FERRANTI
PEGASUS COMPUTER

LIBRARY PROGRAMMES

VOLUME I

This document is a facsimile of the original book, transcribed by Christopher P Burton of the Computer Conservation Society in 2003 by the following method:

- Each page scanned at 200 dpi using Textbridge yielding 1-bit/pixel .tif files.
- Each image was then cropped by eye to have almost no white margins.
- Pages in the original (foolscap paper) which had text longer than A4 were cut and pasted to squeeze on to A4 size.
- Files were then saved as .gif image files.
- Word for Windows was then used to assemble the document, inserting one .gif image per page, with one inch left margin, 0.2 inch top margin, 0 right margin, 0.1 bottom margin on A4 paper. The images were ranged top left against those margins. It was necessary to fractionally reduce the size of each image to be slightly less than 11.38 inches high, rather than allow automatic fitting by Word.
- The document was saved and then output to an Apple Laserwriter II NTX but output to file, not actually printed. Requests to fix margins were not over-ruled. This created a PostScript file of the document, about 250 MB long.
- The PostScript file was then input to Frank Siegert's PStill program which converts to PDF to yield this document.

FERRANTI LTD

INTRODUCTION TO PEGASUS LIBRARY PROGRAMMES

Issue 1
4.58.

1. Uses of Library Programmes

When writing and developing programmes it is sometimes desirable to know rather more about some library subroutine than is available in the specification. Occasionally it may be necessary to modify or re-write a subroutine. It is for these purposes that annotated programmes are provided; they should be regarded as supplementary to the specifications.

Binary tapes of subroutines, punched in a form suitable for use with Assembly, are available from the London Computer Centre. Users are strongly recommended to take these tested tapes rather than to punch for themselves the routines they require.

A letter P is printed in the Library Index (starting with Issue 6) to distinguish subroutines whose programmes have been issued.

2. Layout

Each programme is preceded, where necessary, by a description of the process used. This description is omitted if the process is adequately described in the specification or if it is clear from the nature of the routine.

Where the process requires a large amount of organisation, as in some print routines, the programme is preceded by a flow diagram. Occasionally a comprehensive flow diagram is provided and this, together with a print out of the programme tape, takes the place of the annotated programme sheets.

There are usually two blocks of programme on each page and descriptive titles are printed in heavy type where appropriate. Programmes are printed on only one side of each sheet so that pages may be taken out and laid side by side. Each subroutine is preceded by its cue list and various other Assembly tags. The blocks of the routine itself are often arranged in the order in which they are obeyed, not the numerical order required on the tape.

3. Notation

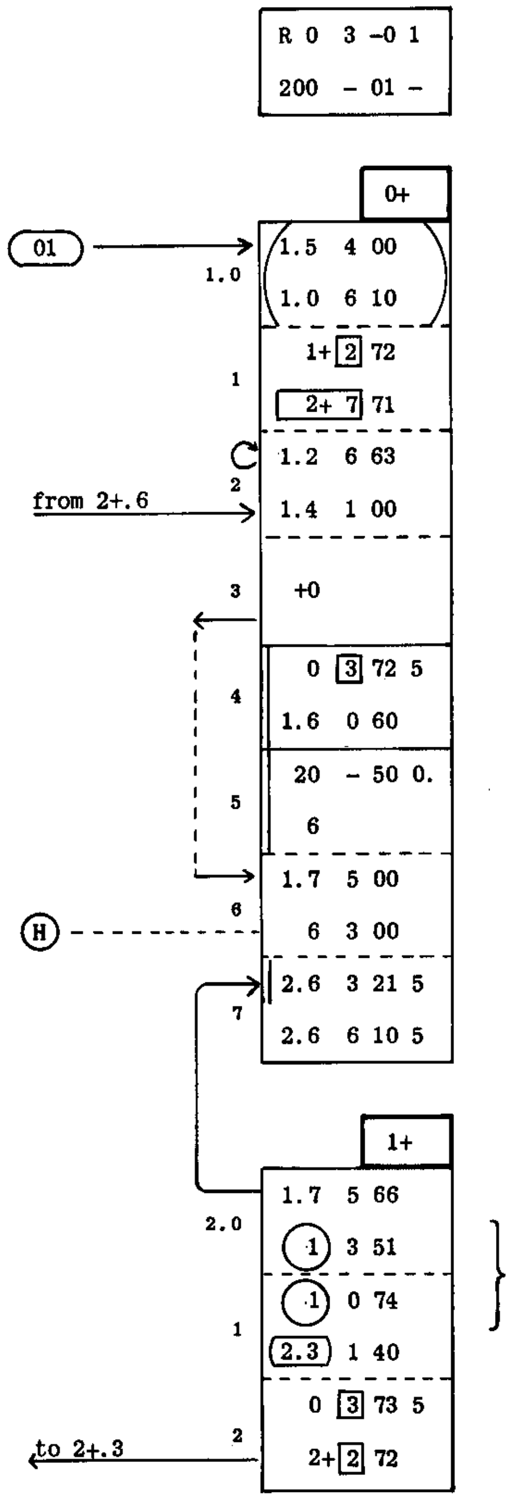
The library programmes are intended to be read in conjunction with the corresponding specifications; whenever possible the notation introduced in the specification is used in the programme.

