series truncated so that the maximum error is not exceeded, then the number of terms may be reduced by one in the following way:

$T_n(x) = 2^n x^n + \bar{T}_n(x)$, where $\bar{T}_n(x)$ is a polynomial of degree less than n. Let the last term retained in the truncated Maclaurin expansion be $a_n x^n$. If $a_n x^n$ is replaced by $-a_n 2^{-n}\bar{T}_n(x)$, then the modulus of the error commited is less than $|2^{-n}a_n T_n(x)| \leqslant 2^{-n}|a_n|$. Thus the term $a_n x^n$ has been replaced by an approximation consisting only of terms of lower degree in x. By a repetition of this process the series is reduced step by step until the sum of the maximum errors reaches the limit allowable.

The sum of the Tchebyschef series

$$f(x) = \sum_{r=0}^{n} a_r T_r(x)$$

can be calculated by a step-by-step method, similar to that for a power series, by means of the 3 term recurrence relations

$$\begin{aligned} p_o &= a_n \\ p_1 &= a_n x + a_{n-1} \\ p_{r+1} &= 2xp_r - p_{r-1} - xa_{n-r} + a_{n-(r+1)} \\ p_n &= f(x). \end{aligned}$$

These are based on the recurrence relation satisfied by T_r, namely,

$$T_r - 2xT_{r-1} + T_{r-2} = 0$$

The use of polynomial approximation is more economical of machine time than other methods but the calculation of the coefficients can be a tedious business if the full accuracy of the machine is required. A routine is being prepared however to reduce the number of terms in truncated Taylor expansions by the above method.

Functions which satisfy an algebraic equation.

If the value f(a) of the function to be calculated satisfies an algebraic equation $F(f(a),a) = 0$, then any method of solving this equation is a method for calculating f(a). An account of the methods available can be found in reference 2 and 3. They can be classified as follows:

1. Iterative, e.g., rule of false position, Newton's method;
2. Repetitive; and
3. Digit by digit method based on a Weirstrassian sub-division process.

In the next section we shall see that repetitive and digital methods are more general and can converge to functions not defined by an algebraic equation but by an algebraic addition law.

1. Iterative methods

An iterative method is one by which the root of the equation - the required value of the function - is obtained by a successive approximation process of the form

$$y_{n+1} = y_n + G(y_n,a),$$

where G is so chosen that $|y_{n+1} - y_n| \to 0$ as $n \to \infty$, and G(y,a) has the same roots as F(y,a).

An iterative process is said to have mth order convergence if the error at any stage, $e_n = y_n - y$, behaves according to a relation of the form

$$e_{n+1} = e_n^m \ (a_0 + a_1 e_n + \cdot \cdot \cdot).$$

The higher the value of m, the more rapid will be the convergence.

1. Taking G = kF usually yields a first order process (m = 1). For example, let F(y,a) = ay-1. Then $y_{n+1} = y_n + k(ay_n - 1)$ and e_n satisfies e_{n+1}: e_n (1+ka). It is necessary for convergence that $|1+ka| < 1$.

There is little advantage in using a first order process unless the value of a_o - called the degree of convergence in this case - is small enough to ensure rapid convergence. For hand computation a_o should be $\frac{1}{10}$ or less: with an electronic commutor values up to 0.95 can be tolerated.

2. Taking $G = -\frac{F}{F'}$, gives Newton's method (sometimes called the Newton - Raphson method). This is a second order process. The form of F should be chosen so that divisions are avoided in the iterative loop (unless the time spent in programmed division can be tolerated). For example let $f(x) = x^{-1}/_p$ where p is a positive integer. We have $F(y,a) = y^{-p} - a$ and the iterative formula is

$$y_{n+1} = y_n - \frac{(y_n^{-p} - a)}{-py_n^{-p-1}} = y_n + \frac{1}{p} (y_n - ay_n^{p+1})$$

No division operation is involved other than the calculation of $\frac{p+1}{p}$ which will presumably be a parameter of the routine.

If e_n is small, i.e., in the neighbourhood of the root, then $y_n^{p+1} = y^{p+1} + (p+1) y^p e_n + \frac{(p+1)p}{2} y^{p-1} e_n^2 + \cdot \cdot$, so that we obtain the relation

$$e_{n+1} = \tfrac{1}{2} (p+1) a^{\frac{1}{p}} e_n^2$$

for successive errors.

The determination of the range of values of the initial approximation for which convergence is assured will not be given here. It is possible that particular iterative processes will converge from any first approximation, but this should never be assumed without further investigation.

The number of iterations required depends on the error of the first approximation. For this reason iterative methods are very suitable for the calculation of functions appearing in the integration of differential equations, where suitable approximations are derived from the values at the previous step. Usually the number of iterations required in such cases is at the most two.

In testing for convergence of an iteration, the most convenient method is to test the difference between successive values of the iterate, and to end the process when the difference is 'sufficiently' small. In adjudging this criterion, it should be remembered that for second (and higher order) iterative processes, the error in the $n+1^{th}$ iterate is approximately equal to the square of that of the nth iterate and hence to the square of the difference between the nth and $n+1^{th}$ iterates. Thus if an accuracy of 2^{-2n} is required from a second-order iteration, the process of calculation need only be carried on until the difference between successive iterates is somewhat less than 2^{-n}, depending on the value of the term a_o in the relation between the errors of successive iterates. There is no advantage in continuing the process further.

The self-checking feature of iterative methods make them very attractive for hand computation; an error in the calculation of y_n at any step is equivalent to starting the next iteration from a nearby point and, if the error is small the total number of iterations will hardly be affected. Such methods can also be used in cases in which the unknown y is an ordered set of numbers - a vector or a matrix. They are suitable for use with computing machines for two reasons; they can be fast and make little demand on storage capacity.

2. Repetitive methods.

We now consider a type of process in which the root of the equation $F(y,a) = 0$ is produced as the limit of a sequence of values defined by a repetitive rule, of which the initial value(s) is a definite function of a. These methods do not possess the self-checking feature of the iterative type since the errors introduced can be cumulative and each process must be separately examined to see how the error behaves.

A repetive process can be derived from the 2nd order iterative process given above by eliminating the parameter a from two consecutive iterations.

We have

$$y_{n+1} = y_n + \frac{1}{p} y_n (1 - a\, y_n^{p})$$

$$= y_n + \frac{1}{p} y_n\, g_n, \text{ where } g_n = 1 - a\, y_n^{p} .$$

Eliminating a between g_n and g_{n+1}, we have

$$(1 - g_{n+1})\, y_n^p = (1 - g_n)\, y_{n+1}^p ,$$

$$= (1 - g_n)\, y_n^p \left(1 + \frac{1}{p} g_n\right)^p .$$

Hence
$$g_{n+1} = 1 - (1 - g_n)\left(1 + \frac{1}{p} g_n\right)^p .$$

The process thus becomes the repetitive process

$$y_{n+1} = y_n + \frac{1}{p} y_n\, g_n$$

$$g_{n+1} = 1 - (1 - g_n)\left(1 + \frac{1}{p} g_n\right)^p$$

in which the parameter a appears only in the initial conditions

$$g_o = 1 - a\, y_o^p ,$$

y_o being arbitrary within some definite range. The process is quadratically convergent because the y_n are equal to the corresponding iterates in the original iterative process. If the initial conditions are altered to:

$g_o = 1 - a$, y_o still being arbitary, then y_n converges to $y_o\, a^{-1}/p$. $\left[\text{This follows from writing } g_o = 1 - a = 1 - \left(\frac{a}{y_o^p}\right) y_o^p \right]$.

The following are the first four members of this class of repetitive formulae

$$y_{n+1} = y_n + \frac{1}{p} y_n\, g_n ,$$

$$p = 1, \quad g_{n+1} = g_n^2 .$$

$$p = 2, \quad g_{n+1} = g_n^2 \left(\frac{3}{4} + \frac{1}{4} g_n\right)$$

$$p = 3, \quad g_{n+1} = g_n^2 \left(\frac{2}{3} + \frac{8}{27} g_n + \frac{1}{27} g_n^2\right)$$

$$p = 4, \quad g_{n+1} = g_n^2 \left(\frac{5}{8} + \frac{5}{16} g_n + \frac{15}{256} g_n^2 + \frac{1}{256} g_n^3\right)$$

$$g_o = 1 - a,$$

$$y_n \rightarrow y_o a^{-1/p} \text{ as } g_n \rightarrow 0.$$

By setting y_o to certain powers of a square roots, cube roots, and a variety of fractional powers may be calculated. These processes converge for $0 < x < \frac{1}{2}$ and, if the final result is within range, then all the intermediate steps will be also. The process corresponding to the case p = 1 is illustrated by examples 5, 2, of Chapter 2 and in the interpretive routine FLOATCODE.

3. Digit-by-digit methods.

These are methods for extracting the successive binary digits of the root, beginning with the most significant digit and ending when the desired (or maximum) accuracy has been obtained. The familiar methods of long division, extraction of square roots, and Horner's process, are examples of digit by digit methods in the decimal system. In the binary system the choice of whether a digit is 0 or 1 may be made to depend on the sign of some numerical quantity.

The general form of this process is as follows. Let F(y,a) have one root in the interval (y_1, y_2) and let $F(y_1, a)$ and $F(y_2, a)$ be of opposite sign. If the interval is halved, then the sign of the mid-ordinate $F(\frac{y_1 + y_2}{2}, a)$ - will indicate which of the two half intervals contains the root. By a repetition of this process (first used by Weirstrass) the interval containing the root can be made as small as desired.

A suitable form for machine use is the following: Let F(y,a) have one root in the interval $(0 \leq x < 1)$ and suppose that F has a positive, non zero, gradient in this range and that $F(0,a) < 0$. A digit-by-digit process for calculating the root is:

$$y_{n+1} = y_n + 2^{-(n+1)}, \text{ if } F(y_n + 2^{-(n+1)}, a) > 0;$$

$$y_{n+1} = y_n \text{ otherwise}$$

where initially $y_o = 0$ and finally $y_{40} = f(a)$, if the full accuracy of the machine is used. It is assumed that F does not exceed the capacity of the machine (± convention) at any stage, i.e., that

$\frac{1}{2} > F \geq -\frac{1}{2}$.

If F(y,a) has a negative gradient and F(0,a) > 0, then the conditions for the addition of $2^{-(n+1)}$ should be reversed. If the process is terminated at any stage, then the resulting approximation is biased, i.e., always in defect or excess of the actual root. This can be avoided by using the non-restoring form of the process which, in the case of a function with a negative slope, is

$$y_o = 0, \quad y_{n+1} = y_n + 2^{-(n+1)} \text{ if } F(y_n, a) \geq 0;$$

$$y_{n+1} = y_n - 2^{-(n+1)} \text{ otherwise.}$$

With this process the last digit of any approximation is a 1.

The error of the process is approximately $2^{-40} \Big/ \frac{dF}{dy}$, which may be quite large, as may be seen by considering the process for the square root function when the argument a is near zero $(F(y) \doteqdot y^2)$.

<u>Functions which satisfy an algebraic addition law.</u>

Digit-by-digit and repetitive processes can be used to calculate the logarithmic and trigonometrical functions; they are usually derived from the addition law satisfied by the function. Some examples are given here.

<u>The Logarithm Function.</u>

This function satisfies the relation

$$\log x^2 = 2 \log x.$$

Let the binary expansion of $\log_2 x$ $(2 > x \geq 1)$ be

$$0.\alpha_1 \alpha_2 \alpha_3 \alpha_4 \;\ldots.$$

Hence $\log_2 x^2 = \alpha_1 . \alpha_2 \alpha_3 \alpha_4 \ldots$.

If $x^2 \geq 2$, then $\alpha_1 = 1$; otherwise $\alpha_1 = 0$.

If $\alpha_1 = 1$, then $\log_2(x^2/2) = 0.\alpha_2 \alpha_3 \alpha_4 \ldots$; otherwise

$$\log_2 x^2 = 0.\alpha_2 \alpha_3 \alpha_4 \ldots .$$

A repetition of this process with one of the last relations determines α_2, and so on.

<u>The inverse trigonometrical functions.</u>

A digit-by-digit process for determining x given the values

$\sin \frac{\pi}{2} x$ and $\cos \frac{\pi}{2} x$ is based on the relations:

$$\sin 2\theta = 2 \sin \theta \cos \theta,$$
$$\cos 2\theta = \cos^2\theta - \sin^2\theta;$$

and the property

$$\sin \theta \geqslant \cos \theta, \text{ if } \theta \geqslant \frac{\pi}{4}; \ \sin \theta < \cos \theta \text{ otherwise.}$$

Let the binary expansion of x be

$$x = 0.\alpha_1 \alpha_2 \alpha_3 \alpha_4 \dots .$$

Let $\sin \frac{\pi x}{2}$ and $\cos \frac{\pi x}{2}$ be denoted by S_o and C_o respectively.

If $S_o \geqslant C_o$, then $\alpha_1 = 1$; otherwise $\alpha_1 = 0$. If $\alpha_1 = 0$ compute $S_1 = \sin \frac{\pi}{2}(2x)$ and $C_1 = \cos \frac{\pi}{2}(2x)$ by means of the relation

$$S_1 = 2S_o C_o,$$
$$C_1 = C_o^2 - S_o^2;$$

otherwise compute $S_1 = \sin\left(\frac{\pi}{2}(2x) - \frac{\pi}{2}\right)$ and $C_1 = \cos\left(\frac{\pi}{2}(2x) - \frac{\pi}{2}\right)$ by means of

$$S_1 = S_o^2 - C_o^2, \quad C_1 = 2C_o S_o.$$

Repeat the process with S_1 and C_1; and so on.

[illegible] process for finding the arcotangent function makes use of the repetitive sequence

$$t_{n+1} = t_n + \sqrt{1 + t_n^2}.$$

If $t_o = \cot \theta$, then $t_n = \cot \frac{\theta}{2^n} = \frac{2^n}{\theta}$ for sufficiently large values of a. Accuracy is lost in the initial stage t_o is large and negative but, without loss of generality, t_o can be limited to positive values.

The Exponential function.

A method for calculating the exponential functions makes use of the recurrence relation

$$y_{n-1} = y_n + \frac{y_n^2}{2^{n+1}},$$

which is satisfied by the function $y = 2^n\left(e^{2^{-n}x} - 1\right)$.

Starting with $y_{39} = 2^{39}\left\{(1 + 2^{-39}x + \ldots) - 1\right\} \div x$ the repetitive sequence is wound backwards to obtain $y_o = e^x - 1$. The method is illustrated by example, 5.1 of Chapter 2.

Interpolation

We conclude this Chapter with one or two remarks about the machine aspect of interpolation.

Interpolation is a process for finding the value y of a function $y = f(x)$ for any value of x given, not the form of the function, but a double entry table (x_1, y_1), (x_2, y_2), . . ., (x_n, y_n), where $y_r = f(x_r)$ and x lies in the range of arguments. In many cases the arguments will be entered at a fixed interval. A method of polynomial interpolation - Neville's method - which is very suitable for machine use is described by W.E.Milne (Numerical Calculus, Princeton University Press, 1949, - p. 72). If the arguments are not given at fixed intervals, then division operations will be encountered which make the process considerably slower than would otherwise be the case.

Inverse interpolation can be performed by Neville's method merely by interchanging the roles of the dependent and independent variables.

On the whole, interpolation is an unsatisfactory method of calculation with the machine as the demands on electronic storage are excessive and, if the tabular values are not calculated by the machine as for example in the solution of a differential equation, but are inserted on punched tape, there is the additional labour of punching and checking the tabular values.

1. Langos, C., Trigonometric Interpolation of Emperical and Analytical Functions, J. Math. Phys. 17, pp. 123-199.

2. Brooker, R.A., The Solution of Algegraic Equations on the EDSAC, Proc. Camb. Phil. Soc., 48.2

3. Olver, F.W.J., The Evolution of Zeros of High-Degree Polynomials, Phil. Trans. Roy. Soc., A, 244, pp. 385-415.

Provisional Summary of the Routines for Calculating Algebraic and Elementary Functions

The routine listed below can be used with either scheme A or scheme B: alternative cues are provided (columns 7 & 8). They are all intended for variable magnetic storage and for electronic storage S1, except in the case of two page routines where the electronic storage is S0 and S1. For one page routines the check characters in the scheme A cues are calculated on the assumption that [/ E] = EE//. The principle line [/ A] is given in the 6th column and in the case of two page routines [/ E] is given immediately below [/ A].

The routines all relate to functions of one, two, or three variables x, y, and z denoting either $[/\ C]_\rho$, $[@\ C]_\rho$, and $[:\ C]_\rho$ or $[L]_\rho$, $[M]_\rho$, and D_ρ according as to whether the routine is called in by R.C.S. or R.C.S/B. We use x_+ and $x_\pm$ to denote $[/\ C]_{+\rho}$ and $[/\ C]_{\pm\rho}$ respectively: similarly for y and z. In general the function value is a single 40 digit number and this will be found in L on leaving the routine. Where the routine calculates two forty digit numbers then these will be found in L and M or /C and @C according as to whether the routine is used with R.C.S. or R.C.S/B.

For all library tapes the name of the routine (1st column) is written both on the tape and the box which contains it. In the case of scheme A the name is also in the titling sequence.

Authors

R. A. Brooker, A. E. Glennie, N. E. Hoskin, R. K. Livesley, C. M. Popplewell, and A. M. Turing.

NAME	EFFECTS	ACCURACY	TIME IN ms.	NOTES	/A /E	A CUE	B CUE	AUTHOR
DIVISION/A	$[L']_{\pm f} = (y_{\pm} + 2^{-40}x_{+})/z_{\pm}$	Number of significant figures is 2 less than the lesser number of significant figures in the divisor & dividend	690-15m	$2^{m} \leqslant \lvert z \rvert < 2^{m+1}$	VA/A	£E@G HEEZ	A@.. ...@	A.E.G.
DIVISION/B	$[@C']_{\pm} + [/C']_{+f} = x_{\pm}/z_{\pm}$ (SCHEME A) $[M']_{\pm} + [L']_{+f} = x_{\pm}/z_{\pm}$ (SCHEME B)	As for DIVISION/A	595-12m	$2^{m} \leqslant \lvert y \rvert < 2^{m+1}$	MA/J	/@:L HEEZ	:@.. ...@	R.K.L.
ARCTAN/A	$[L']_{\pm f} = \frac{1}{2\pi} \arctan(2x_{\pm})$	maximum error = ± 2^{-24}	65		N///	£EY/ ZEEZ	E@.. ...@	R.K.L.
ARTAN/B	If $x_{\pm} = \frac{1}{4} \cos 2\pi\theta + R(\xi)$ $y_{\pm} = \frac{1}{4} \sin 2\pi\theta + I(\xi)$, then $[L']_{\pm f} = \theta$	maximum error = $0.74\xi + 3.2^{-40}$	72	2 page routine for S0 and S1	NS/P OA/J	££JW ZEEZ	@/.. ..@/	A.M.T. C.H.P.
TANGENT/A	$[L']_{+f} = \frac{1}{2} \tan(\pi x_{+}/4)$	maximum error = 2^{-34}	80		/K//	£E/L REEZ	E@.. ...@	R.A.B.
SINANDCOS/A	$[L']_{\pm f} = \frac{1}{2} \sin 2\pi x_{\pm}$ $[L']_{\pm f} = \frac{1}{2} \cos 2\pi x_{\pm}$	maximum error = 16.2^{-40}	75 70	Sine: if $x = \frac{1}{4}$, then $[L'] =$ M£££ ££££; if $x = -\frac{1}{4}$, then $[L'] =$ //// ///T. Cosine: corresponding results are obtained for $x = 0$, & $\pm\frac{1}{2}$ respectively	VA/K	£E@C JEEZ AE@C JEEZ	E@.. ...@ S@.. ...@	A.M.T.

NAME	EFFECT	ACCURACY	TIME IN ms.	NOTES	/A /E	A CUE	B CUE	AUTHOR
EXPONENTIAL/A	If $x_{\pm} < 0$ and $[B2]_{+} = p$, then $[L']_{+f_{1}} = \exp(2^{p}x)$	maximum error $= \pm 2^{-40}(1+2^{p-1})$, This occurs only when $\exp(2^{p}x)$ is near unity.	80 + p(7.2)	$[B2'] =$ V£££	££££	£E@E NEEZ	E@.. ...@	R.A.B.
SQUAREROOT/A	If $x_{+} \geqslant 2^{-38}$, then $[L']_{+f} = \sqrt{x_{+}}$	The number of significant figures in the result is two less than the number of significant figures in the argument.	13o	If x is less than 2^{-38}, then the result is meaningless.	V.$\sqrt{\frac{1}{2}}$	£E@Y FEEZ	E@.. ...@	A.E.G.
CUBEROOT/A	If $x_{+} \geqslant 2^{-38}$, then $[L']_{+f} = 3\sqrt{x_{+}}$		150		GA/F	FEIT CEEZ	E@.. ...@	A.E.G.
SQUAREROOT/B	If $x_{+} \geqslant 3.2^{-40}$, then $[L']_{+f} = x+/\sqrt{y_{+}}$ $[L']_{+f} = \sqrt{x_{+}}$	The number of significant figures in the result is one less than the lesser number of significant figures in x and y	25(n+6)	$2^{-n-1} \leqslant x < 2^{-n}$ If x is less than 3.2^{-40} (or 5.2^{-40}) the result is meaningless	MA/S	@@:0 FEEZ £E:0 FEEZ	S@.. ...@ £@.. ...@	R.A.B.
CUBEROOT/B	If $x_{+} \geqslant 5.2^{-40}$, then $[L']_{+f} = x_{+}/3\sqrt{y_{+}}$		27(n+6)		SHXS	£EVI CEEZ	E@.. ...@	R.A.B.
NTHROOT/A	If B6 = n then $[L']_{+f} = n\sqrt{x_{+}}$	The number of significant figures is approximately $\sqrt{n}$ less that the number of significant figures in the argument.	906 + 480 n		MATA	£EHB KEEZ	E@.. ...@	N.E.H.
LOGARITHM/A	$[L']_{\pm f} = \frac{1}{64}\log_{e} x_{+}$	maximum error is 2^{-34}	80 + 9m	$2^{n}x_{+} \geqslant 1$	////	£EEE TEEZ	E@.. ...@	N.E.H.

B/ROOT/A

Effects

$[L']_{\pm f} = a$, where a is a root of $f(x) - y_{\pm} = 0$ (or of $f(x) - y_{+} = 0$), and $f(x)$ is calculated by an auxiliary sub-routine having the effect $[L']_{\pm f} = f(x_{\pm})$ (or the effect $[L']_{+f} = f(x_{\pm})$).

Initially $[L]_{\pm f}$ must contain an "approximation" x_0 to a such that $-\frac{1}{2} \leqslant x_0 < \frac{15}{32}$, that $|f(x_0)| < \frac{1}{2}| \Delta f/\Delta x_0)$, where $\Delta x_0 = \frac{1}{32}$, and that $-\frac{1}{2} \leqslant x_0 - \frac{\Delta f}{\Delta x_0}(f_0 - y) < \frac{1}{2}$.

Accuracy

See notes.

Time

(200 + T)p ms., where T is the time taken by the auxiliary sub-routine and p is the number of iterations required to satisfy the convergence criterion (see notes below).

Cue.

£ E . . . . . @

Notes

1. The routine has associated with it three 20 digit preset parameters. These are

a_1 b_1 //, giving the DIRECTORY number of the auxiliary sub-routine.

a_2 b_2 //, giving the first of 6 consecutive long lines to be used as working space by B/ROOT/A. These must not be used by the auxiliary sub-routine.

a_3 b_3 //, giving the address of ϵ , a small positive (40 digit) constant to be used in the convergence criterion; which is that two consecutive values of $f(x) - y$ must differ by less than ϵ . When this condition is satisfied the routine is quitted and the value of $f(x) - y$ is left in location (a_2 b_2 + 4)

These parameters must be standing in lines TU, ZU, and LU when B/ROOT/A is read. The routine tape makes use of an interlude (which temporarily occupies lines /U through KU) to select and plant these parameters at appropriate points within the actual routine (which is standing in S5) before this is transferred to the magnetic store.

2. Method. The root is approached by the rule of false position, the maximum size of the step being restricted to $\frac{1}{32}$. Within the 'neighbourhood' of the root the process amounts to inverse (linear) interpolation. Author A.E.G.

NAME	EFFECTS	ACCURACY	TIME IN m.s.	NOTES	/A /E	A CUE	B CUE	AUTHOR
INTERPOLATION/A	$[L']_{\pm f} = f(2^8 x_{\pm})$, where f is given by a table at unit interval, stored in consecutive long lines, the entry in /½ being $f(0)$. The entries will be interpreted on the $\pm f$ convention.	error $< \lvert 0.0005\, \Delta^4 f \rvert$	96	Method. Everett's method with modified 2nd differences. Three points on either side of the entry are used.	OA/D	£E½H LEEZ	E~~L~~G.. ...@	A.E.G.
INTERPOLATION/B	$[L']_{+f} = f(2^8 x_{\pm})$, f being represented as for INTERPOLATION/A, but on the $+f$ convention.	As for INTERPOLATION/..						A.E.G.
B/DIVISION/C	$[L']_{\pm f} = y_{\pm}/z_{\pm}$	The number of significant figures in the result is about 2 less than the lesser number of significant figures in the arguments.	80 + 6m	$2^{-m} \leqslant \lvert z \rvert < 2^{-m+1}$ An 'electronic' routine for use as part of a working PERM. Occupies lines /U to GU inclusive. The punching proper corresponds to lines /J to GJ.			£I.. ..£.	R.A.B.

Chapter 7.

Numerical Solution of Ordinary Differential Equations.

The methodsmost generally used are step-by-step methods, known as such because the values of the dependent variable are calculated one after the other for a sequence of equally spaced values of the independent variable. The majority are based on difference formulae and use previously computed values of the dependent variable to obtain the value at the next point. The accuracy of these formulae depend both on the number of neighbouring values considered and on the size of the interval between successive points. Thus to increase the accuracy of the method one may either use a more accurate formula, which will involve a largernumber of preceding values of the dependent variable, or decrease the size of the interval. However, decreasing the interval increases the number of times a substitution into the differential equation has to be made and, if this substitution is complicated, it may be advantageous to achieve the desired accuracy by using a more accurate formula.

In step-by-step methods it is particularly necessary to guard against errors of calculation as an error introduced at any stage will be carried forward throughout the remainder of the integration. The method described below, due to Milne, should if used correctly discover any error in computation when made and the value at that point may be recomputed. There are many other methods of similar type which may be found by reference to books on numerical analysis

Equations of the First Order.

Consider the equation

$$y' = f(x,y) \qquad y = y_0, \quad x = x_0$$

where y denotes the dependent variable, x the independant variable, and a dash denotes differentiation with respect to x. The equation is being solved at a set of points x_0, x_1, ..., x_n, ... where $x_{n+1} = x_n + h$, and suppose that the values $y_1 \ldots, y_n$ have been computed

Then the method consists of three steps;

1) A "predictor" formula which uses values of y_n, y_{n-1} .... to extrapolate to an approximate value of y_{n+1}

2) Substitution of this approximation into the differential equation to obtain $y'_{n+1} = f(x_{n+1}, y_{n+1})$.

3) A "corrector" formula to give a more accurate value of y_{n+1}, which is assumed to be correct.

The method is then applied to obtain y_{n+2} and the process continued.

Thus, for example the formulae,

$$\text{(a)} \quad y_{n+1} = y_{n-3} + \frac{4h}{3}(2y'_n - y'_{n-1} + 2y'_{n-2}) + \frac{28}{90}h^5y^{(5)}$$

$$\text{(b)} \quad y_{n+1} = y_{n-1} + \frac{h}{3}(y'_{n+1} + 4y'_n + y'_{n-1}) - \frac{1}{90}h^5y^{(5)} \qquad (1).$$

may be used, (a) as a predictor, and (b) as a corrector.

A check on any errors which may occur may be made by adding a fourth step to the three already stated. The terms at the end of each equation are estimates of the error involved in each, and the error in (a) is twenty eight times that in (b) and of opposite sign, if we assume that the fifth derivatives are the same. If E is the error in (b) and the values of y_{n+1} obtained by (a) and (b) are subtracted then the difference is of the order of 29E. This can provide a running check on the integration thus: this difference may be tested and so ascertain whether E has become too large. If so the interval must be shortened and with (a) and (b), cutting the interval by half will divide the error by about 32. If any error in computation has occured this difference will most probably be large and the computation for that point may be carried out again. If the difference is consistently large it may be assumed that the accuracy of the equation is less than required and the interval shortened.

This method, and any other method which uses preceding values of the dependent variable, suffers from the disadvantage of requiring a special starting technique. Equation (a) depends on four previous values and so the method cannot be applied until y_1, y_2, y_3 have been obtained by some other method (e.g., a Taylor's Series). This characteristic of the method which may not be important in hand computing, becomes a serious drawback when the method is applied by an automatic machine. A great deal of time is also taken up by shifting operations since, after the value of y_n has been computed, in order to apply the same set of instructions to calculate y_{n+1}, y_{n-1} must be replaced by y_n, y_{n-2} by y_{n-1}, and so on

There is however the method of solution developed by Kutta which does not involve previous values of y, and so does not suffer from either of these disadvantages. It may be used as the starting routine, or may be used as the main method throughout. If the value of $y = y_o$ is known at $x = x_o$ then to obtain the value of $y_1 = y_o + \Delta y$ at $x_1 = x_o + h$ the formulae in the well known Runge-Kutta form are

$$\begin{aligned}
k_1 &= hf(x_o, y_o) \\
k_2 &= hf(x_o + \tfrac{1}{2}h,\ y_o + \tfrac{1}{2}k_1) \\
k_3 &= hf(x_o + \tfrac{1}{2}h,\ y_o + \tfrac{1}{2}k_2) \\
k_4 &= hf(x_o + h,\ y_o + k_3) \\
\Delta y &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned} \qquad (2)$$

The truncation error associated with these formulae is of order h^5.

There is a modification of this method, due to Gill, which has the advantage of needing less storage space. In this form the formulae become

$$\begin{aligned}
k_1 &= hf(x_o, y_o) \\
k_2 &= hf(x_o + \tfrac{1}{2}h,\ y_o + \tfrac{1}{2}k_1) \\
k_3 &= hf(x_o + \tfrac{1}{2}h,\ y_o + \left\{-\tfrac{1}{2} + \sqrt{\tfrac{1}{2}}\right\} k_1 + \left\{1 - \sqrt{\tfrac{1}{2}}\right\} k_2 \\
k_4 &= hf(x_o + h,\ y_o - \sqrt{\tfrac{1}{2}}\ k_2 + \left\{1 - \sqrt{\tfrac{1}{2}}\right\} k_3. \\
\Delta y &= \frac{1}{6}\left[k_1 + (2 - \sqrt{2})\, k_2 + (2 + \sqrt{2})\, k_3 + k_4\right]
\end{aligned} \qquad (3)$$

Obtaining Δy, y_1 is thus known and the equations may be applied to obtain y_2 from y_1, in a similar manner. This method however has the disadvantages that there is no obvious method of checking whether errors in computation have occured, and that for each step four substitutions into the differential equation have to be made, a serious fault if the differential equation is complicated.

For an estimate of the error associated with (3) reference should be made to the paper by Gill.

Equations of Higher Order.

If all the boundary conditions of an equation, of order higher than the first, are specified at one point then the equation may be reduced to a set of simultaneous first order equations and either method given for a first order equation may be easily modified for solving them. For example an equation of the second order

$$y'' = f(x, y, y') \quad y = y_0,\ y' = y_0,\ x = x_0$$

may be put into the form

$$\begin{aligned} y' &= z \qquad\qquad y = y_0,\ z = z_0,\ x = x_0 \\ z' &= f(x, y, z) \end{aligned}$$

and a solution found to the two simultaneous first order equations.

Dynamical Equations.

Many differential equations of mathematical physics are of the second order with the first derivative not appearing explicitly. There are direct methods of solving these which are less laborious than breaking them down to simultaneous first order equations.

Let the equation be $y'' = f(x, y)$

Most methods are based on the central difference formula.

$$\delta^2 y_n = h^2 \left(f_n + \frac{1}{12}\,\delta^2 f_n + \frac{1}{240}\,\delta^4 f_n \ldots\right) \qquad (4)$$

Ignoring the term in $\delta^4 f_n$ this formula, in terms of ordinates gives

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{12}\left(y''_{n+1} + 10y''_n + y''_{n-1}\right) \frac{-1}{240}\, h^6 y^{(6)} \qquad 5(a)$$

and this formula is given by Milne as a "corrector" together with a "predictor" as follows.

$$y_{n+1} = y_n + y_{n-2} - y_{n-3} + \frac{h^2}{4}\left(5y''_n + 2y''_{n-1} + 5y''_{n-2}\right) \frac{+17}{240}\, h^6 y^{(6)} \qquad 5(b)$$

5(a) may be used as both "predictor" and "corrector" by making a direct estimate of y''_{n+1} from a difference table in order to predict a value of y_{n+1}.

The Gauss-Jackson method uses (4) in the form.

$$y_n = h^2 \left(\delta^{-2} f_n + \frac{1}{12} f_n + \ldots\ldots\right) \qquad (4')$$

where $\delta^{-2} f_n$ is known as the "second sum" of f_n. This is used to predict y_n by estimating f_n, and then also used as a corrector.

A method due to Herrick is similar to the Gauss-Jackson in that it uses the second sum procedure but it is based on backward differences instead of central differences. Thus no predictor is needed and the integration advances by the application of a formula once only. The backward difference formula is as follows:

$$y_n = h^2 \Big\{ \delta^{-2} f_n + \tag{6}$$

$$\frac{1}{12}\left[f_{n-1} + \delta f_{n-\frac{3}{2}} + \frac{19}{20} \delta^2 f_{n-2} + \frac{18}{20} \delta^3 f_{n-\frac{5}{2}} + \frac{1726}{2016} \delta^4 f_{n-3} + \frac{1650}{2016} \delta^5 f_{n-\frac{7}{2}} \cdots \right]\Big\}$$

This may be expressed in terms of function values in a set of formulae (6(m)) where (m) includes the effect of $\delta^m f$

$$y_n = h^2 \ \delta^{-2} f_n + \frac{1}{12} f_{n-1} \tag{6(0)}$$

$$y_n = h^2 \ \delta^{-2} f_n + \frac{1}{12}(2f_{n-1} - f_{n-2}) \tag{6(1)}$$

$$y_n = h^2 \ \delta^{-2} f_n + \frac{1}{240}(59f_{n-1} - 58f_{n-2} + 19f_{n-3}) \tag{6(2)}$$

$$y_n = h^2 \ \delta^{-2} f_n + \frac{1}{240}(77f_{n-1} - 112f_{n-2} + 73f_{n-3} - 18f_{n-4}) \tag{6(3)}$$

. . . .

There is however, with this method, no direct check on errors in computation and so is not to be recommended unless a check can be kept in some other way.

Jury Problems.

If the problem has closed boundaries and the boundary values are divided among them, the method of solving a set of simultaneous first order equations will not apply directly. If however the equation, which is of nth order say, is linear a solution may be obtained by solving the equation n times with differing boundary values all specified at one point and combining the solutions linearly to satisfy all the given conditions. For example, if the equation is of second order and the boundary conditions are $y = \alpha$ at $x = a$, $y = \beta$ at $x = b$, two solutions Y_1, Y_2 satisfying $Y_1(a) = \alpha$, $Y_1'(a) = 1$: $Y_2(a) = \alpha$, $Y_2'(a) = 0$

and can be combined thus

$$p\,Y_1\,(b) + q\,Y_2(b) \;=\; /$$

with $$p + q \;=\; 1$$

to yield a solution satisfying the required conditions.

If the equation is non-linear this method will not give a solution. However it can be used on the example just considered by solving for various Y_1 and thus straddling the condition at $x = b$. By interpolation, a solution can be obtained which satisfies conditions at both points.

Iterative methods.

A finite difference equation may be obtained from the differential equation and an approximate solution found satisfying the boundary conditions but not the equation. By applying the difference equation to this solution a better approximation is obtained and by continuing the process the approximations will converge to the required solution. However, unless care is taken in the construction of the difference equations, the process will most probably converge extremely slowly. It is preferable to use as large an interval as possible, using a finite difference equation which involves a larger number of neighbouring points. This reduces the number of simultaneous equations to be solved, and thus the possibility of ill-conditioning of the equations. The "difference correction" technique of Fox and Goodwin may be used, where an approximate solution is first obtained to a finite difference equation which ignores the correction term, and using this solution to calculate the correction. The full equation of finite difference terms plus correction term is then used and a solution to this is found. The correction terms is recomputed and the process continued until no change occurs in two successive corrections.

Thus for example the equation

$$y'' = f(x,y)$$

may be approximated by a finite difference equation

$$y_1 + y_{-1} - 2y_o - f(x_o,y_o) + \Delta = 0$$

where the correction term $\Delta = -\frac{1}{12}\delta_o^4 + \frac{1}{90}\delta_o^6 - \ldots\ldots$

The first solution is obtained by solving

$$y_1 + y_{-1} - 2y_o - f_o = 0 \text{ at each point}$$

from this $\Delta^{(1)}$ is calculated and the equation to be solved is then

$$y_1 + y_{-1} - 2y_o - f_o + \Delta^{(1)} = 0$$

From this $\Delta^{(2)}$ may be obtained and the process contin ed until $\Delta^{(n)} = \Delta^{(n+1)}$ to the required accuracy.

Organisation of the Method of Solution.

When solving differential equations on an automatic calculator, it must be remembered that operations which seem trivial to a human computer, may assume considerable importance when the method is applied by a machine. An example of this is the shifting operations necessary when using a method involving preceding function values.

For the purpose of illustration, we may consider the Gill-Kutta method of solution of the set of equations

$$\frac{dy_i}{dx} = f_i(x, y_1 \ldots y_n) \quad i = 1, 2 \ldots n.$$

for which a programme is already in the library of routines. In order to be as general as possible this only advances the integration of the equations by one step each time it is called in and needs a sub-routine to compute the f_i's. Thus a main routine is needed to call in the integrating routine, keep account of the number of steps already integrated and finally to print the results.

Gill-Kutta routine.

This routine, when called in as a sub-routine, replaces the $y_i(x_r)$ by the $y_i(x_r + h)$ in accordance with equations (3) which may be written as follows:

$$k_{i,\alpha} = 2^m h f_i (x_\alpha, y_{1,\alpha} \ldots y_{n,\alpha})$$

$$r_{i,\alpha+1} = A_\alpha (k_{i,\alpha} - q_{i,\alpha}) - B_\alpha q_{i,\alpha} \qquad \alpha = 0, 1, 2, 3$$

$$y_{i,\alpha+1} = y_{i,\alpha} + r_{i,\alpha+1} \qquad i = 1, 2, \ldots, n.$$

$$q_{i,\alpha+1} = q_{i,\alpha} + 3r_{i,\alpha+1} - (A_\alpha + 2B_\alpha)k_{i,\alpha}$$

$$y_{i,o} \equiv y_i(x_r) \qquad x_o \equiv x_r$$

$$y_{i,4} \equiv y_i(x_r + h).$$

The constants A_α and B_α are given by the following table

α	A_α	B_α
0	$\frac{1}{2}$	0
1	$1 - \sqrt{\frac{1}{2}}$	0
2	$1 + \sqrt{\frac{1}{2}}$	0
3	$\frac{1}{6}$	$\frac{1}{6}$

If the independent variable appears explicitly in the f_i's it may be obtained either by integrating $x' = 1$, or by adding $h/_2$ to its value at every other stage i.e., at $\alpha = 1$ and $\alpha = 3$. At the end of the step $y_{i,4}$ are the new values of the dependent variables and these are stored in n long lines beginning at /$\frac{1}{2}$. The $q_{i,4}$ become the $q_{i,o}$ for the next step and n long lines beginning at /R are needed to store them. These lines should be cleared at the beginning of the range of integration.

The auxiliary routine needed to calculate the quantities $k_i = 2^m hf_i$ is stored with its first page on the right half of the track holding the Gill-Kutta routine. It should begin at /@ and the full track should be brought down when the Gill-Kutta routine is entered. The auxiliary routine may then be entered by a control transfer instead of the routine changing sequence. The factor 2^m is chosen so that the k_i's are as large as possible without any being larger than 2^{38} and the routine stores the k_i's in n long lines beginning at /N.

References.

(1) Milne. W.E. Numerical Calculus (Princeton University Press)

(2) Gill. S. Proc. Camb. Phil. Soc. 47 (1951) 96.

(3) Herrick. S. M.T.A.C. 5 (1951) 61.

Gill-Kutta Routine.

```
Entry from
master            / I / / /
routine      >    E / / P O                                   set count (κ)
Auxiliary  ←      @ : / / P
routine    →      A C E Q O                                   set count (i)
                  : £ E T :
                  S E : P G
                  I A : / O                   1
                  U O E T K   set  P E   =    -               Test for fourth stage
                  ½ P E T A   at fourth stage 6
                  D D S P G
                  R / N Q ½   add 2^m h y'                                          this cycle
                  J / R Q N   subtract 2^m q                                        of instr. is
                  N V K T A   store 2^m(hy' - q)                                    obeyed for
                  F V K / K                                                         each
                  C O E I N                                                         dependent
                  K O E I N                                                         variable
                  T / R U K                  form r
                  Z P E / D
                  L I S T I
                  W V K / A
                  H V K / K
                  Y H E / N
                  P M K / A   store r.
                  Q / ½ Q ½   add previous y
                  O M K T C   add r
                  B / ½ Q A   plant new y
                  G M K T K
                  " M K T C
                  M M K T A
                  X M K / K
                  V L E / N                                   Form new values
                  £ D / / I                                   of 2^m q
                  / / R U J
                  E / N U K
                  @ O E I ½
                  A O E I ½
                  : P E / ½
                  S P E / ½
                  I / R U A   plant new 2^m q
                  U A : Q G                                   Test for last variable
                  ½ £ / / T                                   ..............
To                D A : P G                                   Test for end of step
Auxiliary  ←      R : / / T
                  J T E T /   *
                  N V S T A                                   Plant link to return
To master  ←      F N S / P                                   to master routine
routine           C           2n -2
                  K
                  T           Link to reenter                 Preset before entering
                  Z           main routine                    routine and may be set
                  L           2^m                             only once.
                  W
                  H           2^(40-m)
                  Y
                  P
                  Q
                  O Y R Y R   1
                  B Y R Y @   12
                  G : T I W   ½(1+√½)                         Constants
                  " U E R "
                  M M K B N   ½(1-√½)
                  X O V Y :
                  V / / / /   ¼
                  £ / / / ½
```

(1) [P E] is zero in stages 1, 2, and 3 but is set $= \frac{1}{6}$ during stage 4.

(2) If m = 0 then $[H\ /]_{0}^{39}$ is replaced by VK/J//T£

(3) Auxiliary routine begins at / @.

*(4) For use with Scheme B, the content of lines J E and N E should be / / T £. The routine is called in as a sub-routine of the master routine.

B/RUNGE-KUTTA

Purpose.

This is a variation of the GILL-KUTTA routine described in the previous pages. It is intended for input with Scheme B. The new feature is the introduction of a set of 7 preset parameters (20 digit lines). These are

(2n-2)//, n being the number of equations
(j)//, j being the address (in PERM) of 2^m
(k)//, k being the address of 2^{40-m}
(h)//, h being the control number for entering the auxiliary sequence
(a)//, a being the address of the 1st of the y_i
(b)//, b being the address of the 1st of the q_i
(c)//, c being the address of the 1st of the $2^m hf_i$

These parameters must be standing in lines BU to £U inclusive when B/RUNGE-KUTTA is read from the tape. This is most conveniently achieved by preceeding B/RUNGE-KUTTA on the tape with a sequence of the form KBUU...etc.. The routine tape makes use of an interlude (see p.7.12) to select and insert these parameters into the appropriate instructions of the routine itself before this (which is standing in S4) is transferred to the magnetic store by a destination sequence.

Notes.

1. Cue is ££...../

2. This version is self-resetting, that is , it does not have to be transferred from the magnetic store to S0 every time it is called in. It can be used as an electronic routine.

3. From the auxiliary sequence the routine is reentered at line :/ by terminating the sequence with ///P (line // contains A/T:)

4. If m = 0 then DS may be used for the address of 2^{40-m} without any loss of accuracy.

5. No lines outside SⲪ other than B6 and B7 are altered.

B/RUNGE-KUTTA (skeleton routine)

```
                               K/UB                            K/½/               K/D/
         |I|                 / X U Q O |U|              |/|  A / T :   /  R / / I  |E|
                             E G ½ Q G    set (a)QA          P E T A   E  (/ / B H)
                             @ G ½ Q B                       F E P O   @  (/ / B T)
                             A V U Q O    set                S / / P   A  O E I ½
                             : N ½ Q G    (b)QN              C E Q O   :  O E I ½
                             S N ½ Q B                       (/ / T ") S  P E / ½
                             I V U Q O    set                E : P G   I  P E / ½
                             U Z ½ Q G    (b)UK              A : / O   U  (/ / B M)
                             ½ Z ½ Q B                       O E T K   ½  A : Q G
                             D V U Q O    set                P E T A   D  / E / T
                             R E D Q G    (b)UJ              D S P G   R  A : P G
                             J E D Q B                       (/ / D Q) J  S / / T
                             N V U Q O    set                (/ / D W) N  N S / P
                             F U D Q G    (b)UA              H E T A   F  I / / /
                             C U D Q B                       H E / K   C  (/ / / /)
                             K £ U Q O    set                O E I N   K  spare
                             T J ½ Q G    (c)Q½              O E I N   T  spare
                             Z J ½ Q B                       (/ / B T) Z  spare
                             L £ U Q O    set                P E / D   L  working
                             W @ D Q G    (c)UK              I S T I   W   space
                   K Y I J   H @ D Q B                       H E / A   H  working
set (j)/N          G U Q O   Y M U Q O                       H E / K   Y   space
                   £ ½ Q G   P S ½ Q G                       (/ / / H) P  working
                   £ ½ Q B   Q S ½ Q B                       L E / A   Q   space
set (k)/N          " U Q O   O N @ / P  ←return              (/ / D Q) O  Y R Y R
                   P ½ Q G   B 2n-2/ /   to                  L E T C   B  Y R Y @
                   P ½ Q B   G  j  / /   B.INPUT             (/ / D M) G  : T I W
set (2n-1)         B U Q O   "  k  / /                       L E T K   "  U E R "
                   C D Q B   M  h  / /                       L E T C   M  M K B N
set (a)Q½          X U Q O   X  a  / /                       L E T A   X  O V Y :
                   O ½ Q G   V  b  / /                       L E / K   V  / / / /
                   O ½ Q B   £  c  / /                       (/ / / H) £  / / / ½
                               THI
```

The punching proper of the interlude is terminated by the meaningful sequence THI which transfers control to YI, the first instruction of the interlude. The last instruction of the interlude N @ / P, returns control to the B.INPUT routine. The skeleton instructions which are modified are enclosed by brackets.

B/QUADRATURE/A

Purpose Quadrature

Cue £E.....@

Magnetic Storage Variable

Electronic Storage S1

Effects

$$[L']_{\pm f} = \int_{y_\pm}^{x_\pm} f(t)\,dt \quad \text{or} \quad \int_{y_+}^{x_+} f(t)\,dt, \text{ where}$$

$x_\pm = [L]_{\pm f}$, $x_+ = [L]_{+f}$, $y_\pm = [M]_{\pm f}$, $y_+ = [M]_{+f}$, and $x > y$.

An auxiliary sub-routine for calculating $f(t)$ has to be prepared by the user. This should have the effect

$$[L']_{\pm f} = f(L_f).$$

Parameters

There are two preset parameters which must be placed in lines TU and ZU before B/QUAD./A is read from the tape. Thus

```
KTU@(h)//
   (2u)//,
```

where 2n is the DIRECTORY number of the auxiliary sub-routine, and h defines the first of 13 consecutive (short) lines which are needed as working space by B/QUAD./A. (h should be an even numbered location).

Accuracy

Round-off error: $2^{-38} + (L-M)\,2^{-36}$

bias: the result is in defect by about 6.2^{-40}

truncation error: zero for polynomials of degree less than 26.

Time

(700 + 13T) ms., where T is the time of operation of the auxiliary routine.

Method

The routine makes use of Gauss' 13 point formula. See Bulletin Amer. Math. Soc., 48, 10, pp 739-743 (Oct. 1942)

Author A.E.G. 18/11/52.

Chapter 8.

Fault Diagnosis in Programmes.

(With particular reference to Scheme B).

This chapter is concerned with the problem of finding mistakes in programmes. The incidence of such mistakes and the difficulty of locating them are serious obstacles to the efficient use of the machine Moreover experience with machines has shown that the occurence of mistakes is not a temporary evil due to lack of experience but one likely to remain as long as programmes are drawn up by human users; that, whilst a lot of mistakes can be avoided by reasonable care and attention, there frequently remain mistakes which could only have been detected in the early stages by prolonged and laborious study.