Most of the conventions used in annotating library programmes are introduced in the Pegasus Programming Manual. Lists of the more important symbols and abbreviations are given below:

B 2+	Block 2+ in the Main Store
B.P	Block and Position Address
C(2)	The content of Accumulator 2
Er	Tape Character Erase
I.O.	The Initial Orders
l.s.	Least significant (or less significant)

m.s.	Most significant (or more significant)
OVR	The Overflow Indicator
P	Accumulator 6 (P for Product)
p	The content of Accumulator 6
P.P.	Preset Parameter
PQ	The double-length Accumulator, 6 and 7
(pq)	The double-length number in PQ.
Q	Accumulator 7 (Q for Quotient)
q	The content of Accumulator 7
r.o.c.	Round-off constant
Sp sp	Tape Character Space
U 2.1	Register 2.1 in the Computing Store. The U is often omitted.
W 2	Block 2 in the magnetic tape buffer
X 2	Accumulator 2. The X is usually omitted.
x_2	The content of Accumulator 2
α	The α-search is the section (or setting) of an input routine which ignores blank tape
β	The β-search is the section (or setting) of an input routine which searches for the start of a number or a letter shift
γ	The γ-search is the section (or setting) of an input routine which searches for a warning character or directive.
ϕ	Tape Character Figure Shift (≡ Blank Tape)
λ	Tape Character Letter Shift
$\odot$ •	Tape Character Full Stop (≡ Decimal Point)
3_c or 3_C	The counter in Accumulator 3
3_m or 3_M	The modifier in Accumulator 3
3_B	The block part of the modifier
3_P	The position part of the modifier
(4.3, 20)	A modifier of 4.3 and a counter of 20.

4. Symbols on Programme Sheets



Main Store Block Number

Oval indicates cue 01

Brackets show that the order pair is overwritten

Box round U block number

Box round single word address

Arrow marks a loop stop

Arrow indicates a jump to this order

+ cue — Solid line under the cue indicating an unconditional jump

Vertical line marks a LINK

Solid line under an unconditional jump

Vertical line marks a pseudo order-pair

Arrow indicates return from subroutine

Letter in circle refers to flow chart

Vertical line (may be a broken line) marks an order which is used as a number

Line and arrow mark a jump

Rings indicate that the *N* part of the order is a number, not an address

Ring indicates that the address 2.3 is set in 1_C, not C(2.3)

Solid line under an unconditional transfer of control to B2+.3

GENERAL PURPOSE NUMBER PRINT

A self-preserving subroutine for printing scaled numbers in various styles. The style is controlled by two parameters. The first of these, known as the Indicator I, controls the digit layout; the second, k, is used as a scaling factor. Both are described in the Specification, but further details of I are given below to explain the operation of the programme.