The incidence of mistakes varies a lot with the type of programme; a complicated routine concerned almost entirely with logical operations, e.g., an interpretive routine, will probably contain a higher proportion of mistakes than a routine of the same size which is concerned mainly with straightforward arithmetical operations.

Hithertoo the main method for locating programme mistakes has been "peeping". This is a derogatory term for the following procedure. Seated at the console and with the aid of the single prepulse key the mathematician works through his programme, or a section of it, instruction by instruction observing from the monitors the behaviour of certain storage locations. This procedure is regarded unfavourably because the machine normally obeys instructions at the rate of about 850 per sec and whilst "peeping" this is reduced to (say) 1 every few seconds - an extravagent waste of machine time. In ref. 1, which deals with the subject of fault diagnosis with particular reference to the EDSAC, the author goes further and remarks "All these facts make the process extremely inefficient one for the checking of programmes". This statement needs to be qualified slightly before it can be applied to the Manchester MK II machine.

The facts referred to are not all characteristic of this machine and in addition the facilities on the console of the Manchester machine enable one to correct errors on the spot so that a routine can usually be made to work at one sitting provided the errors are "slips" rather than errors of design. Furthermore the effects of programme errors are likely to be confused with those of faulty magnetic transfers: both can be consistent * . However we are in agreement with the views and conclusions set forth in ref. 1, and it is proposed to eliminate "peeping" as far as possible. Although we recognise that there are occasions when the console facilities are the appropriate means for fault diagnosis and correction.

* The fact that the track(s) in question have been passed as satisfactory is not sufficient evidence of their reliability: malfunctioning may be characteristic of the pattern of 0's and 1's which are stored on it, i.e., characteristic of the programme. This is probably the main cause of "programme sensitiveness" which has been a feature in the life of the machine.

Ref. 1 describes a method by which the machine - operating at a speed limited mainly by the teleprinter - can be utilised to print out the information necessary for fault diagnosis. The basic idea is to use an interpretive routine which, treating the routine under test as a slave routine, simulates the behaviour of the machine and prints out useful information as it proceeds.

There are two principle forms which this idea can take

1. Instruction Sequence Check.

In this scheme the interpretive routine is arranged so that, as each instruction is 'obeyed', a symbol is printed which identifies that instruction. The aggregate of symbols thus forms a printed record of the course of the routine which can be torn off, taken away, and studied at leisure. Such a record is extremely useful when testing any routine for the first time.

The need for a checking routine of this kind has now been eliminated because, at the time of writing, a modification which enables this information to be printed automatically is being incorporated into the MK II machine. It takes the following form. A switch on the console enables the operator to change from the normal mode of operation to an instruction checking mode in which the 4th character (i.e., the five most significant digits) of each actual instruction is printed and/or punched immediately after it has been obeyed. If the printer is used then the operator must provide carriage return, line feed, and, if necessary, letter shift manually. The last may be necessary if the routine under test calls for a figure shift. Symbols that would normally be printed by the routine under test will be interspersed with the main printing. If the punch only is used, then it is possible to print the resulting length of tape on a printer which provides an automatic carriage return and line feed after every 32 symbols.

2. Numerical Check. (Scheme B).

This scheme is used when the routine is going through the motions more or less correctly, and it is required to check the numerical value of the number in the accumulator at preassigned points in the routine. A checking routine - NUMBERCHECK - based on these ideas is available. The details are as follows. The programme is first rearranged so that it does not require to use S3 except for the list of links which are normally kept there. This means that any of the time economy measures described on page 4.12, which make use of S3 must be dispensed with. Next the cues of the routine which it is desired to check are marked by inserting a 1 in digit position 15 of the magnetic half of the cue. Then the particular instructions at which, before being 'obeyed', it is required to know the value of the accumulator, are marked with a 1 in the spare digit position 13.

The programme is then run with a special two page PERM - part of NUMBERCHECK - in S2, S3. Provision is made for the list of links to extend 4 deep (j = 4) which should be ample for most purposes. The programme operates in the usual way until a routine with a marked cue is encountered. In this case control is transferred not to the routine but to an interpretive checking sequence which causes the routine to operate as it would normally - with certain restrictions listed below - and in addition causes the content of both halves of the accumulator to be printed each time that a marked instruction is encountered and before 'obeying' that instruction.

The properties and restri ons of NUMBERC CK are:

1. The more significant f of the acc ulator is printed on the ± fractional convention fo wed by, a parated by a space, the lesser significant half printe n the + f onal convention. A carriage return and line feed urs before pair of numbers is
urther line f urs a fer of control
der test.
checking i tions appear to fail.
So B check fac on t unch.
ne of the dig s 16, r 19 of a magnetic
h not zero n that cue will be treated as a

ECK will to interpret correctly, or fail
he follow circumstances:
Z instruc curs in the routine under test;
address presumptive instruction lies between
ive, exce he case of the instructions NS/P,
GS/P, which cau routine changing sequence to be
entered in the usual wa c tional entries to the routine
changing sequence, e.g /H, will also have disastrous effects;
and

(c) If the last 6 ts of the presumptive instruction differ from those of the actu struction and either (i) the actual instruction would cause nsfer of control. or, (ii) the presumptive instruction is one of the uence /T, /Z, ..., /£.

It is to be hoped t se exceptions will not prove too restrictive in practise.

The routine NUMBERC s 4 pages. Pages 1 and 2 which replace the usual PERM, are intended to be stored on magnetic track 98, and pages 3 and 4 on track 99. The tape for NUMBERCHECK, if read by INPUT, does this automatically and can be copied on the head of the programme tape in place of the usual PERM setting sequences. The rest of the programme tape takes the usual form.

The mechanism of NUMBERCHECK.

The foregoing description of NUMBERCHECK should be sufficient to enable the routine to be used. In this section the mechanism of the interpretive sequence is explained.

An interpretive routine which would accurately simulate every feature of the machine's possible behaviour would extend to several hundred instructions. Fortunately however if certain conventions have been followed in the coding of the routine under test, then the necessary instructions can be contained on a single page (S3). A number of auxiliary constants and a printing routine are also needed however. These together with a modified form of the usual S2 PERM bring the total amount of material involved to 4 pages. Of these only one, the interpretive sequence in S3, is strictly permanent information, the other being brought down to S2 when needed. The details of these pages are given below. The general principles are as follows.

The complexities that would be involved in simulating the B-sign flip-flop are avoided by not using any B tube instructions. For a similar reason the D register is not used except in the printing routine. Thus all the necessary operations must be confined to the accumulator and the information which would normally be contained in the accumulator is stored in line pairs ½I and RI. Before each instruction is actually obeyed the accumulator is reassembled and afterwards returned to these locations.

The 'control', i.e., the task of selecting the current instruction, is the major part of the routine and entails sorting out those instructions likely to cause a transfer of control; for if these instructions were actually obeyed they might direct control to the original routine and the interpretive sequence would cease to operate. The task is simplified if we assume that in the case of such instructionsthe function digits of the presumptive and actual instructions are identical. This means that it is only necessary to test for the presumptive instructions /T, /H, /P, /Q, /O, and /M. If we are prepared to ignore the possibility of a /Z instruction, then the task is further simplified by the fact that the instructions /T to /£ inclusive are all either transfers of control or addressless (the significance of this will be revealed below). The instructions in this group are easily recogonised because they are negative and digit 14 is a 0. With in this group it is necessary to distinguish the relative from the direct transfers of control and to recognise the particular instructions NS/P, QS/P, and GS/P, which occur when changing routines; these last are actually obeyed as they stand. The addresses of the direct and relative control transfer instructions are modified so that if, when they are obeyed, a transfer of control takes place, then it will not be to the original programme but to one of two points within the interpretive sequence which is so arranged that the address of the selection instruction is modified appropriately.

Before being obeyed each (presumptive) instruction is tested for the marking digit. If the instruction is marked, then the printing routine (page 4) is brought down to S2 and entered. The properties of the printing routine have already been described. The only feature of interest perhaps is the temporary storage of the content of the D register (lines CS to BS) so that this register can be used in the printing sequence. When this sequence has finished its task the D register is reassembled and control returned to the interpretive routine in a way which causes page 3 - the constants - to be restored to S2.

To summarise NUMBERCHECK involves 4 pages of material of which only 2 - page 2 and 1 other - are held in the electronic stores S2 and S2 at any stage.

Page 1 (98L, S2). This page is identical with the normal S2 PERM except that the last instruction (VSEP) is changed to VS/P. This enables the routine changing sequence to be extended to S3. Page 1 occupies S2 when the checking routine is non-operative.

Page 2 (98R, S3). This page contains the interpretive sequence which tests for marked cues. Space for the list of links to extend 4 deep is provided. Page 2 occupies S3 for the duration of the programme.

Page 3 (99L, S2). Lines /: to JS inclusive of this page and page 1 are identical. The remaining lines of this page are constants used by the interpretive sequence. Page 3 is brought down to S2 after a marked cue has been encountered.

Page 4 (99R, S2). This page contains the printing routine. It is temporarily brought down to S2 when a marked instruction is encountered.

If these pages are stored in their allocated tracks, then the effect of the final sequence on the programme tape is to bring down the contents of the PERM track (pages 1 and 2) to S2 and S3 and to start the programme in the usual way.

Page 1

```
:| / / / / | / | / / / / |S
 | E / / / | E | / / / @ |
 | / / / / | @ | / / / / |
 | @ / / / | A | / / / : |
 | / / / / | : | / / / / |
 | : / / / | S | / / / ½ |
 | / / / / | I | / / / / |
 | ½ / / / | U | / / / T |
 | / / / / | ½ | / / / / |
 | T / / / | D | £ £ £ £ |
 | / / / / | R | £ £ £ £ |
 | / E / / | J | / / / / |
 | / / / / | N | " S / L |
 | / @ / / | F | @ E / / |
 | / / / / | C | Z S Q G |
 | / : / / | K | A : Z G |
 | / / / / | T | S : / Q |
 | / ½ / / | Z | V £ P O |
 | / / / / | L | M £ £ £ |
 | / T / / | W | Z S Q G |
 | / / / / | H | V S Z Z |
 | / / E / | Y | / / Q T |
 | / / / / | P | @ I E A |
 | / / @ / | Q | F S / : |
 | / / / / | O | / / U F |
 | / / : / | B | / I E A |
 | / / / / | G | L S Z G |
 | / / ½ / | " | V S E J |
 | / / / / | M | A : Z G |
 | / / T / | X | £ S E : |
 | / / / / | V | J I / G |
 | / / / E | £ | V S / P |
```

Page 3

```
:|         | / |         |S
 |         | E |         |
 |         | @ |         |
 |         | A |         |
 |         | : |         |
 |         | S |         |
 |         | I |         |
 |         | U |         |
 |         | ½ |         |
 |         | D |         |
 |         | R |         |
 |         | J |         |
 |         | N | @ A / : |
 |         | F | T S / : |
 |         | C |         |
 |         | K |         |
 |         | T | A A E : |
 |         | Z | £ £ £ £ |
 |         | L | / / / £ |
 |         | W | M S / / |
 |         | H | / / / / |
 |         | Y | / / / E |
 |         | P | V I T I |
 |         | Q | ½ I / U |
 |         | O | Q F / P |
 |         | B | £ £ £ / |
 |         | G | / / T / |
 |         | " | R S T R |
 |         | M | £ £ / / |
 |         | X | / / / / |
 |         | V | R I / U |
 |         | £ | X U / / |
```

N.B. lines /:
to JS incl.
are identical
with those of
page 1.

Ref.1. Gill, S., The diagnosis of mistakes in programmes on the EDSAC, Proc. Roy. Soc., A, 206, 538-554 (1951).

```
LI  /                  space reserved for
    E                  possible list of
    @                  links
    A
    :
    S
    I
    U
    ½   m.s.           accumulator
    D
    R   e.s.
    J
--> N  £ : Q O         test for marked
    F  £ S Z L         cue
<-- C  V S E T
    K  V S Z T         set initial address
    T  V I Q B         of selection instr.
    Z  M V W £
    L  H I / :         page 3 to S2
    W  E U / P         store accumulator
    H  A A / :         these instr: are
--> Y  W U T :         obeyed in the case of
    P  V I T A         a relative transfer
                       of control
--> Q  Z I / :         line feed
    O  / / T /
    B  V I T I         add 1 to address of
    G  E : T C         selection instr.
    "  M S T R
    M  G S T I
    X  V I T A
    V  / / T /         selection instr.
    £  R S T R
```

```
LU  /  X S T I         plant current
    E  G U / S         instruction
    @  B S T R         test if digit
    A  X : T N         14 is a 1
    :  Y I / H
    S  Y S T I         plant a selection
    I  O I / S         instruction in
                       case of a transfer
                       of control
    U  M S T R         test for NS/P
    ½  W S T N         QS/P and GS/P
    D  A : / M
    R  D : T C
    J  £ S / H
    N  B U T ½         test for transfers
    F  Y I / H         of control /L,/G
                       /W,/Z and /V.
    C  Z S T R         test for relative
    K  M U T N         transfers
    T  E : / M
    Z  / : T C         Manufactive
    L  Q S T C         special address
    W  B U T A         for transfer
                       instructions
    H  B U T /         test for marked
    Y  G : T R         instruction
    P  G : T N
    Q  V U / H
    O  R I T /         reassenble
    B  ½ I / J         accumulator
    G  (/ / / /)       current instr.
    "  R I / U         restore
    M  ½ I / U         accumulator
    X  I U / P
    V  N S / :         case of NS/P,
    £  Y I / P         QS/P and GS/P.
```

<-- relative transfer of control <--

<-- direct transfer of control <--

```
:  /      H       figure shift
   E  ½ I T ½     test sign of m.s. half
   @  ½ : / H     of accumulator.  Print
   A  H S T :     "-" if -ve; otherwise
   :  Y S / J     space
   S  Y S / :
   I  @ : / P
   U      Z       space
   ½  U : / :
   D  ½ I T /     set number for
   R  / : T A     printing
   J  R : T :
   N  / S / C
   F  / : / N
   C  / : / A
   K  / : / C     digit cyclo
   T  / : T A
   Z  E S / N
   L  @ / / I
   W  D : / :
   H  V : T K
   Y  V : / S
   P  J : / H     examine digit layout
   Q  V : T K     constant
   O  V : / S
   B  A : / H     leave printing sequence
   G  R I T /     prepare to print E.s.
   "  U : / :     half of accumulator
   M  U : / :     separated by 2
   X  W : / P     spaces
   V  / T / /     digit layout
   £  T E / /     constant
```

```
S  /  R / / /     10
   E  / / / /
   @  / / / E
   A  E / / /
   :  / / / /
   S  / / L /     carriage return
   I
   U
   ½
   D
   R
   J
   N
   F
   C  A S / N     store content of
   K  C S / U     D register
   T  X S / U
   Z  / : / :     figure shift
   L  S S / :     carriage return
   W  Z I / :     line feed
   H  Z S / P     switch to printing
                  sequence
   Y  X S T ½
   P  L : / M
   Q  C S / K     reassemble D
   O  A S / Q     register
   B  C S / C
   G  B U T /
   "  X S T A     reassemble
   M  R I T /     accumulator & plant
   X  / / / /     current instruction
   V  / / / /     current instruction
   £  Q I / P     return to interpret-
                  ive sequence.
```

Chapter 9.

Measures Against Machine Breakdown.

Warning

It should be clearly understood that the machine is liable to carry out certain operations incorrectly. This malfunctioning can be either chronic or transient. Whilst everthing is being done on the ehgineering side to eliminate these undesirable features, nevertheless in the meantime programmers are obliged to minimise the effects of machine errors by devices in the programme.

Nature of faults.

Every part of the machine is of course liable to function incorrectly but in the case of many units, e.g., the power supplies, the effects of malfunctioning are usually obvious at once. When a chronic fault is detected, then the maintenance engineer should be informed immediately.

It is the effect of faults - particularly transient faults - in the units concerned with the processing and transfer of information that are liable to pass unnoticed. For this reason checks of some kind, preferably overall mathematical checks, should be incorporated in the programme.

A further kind of fault consists of the appearance of spurious digits in one or more of the electronic stores. This is referred to as "clodding". The effects of a fault of this kind - particularly when they concern the B-tube - are not usually confined to numerical errors but can cause the programme to breakdown.

The most common fault of a transient nature are those of faulty magnetic transfers. If the transfer refers to numerical information, then only a mathematical check will detect it. If a routine is transferred incorrectly, then the programme will probably breakdown. The detection of faulty magnetic transfers has received a great deal of attention and is discussed below in a separate section.

Error detection and precautions

One technique for detecting and minimising the effects of machine fault is the following.

The calculation is broken down into steps and the programme arranged so that a decision as to whether each step has or has not been carried out correctly can be taken by an operator after the inspection of information printed at the end of each step (this information should be kept to a minimum); and that, if necessary, the calculation can be restarted at the beginning of that step with the minimum of trouble, e.g., by simply operating K.E.C. (see appendix).

All the steps should be of approximately the same length so that, if the programme breaks down, then the operator will be informed by the failure of the regular printing interruption. It is this last possibility - usually the result of a clod - which prevents the decision being left to the machine itself.

Each step should be so arranged that reliance on the correct operation of the electronic stores for periods exceeding 2 mins. should be avoided.

A programme recently coded in this Laboratory for the solution of a certain partial differential equation provides an example of the foregoing suggestions. The problem involved the solution by iteration of a set of simultaneous equations associated with a mesh of points. Each sweep of the mesh occupied 1 min. 35 sec. At the end of each sweep the sum of the squares of the residuals (E) was printed and the new approximation transferred to the magnetic store. The iterative process was such that E should take progressively smaller values. Thus if, at any stage, a relatively large value of E is printed, then the operator would repeat the last step merely by the operation of K.E.C.

The reliability of the magnetic store.

Experience with the magnetic wheel has shown that certain, apparently random, patterns of 0's and 1's are not transferred correctly but lose or gain one or more digits. This applies both to reading and writing transfers. Little can be done about this phenomenon and it is accepted that 40% of all tracks have failed in this way at some time in their history.

If a track has behaved successfully with a given routine, i.e., the pattern of 0's and 1's corresponding to the routine is not critical for the track, then it is usually safe to use the same track to store the routine on subsequent occasions.

On the other hand it is unwise to store numerical results, or other unknown patterns, on any but the recommended tracks particularly if it is not intended to use a device in the programme (see below) to detect faulty transfers

In view of these statements users are advised to arrange their programmes so that they are not tied to any fixed tracks, that is to say, he should think in terms of entries in a directory, a list of working tracks which can be varied either by himself or by the machine in accordance with the current state of the wheel. The half cue directory of scheme A and the cue directory of scheme B provide examples of this technique.

Track testing techniques

The list of recommended tracks has grown as a result of experience gained with various track testing routines. These operate by writing a pattern on to the suspected track and then comparing, in some way, the pattern on the track with the pattern in the electronic store. Some tests use a pattern of random numbers; others use special patterns surmised for engineering reasons to be a stringent test of the storage properties of the track. The comparison can be effected either by means of the magnetic write check operation, or by transferring the pattern on the track to a part of the electronic store, other than that occupied by the original pattern, and then comparing the two patterns in the electronic store.

The detection of faulty magnetic transfers.

The times of the various magnetic operations are stated here for reference.

Magnetic	write	90	milliseconds.	
"	" read	35	"	"
"	" write check	35	"	"
"	" read check	35	"	"

The basic difference between read checks and write checks is that the so-called 65th line on a tube is compared with its counterpart on the wheel in the former case, but not in the latter: this, it was thought, would ensure that the track selecting mechanism was operating correctl

It has been found in practice that the track-selecting mechanism is sufficiently reliable for this check not to be needed. The difficulty with the use of read check is that the signal at its most vulnerable level passes through exactly the same mechanism as it does during the read operation: the effectiveness of the read check for decting transfer errors is thus very questionable. The write check (following a writing transfer), on the other hand, is free from this objection - i.e., it can be used as a safeguard to ensure that information has reached the wheel successfully. In fact, with lengthy calculations and during input, it is as well to ensure confidence by the successive use of several write checks. This is especially true if for any reason a track described in the directory as being reliable has become marginal.

Typical writing sequences are given in the Examples 3.

Operations to be followed in the event of a check failure will be discussed below. For the moment, it is sufficient to note that, once a failure has been recorded, then at least with the information under test, continued writing will not produce a reliable problem on the wheel. Even if the check works subsequently, the transfer should be regarded as unsafe; tests indicate that that track will be no less subject to marginal operation because repeated writing has taken place

In the case of reading transfers the only means of ensuring a c rrect transfer is to have available some extra information regarding the contents of the track. Thus if the check sum is available, then this can be compared with the number obtained by summing (mod. 2^{40}) the long lines of the page or pair of pages transferred. The check sum can be stored on the track itself. Alternatively the content of a spare line pair can be adjusted so as to make the check sum zero.

It is the author's opinion that transfers involving numerical information should be checked in some way, particularly if the results are anything more than experimental.

The correction of faulty magnetic transfers.

The action to be programmed when a failure is recorded depends largely on the circumstances. In cases where the failures are transient and the checks sufficiently stringent, it is sufficient to continue attempts to write or read until the checks are satisfied. This may be time consuming if the failures are chronic. In that case it would be better to draw up a new directory for the problem.

It should be possible to arrange the programme so that failure of a check leads directly to the replacement of the bad track wherever it occurs in tho the directories. However the author's have never had any need to adopt such sophisticated techniques.

Conclusion.

Experience has shown repeatedly that, if long production runs are contemplated, then there is every advantage in making the programme as far as possible self checking. The time lost in this way is negligible compared with that which would otherwise be lost by e.g., the necessity of repeating certain runs due to lack of confidence in previous results.

APPENDIX.

The Console.

This is a control desk by means of which the behaviour of the machine can be controlled manually. In practice this is best studied in connection with the machine itself. For this reason only a brief description giving its essential logical features will be given here.

A switch is a 2-position manual control which is stable in either position:

A key is one which must be kept pressed if it is to remain in the 'active' position.

The various keys and switches are classified as follows.

1. Dummy Stop switches and the hand switches.

The purpose of these are described on pp. 1.20.

2. MAN-AUTO switch.

Manual instruction switches.

When the MAN-AUTO switch is in the AUTO position the instructions from the electronic store are selected and obeyed in a manner determined by the switches in group 3 (below). When this switch is in the MAN position a different arrangement applies. In this case the actual instruction is the combination set up on the set of 20 manual instruction switches. A number of restrictions are associated with this facility and its use by programmers is not recommended; although it is useful to the maintenance engineer.

3. Switches for the control of completion signals.

With the MAN-AUTO switch in the AUTO position the instructions can be selected and obeyed from the electronic store in the following ways.

(a) Full speed ahead, i.e., in the manner already explained in detail in Chapter 1.

(b) 50 per second.

(c) Singly under the operation of the single prepulse key.

X (d) Printing and/or punching a record of the 4th character. This mode of operation is described in Chapter 8.

4. Clearing keys.

By means of these the operator can clear the entire electronic store or its individual units, that is, the accumulator, the B-tube, the control lines, the multiplicand register, and all 8 storage tubes (individual storage tubes cannot be cleared).

5. Manual writing keys.

These provide means for inserting or removing digits from any line of the electronic store.

6. Input and Output control switches.

These enable the operator to select the printer and/or the punch as the output unit and to inhibit the operation of the output unit and/or the tape reader.

7. Writing suppression keys.

The effect of undesired magnetic writing transfers can be disastrous. In many problems no such transfers need be made. During the solution of these problems it is usual to suppress all writing transfers by switching off the write power. Unfortunately it is not (easily) possible to suppress the write power on individual tracks

In addition to the various keys and switches there are monitor tubes and neons on the console by means of which the behaviour of the machine can be observed. The contents of the accumulator, B-tube, control, and multiplicand register are displayed on different tubes and the content of any 2 stores can be displayed on two further tubes. Two neon lamps indicate the sign digits of the multiplicand register and the last B-line altered (on which /T discriminates). A neon is bright if the corresponding digit is a 1.

Numerous special points arise in connection with the precise behaviour of certain keys and switches, and with the interpretation of the monitors, in special circumstances. No attempt will be made to list them here and reliance on these facilities to check programmes should be regarded only as a last resort.

The Tape Preparation Equipment.

This equipment is concerned with the physical preparation of programme tapes for the machine.

There are two keyboard perforators. These are instruments, similar to typewriters, for punching combinations of holes in paper tape. They have 32 keys on which are engraved the characters /,E,

..., £, and which when depressed, punch the corresponding combination of holes on the tape. At the same time the teleprint signal corresponding to the character in question is transmitted along a teleprint line. There are also keys engraved with 0, 1, 2, 3, 4, 5, 6 7, 8,9, +, -, .,whose effects are respectively duplicates of /, E, @, A, :, S, I, ½, D, P, M, F. There is also a reperforator which accepts signals from a teleprinter line and punches the corresponding combinations of holes. There are two tape readers which accept an input of tape and provide an electrical output of teleprint signals. These units may be coupled in various ways. For instance one may connect both a reader and a keyboard perforator to a reperforator. This enables one to copy a tape with interpositions of new material, and possibly with omissions. In another arrangement the tape from the reperforator may be threaded through the reader, which is electrically connected to the reperforator. Under these circumstances a given sequence of characters can be repeated indefinitely often on a tape.

There is also a mechanism for printing out the contents of tapes. In the letter shift position this gives the standard characters used in this handbook; on figure shift the equivalents

0	1	2	3	4	5	6	7	8	9	.	Space	CR	LF	+	"	-	£
/	E	@	A	:	S	I	U	½	D	F	Z		L	W	P	"	M	£.

will apply. The changes from figure to letter shift and back are manually controlled. The printer stops automatically at the end of a line and the carriage return and line feed can be operated by the appropriate keys on the keyboard. In the figure shift position L and W cause automatic carriage return and line feed whenever they occur on the tape.

Abbreviated Instruction Code.

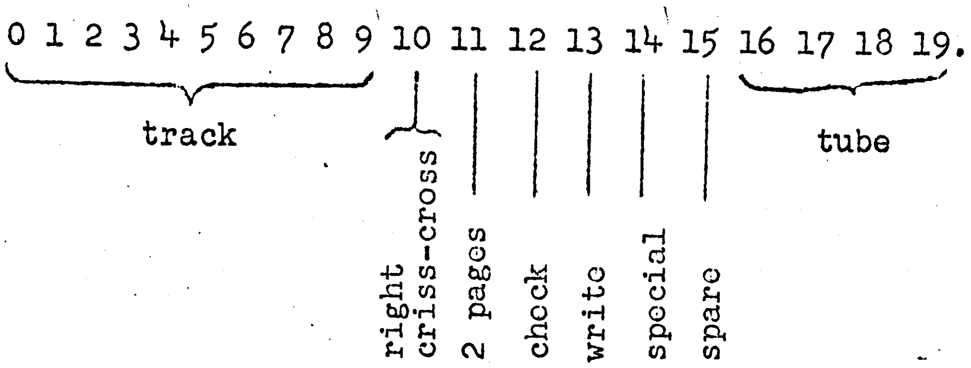
/ /	H as mag. instr.	T /	$A' = S_+$	
/ E	$S' = M$	T E		
/ @	standardise	T @		
/ A	$S' = M$, $M' = 0$	T A	$S' = L$, $A' = 0$	
/ :	S as mag. instr.	T :	$A' = 0$	
/ S	$S' = L$	T S		
/ I	$L' = M$, $M' = L$	T I	$A' = A + S_+$	
/ U	$S' = L$, $L' = M$, $M' = 0$	T U		
/ ½	$A' = A - D\ S_+$	T ½	$A' = S_\pm$	
/ D	$A' = A - D\ S_\pm$	T D	$A' = A \vee S_\pm$	
/ R	Sideways adder	T R	$A' = A \,\&\, S_\pm$	
/ J	$M' = M + S$	T J	$A' = A \not\equiv S_\pm$	
/ N	$A' = A + D\ S_+$	T N	$A' = A - S_\pm$	
/ F	$A' = A + D\ S_\pm$	T F	$A' = -S_\pm$	
/ C	$D' = S_+$	T C	$A' = A + S_\pm$	
/ K	$D' = S_\pm$	T K	$A' = 2S_\pm$	
/ T	B-cond. direct trans. control	T T	$B' = S$	
/ Z	$S' = H$	T Z	$S' = B$	
/ L	Stop	T L	$B' = B - S$	
/ W	Random number generator	T W	$B' = B - S$	
/ H	A-cond. direct trans. control	T H		
/ Y		T Y		
/ P	Uncond. direct trans. control	T P		
/ Q	Uncond. rel. trans. control	T Q		
/ O	B-cond. rel. trans. control	T O	$B' = S$	no B-addition
/ B		T B	$S' = B$	
/ G	Stop	T G	$B' = B - S$	
/ "		T "	$B' = B - S$	
/ M	A-cond. rel trans. control	T M		
/ X		T X		
/ V	Hoot	T V		
/ £		T £	Dummy	

The reader will observe that the instructions with positive codes, i.e., which lie in the upper half of the table, refer mainly to the accumulator, whilst the remainder are transfers of control or refer to the B-tube. With certain exception the former are 5 beat codes, the latter take 4 beats.

The exceptions are the multiplication codes /½, /D, /N, and /F which each take 9 beats, the random number instruction which takes 24 beats, and the magnetic instructions which take times as follows: read, read check, write check 35 ms; write 90 ms.

Magnetic Instructions.

The purpose of the 20 digits of a magnetic instruction can be represented as follows



A special magnetic instruction (input-output) is interpreted according to its 3rd character as follows

T	Print and Punch
Z	Space
L	Carriage return
W	Line feed
H	Figures
Y	Letters
O	Input
B	Check output
£	Official Dummy.

The Teleprinter Code.

0	00000	/	0	11	11010	J		22	01101	P	+
1	10000	E	1	12	00110	N		23	11101	Q	
2	01000	@	2	13	10110	F	.	24	00011	O	
3	11000	A	3	14	01110	C		25	10011	B	
4	00100	:	4	15	11110	K		26	01011	G	
5	10100	S	5	16	00001	T		27	11011	"	
6	01100	I	6	17	10001	Z		28	00111	M	-
7	11100	U	7	18	01001	L		29	10111	X	
8	00010	½	8	19	11001	W		30	01111	V	
9	10010	D	9	20	00101	H		31	11111	£	
10	01010	R		21	10101	Y					

Where the figure shift equivalent is not shown there may only be a smudge.

Aids to Calculation in the Scale of 32.

Fig A.1

Multiplication by powers of 2. The product is given one character at a time. Each character is determined by two adjacent characters of the multiplicand. The less significant is shown on the left of the table, the more significant above it. Thus, e.g., to multiply the array IRELAND by 8, the pattern is first bordered by a couple of /'s, thus /IRELAND/, and the characters in the result are then given as follows

D and / give @,
N " D " J,
A " N " /, and so on,

the final result being TZRTM/J@

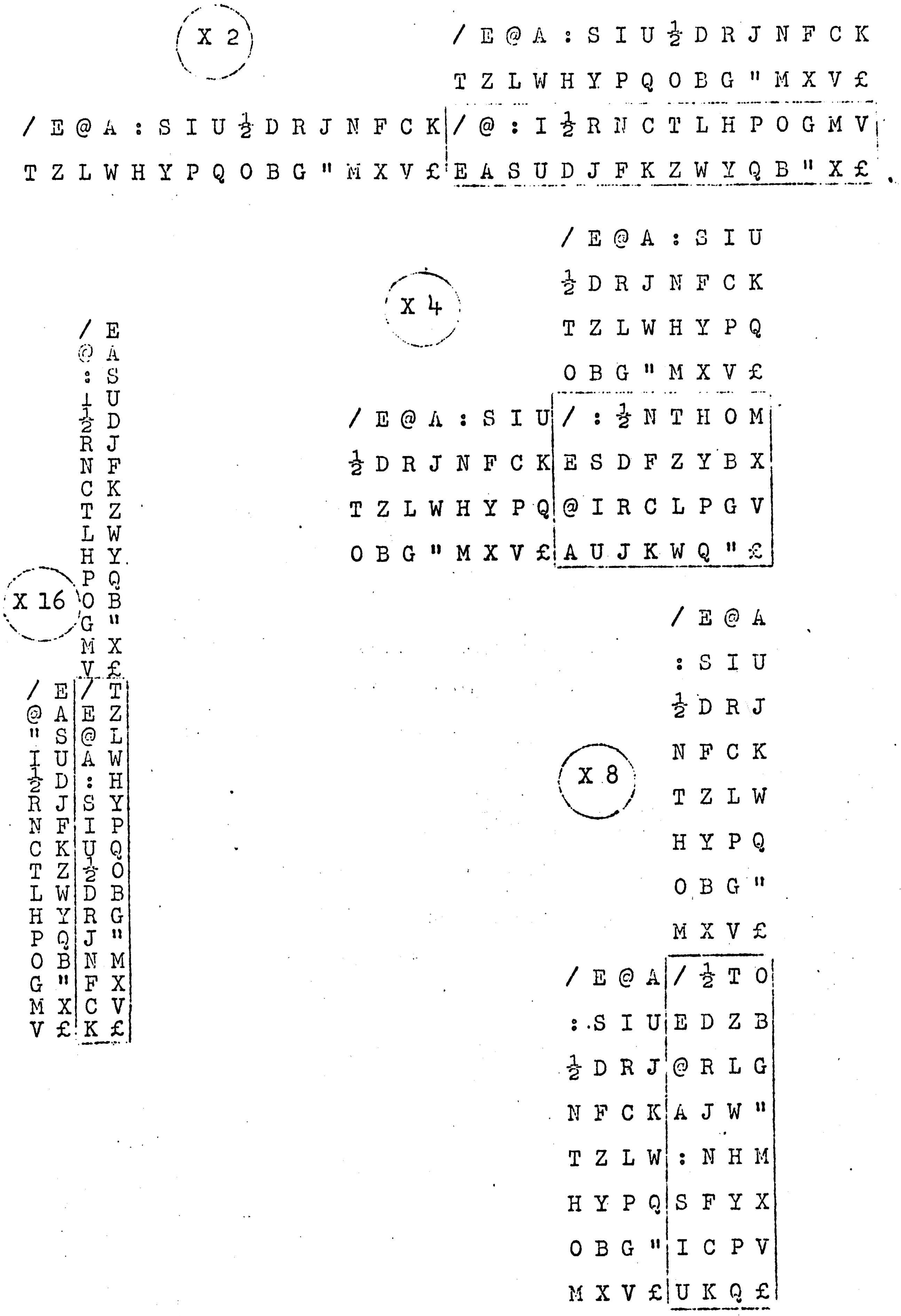
Fig A.2.

This table gives the integral equivalents of pairs of teleprint characters, E.g., that for YR = 1010101010 is given at the intersection of Y on the horizontal scale with R on the vertical scale, which is 341.

It may be used to facilitate the binary conversion of a decimal fraction. The fraction is multiplied by 1024 and the pair of t teleprint characters corresponding to the integral part looked up in the table. The process is repeated with the fractional part.

Fig A.3.

This table gives the teleprint equivalents of some useful powers of 10.



```
                                 / E @ A : S I U ½ D R J N F C K
                                 T Z L W H Y P Q O B G " M X V £
/ E @ A : S I U ½ D R J N F C K  / @ : I ½ R N C T L H P O G M V
T Z L W H Y P Q O B G " M X V £  E A S U D J F K Z W Y Q B " X £
```

```
                 / E @ A : S I U
                 ½ D R J N F C K
                 T Z L W H Y P Q
                 O B G " M X V £
/ E @ A : S I U  / : ½ N T H O M
½ D R J N F C K  E S D F Z Y B X
T Z L W H Y P Q  @ I R C L P G V
O B G " M X V £  A U J K W Q " £
```

```
         / E @ A
         : S I U
         ½ D R J
         N F C K
         T Z L W
         H Y P Q
         O B G "
         M X V £
/ E @ A  / ½ T O
: S I U  E D Z B
½ D R J  @ R L G
N F C K  A J W "
T Z L W  : N H M
H Y P Q  S F Y X
O B G "  I C P V
M X V £  U K Q £
```

```
    / @ : I ½ R N C T L H P O G M V
    E A S U D J F K Z W Y Q B " X £
/ @ " I ½ R N C T L H P O G M V
E A S U D J F K Z W Y Q B " X £
/ E @ A : S I U ½ D R J N F C K
T Z L W H Y P Q O B G " M X V £
```

Fig. A.1.

Binary-Decimal Conversion Table.

/	/	E	@	A	:	S	I	U	½	D	R	J	N	F	C	K	T	Z	L	W	H	Y	P	Q	O	B	G	"	M	X	V	£	/		.00000	/
E	32								40										50										60			63	E	.015625	.03125	E
@	64						70										80										90					95	@	.046875	.0625	@
A	96				100										110										120							127	A	.078125	.09375	A
:	128		130										140										150									159	:	.109375	.125	:
S	160										170										180										190	191	S	.140625	.15625	S
I	192								200										210										220			223	I	.171875	.1875	I
U	224						230										240										250					255	U	.203125	.21875	U
½	256				260										270										280							287	½	.234375	.25	½
D	288		290										300										310									319	D	.265625	.28125	D
R	320										330										340										350	351	R	.296875	.3125	R
J	352								360										370										380			383	J	.328125	.34375	J
N	384						390										400										410					415	N	.359375	.375	N
F	416				420										430										440							447	F	.390625	.40625	F
C	448		450										460										470									479	C	.421875	.4375	C
K	480										490										500										510	511	K	.453125	.46875	K
T	512								520										530										540			543	T	.484375	.50	T
Z	544						550										560										570					575	Z	.515625	.53125	Z
L	576				580										590										600							607	L	.546875	.5625	L
W	608		610										620										630									639	W	.578125	.59375	W
H	640										650										660										670	671	H	.609375	.625	H
Y	672								680										690										700			703	Y	.640625	.65625	Y
P	704						710										720										730					735	P	.671875	.6875	P
Q	736				740										750										760							767	Q	.703125	.71875	Q
O	768		770										780										790									799	O	.734375	.75	O
B	800										810										820										830	831	B	.765625	.78125	B
G	832								840										850										860			863	G	.796875	.8125	G
"	864						870										880										890					895	"	.828125	.84375	"
M	896				900										910										920							927	M	.859375	.875	M
X	928		930										940										950									959	X	.890625	.90625	X
V	960										970										980										990	991	V	.921875	.9375	V
£	992								1000										1 010										1020			1023	£	.953125	.96875	£
																																		.984375		

Fig. A.2.

<u>Teleprinter Equivalents of Positive & Negative Powers of Ten.</u>

	10	R	2^{40}	10^{-1}	G N I W B N I A
	10^2	: A	2^{45}	10^{-2}	O Y U R O Y U R
	10^3	½ £	2^{45}	10^{-3}	B J G F L O / E
	10^4	T O D	2^{50}	10^{-4}	M O G S C " ½ A
	10^5	/ Y E A	2^{55}	10^{-5}	C M ½ N F Z K R
	10^6	/ L T V	2^{55}	10^{-6}	" @ H Q Q Z E E
	10^7	/ H S Z D	2^{60}	10^{-7}	X Y I S £ J J A
	10^8	/ ½ O J £ @	2^{65}	10^{-8}	@ I @ Q A W Q R
	10^9	/ T L Y B X	2^{65}	10^{-9}	W / X J T J @ E
	10^{10}	/ / B Q / R D	2^{70}	10^{-10}	O C P £ F V F A
2^{-10}	10^{11}	G F U : X @	2^{75}	10^{-11}	I K E £ £ G £ R
2^{-10}	10^{12}	: R R R A X	2^{75}	10^{-12}	Q : T B M S A E
2^{-10}	10^{13}	½ S U U E A D	2^{80}	10^{-13}	D @ G : M L T A
2^{-10}	10^{14}	T H U ½ N V G @	2^{85}	10^{-14}	S H I M I D ½ J
2^{-10}	10^{15}	/ F N L G K F M	2^{85}	10^{-15}	H " W K X / : E
2^{-15}	10^{16}	@ M Q D V I M ½	2^{90}	10^{-16}	B V S S J D W A
2^{-20}	10^{17}	B C E K S G O @	2^{95}	10^{-17}	" ½ I R X X T J
2^{-20}	10^{18}	P W C P P S O "	2^{95}	10^{-18}	D U G X L M : E
2^{-25}	10^{19}	S L / A B Z Y ½	2^{100}	10^{-19}	Q X W L M E P A
2^{-30}	10^{20}	P S [illegible] G Z Q P @	2^{105}	10^{-20}	D L N ½ ½ [illegible] B J

The figures given have an accuracy of ± 1.

<u>Fig. A.3.</u>

Specimen Coding Sheets.

Programme Sheet 1 is intended for recording the official account of a library routine. This should contain information sufficient to enable it to be treated as a 'black box'.

Programme Sheets 2(a) & 2(b) are intended to contain a record of the instructions for one or two-page routines.

Check Sheets.

The purpose of these are to provide a complete record of the state of every relevant line in the electronic store at any stage of the routine. They should contain information sufficient to enable the details of the routine to be understood at a date subsequent to that on which it was coded. This information is entered as follows:

Column 1 contains the instructions.

Columns 2 and 3 represent the two halves of the accumulator.

Column 4 contains the name of the line altered (if any) as a result of the obeying the instruction.

Column 5 contains the new content of the line given in the corresponding entry in Column 4.

Column 6 is intended for notes.

At any stage in the routine the content of any store line can be found by glancing back along column 4 for the name of the line in question. It is necessary to bear in mind that the name of a short line may be contained implicitly in the name of a line pair involving that short line.

The information can be entered in any private notation that the programmer prefers, e.g., numerical values in teleprint or decimal form, algebraic symbols, English words. As an illustration Example 1.8 (see p.2.5) is interpreted on the specimen check sheet.

MANCHESTER UNIVERSITY COMPUTING MACHINE LABORATORY.

Programme Sheet 1.

Name of Routine. ____________ Date. ________

Purpose

Cues.

Sub-routines

Principal Lines.

Tapes.

Magnetic Storage.

Electronic Storage.

Stores Altered.

Effects.

MANCHESTER UNIVERSITY COMPUTING MACHINE LABORATORY.

Programme Sheet 2 (a).

ROUTINE ____________________

/
E
@
A
:
S
I
U
½
D
R
J
N
F
C
K
T
Z
L
W
H
Y
P
Q
O
B
G
"
M
X
V
£

MANCHESTER UNIVERSITY COMPUTING MACHINE LABORATORY.

Programme Sheet 2 (b)

ROUTINE

/	/
E	E
@	@
A	A
:	:
S	S
I	I
U	U
½	½
D	D
R	R
J	J
N	N
F	F
C	C
K	K
T	T
Z	Z
L	L
W	W
H	H
Y	Y
P	P
Q	Q
O	O
B	B
G	G
"	"
M	M
X	X
V	V
£	£

MANCHESTER UNIVERSITY COMPUTING MACHINE LABORATORY

Check Sheet.

ROUTINE. Example 1.8

T :	clear	clear		
/ C / K			$D_{\pm f}$	a
: C / F	(ac)ls	(ac)ms		
/ C / U	(ac)ms	clear	/C	(ac)ls
: C / U	clear	clear	:C	(ac)ms
X : / C			D_+	2^{14}
: C / F	2^{14}(ac)ms	clear		
/ I	clear	2^{14}(ac)ms		
/ C / N	(2^{14} ac)ls	(2^{14}ac)ms		
I S T I	round-off			
/ C / A		clear	/C	2^{14}ac(rounded-off)
J : T /	set round-off			
F : / C			D_+	2^6
@ C / F	2^6b			
@ C T A	clear	clear	@C	2^6b
@ C / K			$D_{\pm f}$	2^6b
/ C / F	2^{20} abc			
I S T I	round-off			
/ C / A		clear	/C	2^{20}abc (rounded-off)

Supplement to the Programmers' Handbook

(2nd Edition).

Table of Contents.

Preface

Numerous alterations and additions have followed the first issue of the second edition of the Programmers' Handbook. The two most recent additions - some examples illustrating the construction of a complete programme and a chapter on output routines - have been large enough to justify issuing them separately as a supplement to the original Handbook.

The examples were originally intended as an appendix to Chapter 4 and hence bear the page numbers 4A.1 - 4A.37. The chapter on output routines was written as chapter 10 and is numbered accordingly.

We take this opportunity of describing some useful extensions which have been made to the basic routine B.INPUT. Since this routine was introduced various suggestions for new warning characters have been made, As a result two new characters M and X have been introduced. They are described below. A further suggestion, resulting in the warning character P, was furnished by a study of the input routine developed for FERUT - the computing machine, identical to this, which has been installed at the McLennan Computation Centre, Toronto University.

R. A. Brooker.

Extensions to the routine B.INPUT (17.11.52)

The routine B.INPUT has been extended to treat 3 new warning characters X, M, and P. One of these , P, necessitates starting a 3rd. page of instructions and for this purpose the right half of track 97 is used. The reader will recall (p.4.23) that only the first line pair of this page is at present used. This magnetic page is transferred to S0 whenever the warning character P is encountered and when the P sequence has been treated then the magnetic page 96L is restored to S0.

Warning character X.

This is a one character sequence and corresponds to the closure when B.INPUT is used as a sub-routine. The effect is to transfer control to MS. This character may thus only, sensibly, be used when PERM stands in S2, 3.

Warning character M (3 character sequence)

The effect of Mab is to obey the content of the short line ab as a magnetic instruction.

One application would be to "set" one or more of the electronic stores 4, 5, 6, and 7 directly from the tape before entering a routine. A means for effecting this has already been described (p.4.31) but M sequences provide a very much neater and more easily understood method of achieving such operations. In particular use can be made of the three lines K@, T@, and Z@ within B.INPUT itself (see p.4.22).

Warning character P (3 character sequence)

The effect of Pab is to replace the (short) line a b by (//rs) + [pq] , where [ab] = pqrs. Thus, e.g., if, when the sequence PD½ is read from the tape, [D½] = VFT½ and [VF] = /C//, then the effect is to replace VFT½ by /CT½.

Applications.

P sequences may be used to insert addresses into instructions and will thus be useful in connection with preset parameters such as are required by the routines B/RUNGE-KUTTA, B/ROOT/A, and B/QUAD/A. When reading these routines the parameters are first read into certain storage locations, usually in column U, and, after the punching proper has been read, an interlude is used to pick up these parameters and insert them into the the appropriate lines of the routine as it stands in S4 and / or S5, before transferring it to the magnetic store. Essentially the same technique is used, but in place of the interlude, which is included on the library tape, a set of P sequences provide a very much neater method of inserting the parameters into the routine. Thus, e.g., consider the case of B/RUNGEKUTTA, for which the routine and the interlude are described on pages 7.11 and 7.12. If it were intended to repunch this tape, then the following alterations would be made to the punching proper:

Line	S½	becomes	BUT:
"	J½	"	£UQ½
"	N½	"	VUQN
"	Z½	"	VUUK
"	P½	"	"U/N
"	O½	"	XUQ½
"	G½	"	XUQA
"	£½	"	GU/N
"	ED	"	VUUJ
"	@D	"	£UUK
"	UD	"	VUUA
"	CD	"	BU//

and in place of the interlude KYIJ,......, N@/PTHI would be punched the P sequences

PS½, PJ½, PN½, ..., PCD.

The effect, e.g., of PJ½ is to replace the line £UQ½ by (C)Q½, where (C)// has been preset in £U.

A further example of the use of P sequences is to transfer magnetic entries from a reading- in DIRECTORY to other pages of material. Thus, e.g., when using an electronic cue DIRECTORY as part of a WORKING PERM, with cues standing (say) in column I, the technique hithertoo used to read in the routines is to copy the WORKING PERM up to track 34 to serve as a reading-in DIRECTORY and follow the routines with destination sequences of the form Y.@ The warning character P however enables one to use a separate reading-in DIRECTORY consisting only of magnetic entries and to regard the magnetic halves of the cues in the cue DIRECTORY as parameters for the WORKING PERM. By including appropriate P sequences among the PERM alteration sequences these parameters can be selected from the reading-in DIRECTORY and planted in the appropriate lines of the WORKING PERM before transferring this to track 98.

The advantage of this technique is that a single reading-in DIRECTORY can serve for a number of different programme tapes, all of which have routines in common, but each of which uses a distinctive cue DIRECTORY. Indeed an isolated track containing an entry for each routine in the library could be made the basis for the manufacture of a WORKING (reading-in) DIRECTORY, by a device analogous to that used in making a WORKING PERM. Thus such a list would first be transferred to S4, S5, then entries inserted which correspond to the routines specially made for the programme, and finally the result transferred to track 34.

Alterations and Additions to the Isolated Material on Tracks 96 & 97.

(Compare with pages 4.34 & 4.35: the new contents of lines altered are shown below).