1. The Layout-Parameter (see Specification, Section 6)

The binary digits of the layout-parameter are allocated as follows

I =

	N	X	S	$(p_0\ q_0)$
i_0	i_1 0 i_3 i_4 i_5 i_6 i_7	X	S	p_{00} p_{01} p_{02} q_{00} q_{01} q_{02}
	0 p_{10} p_{11} p_{12} q_{10} q_{11} q_{12}	p_{20} p_{21} p_{22} q_{20} q_{21} q_{22}		p_{30} p_{31} p_{32} q_{30} q_{31} q_{32}
	$(p_1\ q_1)$	$(p_2\ q_2)$		$(p_3\ q_3)$

2. Group Counters

For the purpose of printing, the decimal digits of each number are treated in four groups. Each of these has a 3 bit counter p_j and a 3 bit indicator q_j.

The four groups are taken in order. If $p_j \neq 0$ the routine will print (or suppress or omit) the next p_j $(1 \leqslant p_j \leqslant 7)$ decimal digits, as specified by various indicators.

$p_j = 0$ indicates the end of the number and, in this case only, the octal number given by the group indicator q_j gives the number of spaces which are to be punched after the number. When these final spaces have been punched, the LINK is obeyed.

3. Group Indicators

q_{j0} = 0 Digits not printed will be suppressed. (i.e. replaced by Spaces)

q_{j0} = 1 Digits not printed will be omitted altogether.

q_{j1} and q_{j2} are used to indicate the action to be taken at the end of group j.

$q_{j1}\,q_{j2}$ = 00 Initiate 'special operation', see Section 6, then proceed to next group.

$q_{j1}\,q_{j2}$ = 01 Punch two spaces, then obey LINK.

$q_{j1}\,q_{j2}$ = 10 Punch one space, then look at next group indicator.

$q_{j1}\,q_{j2}$ = 11 Look at next group indicator immediately.

4. Initial Printing

i_0 = 0 gives CRLF before the number
i_0 = 1 has no effect

i_1 = 0 sign is punched immediately after the initial space.
i_1 = 1 has no effect. This is used for unsigned numbers, or for numbers printed with delayed sign.

S Immediately following the CRLF (if any) $7-S$ spaces are punched before anything else.

X All numbers are punched ultimately as integers in the range $0 \leqslant z < 2^{38}$. If m is the maximum number of decimal digits to be printed, where $m \leqslant 12$, then $X = 12 - m$ $(0 \leqslant X \leqslant 7)$ i.e. X decimal digits are ignored completely. The integer to be punched is multiplied by 10^X before entering the printing sequence so that the first X significant digits are ignored.

5. Zero Suppression

There are two indicators, σ and μ, set during the subroutine, which control the suppression, omission and punching of digits.

$\sigma = 0$ causes the digit to be punched.
$\sigma = 1$ causes the digit to be suppressed or omitted.

$\mu = 0$ there is no R.H.Z. suppression or omission.
$\mu = 1$ the remainder, r, is examined after punching each digit; if it is zero, puts $\sigma = 1$.

$i_3 i_4 = 00$ Start with $\sigma = 1$, $\mu = 0$. L.H.Z. suppression or omission. Set $\sigma = 0$ and print as soon as first significant digit reached.

$i_3 i_4 = 01$ As above, but print Sp or – before first significant digit.

$i_3 i_4 = 10$ Start with $\sigma = 0$, $\mu = 0$. Print all digits.

$i_3 i_4 = 11$ Start with $\sigma = 1$, $\mu = 0$. L.H.Z. suppression or omission terminated by the 'special operation' group (i_5, i_6, i_7). See below.

6. Special Operations

Special operations are initiated at the end of a group if the group indicators q_{j1} and q_{j2} are both zero.

i_5 Only consulted if $i_3 = 0$.

$i_3\ i_5 = 00$ Treat the last digit of the group as a significant digit, even if it is zero.

$i_3\ i_5 = 01$ No special treatment for last digit.

$i_6\ i_7 = 00$ Set $\sigma = 0$, $\mu = 0$ to stop further zero suppression (or omission).

$i_6\ i_7 = 01$ Punch decimal point. Set $\sigma = 0$, $\mu = 1$ to stop L.H.Z. suppression but allow for R.H.Z. suppression later.

$i_6\ i_7 = 10$ Punch decimal point. Set $\sigma = 0$, $\mu = 0$ to stop further zero suppression.

$i_6\ i_7 = 11$ If remainder $r = 0$, punch Sp. Set $\sigma = 1$, $\mu = 1$ for immediate R.H.Z. suppression.
If remainder $r \neq 0$, punch decimal point. Set $\sigma = 0$ and $\mu = 1$ to allow for R.H.Z. suppression later.

NOTATION

3_f	five least significant binary digits of accumulator 3.
5_s	sign bit of accumulator 5.
R.H.Z.	right hand zeros
L.H.Z.	left hand zeros
σ	Controls zero suppression, see section 5.
μ	Controls R.H.Z. suppression, see section 5.
i_j	Layout parameter, see sections 1, 5, 6.
p_j	Group counter, see section 2.
q_{jk}	Group indicator, see section 3.
s	Style number.
k_s	Scaling constant.
I_s	Layout-parameter, see section 1.
z	the number to be scaled and punched.
ζ	$\lvert\zeta\rvert = \lvert kz\rvert$ where k is the scaling factor.
δ	Next decimal digit to be punched where $\delta = [\zeta.10^I/10^{11}]$
r	Remainder after forming δ; replaces ζ to form next decimal digit.