Isolated on 96R

C								A
				/				
				E				
				@	~~E~~ C	~~A~~ /	E	/
				A				
				:				
				S				
				I				
				U	F	A		
				½				
				D	/	/	I	:
				R	N	@	/	P
				J				
				N				
				F	½	A		
				C	@	A	/	:
				K	N	@	/	P
				T				
				Z				
				L				
				W				
				H				
				Y				
				P				
				Q	L	/		
				O				
				B				
				G				
				"				
				M				
				X	L	/		
				V	"	S		
				£				

Isolated on 97R

/								E
				/				
				E				
				@				
				A				
£	£	£	£	:				
£	£	/	/	S				
/	/	/	/	I				
/F	A/	/	/	U				
U	/	/	:	½				
/	E	U	P	D				
/	/	P	/	R				
S	/	T	R	J				
@	/	T	A	N				
@	/	Q	O	F				
/	/	Q	/	C				
/	/	P	I	K				
@	/	T	N	T				
@	/	T	A	Z				
@	/	Q	O	L				
:	/	P	G	W				
£	£	P	Z	H				
½	/	/	P	Y				
				P				
				Q	D	/	/	/
				O				
				B				
				G				
				"				
				M				
				X				
				V				
				£				

Example 1.

Calculation of the period of a single pendulum swinging through a finite angle

If a is the length of the radius vector from the centre of the circle to the particle and $\alpha = 2\pi R$ is the maximum angle which it makes with the downward vertical, then it can be shown (H.Lamb, Dynamics, p.107, 2nd. edition, Camb. Univ. Press, 1942) that the period of the motion is

$$4\sqrt{\frac{a}{g}}\int_0^{\frac{\pi}{2}} \frac{d\phi}{\sqrt{\cos^2\phi + \left(\frac{1+\cos\alpha}{2}\right)\sin^2\phi}} = 4\sqrt{\frac{a}{g}}\,K.$$

The ratio which this bears to the period of an infinitely small arc is thus

$$\frac{2}{\pi}\,K.$$

The programme is to read values of R from the tape and print the corresponding values of $\frac{2K}{\pi}$. The complete elliptic integral K is computed by a repetitive method based on the Gaussian form of Landen's transformation (Whittaker & Watson, Modern Analysis, p 533, 4th. edition, Camb. Univ. Press, 1950),

$$\int_0^{\frac{\pi}{2}} (a_1^2\cos^2\phi + b_1^2\sin^2\phi)^{-\frac{1}{2}}\,d\phi = \int_0^{\frac{\pi}{2}} (a^2\cos^2\phi + b^2\sin^2\phi)^{-\frac{1}{2}}\,d\phi$$

where $a_1 = \frac{1}{2}(a + b)$ and $b_1 = \sqrt{a\,b}$.

Thus if $a_{n+1} = \frac{1}{2}(a_n + b_n)$ and $b_{n+1} = \sqrt{a_n\,b_n}$ and λ is the common limit of a_n and b_n, then

$$\int_0^{\frac{\pi}{2}} (a^2\cos^2\phi + b^2\sin^2\phi)^{-\frac{1}{2}}\,d\phi = \frac{\pi}{2\lambda}\,.$$

The required ratio of the periods is then simply the reciprocal of the limit of the sequence with initial values

$$a_0 = 1 \quad\text{and}\quad b_0 = \sqrt{\frac{1+\cos\alpha}{2}}\,.$$

The MASTER Routine

Before attempting the detailed coding of the master routine it is first necessary to enumerate any library sub-routines which it is intended to use.

This example requires the following routines

```
(@ /   MASTER)
 : /   INPUT/B
 I /   PRINT/A
 ½ /   SINANDCOS/A (cosine entry)
 R /   SQUAREROOT/A
 N /   DIVISION/A
```

It is conventional to reserve / / for B.INPUT in case it should be needed.

The details of the MASTER routine are as follows.

K/½/

```
/ | / / T : | clear A
E | D S T / | set a0 ÷ 1
@ | @ U T A |
A | A / Q O | calls in INPUT/B:
: | G S / P | [M']+ƒ = R
S | : / / V |
I | / / / U | [L']+ƒ = R
U | U / Q O | calls in SINANDCOS/A:
½ | G S / P | [L']+ƒ = ½cos2πR
D | ½ / / V |
R | I S T I | [L']+ƒ = ½(1+cos2πR)
J | S E / P | Jump to set b0
N | I S T / | = √½(1+cos2πR
F | I S / C | form
C | / U / N | ½(an + bn) in M
K | @ U / N |
T | @ U / C |
Z | @ U / A | plant an+1
L | / U / N | form an bn
W | / / / U | [L']+ƒ = √an bn
H | H / Q O | calls in SQUAREROOT/A.
Y | G S / P | [L']+ƒ = √an bn
P | R / / V |
Q | / U T A | plant bn+1
O | @ U T ½ | an+1- bn+1
B | / U T N |
G | S : T N | an+1- bn+1- 2^-37
" | R E / H | test for convergence
M | I S T / |
X | I S / C | Dƒ' = λ/2
V | / U / N |
£ | / U / A |
```

K/DK

```
/ | / U / K |
E | N E / J | [M']±ƒ = ½ 10^-2
@ | @ E Q O | calls in DIVISION/A;
A | G S / P | [L']±ƒ = 10^-2/λ
: | N / / V |
S | W / / I |
I | F E / C | set digit layout
U | U E Q O |
½ | G S / P | print result
D | I / / V |
R | J / / L | dummy stop
J | D S / P | return to read tape
N | " A S M | ½ 10^-2
F | G A S / |
C | / / A ½ | digit layout const-
                  ant
```

The DIRECTORY

It remains to draw up the cue DIRECTORY. First magnetic storage is allocated to each routine and the appropriate track number inserted into the skeleton cue which is found from the list of library routines.

ROUTINE	MAG.STORAGE	DIRECTORY ENTRY
		K/½C
B.INPUT	96	/ G E / / E / A @ /
MASTER	4L	@ £ £ / / A : / / /
INPUT/B	5L	: / / / / S S / / /
PRINT/A	5R	I C E / / U S / E /
SINANDCOS/A	6L	½ S @ / / D I / / @
SQUAREROOT/A	6R	R E @ / / J I / E @
DIVISION/A	7L	N A @ / / F U / / @

TAPE ASSEMBLY

For demonstration purposes a main tape bearing all the routines and an auxiliary number tape, with the required values of R, will be the most convenient arrangement for input.

The main tape can be terminated with Z's so that it can be halted on the / G stop after reading in with / L off. The number tape can then be inserted. This should carry a starting sequence at its head. The general layout of both tapes is given on p.4A.4.

MAIN TAPE

DIRECTORY

YTP

MASTER

Y@/

INPUT/B

Y:/

PRINT/A

YI/

SINAND
COS/A

Y$\frac{1}{2}$/

SQUARE
ROOT/A

YR/

DIVISION/A

YN/

ZZ.....

NUMBER TAPE

Q@/
01+
02+
03+
04+
05+

etc

This number tape bears values of R = .01, .02, .03, etc..

Example 2

Solution of an Algebraic Equation with Real Roots using the Floating Decimal Interpretive Routine FLOATCODE.

General

The method used is to locate a root by an iterative method, divide the equation by the corresponding factor and repeat the process on the reduced equation. When finding the first root, zero is used as an initial approximation, thereafter each new root is used as a first approximation for finding the 'next' root. This device is useful in the case of double or multiple roots.

The iterative formula used is

$$x_{s+1} = x_s - \frac{f(x_s)\ f'(x_s)}{f'(x_s)^2 - a\,f(x_s)\ f''(x_s)} \qquad (1),$$

where a takes the values $\frac{1}{2}$ or 1. (See Bodewig, E., Quart. Appl. Math., 7, p.328 (1949).). With a = $\frac{1}{2}$, the formula yields cubic convergence in the 'neighbourhood' of a simple root, that is, a root of multiplicity one, and linear convergence in the case of a multiple root. With a = 1 the formula always yields quadratic convergence - whatever the multiplicity of the root.

The programme is so arranged that a can take either value simply by changing one 'instruction' in the principal sub-routine. This is effected by means of a correcting sequence (see p.4.25) at the end of the main tape which can be omitted or otherwise from the reading in process.

Details.

The programme will solve equations of degree ≤ 31.

The coefficients of the equation are read with FC/DECINPUT and results printed with FC/DECPRINT/A.

The material describing an equation is stored on a single electronic page. For example, if the page in question is S7 and the equation is

$$x^n + a_1 x^{n-1} + a_2 x^{n-2} + \ldots\ldots + a_{n-1}x + a_n = 0, \qquad (2)$$

then $[/\ C]_0^{19}$ = (2n-2)// and a_r = F(@C + 2n - 2r).

During the iterative process this material describing the equation being solved (that is the original equation or one of the subsequent reduced equations) is held in S7; S6 is used as working space; S2, S3, S4, and S5 contian FLOATCODE; the REDUCTION routine stands in S0; and S1 is not used.

All other routines when required are read down to S0 except FC/DECINPUT which replaces part of FLOATCODE (see p.5.13B)

In addition to the library routines FLOATCODE, FC/DECINPUT, and FC/DECPRINT/A, the programme is based on two new routines, the REDUCTION routine and a MASTER routine. They will be described in this order.

The REDUCTION routine.

This is the principle sub-routine. Its effect can be described as follows.

If the material describing an equation, $f(x) = 0$, is held in S7 and F(WI) is an initial approximation to a root, then F'(WI) is the final approximation (x_f) and the material describing the corresponding reduced equation, $f(x) / (x-x_f) = 0$, will be found in S6. In addition $[E\ N]^{19}_{0}$ contains the twenty most significant digits of the standard (floating) representation of the residual $f(x_f)$.

The routine is based on the following algorithms.

The quantities $f(\xi)$, $f'(\xi)$, and $f''(\xi)$ which appear in the iterative formula (1) are computed by means of the recurrence relations (see p.6.1)

$$p_1 = \xi + a_1 \qquad q_1 = 1 \qquad \rho_1 = 0$$

$$p_{r+1} = p_r \xi + a_{r+1} \quad q_{r+1} = q_r \xi + p_r \quad \rho_{r+1} = \rho_r \xi + q_r$$

$$p_n = f(\xi) \qquad q_n = f'(\xi) \qquad \rho_n = \frac{f''(\xi)}{2!}$$

Furthermore the p_r calculated by the first set of recurrence relations are also the coefficients of the quotient

$$q(x) \equiv \frac{f(x)}{x-\xi} = x^{n-1} + p_1 x^{n-2} + \ldots\ldots p_{n-2} x + p_{n-1},$$

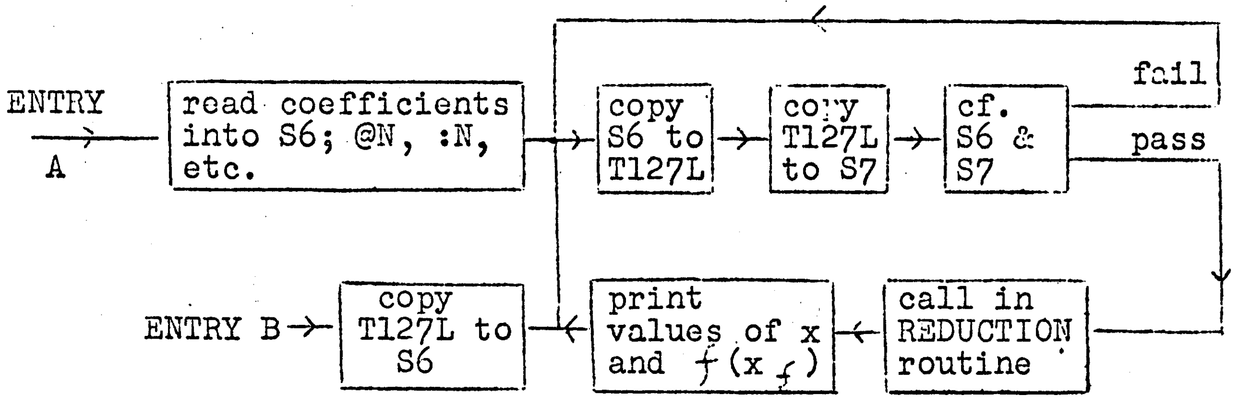
where $f(x) = q(x)(x-\xi) + f(\xi)$.

During the course of the calculation $f(x)$ is computed for a sequence of values of x terminating eventually at a value x_f such that $(x_f - x_{f-1})/x_f$ is less than 2^{-38}. At each step the p's are recorded in /N, @N, etc., each set being overwritten by the succeeding set. The set of p's which is finally recorded provides the coefficients of $f(x) / (x - x_f)$, the reduced equation. The residual $f(x_f)$ stands in /N and finally, before returning to the MASTER routine,the value of 2n-4, the degree of the reduced equation, is copied into the short line /N.

The detailed instructions are given on p. 4A.8

The MASTER routine

This is best explained with the aid of a flow diagram.


ENTRY
A
read coefficients into S6; @N, :N, etc.
copy S6 to T127L
copy T127L to S7
cf. S6 & S7
fail
pass
call in REDUCTION routine
print values of x and $f(x_f)$
copy T127L to S6
ENTRY B

The detailed instructions and 'instructions' are given on p.4A.9 It will be necessary to study the DIRECTORY in order to follow some of the details.

Notes.

1. The MASTER routine is called in from the tape as a sub-routine of B.INPUT. Immediately prior to this (2n-2)//, the 20 digit line defining the degree of the equation, will have been read into line /N (by means of sequences at the head of the number tape). Thus when the coefficients have been read all the material describing the equation is standing in S6.

2. Of the value printed for the residual of any root, only the exponent and the first three decimals will be significant.

3. The alternative entry B. This proves useful in the event of an interruption because of machine breakdown (or for any other

reason). It enables the process of root finding to be restarted with the last reduced equation which is described on track 127L.

REDUCTION ROUTINE

```
ENTER   K/½/
 /   /  | Q E T / | set q_1 = 1
     E  | Y I T A |
     @  | A / T / | set ρ_1 = 0
     A  | V E T A | (see note 1)
     :  | / C P O |  [B6'] = 2n-2
     S  | S / Q O | enter FLOATCODE
     I  | E I / P |
     U  | W I / G |  F'(R) = x_s
     ½  | W I / W |
     D  | @ C I W | set p_1 = x+a_1
     R  | / N I Y |
     J  | X / / " |
     N  | Y I T / | set q_r(r ≥ 1)
     F  | M E T A |
     C  | A : P G | count on r
     K  | @ / / O |
     T  | W I T / | transfer x_s to YI
     Z  | Y I T A |
     L  | I I / P | reenter FLOATCODE at
                    'instruction' standing
                                      in £/
     W  | W / Q O | enter FLOATCODE
     H  | E I / P |
     Y  | @ N I Q |
     P  | @ C I W | p_{r+1} = p_r x + a_{r+1}
     Q  | / N I Y |      (r ≥ 1)
     O  | M E / Q |
     B  | @ N I W | q_{r+1} = q_r x + ρ_r
     G  | Y I / Y |      (r ≥ 1)
     "  | V E / Q |
     M  | M E / W | ρ_{r+1} = ρ_r x + q_r
     X  | V E / Y |      (r ≥ 1)
     V  | J / / £ | quit FLOATCODE
     £  | M E / G |
```

```
        K/DB
 E   /  | M E / Q |  F'(A) = ρ'^2
     E  | / N / G |
     @  | V E / O | F'(A) = ρ'^2 - ρρ''
     A  | V E / O | (see note 2)
     :  | V E / Y |
     S  | M E / Q |  F'(A) = ρρ'
     I  | V E / B |  F'(R) = 1/(ρ'^2 - ρρ'')
     U  | V E / Y |
     ½  | V E / O |  F'(A) = - ρρ'/(ρ'^2 - ρρ'') = α
     D  | Y I / W | x_s + α = x_{s+1} to WI
     R  | W I / Y |
     J  | J E / £ |  quit FLOATCODE
     N  | W I T ½ | form difference
     F  | Y I T N |    x_{s+1} - x_s
     C  | A : / M |  test sign of difference
     K  | D S T J | change sign if -ve
     T  | E : T C |
     Z  | S : T N |  |x_{s+1} - x_s| - 2^-38
     L  | D S / H |  repeat iteration
     W  | / C P O |  set degree of
     H  | A : P G |  reduced equation
     Y  | / N P B |  (2n-4) in /N
     P  | N S / P |  closure
     Q  | / / / / |  F(QE) ≡ 1
     O  | / ½ : / |
     B  | spare   |
     G  | spare   |
     "  | spare   |
     M  |   q_r   | working q_n = ρ'
     X  |         |
     V  |   p_r   | space   p_n = ρ''/2
     £  |         |
```

Notes

1. F(A/) is approximately 2^{-115}, i.e., 'zero'.
2. If the 'instruction' standing in line AE is replaced by a dummy (../P), then the effect is to form $\rho'^2 - \frac{1}{2}\rho\rho''$ for the denominator of α. This results in the iteration formula which is cubically convergent for simple roots.

MASTER ROUTINE

```
ENTRY     K/½/                                        K/DE
  /   / | / E Y O ⌐ calls in FC/DECINPUT;/|@ N / /E  parameter for
      E | E / £ O | reads coefficients                 FC/DECINPUT
      @ | E I / P | into @N, :N, etc
      A | R / / V ┘
ENTRY←: | M / / P
  →   S | X / / :
  ε  →I | V / / : ⌐ copy coefficients of
      U | £ / / : | reduced equation into
      ½ | / N Q O   magnetic store & into
      D | E : / O   test for end of process       S7
      R | N S / P   return to B.INPUT
      J | V D Q G
      N | / N Q C ⌐ compare relevant
      F | / C Q N | lines of S6 & S7:
      C | A : Q G | repeat writing
      K | B I / O | operation until
      T | / / / A | check sums agree
      Z | E : T N | (see Example 3.2)
     ←L | M / / H ┘
      W | W / Q O ⌐ call in REDUCTION
      H | E I / P | routine
      Y | N / / V ┘
      P | W I / W   F'(A) = new root               ⌐ print value
      Q | I / / V   call in FC/DECPRINT/A          | of root on
      O | / / / Y   clear 'accumulator'            ┘ new line
      B | / N / W ⌐ print value of residual
      G | ½ / / V | similarly, but on same line
      " | / / / Y ┘
     └M | S / / £
      X | £ A / N   127L to S6  )  magnetic
      V | £ A ½ N   S6 to 127L  )  instructions
      £ | £ A / C   127L to S7  )
```

The DIRECTORY

```
ROUTINE          MAGNETIC STORAGE                DIRECTORY ENTRY

                                                 K/½T
                                              | / | G E / / |
B.INPUT                96                     | E | / A @ / |
                                              | @ | £ £ / / |
MASTER                 4L                     | A | : / / / |
(ENTRY A)                                     | : | : / / / |
DITTO                                         | S | : / / / |
(ENTRY B)                                     | I | : / / / |
FC/DECPRINT/A          5                      | U | S / / / |
                                              | ½ | ½ / / / |
DITTO (alternative entry omitting             | D | S / / / |
       carriage return & line feed)           | R | G J / / |
FC/DECINPUT            6                      | J | I / @ ½ |
                                              | N | £ £ / / |
REDUCTION              4R                     | F | : / E / |
Entry used only for reading in FC/DECPRINT/A  | C | S / @ / |
"     "    "    "   "       "  " FLOATCODE,Pt 2. | K | A A @ / |
```

For further details about the purpose of entries C/ and K/ see Note 2 on p. 5.13A and the first paragraph on p.5.12

LAYOUT OF TAPES

MAIN TAPE

DIRECTORY

YTP

FLOATCODE pt 1

FLOATCODE pt 2

WK/

FC/DEC PRINT/A

WC/

FC/DEC INPUT

KHDEAAA@D (see p.5.13B)

YR/

REDUCTION routine

YN/

MASTER routine

Y@/

Z....(/G stop)

HN/
KADE///P
YN/

} This correcting sequence to the REDUCTION routine is to be read only in the case $a = \frac{1}{2}$

Z....

NUMBER TAPE

LK/
K/NE(2n-2)//
G@/

} See p.5.12, paras. 2 and 3, and the note below

a_n
a_{n-1}
a_{n-2}
:
a_1

} coefficients of equation punched as described on p. 5.13A

" (see p.5.13B)

LK/G:/
LK/G:/
LK/G:/
etc..

} restarting sequences (entry B of MASTER routine)

For example the number tape for the equation

$$x^3 + 12x^2 + 44x + 48 = 0$$

is punched as follows

LK/
K/NE:///
G@/
48+1+
44+1+
12+1+
"
LK/G:/
LK/G:/
LK/G:/etc..

It is necessary to arrange the first two meaningful sequences of the number tape in this order otherwise [/ N] would be overwritten when the DIRECTORY is brought down to S6 and S7 to treat the L sequence.

The Compound Tape SIMULTEQN for the Solution of Sets of Linear Simultaneous Algebraic Equations

(For use with SCHEME B) by R. K. Livesley.

General

The compound tape contains the routine SIMULTEQN, a master routine, and the library sub-routines DIVISION/B and DEC.OUTPUT/B. It may be used for solving any set of linear equations $a_{ij}\ x_i = b_j$, where the number of variables n is $\leqslant$ 31, provided that a unique solution exists.

The method is the elementary one of elimination and back-substitution (Milne. Numerical Calculus. p.15) The programme is so arranged that on entry the equations may be either printed out or solved, while on subsequent re-entry they may be checked by direct substitution.

Input Organization

The Input Organization is described for a set of n equations. The equations are referred to by a suffix j, where j takes the values 0, 1, .. n-1, (instead of the more usual 1,...n,)

Before input of either the compound tape or the equations themselves a directory must be made up. This contains, besides the normal cues for the routines, a list of transfers for the equations themselves, each of which occupies a separate half-track in the magnetic store.

The make up of the Directory is as follows:-

Cue for Master routine		/	/ @ / /	/	. . . ½	E
Single page brought down only			α E @	E	. . . ½	
Cue for sub-routine			/ / / /	@	. . . ½	
'SIMULTEQN'			β @ /	A	. . . ½	
Cue for checking sub-routine			/ / / /	:	. . . ½	
(Part of Master routine)			α @ /	S	. . . ½	See note (I)
Cue for equation printing			/ E / /	I	. . . ½	
sub-routine (part of Master routine)			α @ /	U		
Space available for insertion				½		
of cue to private sub-routine				D		
Cues for	Fractions		@ / / /	R		
DEC.OUTPUT/B			γ @ /	J		
	Integers		½ / / /	N		
			γ @ /	F		
Cue for DIVISION/B			: @ / /	C		
			δ , . @	K		
Cue for B.DEC.INPUT			/ / / /	T		See note (2)
			ε @ /	Z		
				:		
(Track numbers α, β, γ, δ, ε must be assigned by the user)				£	. . . R	

Note. 1. Cues for equations. Single page reading transfers to Store 4. n transfers in all, standing in lines /E to n-1, E (Note. Transfer for j th equation stands in line j, E.)

Note. 2. All lines n,E - VE must be clear. Line £E is a single page reading transfer to Store 5. Half-track specified is used to store calculated values of the unknowns x_i.

The routines are designed to deal with equations in which the coefficients a_{ij} and the constants b_j are integers. (The equations will be solved correctly if both a_{ij}'s and b_j's are entered as fractions, but the equation-printing and checking sub-routines will then print incorrectly). It is also assumed that the unknowns x_i are fractions in the range $\pm \frac{1}{2}$, (if on solution any x_i is found to exceed this range the routine enters a hoot stop in column A). These conditions may always be met by multiplying individual equations or the constants be appropriate scale factors.

The programme will also fail, (by entering a closed loop in the division routine) if any coefficient on the leading diagonal of the matrix $[a_{ij}]$ is zero, or becomes reduced to zero during the course of the calculation. Unless the matrix is actually singular this may be avoided by rearranging the order of the equations.

As mentioned above, each equation is stored in a separate half-track. When any such half-track is read to S4 the coefficients a_{ij} - a_{nj} stand in the long lines /½ .... 2(n-1), ½; and the constant b_j appears in the long line VD.

It is assumed that the equations will normally be read in using B.DEC.INPUT, and for this reason the cue for this routine is included in the directory.

The equation tape will then take the form ;-

GT/	Enters B.DEC.INPUT
W/½,n	reads a_{ij} .... a_{nj} to lines /½ .... 2(n-1), ½
WVDE	reads b_j to line VD.
C	returns to B.INPUT
W j,E	Writes up j th equation using line j E of directory.
.	
.	
.	
GT/ etc..	commences reading (j + 1) th equation.

The Routines. General Effects.

The routines use S0, S1. S2 is used for PERM. The routine uses lines TI - £I as working space, together with S4 & S5. The directory is kept in S7, and S6 is not used. If the equations are read in as above with the dummy stop /L 'on', the tape terminating in Q//, the machine will eventually reach a /L stop in line N@ of the master routine. Subsequent behaviour depends on the setting of the hand-switches. The effects are as follows:-

(1) Set H = ////, operate KCS.

Enters routine SIMULT-EQN. The equations are solved and the unknowns, if in the range ± ½, are printed out as fractions on the ± convention. If the range has been exceeded at any point a hoot stop is entered in column A.

Printing is normally to 9 decimal places, the numbers being arranged in 5 columns. Modifications to this scheme may be made by altering certain lines in DEC.OUTPUT/B in accordance with the library account of that routine.

The routine ends on a /L stop. The unknowns will be found in S5, and this page will also have been written up to the half track specified by line £E of the directory. If a prepulse is given, the machine returns to the /L stop in the master routine.

(2) Set H = @///, operate KCS.

Enters the checking routine, which forms part of the master routine. The routine brings down the track specified by line £E of the directory, (which will contain the unknowns x_i if the routine SIMULT-EQN has previously been operated) and forms and prints the quantities $b_j - a_{ij}x_i$, b_j, for each equation. Each pair of values occupies a separate line of printing, and numbers are printed as integers.

It should be remembered that the routine SIMULTEQN alters the original equations, so the most satisfactory check is obtained if these equations are first reformed by re-running the equation tape.

Control eventually returns to the /L stop in the master routine.

(3) Set H = ;///, operate KCS.

Enters the equation-printing routine, which forms part of the master routine. The constants b_j and the coefficients a_{ij} are printed out in order as follows:-

⋮

2 line feeds

(j) Single character (letter shift) identifying equation.

2 line feeds

$a_{ij} - a_{nj}$ Normally printed in 5 columns. Printing as integers.

2 line feeds

b_j Printed as integer

2 line feeds

(j+1) Next equation

⋮ ⋮

control eventually returns to the /L stop in the master routine.

(4) Set H = I///, operate KCS.

The routine whose cue stands in line ½/ of the directory is entered as a sub-routine. This cue must be inserted previously by the operator when making up the directory. It enables some operation to be performed on the calculated x_i's.

Notes.

(1) Other settings of the hand switches will initiate various effects.

(2) If the programme is used with the punch switched on and the tape produced is subsequently read on the teleprinter on figure-shift, the printing will correspond to that obtained directly on the printer attached to the machine. The only difference will be that when printing out equations the character j denoting the equation number will also be printed on figure shift, and will thus not appear as a number for $j > 9$.

(3) Accuracy. This depends on the initial set of equations and the condition of the matrix $[a_{ij}]$. The errors $b_j - a_{ij} x_i$ printed out by the checking routine should never exceed $^n/_2$, (provided of course no machine errors have been made) and since

this is due to rounding off will probably be much less. In practice the error has rarely exceeded the range $\pm$ 5 for any set of equations.

It will be noted that the error is independent of the magnitude of the numbers involved. It is therefore advantageous to scale up the equations as far as possible. It should however be remembered that the equations are altered during the process of solution, and this may increase some of the coefficients, so that a margin of safty should be left. It will generally be sufficient if initially all numbers are in the range $\pm 2^{30}$.

(4) During solution read and write transfers are not checked. It is therefore desirable to use reliable tracks for storing equations.

The Routines. Detailed method of Operation.

The instructions of the MASTER routine and the routine SIMULTEQN are given. The sub-routines for checking and equation printing are straightforward and are therefore not included

MASTER ROUTINE

```
|@| @ E / C | /
    / @ / : | E
    / : P O | @
    S S Q O | A
    E : P G | :
    £ K P L | S
    N @ / T | I
    Y @ P G | U
    T I P B | ½
    Z I T B | D
    T T T K | R
    L I T A | J
    @ @ / L | N
    £ A / Z | F
    L @ T / | C
    V A T I | K
    L @ T A | T
    Z @ Q O | Z
    G S / P | L
    @ / / V | W
    P @ / P | H
    E £ / / | Y
    J @ / / | P
```

Bring down directory to S7.

Counting sequence. Initially B6 = //E/

B7 = ///½

B6 = ££//. Reduced by 1 each cycle

B7 = ///½ - line in directory corresponding to 1st ch. (of B6.

Test B7. exit from cycle when line subtracted (contains a transfer

B6 = n-1

Long line TI = n-1.

Long line LI = 2(n-1).

Dummy stop. Hand switches set here.

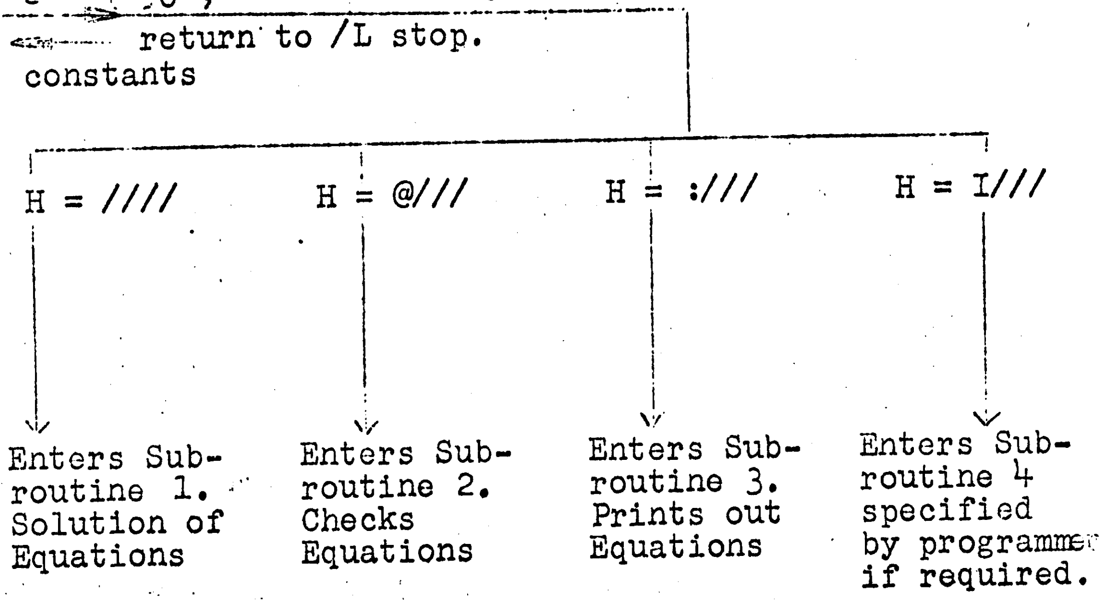
$[£\ A]_0^{19} = H$

Alter short line W@ to @ + $[H]_0^4$, //V

Enter routine whose cue lies in line @ + $[H]_0^4$, / of directory as sub-routine

return to /L stop.

constants



H = ////

Enters Sub-routine 1. Solution of Equations

H = @///

Enters Sub-routine 2. Checks Equations

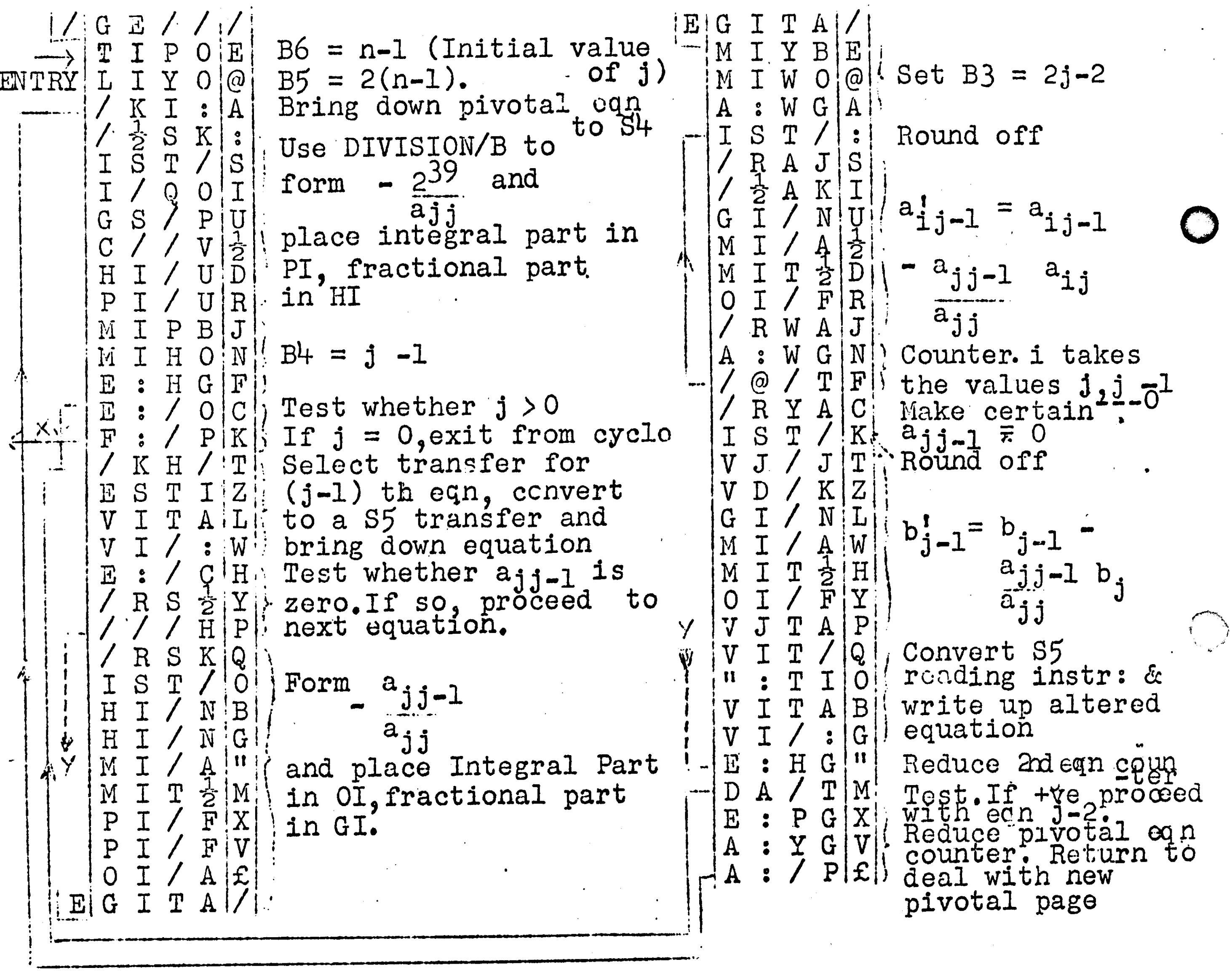
H = :///

Enters Sub-routine 3. Prints out Equations

H = I///

Enters Sub-routine 4 specified by programmer if required.

ROUTINE SIMULTEQN PAGE 1.



```
/ G E / / /
  T I P O E
  L I Y O @
  / K I : A
  / ½ S K :
  I S T / S
  I / Q O I
  G S / P U
  C / / V ½
  H I / U D
  P I / U R
  M I P B J
  M I H O N
  E : H G F
  E : / O C
  F : / P K
  / K H / T
  E S T I Z
  V I T A L
  V I / : W
  E : / Ç H
  / R S ½ Y
  / / / H P
  / R S K Q
  I S T / O
  H I / N B
  H I / N G
  M I / A "
  M I T ½ M
  P I / F X
  P I / F V
  O I / A £
E G I T A /
```

B6 = n-1 (Initial value of j)
B5 = 2(n-1).
Bring down pivotal eqn to S4

Use DIVISION/B to form $-\frac{2^{39}}{a_{jj}}$ and place integral part in PI, fractional part in HI

B4 = j -1

Test whether j > 0
If j = 0, exit from cyclo
Select transfer for (j-1) th eqn, convert to a S5 transfer and bring down equation
Test whether a_{jj-1} is zero. If so, proceed to next equation.

Form $-\frac{a_{jj-1}}{a_{jj}}$ and place Integral Part in OI, fractional part in GI.

```
E G I T A /
  M I Y B E
  M I W O @
  A : W G A
  I S T / :
  / R A J S
  / ½ A K I
  G I / N U
  M I / A ½
  M I T ½ D
  O I / F R
  / R W A J
  A : W G N
  / @ / T F
  / R Y A C
  I S T / K
  V J / J T
  V D / K Z
  G I / N L
  M I / A W
  M I T ½ H
  O I / F Y
  V J T A P
  V I T / Q
  " : T I O
  V I T A B
  V I / : G
  E : H G "
  D A / T M
  E : P G X
  A : Y G V
  A : / P £
```

Set B3 = 2j-2

Round off

$$a'_{ij-1} = a_{ij-1} - \frac{a_{jj-1}}{a_{jj}} a_{ij}$$

Counter. i takes the values j, j-1, ... 0
Make certain $a_{jj-1} \neq 0$
Round off

$$b'_{j-1} = b_{j-1} - \frac{a_{jj-1}}{a_{jj}} b_j$$

Convert S5 reading instr: & write up altered equation
Reduce 2nd eqn counter
Test. If +ve proceed with eqn j-2.
Reduce pivotal eqn counter. Return to deal with new pivotal page

Notes. Initially TI = n-1, LI = 2(n-1) and S7 contains the Directory. (effects of Master Routine).

Each equation is brought down to store 4 in turn as the pivotal equation. The cycle is as follows.

(1) Bring down j th equation to S4 as pivotal equation

(2) Bring down (j-1) th equation to S5

(3) Test whether a_{jj-1} is zero. If so pass on to (j-2) th equation (5).

(4) If not, reduce a_{jj-1} to zero by multiplying S4 by $-\frac{a_{jj-1}}{a_{jj}}$ and adding to S5. Write up new j-1 th equation.

(5) Repeat for equations j-2, j-3 ..... , 0.

(6) Bring down (j-1) th equation as pivotal equation, etc..

ROUTINE SIMULTEQN PAGE 2.

```
   |@|A E / /|/
X→ | / R T A |    Clear remaining line of S5
   | T I W O |@   Set B3 = n-1
   | / K I : |A   Bring down jth eqn (Initially B6 = 0)
   | M I Y B |:   } Set B4 = 2j
   | M I H O |S   }
   | I S T / |I   Round off j-1
   | V D / J |U   Form d_j - Σ_0 C_ij x_i.
   | / R : K |½   Actual cycle sums
   | / ½ : D |D   Σ_0^j C_ij x_i, but since x_j
   | A : H G |R
   | ½ A / T |J   has not yet been worked
   | O I / U |N   out and store line contents are ∴ =0, effect is
   | / ½ S K |F   the same.
   | C @ Q O |C   Divide by C_ij & plant
   | G S / P |K   answer in S5 i.e.
   | C / / V |T   x_j = 1/C_jj [d_j - Σ_0^(j-1) C_ij x_i]
   | / R S S |Z
   | A : / M |L   Replace x_j in A by
   | D S T J |W   its modulus
   | E : T I |H
   | I S T C |Y   Subtract ½
Z← | Q @ / H |P   If x_j ≥ ½ enter hoot stop
   | J A T £ |Q
   | D S Y G |O   Increase B5 to 2(j+1)
   | D S Y G |B   and B6 to (j+1)
   | D S P G |G
   | E : W G |"   Reduce B3
   | £ @ / T |M   Test. If +ve, treat next eqn
   | R A T / |V   Setting for printing n
   | T I T I |X   values x_j.
   | @ @ / : |£   Line feed.
```

```
|A| / A Q O|/
   | G S / P|E   Print out x_j' s.
   | R / / V|@
   | £ K T /|A
   | " : T I|:   Write up x_j' s to magnetic store
   | V I T A|S
   | V I / :|I
   | N S / P|U → Return to master Routine
   | U @ / /|½
   | K / / /|D
   | E / / /|R   Basic setting for printing x_j' s
   | / R / /|J
   | / / / V|N   ← Z
   | / / / V|F   Hoot stop if |x_j| > ½
   | Q @ / P|C
```

When all equations have been treated as pivotal equations, the set of equations has been altered to the following form:-

$$C_{00}\,x_0 = d_0$$

$$C_{01}\,x_0 + C_{11}\,x_1 = d_1$$

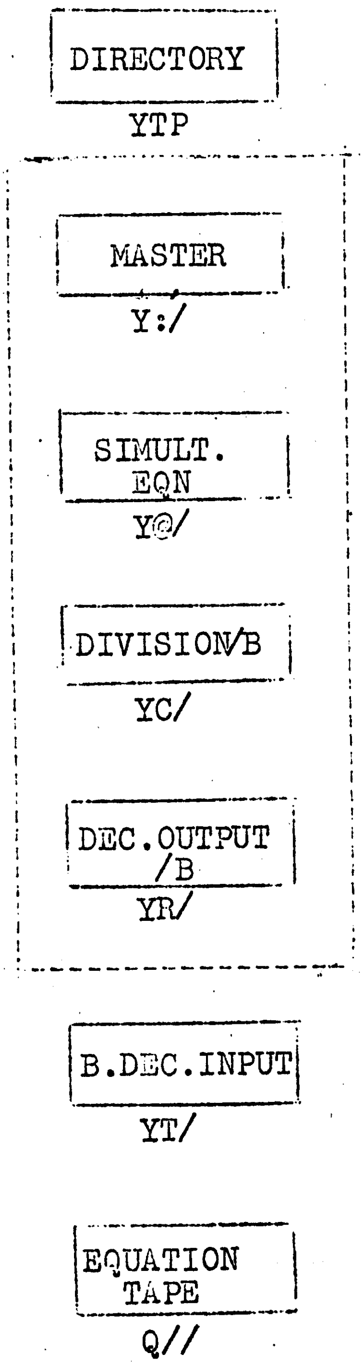
$$\vdots$$

$$C_{0n-1}x_0 \;\cdots\cdots\; C_{n-1n-1}\,x_{n-1} = d_{n-1}$$

hence $x_0 = d_0/C_{00}$, $x_j = \frac{1}{C_{jj}}\left[d_j - \sum_0^{j-1} C_{ij}\,x_i\right]$

Make up of Tape

(H = /A@/)



Example 4

Solution of a pair of differential equations with a closed boundary condition.

General

This relates to the solution of a pair of differential equations which arise in the theory of the flow of water through a manifold.

The equations are

$$\frac{dh}{dx} = -0.00109\, v\sqrt{h} + k\,v^2$$

$$\frac{dv}{dx} = 0.0349\sqrt{h}, \qquad k \text{ being a parameter.}$$

It is required to find a solution satisfying the boundary conditions

$$\text{at } x = 0,\ v = 0,\ h = ?$$

$$\text{at } x = 100,\ h + v^2/64 = 50,$$

in particular to find h_0, h_{100}, and v_{100}.

Solutions are required for various values of the parameter k (and if necessary alternative values for the other coefficients).

It is known that $0 \leqslant h, v < 56$. Thus we cannot represent these quantities directly inside the machine: but by introducing new variables ϕ, θ where

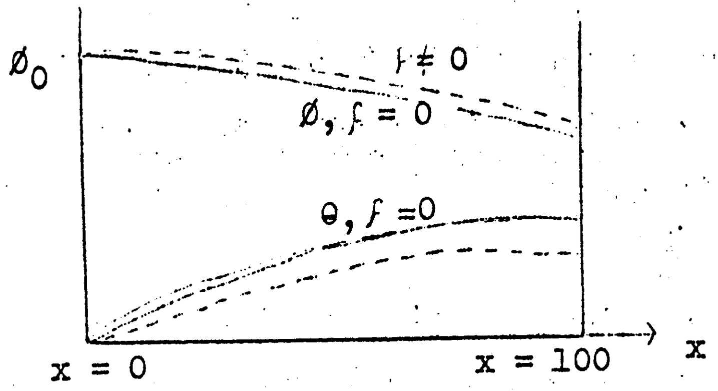
$$h = 256\,\phi \quad \text{and} \quad v = 256\,\theta,$$ the equations and boundary conditions become

$$\frac{d\phi}{dx} = -0.01744\,\theta\sqrt{\phi} + f\,\theta^2,$$

$$\frac{d\theta}{dx} = 0.00218125\sqrt{\phi},$$

$$\theta_0 = 0,\ \phi_{100} + 4\,\theta_{100}^2 = 50/256,$$ in which

all relevant quantities lie within a suitable range, k being such that f lies in the range $0 \leqslant f < 1$. The general behaviour of the equations for a given value of ϕ_0 is shown in the accompanying figure.



The broken curves correspond to a small positive value of f.

Method

The method is to carry out integrationsover the whole range using trial starting values of $\emptyset$. At the end of each run the value of the difference $\emptyset + 4\theta^2 - 50/256$ is computed and used to readjust the starting value $\emptyset_0$.

The programme makes use of the following library routines:

B.DEC.INPUT	for reading in the coefficients occurring in the equation;
B.RUNGE-KUTTA	for advancing the integration;
SQUAREROOT/A	for calculating $\sqrt{\emptyset}$;
B/ROOT/A	for adjusting the value of $\emptyset_0$; and
PRINT/A	for printing the final value of $\emptyset_0$.

The programme itself - the 'cementing' instructions - are embodied in two further routines. The MASTER routine and an AUXILIARY routine (for B/ROOT/A). These are briefly described below. The detailed instructions are given on the following pages together with the distribution of working space and the make up of the complete tape. In order to follow all the details it will be necessary to study the specifications of the library routines involved, particularly those of B/RUNGE-KUTTA (p.7.11) and B/ROOT/A (p.6A.4)

The MASTER routine

This stands in column U, thus forming part of a WORKING PERM. The DIRECTORY stands in column I.

The MASTER routine initiates, in the order given, the following operations:

1. Calls in B/ROOT/A to compute $\emptyset_0$;
2. Prints the values of $256\emptyset_0/100$, $256\emptyset_{100}/100$, and $256\theta_{100}/100$;
3. Calls in B.INPUT (as an ad-routine) to read a further value of the parameter f, or further values of the other coefficients.

The AUXILIARY routine

This sub-routine itself consists of two parts.

Part 1 (which stands in column $\frac{1}{2}$) is a main sequence having the following effects:

1. Sets initial values ϕ_0 and θ_0;
2. Clears the q locations (see spec. of B/RUNGE-KUTTA)
3. Calls in B/RUNGE-KUTTA for 100 steps;
4. Forms $\phi + 4\theta^2$ in L;
5. Returns to B/ROOT/A.

Part 2 (column D) is the suxiliary sequence required by B/RUNGE-KUTTA for calculating the values of $2^m h\phi'$ and $2^m h\theta'$, where h is the size of step and m is so chosen that these quantities are represented to a reasonably good precision, but without exceeding capacity. The values h=1 and m=0 were chosen for this programme.

The DIRECTORY

An electronic cue DIRECTORY is used. This stands in lines NU to MU inclusive. In the interests of time economy certain routines are called in as electronic routines and in these cases separate entries are needed for reading in the routines initially.

ROUTINE	No.	MAGNETIC STORAGE	ELECTRONIC STORAGE	TYPE	CUE
B.INPUT	//				GE// /A@/
MASTER	@/		Column U	'electronic'	£I// //£/
B/ROOT/A	:/	5L	S1	usual	£E// S//@
AUXILIARY	I/	5R	S4	usual	£U// S/E½
B/RUNGE-KUTTA	½/	4L	S0	'electronic'	££// //£/
SQUAREROOT/A	R/	4R	S1	'electronic'	E@// //£/
PRINT/A	N/	6L	S0	usual	CE// I///
B.DEC.INPUT	C/	7	S0 & S1	usual	//// U/@/
B/RUNGE-KUTTA & SQUAREROOT/A	entry used for the purpose of reading these routines initially and subsequently for transferring them to S0 & S1 (see note below).				:/@/

Note. The routine B/RUNGE-KUTTA and SQUAREROOT/A are transferred to S0 and S1 by the instruction standing in line :½ of the AUXILIARY routine. They remain in the electronic store for the duration of the integration.

WORKING PERM

(DIRECTORY and MASTER ROUTINE)

```
    K Q D @
    F S T £   sée p.4.28
    N I U F
 I                K / J M U
            | / | G U T / |  set L and M as required
            | E | O U / J |  by B/ROOT/A
            | @ | @ U Q O |
            | A | G S / P |  call in B/ROOT/A
            | : | : / / V |
            | S | @ R T A |
            | I | S : P O |  set counter to print 3 numbers
            | U | @ R I C |
            | ½ | P U / N |  form  256Ø    256Ø        and  256θ100
            | D | P U / N |        100 0,  100 100,         100
            | R | / R / E |  in M for printing
  | K N R Z | J | / R / J |
  | G E / / | N | H U / C |  set digit layout constant
  | / A @ / | F | F U Q O |
  | £ I / / | C | G S / P |  call in print routine
  | / / £ / | K | N / / V |
  | £ E / / | T | A : P G |  test for end of printing cycle
  | S / / @ | Z | Y U / T |
  | £ U / / | L | L U Q O |  return to read
  | S / E ½ | W | Q S / P |  tape
  | £ £ / / | H | / / / V |
  | / / £ / | Y | I U T E |  control number: also digit layout constant
  | E @ / / | P | Z J K H |  0.64
  | / / £ / | Q | T J K H |
  | C E / / | O | / / / / |   50
  | I / / / | B | / / ½ I |  256
  | / / / / | G | / / / / |  initial approx. (see specification
  | U / @ / | " | / / ½ I |  of B/ROOT/A)
  | : / @ / | M |
  |         | X |
  |         | V |
  |         | £ |

DIRECTORY entries
```

AUXILIARY ROUTINE for B/ROOT/A

```
                              Part 1      Part II
                            ½ |K / ½ Y   K / D T|D
ENTER--> set Ø0               |: R T A|/|: R T /|  [L'] +f  =  Ø <-- ENTER
         set θ0 = 0           |I R T A|E|E D Q O|  call in SQUAREROOT/A:
             clear            |/ R T A|@|G S / P|
           q locations        |@ R T A|A|R / / V|  [L'] +f  = √Ø
  S0<-B/RUNGE-KUTTA           |M I / :|:|R R T A|  form and plant
  S1<-SQUAREROOT/A            |W ½ Y O|S|R R / C|  .00218125√Ø
             set count        |I ½ Q O|I|I S T /|
               call in        |G S / P|U|C D / N|    (= θ')
         B/RUNGE-KUTTA        |½ / / V|½|R R / A|
                              |E : Y G|D|T D / ½|  form and plant
  test for 100 steps          |H ½ / T|R|I R / C|
                              |I R / C|J|L D / N|  -.01744θ√Ø + fθ²
              form 4θ²        |I R / N|N|½ R / A|
                              |I R / N|F|½ R / F|    =  Ø'
                              |I R / N|C|½ R / A|
                              |I R / N|K|/ / / P|  return to RUNGE-KUTTA routine
              form            |/ R / U|T|.00218125|  These constants are
              Ø + 4θ²         |: R T I|Z|       |    read from a number
  return to B/ROOT/A          |N S / P|L|  01744|    tape with B.DEC.INPUT
                   99=        |A A / /|W|       |
                              |S ½ / /|H|   f   |
                              |       |Y|       |
                              |       |P|       |
                              |       |Q|       |
                              |       |O|       |
                              |       |B|       |
                              |       |G|       |
                              |       |"|       |
                              |       |M|       |
                              |       |X|       |
                              |       |V|       |
                              |       |£|       |
```

WORKING SPACE: $q_Ø$, $q_θ$, Ø, θ. Ø', and θ' are stored in line pairs /R to RR respectively. Line pairs NR to PR inclusive are used as working locations by B/ROOT/A.

Layout of Tape.

Main tape

```
           HCP
    ---------------
    |  WORKING    |
    |   PERM      |
    ---------------
          YKP
          YTP
    ---------------
    | B.DEC.INPUT |
    ---------------
          YG@
    ---------------
    |   PRINT/A   |
    ---------------
          YO@

  KTUAI///              parameters
      NR//              needed by
      R://(ε = 2^-15)   B/ROOT/A
    ---------------
    |  B/ROOT/A   |
    ---------------
          Y:@
    ---------------
    |  AUXILIARY  |
    | SUB-ROUTINE |
    ---------------
          YI/

  KBUU@///(n=2)
      E://(m=0)
      DS//            parameters
      £½//            needed by
      :R//            B/RUNGE-KUTTA
      /R//
      ½R//
    ---------------
    |  B/RUNGE-   |
    |    KUTTA    |
    ---------------
    ---------------
    |   SQUARE    |
    |   ROOT/A    |
    ---------------
          WM@

          ZZ....
```

Number tape

```
HI/      read AUXILIARY routine to
GC/      call in B.DEC.INPUT     (S4
FCDA        insert
00218125+   coefficients
01744+      of diff. eqns.
:
C        return to B.INPUT
YI/      rewrite AUXILIARY routine
Q@/      calls in MASTER routine
 .
 .
 .
 .
Q@/
Q@/      restarting sequences
Q@/
```

Example 5

Forward Intregration of a set of Simultaneous Differential Equations

General

This example relates to the solution of a set of differential equations which arise in the subject of internal ballistics.

The problem has already been programmed for the EDSAC - the electronic digital computer at the University Mathematical Laboratory, Cambridge - and the following account is taken from pages 39 - 41 of Memo No. 7/51 by K. N. Dodd and A. E. Glennie, Armament Research Establishment, Fort Halstead, Kent.

Simultaneous Differential Equations

"The second problem is to solve the equations:

$$\frac{d^2\xi}{dt^2} = \pi - r_o(1 - \xi) \quad \text{for} \quad 0 \leq \xi \leq 1 \qquad \ldots\ldots(1)$$

$$= \pi \quad \text{for} \quad \xi > 1 \qquad \ldots\ldots(2)$$

$$\Phi = (1 + \xi - B\,\Phi)\pi + (\bar{y} - 1\int_0^{\xi} \pi \, d\xi \qquad \ldots\ldots(3)$$

$$\frac{d\Phi}{dt} = \frac{4\theta}{M}^{\frac{1}{2}} (1 - \Phi)^{\frac{1}{2}}\pi \qquad \ldots\ldots(4)$$

where π, ξ, Φ are functions of r.

It is given that when $t = 0$, $\pi = r_o$ and $\xi = \frac{d\xi}{dt} = 0$. B, $\bar{y}$, r_o $\frac{M}{\theta}$ are constants.

It is required to find the maximum value of π and the value of Φ at which it occurs and also the value of ξ, $\frac{d}{dt}$ and π when $\Phi = \frac{5}{9}$, $\frac{40}{49}$, $\frac{15}{16}$, $\frac{80}{81}$ and 1. These things are required for various values of the constants B, $\bar{y}$, r_o and $\frac{M}{\theta}$.

It is known that Φ starts near 0 and runs to 1 and ξ, $\frac{d\xi}{dr}$ and π are likely to obtain the value of several units. All are positive.

It was decided to use the sub-routine G.1$^{(1)}$ to solve the equations. This necessitated reducing the equations to the form

$$y_r' = f_r\,(y_1, y_2, \ldots, y_n) \quad r = 1, \ldots, n.$$

Also, the values of the variables must not exceed unity (i.e., the capacity of the machine) so it was necessary to change the variables in the equations.

First, equation (3) is differentiated with respect to t:

$$\frac{d\Phi}{dt} = (1 + \xi - B\,\Phi)\,\frac{d\pi}{dt} + \left\{\frac{d\xi}{dt} - B\,\frac{d\Phi}{dt}\right\}\pi + (\bar{y} - 1)\,\frac{d\xi}{dt}\,\pi$$

so
$$\frac{d\pi}{dt}\,(1 + \xi - B\,\Phi) = \frac{d\Phi}{dt}\,(1 + B\pi) - \bar{y}\,\frac{d\xi}{dt}\,\pi \qquad \text{......(3a)}$$

The following changes of variables are made

$$\frac{d\xi}{dt} = 100\ n$$

$$\xi = 100\ X$$

$$\pi = 100\ P$$

$$\Phi = 10\ S$$

Powers of 10 are used as factors so that the printed results contain the same digits as the values required.

Inserting these changes in equations (1), (2), (3a) and (4) and writing $\frac{M}{\theta} = K$ for simplicity, we obtain the equations

$$\frac{dn}{dt} = P - r_o\left\{\frac{1}{100} - x\right\} \quad \text{for} \quad 0 \leqslant X \leqslant \frac{1}{100} \qquad \text{......(1b)}$$

$$= P \quad \text{for} \quad X > \frac{1}{100} \qquad \text{......(2b)}$$

$$\frac{dP}{dt} = \left[\frac{1}{10}\,\frac{dS}{dt}\left\{\frac{1}{100} + BP\right\} - \bar{y}\,\eta\,P\right] \Big/ \left[\frac{1}{100} + X - \frac{BS}{10}\right] \qquad \text{...(3b)}$$

$$\frac{dS}{dt} = \frac{20}{\sqrt{K}}\,(1 - 10\ S)^{\frac{1}{2}}\,P. \qquad \text{......(4b)}$$

We also have the equation

$$\frac{dX}{dt} = \eta \qquad \text{.....(5b)}$$

The initial values are $\eta = X = 0$, $P = \frac{1}{100}\,r_o$.

From equation (3), putting $\xi = 0$ we obtain

$$\Phi = (1 - B\,\Phi)\pi = (1 - B\,\Phi)\,r_o$$

$$\therefore\ \Phi = r_o/(1 + B\,r_o)$$

$$\therefore\ S = \frac{1}{10}\,\frac{r_o}{(1 + B\,r_o)}\ , \text{ initially}$$

$$= \frac{r_o/100}{\left\{\frac{1}{10} + \frac{B\,r_o}{10}\right\}}$$

This last step is taken because otherwise the denominator would exceed capacity as B and r_o are positive.

Having got the equations into a form suitable for machine computation, it must be decided what printing is required. It is a waste of machine time to print out unnecessary values so arrangements must be made to print out values only at certain stages of the calculation.

It was decided to print η, X, S and P in four columns, to 10 decimal places, with a space after the fifth digit and three spaces between columns. P.11 was used for printing $^{(2)}$ The value of Φ is tested after each step of the integration. At first it is tested to see if $\Phi = \frac{5}{9}$. When this occurs, the values of η, X, S and P are printed for this and the previous step. The values of the variables at $\Phi = 5/9$ can then be found by a linear interpolation on a desk machine. This interpolation might have been done in the machine but there was insufficient storage space available. After that, the test is altered so as to test for $\Phi > \frac{40}{49}$. To test for the maximum of P, $P_n - P_{n-1}$ is calculated after each step and the sign of this is tested. As soon as it becomes negative, P_{n-1} is printed together with the corresponding values of n, X and S. To distinguish this line of printing, $\neq$ is also printed.$^{(3)}$ ..... the values of B, $\bar{y}, r_o$, and K were fed in at the beginning of the calculation of each set of solutions. By a "set of solutions" is meant all the required information for a given value of each of the parameters B, $\bar{y}$, r_o and K."

(1) corresponds to B/RUNGE-KUTTA

(2) " " " " DEC.OUTPUT/B (but the style of printing will be slightly different)

(3) a different device is used in the programme given here.

With these exceptions the whole of this account remains valid.

In addition to the two routines already mentioned the following library routines are needed:-

B.DECINPUT to insert decimal information into routines;

B/DIVISION/C to calculate S_o, and P' at each step;

SQUAREROOT/A to calculate S' at each step.

The operations described in the last paragraph of the account are embodied in a MASTER routine. The calculation of the derivatives is the task of the AUXILIARY sequence which must be prepared for use with B/RUNGE-KUTTA. To follow these routines it will be necessary to study the specifications of the routines involved and the distribution of numerical information and working locations (which is given on pages 4A.35 and 4A.36.

The AUXILIARY sequence

This sequence of instructions is intended to be stored in S4 and a few lines of S5. It is transferred to these stores at the beginning of each run (by the sequence LO@ on the number tape) and remains there for the duration of the run. The instructions are given on p. 4A.35, together with explanatory annotations. The effects can be summarised as follows:-

Given values of γ, X, S, and P in the line pairs ½C, RC, NC, and CC, the routine computes and places the values of $2^m h\gamma'$, $2^m hX'$, $2^m hS'$, and $2^m hP'$ (defined by equations 1b to 5b) in OC, GC, MC, and VC. h is $\frac{1}{32}$ so that if $m = 5$, then $2^m h = 1$. In order to alter the integration step it is only necessary to alter a preset parameter. Thus a set of solutions with $h = \frac{1}{64}$ can be obtained by changing m from 5 to 6.

The sequence uses SQUAREROOT/A and B/DIVISION/C as electronic sub-routines.

The MASTER routine (corresponds to electronic storage S0 and S1)

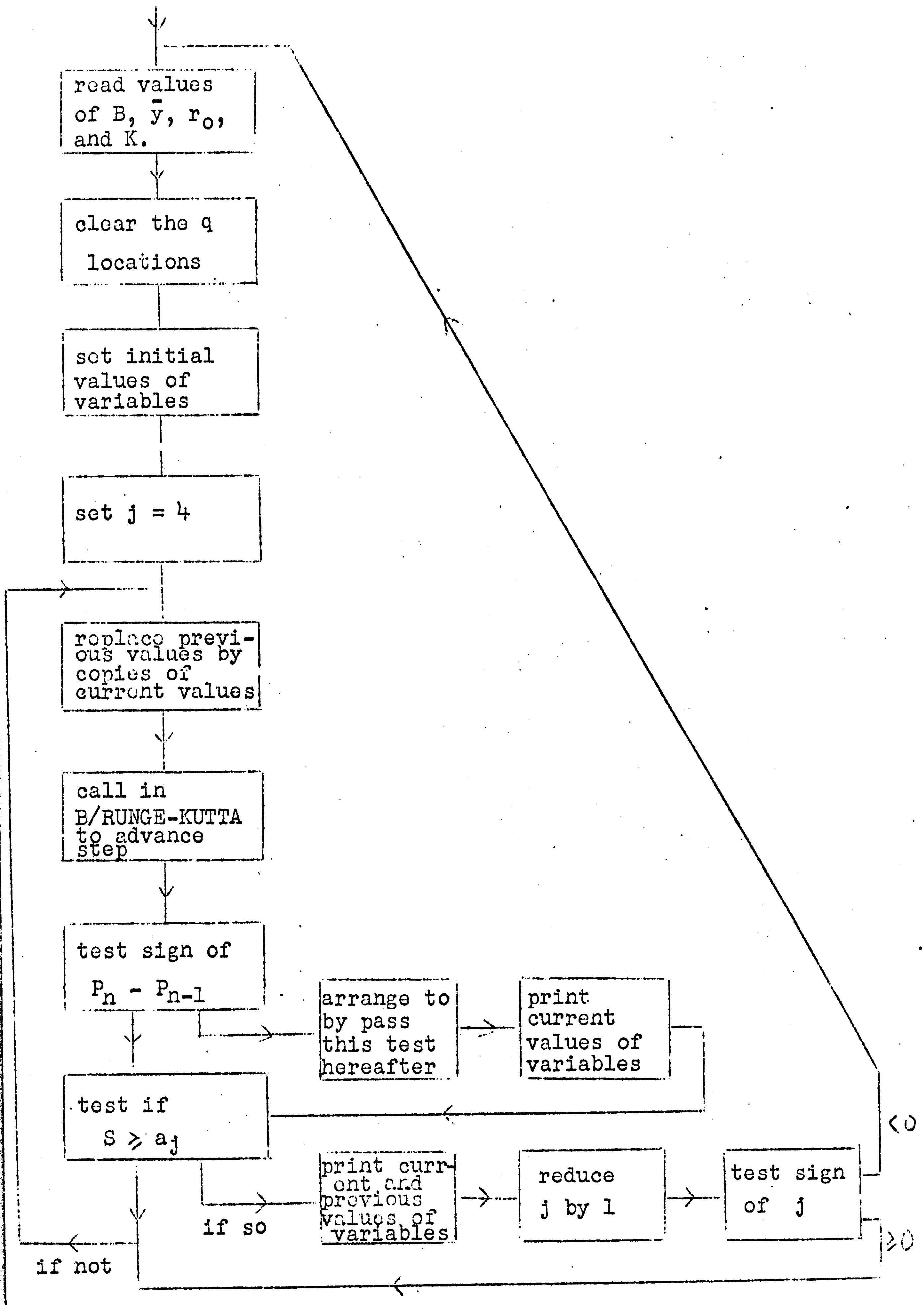
The instructions are given on p.4A.34. These are best followed with the aid of the flow diagram given below. The correspondence should be fairly clear with the exception of the arrangement whereby the test of the sign of $P_n - P_{n-1}$ is by passed once this becomes negative. This is achieved by making the actual test instruction, FE@H in line @E, subject to modification by B2. Initially B2 is cleared. When the maximum has been reached B2 is altered to ///@ (by instruction QELG in line RE). Thereafter the actual instruction corresponding to FE@H is FE@P - the unconditional transfer.

The printed information corresponding to the maximum of P is most conveniently distinguished by the fact that it is only necessary to print the values of the current variables - not the current and previous values. (If interpolation for the maximum is contemplated, then it would be necessary to print at least three sets of values).

The storage space and time requirements are almost identical with those of the programme prepared for the EDSAC.

	EDSAC	M.U.E.C/MK II
MASTER routine	72 orders	68 instructions
AUXILIARY routine	75 orders	69 instructions

In both cases the average time for the calculation of a set of solutions is about 10 mins.


read values of B, $\bar{y}$, r_o, and K.
clear the q locations
set initial values of variables
set j = 4
replace previous values by copies of current values
call in B/RUNGE-KUTTA to advance step
test sign of $P_n - P_{n-1}$
arrange to by pass this test hereafter
print current values of variables
test if $S \geqslant a_j$
if so
if not
print current and previous values of variables
reduce j by 1
test sign of j
<0
≥0

$$a_4 = \frac{1}{10} \cdot \frac{5}{9} \quad a_3 = \frac{1}{10} \cdot \frac{40}{49} \quad a_2 = \frac{1}{10} \cdot \frac{15}{16} \quad a_1 = \frac{1}{10} \cdot \frac{80}{81} \quad a_0 = \frac{1}{10}$$

The DIRECTORY.

An electronic cue DIRECTORY is used This forms part of a WORKING PERM and occupies lines RI to OI of S3 inclusive. (The electronic routine B/DIVISION/C, standing in column U, is the other material in S3). The following table summarises all the information required to form the cue DIRECTORY.

ROUTINE	No.	MAGNETIC STORAGE	ELECTRONIC STORAGE	TYPE	CUE
B.INPUT	//				GE// /A@/
MASTER	@/	4	S0 & S1	usual	££// :/@/
DEC.OUTPUT/B	:/	5	S0 & S1	usual	@/// S/@/
B.DEC.INPUT	I/	6	S0 & S 1	usual	//// I/@/
B/RUNGE-KUTTA	½/	7L	S0	usual	££// U/@/
B/DIVISION/C	R/	Part of WORKING PERM	column U	electronic	£I// //£/
SQUAREROOT/A	N/	7R	S1	electronic	E@// //£/
AUXILIARY	entry needed only for reading in initially				½/@/

Note: The magnetic half cue of B/RUNGE-KUTTA is chosen so that SQUAREROOT/A is also transferred to S1 for subsequent entry as an 'electronic' routine.

WORKING PERM

DIRECTORY and B/DIVISION/C

```
 I K R R K    K / J " U
            /|Q U / U
            E|I S / N
            @|I S / N
            A|E : / M
            :|D S T J
            S|A : / Q
            I|B U / J
            U|Q U T I
            ½|Q U / S
            D|B U / E
/|G E / /   R|J U / H
E|/ A @ /   J|S U T £
@|£ £ / /   N|I S T /
A|: / @ /   F|B U / K
:|@ / / /   C|Q U / J
S|S / @ /   K|Q U / D
I|/ / / /   T|Q U / A
U|I / @ /   Z|B U / E
½|£ £ / /   L|B U / E
D|U / @ /   W|E : / M
R|£ I / /   H|R U / P
J|/ / £ /   Y|Q U T /
N|E @ / /   P|N S / P
F|/ / £ /   Q|  a_n
C|½ / @ /   O|
            B|  c_n
            G|
            "|
            M|
            X|
            V|
            £|
```

test sign of divisor

change sign of divisor & dividend if former is -ve

shifting cycle: shifts divisor and dividend until divisor > ½. $[BU]_{+f} =$ b

$[BU]_{\pm f}$ = b-1 = c_o

repetitive cycle

$a_{n+1} + a_n (1 - c_n)$

$c_{n+1} = -c_n^2$

test for end of process

$[L']_{\pm f} = a/_b$

closure

working space

MASTER ROUTINE

```
        K / ½ /
  /  |  / | / : L O |  clear B2
     |  E | U : W O |  set B3 (2j=8)
     |  @ | @ / Q O |  call in B.DEC.INPUT:
     |  A | G S / P |  reads B, $\bar{y}$, $r_o$ & $1/\sqrt{K}$
     |  : | I / / V |
     |  S | I S T / |  set round-off
     |  I | / C / A |
     |  U | @ C / A |  clear the q
     |  ½ | : C / A |  locations
     |  D | I C / A |
     |  R | ½ C / A |  set $\gamma_o = 0$
     |  J | R C / A |  set $X_o = 0$
     |  N | : K / C |
     |  F | / K / N |
     |  C | / K / N |  form and plant $r_oB$
     |  K | V K / A |
     |  T | V K / C |
     |  Z | @ J / N |  form and plant
     |  L | @ J / J |  $\frac{1}{10} + \frac{r_oB}{10}$
     |  W | V K / A |
     |  H | : K / C |  form $r_o/_{100}$ in M
     |  Y | N J / N |
     |  P | C C / E |  $D'_{\pm f} = \frac{1}{10} + \frac{r_oB}{10}$
     |  Q | V K / C |
     |  O | O / Q O |  call in B/DIVISION/C:
     |  B | G S / P |
     |  G | R / / V |  $[L']_{\pm f} = \frac{r_o/100}{\frac{1}{10} + \frac{Br_o}{10}}$
     |  " | N C T A |
     |  M | @ E / P |
     |  X | X / Q O |
     |  V | G S / P |  call in B/RUNGE-KUTTA
     |  £ | ½ / / V |
```

```
        K / D /       test sign of $P_n - P_{n-1}$
  E  |  / | C C T ½ |  if -ve, then maximum
     |  E | P C T N |  has been reached
     |  @ | F E @ H |
     |  A | ½ E T / |  set block constant
     |  : | : E Q O |  call in DEC.OUTPUT/B:
     |  S | G S / P |  prints $\gamma_n$, $X_n$, $S_n$,
     |  I | : / / V |  and $P_n$
     |  U | A : / Q |
     |  ½ | : / / / |  block constant
     |  D | ½ C / / |
     |  R | Q E L G |  $[B2]' = ///@$
     |  J | N C T ½ |  compare $S_n$ with $\frac{1}{10}\cdot\frac{5}{9}$,
     |  N | @ J W N |  $\frac{1}{10}\cdot\frac{40}{49}$, $\frac{1}{10}\cdot\frac{15}{16}$, $\frac{1}{10}\cdot\frac{80}{81}$,
     |  F | R E / H |
     |  C | ½ C T / |  or $\frac{1}{10}$
     |  K | T C T A |  replace $\gamma_{n-1}$, $X_{n-1}$,
     |  T | R C T / |  $S_{n-1}$, & $P_{n-1}$, by
     |  Z | L C T A |  copies of $\gamma_n$, $X_n$,
     |  L | N C T / |
     |  W | H C T A |  $S_n$, and $P_n$.
     |  H | C C T / |
     |  Y | P C T A |
     |  P | A @ / P |  return to B/RUNGE-
     |  Q | / / / V |                KUTTA
     |  O | E @ T / |  print current and
     |  B | B E Q O |  previous values of
     |  G | G S / P |  $\gamma$, X, S, and P.
     |  " | : / / V |
     |  M | A : W G |  reduce value of 2j by
     |  X | @ E / T |  2; test for end of run
     |  V | M E / : |  line feeds to space
     |  £ | M E / : |  complete solutions

        K / R :
  @  |  / | D S / P |  return to read new
     |  E | ½ / / / |  set of parameters
     |  @ | ½ C / / |
     |  A | M / / / |
```

AUXILIARY SEQUENCE

```
½       K / ½ /
    /   ½ C T /    X'
    E   G C T A
    @   N J T ½    form $\frac{1}{100}$ - X in L
    A   R C T N
    :   A : / M    test if $X > \frac{1}{100}$
    S   / / T :
    I   S : / Q
    U   V K T A
    ½   V K / C    $[M']_f = -r_o\{\frac{1}{100} - X\}$
    D   : K / ½
    R   I S T I
    J   C C / J    P or $P-r_o\{\frac{1}{100} - X\}$ (=$\gamma'$)
    N   O C / A    plant $\gamma'$
    F   @ J T ½    $\frac{1}{10}$ - S
    C   N C T N
    K   E : / M
    T   N S / P    return to MASTER routine
    Z   V K T A
    L   V K / C    $[L']_f = 1-10S$
    W   / J / N
    H   H ½ Q O
    Y   G S / P    $[L']_f = \sqrt{(1-10S)}$
    P   N / / V
    Q   V K T A
    O   V K / C    form and plant
    B   I K / N    $\frac{1}{\sqrt{K}}\sqrt{1 - 10S}$
    G   I S T I
    "   V K / A
    M   V K / C    form and plant
    X   C C / N    $\frac{2P}{\sqrt{K}}\sqrt{1 - 10S} = \frac{1}{10}S'$
    V   V K / E    in VK
    £   V K / J
```

```
D       K / D /
    /   E : / J    round ff
    E   V K / A
    @   / / T :
    A   V K / C    S'
    :   / J / N
    S   M C T A
    I   N C / C
    U   / K / N    form and plant BS
    ½   / K / N
    D   I S T I
    R   G K / A
    J   G K / C
    N   @ J / ½    form and plant
    F   R C / J    $-\frac{BS}{10} + X + \frac{1}{100}$
    C   N J / J
    K   G K / A
    T   C C / C    form
    Z   / K / N    $BP + \frac{1}{100}$ in M
    L   / K / N
    W   N J / J
    H   V K / C    $\frac{1}{10} S'\{BP + \frac{1}{100}\}$
    Y   V K / A
    P   V K / N
    Q   M K / A
    O   C C / C
    B   @ K / N    $P\bar{y}$
    G   @ K / N
    "   V K / A
    M   V K / C    $-\gamma P\bar{y}$
    X   ½ C / ½
    V   M K / J    $-\gamma P\bar{y} + \frac{1}{10}S'(BP + \frac{1}{100})$
    £   G K / K    set divisor
```

The following constants are stored in column J.

```
        K / R S    call in B/DIVISION/C:
R   /   / R Q O    $[L']_{\pm f} = P'$
    E   G S / P
    @   R / / V
    A   V C T A
    :   / / / P    return to B/RUNGE-KUTTA
```

$[/\ J] = 10$

$[@\ J]_f = 0.1$

$[:\ J]_f = 0.1(80/81)$

$[I\ J]_f = 0.1(15/16)$

$[\frac{1}{2}\ J]_f = 0.1(40/49)$

$[R\ J]_f = 0.1(5/9)$

$[N\ J]_f = 0.01$

The constants are read into these locations by torminating the punching proper with the following sequences

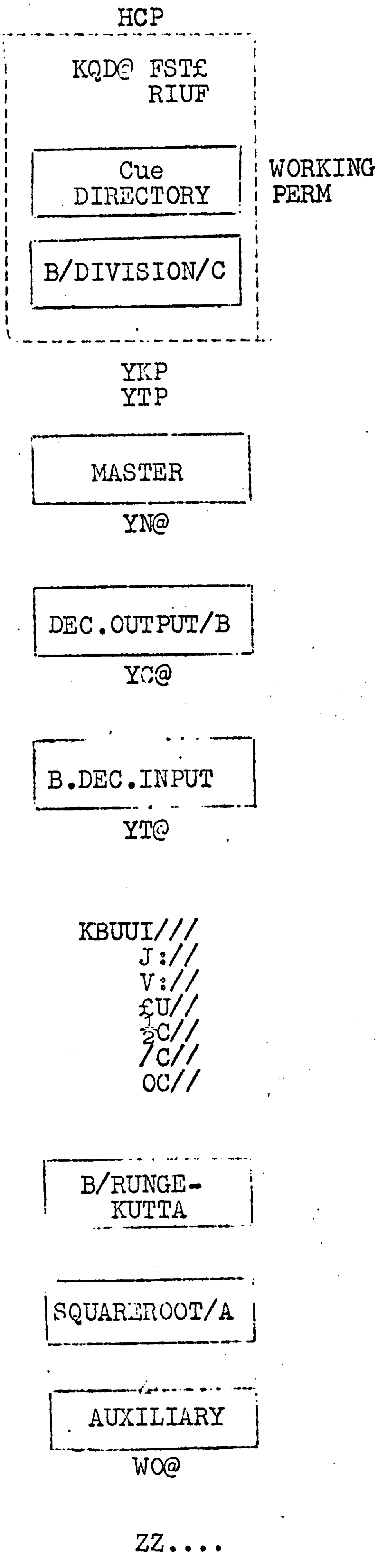
```
K/J@R///
    ////
 GI/               calls in B.DEC.INPUT
F@JI 1+
     09876543210+  these sequences are
     09375+        read by B.DECINPUT
     08163265306+
     05555555555+
     01+
     C             returns to B.INPUT
```

DISTRIBUTION of WORKING LOCATIONS in columns C and K.

	C		K	
	q_{ψ}	/ E	B/2	
	q_X	@ A	$\bar{y}/2$	
	q_s	: S	r_o	
	q_P	I U	$1/\sqrt{K}$	
current values of the variables	ψ_n	½ D		
	X_n	R J		
	S_n	N F		
	P_n	C		
previous values of the variables	ψ_{n-1}	T Z		
	X_{n-1}	L W		
	S_{n-1}	H Y		
	P_{n-1}	P Q		
	ψ'	O B		
	X'	G "		Miscellaneous working locations
	S'	M X		
	P'	V £		

Layout of Tapes.

MAIN TAPE



```
        HCP
 KQD@ FST£
      RIUF

   Cue
 DIRECTORY          WORKING
                    PERM
 B/DIVISION/C

        YKP
        YTP

   MASTER
        YN@

 DEC.OUTPUT/B
        YC@

 B.DEC.INPUT
        YT@

   KBUUI///
        J://
        V://
        £U//
        ½C//
        /C//
        OC//

   B/RUNGE-
    KUTTA

 SQUAREROOT/A

  AUXILIARY
        WO@

        ZZ....
```

NUMBER TAPE

```
        LO@  AUXILIARY to S4 & S5
        Q@/  enter MASTER routine
        F/K:
         +      (B)
        65+     (ȳ/2)
        6+      ( 9)
35355339059+    (1/ K)
        C
```

Chapter 10

OUTPUT ROUTINES

(by R.K.Livesley)

In preceeding chapters various references have been made to routines for printing or punching out information contained in the store. The purpose of the present chapter is to give a more systematic account of the library output routines available.

These routines fall naturally into three groups, the effects of which are as follows:-

Group 1

Numerical data or results are printed out in decimal form - either as integers, fractions, or with floating decimal point.

Group II

The contents of certain stores or store lines are printed out in teleprint form. The main use of such routines is in the checking of programmes for punching errors.

Group III

The contents of certain stores or store lines are punched out in a form suitable for use with one of the normal input routines. These routines are useful in producing programme tapes.

The following system of titling has been adopted:-

(1) Routines with the title PRINT give useful output on the printer only.

(2) Routines with the title PUNCH give useful output on the punch only.

(3) Routines with the title OUTPUT give useful output on both printer and punch.

(4) A prefix B or A indicates that the routine is only suitable for R.C.S/B or R.C.S/A respectively. Routines with no prefix are suitable for either R.C.S, although the correct entry points may be different.

(5) Separate routines in a given class are denoted by a letter suffix. In general the suffix only indicates the chronological order of composition.

Group I. Output of Numerical .results.

Various routines have already been issued to give certain types of numerical output. The routines FC:DECPRINT/A and B/DEC.PRINT/A are described elsewhere in the handbook (p.5.13A and p.4.8). A set of routines bearing the general title, DECOUTPUT, are described below. It is hoped that they will cover most printing requirements.

In connection with these output routines certain conventions have been introduced. They are as follows:-

(1) All operations connected with printing layout (i.e., space, carriage return, line feed) are produced by sending the appropriate figure shift character (Z, L, W) to M, and obeying special magnetic function T. This takes more space in the routine than merely obeying Z , L, W, as special magnetic functions, but has the advantage that if output is by punch, these characters are actually punched on the tape. Thus if the punched tape is subsequently read by an ordinary teleprinter (on figure shift), the layout originally produced is preserved.

This facility enables programmes in which printing takes a large proportion of the total time to be speeded up by using the punch output. The punch is about twice as fast as the printer, and the tape produced can be printed out afterwards.

(2) No stores (other than A and D) are altered by the routines.

(3) The accumulator is left clear at the end of the routine.

(4) Routines operate with both R.C.Sequences. As in the case function routines, the symbols x and y denote either [L] and [M] or [/ C] and [@ C] , according to which R.C.S. is used.

(5) All one page routines occupy S1.

(6) A suffix P indicates printing on + convention. Routines are otherwise assumed to print on the ± convention.

There are at present 6 standard routines of the above type. The first two, DECOUTPUT/A and B, are general purpose routines designed to print both integers and fractions. These routines contain devices for the suppression of irrelevant printing (such as the initial zeros before a small integer) and are therefore more suitable for cases where time is important than for layout. The other routines, DECOUTPUT/C - /F are designed for layout work, digit and page layout being provided.

In all routines the layout constants (i.e., no. of decimals, position of spaces, no. of columns) are preset parameters contained in the last few lines of the routine. The style of printing is specified on the box containing the library tape, and this may be altered by the programmer when copying the tape if required.

Routines for Printing Single Numbers.

These are all one-page routines.

Name of Routine. DECOUTPUT/A.

Cues R.C.S.A /@£BPEEZ.
R.C.S.B A@//...@.

Principal lines $[/ A]_0^{19}$ = EA/: Preset lines $[£ A]_0^{19}$. (see effects)

Purpose To print single numbers with both integral and fractional part.

Effects.

(1) The routine prints $x_{+f} + y_{\pm}$. That is, y is printed as an integer on ± convention, followed by x as a fraction on the + convention. It is assumed that $| y_{\pm} | < 10''$. If this is not so the integral printing will be incorrect.

(2) No carriage return or line feed instructions are included- these must be provided by tha programmer in the calling-in routine.

(3) Printing is as follows:-

(a) The integral part is printed with suppression of initial irrelevant zeros and preceeded by correct sign.

If y = 0, then '+ 0' is printed.

(b) The decimal point is printed.

(c) The fractional part is printed to n decimal places where $n - 1 = [£ A]_0^{19}$. If however the residue (i.e., the quantity remaining to be printed) is zero at any stage, printing ceases and the routine is left. Thus if $[£ A]_0^{19}{}_{\pm} \geqslant 39$, the fractional part is printed to its terminating decimal figure.

Notes

(1) The routine is useful in conjunction with DIVISION/B, the sign convention being the same in each case.

(2) To print numbers as fractions on the $\underline{+}$ convention they should be sent to the accumulator with a T/ instruction. To obtain printing on the ± convention a T$\frac{1}{2}$ instruction should be used.

(3) To print numbers as integers on the $\underline{+}$ convention, clear the accumulator and plant with a /J instruction. Printing will then cease after the decimal point.

Examples.

$\underline{x}_{+f}$	$\underline{y}_{\pm}$	Printed as
///////T,	££££££££	-0.5
///////T,	////////	+0.5
///////:,	S£££££££	-26.875
///////M,	G///////	+26.875
////////,	££££££££	-1.

The instructions of DECOUTPUT/A are given as an example of both integer and fraction printing cycles.

Name of Routine. DECOUTPUT/C.

Cues. R.C.S.A (1) /@VLPEEZ
(2) E@VLPEEZ

R.C.S.B (1) :@//...@
(2) S@//...@

Principal line. $[/\ A]_0^{19}$ = @@/N

Preset lines. $[M\ A]_0^{39}$, $[V\ A]_0^{19}$, $[£\ A]_0^{19}$. (See effects)

Purpose.

To print single long lines as integers with digit layout.

Effects.

(1) $x_{\pm}$ is printed as an integer, preceeded by correct sign and followed by decimal point. If entry is by cue (1), a carriage return and line feed occurs after printing. If by cue (2), two spaces occur after printing.

(2) n digits are printed, where $n \leqslant 12$. The following lines must be preset:-

$[V\ A]_0^{19}$ = n-1, $[M\ A]_0^{39}$ = 10^{n-1}. (Values are given in Fig. A.3)

If a setting is made for a value of $n < 12$, then a number greater in modulus than $10^{n}-1$ will not be printed correctly.

(3) Layout is determined by examining the line $[\pounds\ A]_0^{19}$ commencing at the most significant end. Only that part of the line containing the n most significant zeros is relevant. Each digit is examined in turn and the following action taken:-

(a) If digit is a 0 the next decimal digit is printed.

(b) If digit is a 1 the teleprinter is spaced.

Examples

$[V\ A]_0^{19}$	$[M\ A]_0^{39}$	$[\pounds\ A]_0^{19}$	Style of Printing
J///	//GFU:X@	T½:@	12 digits, 4 blocks of 3
S///	/YEA////	..:@	6 digits, 2 blocks of 3 (If $\lvert x\rvert < 10^{6}$)
I///	/LTV////	..T½	7 digits, 1 + 2 blocks of 3 (If $\lvert x\rvert < 10^{7}$)

Note. When using R.C.S.B. the number should be placed in L with a T ½ instruction.

Name of Routine. DECOUTPUT/D

Cue. R.C.S.A (1) /@@ZPEEZ
(2) E@@ZPEEZ

R.C.S.B (1) :@//...@
(2) S@//...@

Principal line $[/\ A]_0^{19}$ = V:/J Preset line $[\pounds\ A]_0^{19}$ (See effects)

Purpose.

To print single long lines as fractions with digit layout.

Effects

(1) $x_{\pm}$ is printed as a fraction, proceeded by correct sign and decimal point. If entry is by cue (1) a carriage return and line feed occur after printing. If by cue (2), two spaces occur after printing.

(2) Layout is determined by examining the line $[\pounds\ A]_0^{19}$ commencing at the most significant end, Each digit is examined in turn and the following action taken:-

(a) If digit is a 0 the next decimal digit is printed.

(b) If digit is a 1 the teleprinter is spaced.

(c) If two consecutive 1's occur together the routine is left.

(This layout scheme is the same as that used in B/DEC.PRINT/A).

Note. When using R.C.S.B the number should be placed in L using a T½ instruction.

Routines for Printing Sequences of Numbers.

These routines are all two-page routines. Numbers are printed in m columns, where m is a preset paramoter. The following properties are common to all these routines.

(1) The numbers to be printed are specified as follows:-

$[x]_0^{19}$ = number of successive long lines

$[x]_{20}^{39}$ = address of first line

(2) Each number is preceeded by correct sign, and the decimal point is printed.

(3) A carriage return and line feed occurs before and after printing. Hence, if the routine is entered repeatedly a double line feed separates the different blocks of printing.

Name of Routine. DECOUTPUT/B.

Cues.. R.C.S.B (1) ////QEEZ
(2) I///QEEZ

R.C.S.B (1) @///..@/
(2) ½///..@/

Preset lines $[V\ @]_0^{39}$, $[V\ A]_0^{19}$, $[\pounds\ A]_0^{19}$.

Purpose. To print sequences of Integers or Fractions.

Effects.

(1) The routine prints a sequence of numbers as fractions or integers, according as to whether entry is by cue (1) (fractions) or cue (2) (integers).

(2) If the content of a long line is zero, then in either case '+ 0'only is printed. When printing fractions this effect causes lines to appear offset and may therefore be undesirable. It may be suppressed by altering line $[D\ E]_0^{19}$ to A:T£.

(3) Printing is in m columns, where $m-1 = [\pounds\ A]_0^{19}$. Each column is separated by two spaces. (But see Note 4).

(4) Printing of Integers. Printing commences with the first non-zero decimal digit (but see note 2). This means that integers do not appear in columns unless they all have the same number of decimal digits.

(5) Printing of Fractions. Each number is printed to n decimal places, where $n-1 = [V\ A]_0^{19}$. If round-off is desired, the long line $[V\ @]_0^{39}$ should be set to 5×10^{-n} - otherwise it should be cleared.

The following table gives the maximum number of columns which may be allowed if the printer is not to exceed the capacity.

No. of decimal places.	No. of columns	Round-off
		$[V\ @]_0^{39}$
n	m	
2	11	FIWBNIWE
3	9	"ASMGAS/
4	8	IXIDNT//
5	7	V@QFHE//
6	6	PFOUS///
7	6	M"OT////
8	5	/PYE////
9	5	GJS/////
10	4	IZ//////
11	4	QE//////
12	4	I///////

Name of Routine. DECOUTPUT/E.

Cue. R.C.S.A. //ANQEEZ

R.C.S.B. @///..@/

Preset Lines. $[\text{"}\ A]_0^{19}$, $[M\ A]_0^{39}$, $[V\ A]_0^{19}$, $[\pounds\ A]_0^{19}$.

Purpose. To print sequences of long lines as integers, with digit layout.

Effects.

(1) This routine is the sequence version of DECOUTPUT/C. The purpose of the lines MA - £A is the same as in that routine, and the layout scheme is also the same.

(2) The number of columns n is determined by the contents of $[\text{"}\ A]_0^{19}$. This line should be set so that $[\text{"}\ A]_0^{19} = m-1$. Care should be taken not to exceed the capacity of the printer.

Name of Routine. DECOUTPUT/F.

Cues. R.C.S.A. //ANQEEZ

R.C.S.B. @///..@/

Preset lines $\left[\text{" A}\right]_0^{19}$, $\left[\text{£ A}\right]_0^{19}$

Purpose. To print sequences of long lines as fractions with digit layout.

Effects.

(1) This routine is the sequence version of DECOUTPUT/D. The layout line $\left[\text{£ A}\right]_0^{19}$ has exactly the same properties as in that routine.

(2) The number of columns m is given by $m-1 = \left[\text{" A}\right]_0^{19}$ as in DECOUTPUT/E.

Routines for printing on the Plus Convention.

It is occasionally desirable to print numbers as fractions or integers on the + convention. Modified versions of the routines described above exist for this purpose. Their properties are in all cases similar to the appropriate ± routines, and all layout constants, etc., have the same effect. The cues are in all cases the same. The routines are :-

DECOUTPUT/BP
DECOUTPUT/CP
DECOUTPUT/DP
DECOUTPUT/EP
DECOUTPUT/FP.

The following points should be noted:-

(1) Routines for printing integers (i.e., DECOUTPUT/BP, CP, EP) will only print correctly integers less than 10^{12}. Numbers in the range $10^{12} < x < 2^{40}$ will be printed with a space in front of the last 11 digits instead of the correct printing '10'.

(2) Routines for printing single integers and fractions i.e., DECOUTPUT/C & D, may be used, with R.C.S.B, to print on the + convention. It is merely necessary to send the number to L with a T/ instruction. If however the + version is used, printing will be on the + convention irrespective of whether the instruction is T/ or $T\frac{1}{2}$.

Group II Printing in Teleprint Form.

There are several routines used for printing out instructions in teleprint form. Two such routines for printing out the contents of a single store are PAGEPRINT/A and the routine generally known as ENGINEER'S OUTPUT. A description of the former has been issued. The latter routine has the advantage of being kept permanently in the magnetic store.

It is normally used manually as follows:-

(1) The required page is brought to S4

(2) The routine is brought to SO from isolated track 15R.
(Set H = K/E/).

(3) The /L stop is switched on.

(4) KAC, KBC, KMC, KCC are operated.

(5) Completion signals are switched on.

Printing is in 8 lines, each line containing 8 short lines printed in teleprint form and each separated by a space. The routine stops on a /L instruction. A new page may now be brought down to S4 and printed out by operating KAC, KBC, KMC, KCC, and switching on completion signals again.

Group III

Output of Punched material for use with
one of the existing Input Routines.

Various routines have been issued from time to time for the output of routines in a form suitable for use with an input routine. In particular the routines Output C and Output D will punch out the contents of S4 for use with A/INPUT. The reader is referred to the appropriate specification sheets for descriptions of these routines.

The routine B/PUNCH/A, (for use with scheme B only), is now available on a wired-off track. The details are as follows:-

Name of Routine B/PUNCH/A. (Author A.E.G.).

Cue E///EAE/.

Purpose To make punched tape copies of routines, or other material, stored in the magnetic store, in the (teleprint) form suitable for re-input with B.INPUT.

Magnetic storage. Isolated on 106R (19.10.52).

Electronic storage. S0.   Stores altered. S1.

Effects.

If the DIRECTORY is standing in S6, S7 when B/PUNCH is called in, the effect is first to scan the input tape in search of a meaningful sequence of the form x a b, where x is any negative (i.e., $\geqslant 16$) character. For each such sequence encountered the subsequent effect is to punch the "punching proper" of the routine corresponding to the destination sequence Y a b, which is also punched immediately after the punching proper. Finally B/PUNCH returns to scan the input tape for another sequence. The routine has no exit and this process continues indefinitely unless interfered with manually.

For one page routines the punching takes the form

```
K/½/  (1st column)
K/D/  (2nd column)
Y ab
```

for S0 (or even page) routines, and

```
K/R/  (1st column)
K/J/  (2nd column)
Y ab
```

for S1 (or odd page) routines.

For two page routines it takes the form

```
K/½/  (1st column)
K/D/  (2nd column)
K/R/  (3rd column)
K/J/  (4th column)
Yab
```

B/PUNCH does not treat (correctly) routines for which the magnetic half cuo o entry specifies, in digits 15 and 16; columns other than the first, or which specify "criss-cross" transfers e.g., ..@E, ..@@, ../E, ../A, or ..A/.

To punch out a set of routines with B.PUNCH it is only necessary to prepare a small input tape bearing the corresponding destination sequences. Thus, e.g., to punch a copy of the programme tape for Example I (see p.4A.1) (assuming that the complete programme and DIRECTORY is already in the magnetic store) the following steering tape is prepared for reading with B.INPUT in the usual way.

```
HTP              } inserts cue to B.INPUT in line pair
KVD@E///EAE/     } VE of DIRECTORY
YTP                calls in B.PUNCH
QVE
YTP
Y@/
Y:/                destination sequences of
YI/                required routines and
Y½/                other material (see p.4A.4)
YR/
YN/
```

N.B. Since B.PUNCH has no exit it will continue to read tape until the machine is stopped manually. Operation of the "run-out" button on the punch will cause the tape to be extended by copies of the last character punched - the 3rd character of the last destination sequence. This is usually / or E and hence will be ignored by B.INPUT. To complete the programme tape Z's can be inserted manually.

DECOUTPUT/A.

```
@ | A / H / | /
  | V S T A | E    Entry for R.C.S/A.
  | / C T / | @  } Load accumulator
  | @ C / J | A  }
  | / @ / : | :    Entry for R.C.S/B Figs.
  | G @ / H | S    Test Accumulator
  | D S T J | I
  | E : T I | U    Reverse sign if -ve.
  | @ @ / U | ½    print -, planting
  | N A / J | D    |x±| in @@
  | E A / : | R
  | / @ / Q | J
  | @ @ / U | N  } plant x+ in @@
  | O A / J | F    and print +
  | E A / : | C    plant y± in :@
  | : @ / U | K
  | M A / C | T    Set D+ = 10
  | / : Q O | Z    Set B7 = 0
  | : @ T F | L  } Test for y ≡ 0
  | R A / H | W  } If so, print 0
  | M A Q O | H    Set B7 = 10
  | V : T F | Y    A = ///////£,££££££££
  | : @ / J | P    A = ///////£, y - 1
  | G A / J | Q    Subtract 10^10 from M &
  | V : T I | O    add 2^35 to L. Eventually
  | C A / H | B    L contains 2^35 x dec. digit.
  | J @ / I | G    Reverse accumulator
  | G A T N | "    add 10^10 to L. L now contains residue.
  | I S / ½ | M  } Test whether dec.digit = 0
  | E : / M | X  } If so & no digit previously printed
  | / @ / Q | V  } skip to @A.
  | S : / J | £    restores M to original value
A | E A / : | /    prints decimal digit
  | / @ T B | E     Blots out transfer in line V@.
  | : @ T A | @  }
  | : @ / N | A  } Multiply residue by 10
  | : @ T A | :  }
  | E : Q G | S  } Test multiplying counter. Note
  | H A / T | I  } that B7 = 0 if y = 0
  | Z @ / J | U  } Print Decimal Point.
  | E A / : | ½  }
```

```
A | £ A Q O | D    Set no.of dec.places
  | £ @ T : | R    Clear A
  | M A / C | J    Set D+= 10
  | @ @ / ½ | N  } Test if x ≡ 0.  If
  | M A / M | F  } so leave routine.
  | P @ T : | C    Clear A
  | @ @ / N | K    A = 10 x
  | I @ / A | T    Plant Integral part.
  | @ @ T A | Z    Plant residue
  | V : / C | L  } Multiply Integral
  | I @ / N | W  } part by 2^35
  | H @ / I | H    Reverse A.
  | E A / : | Y    Print digit
  | E : Q G | P  } Test for no. of
  | O A / T | Q  } decimal places
  | D A T : | O    Clear A
  | N S / P | B     Leave routine
  | / / U ½ | G  }
  | £ Y P £ | "  }   - 10^10
  | R / / / | M  }
  | / / / / | X  }   10
  | / / / / | V
  | ½ / / / | £    No.of decimal places
                   -1.
```