PRINTING CYCLE

Contents of Accumulators on entry to Printing Cycle

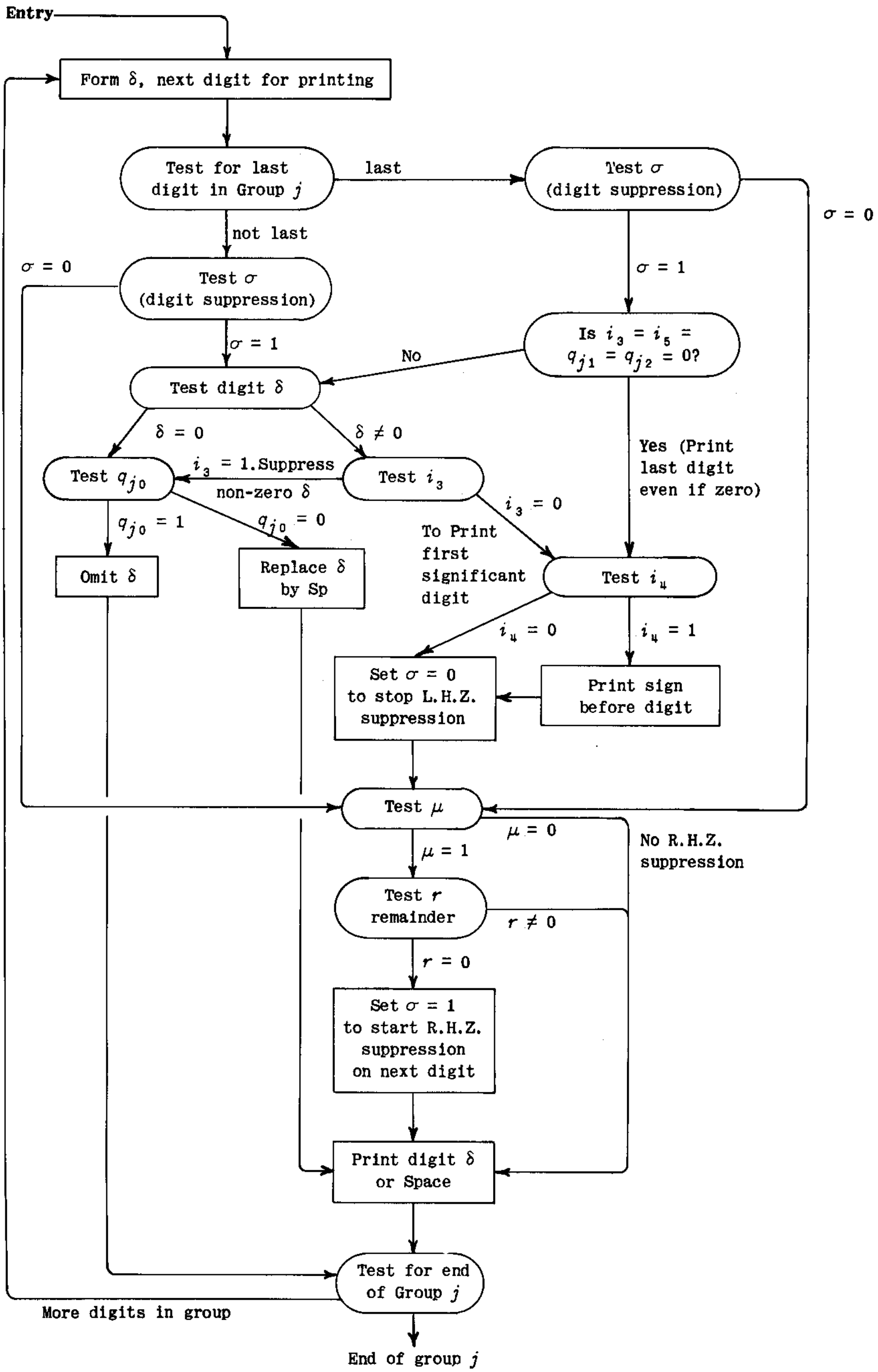
1st Entry

x_1	$1_m = p_1\ q_1$
x_2	$i_3 + (q_{11}\ q_{12}).2^{-13} + i_5.2^{-38}$
x_3	$i_4.000000i_7 00000i_6 00\ldots0$ [Space / ····· / or –]
x_4	$4_m = -0.1$; $4_c = X$; 4_s becomes $\mu = 0$ after 66 order in 5+.4+
x_5	Sign bit $= \sigma$; $5_m = q_j$; $5_c = p_j$
x_6	$\lvert\zeta\rvert$ where $\lvert\zeta\rvert = \lvert kz\rvert$
x_7	Overwritten

Subsequent Entries

x_1	$1_m = p_j\ q_j$
x_2	$i_3 + (q_{j1}\ q_{j2}).2^{-13} + i_5.2^{-38}$
x_3	$i_4.000000i_7 00000i_6 00\ldots0$ [Space / ····· / or –]
x_4	Sign bit $= \mu$ $4_m = 4_c = 0$
x_5	Sign bit $= \sigma$; $5_m = q_j$; $5_c = p_j - N$
x_6	Remainder r
x_7	Overwritten

FLOW CHART OF PRINTING CYCLE



	R 0 0 -0 3 1 - 28 -	Title of cue list
01	0+ [0] 72 0.2+ 0 60	Normal cue
02	0+ [0] 72 0.3+ 0 60	Cue used only by R3
03	0 10+ 6 00 0.	Address -0.2 of Parameter List *

LN
φ NUMBER PRINT

R 0 5 -0 3 1 - 01 -	Call for address of Parameter List
R 4 6 -0 2 3 - 01 -	Call for cue 02 to R3 or R03 †
R 8 3 -0 3 1 - 01 -	Call for address of Parameter List
R11 0 -2 0 1 - 06 -	Title of Optional Parameter List

* Note that if the optional parameter list is not used, the Interlude will alter cue 03 to be the address -0.2 of the programmer's parameter list.

† When the Master Programme calls for R3 this tag will cause the order in 4+.6+ of R1 to restore B0+ of R3 to U0. R3 uses R1 as a subroutine and this is the means of returning to R3. When the Master Programme does not call for R3, this tag will cause the dummy subroutine R03 to be accepted. Cue 02 to R03 is such that the order in 4+.6 of R1 will then restore B0+ of R1 to U0.

(0+)

Address	Order	Notes
0.0	1 0 00	Mask
	0.1 6 10	
		ENTRY
(01) →	0 [7] 73	Store Accumulators
0.3	1 6 00	Number to print, z to 6
(02) → (Used only by R3)	(26) 7 52	$2s$ to 7_m, s = style number
0.4	2+ [1] 73	Store U1
	10+ [2] 73	Store U2
0.5	0.0 3 00	Mask (= 1.000000100....0) to 3
	[0 1] 70 7	+ cue 03 to R1. Read $-k_s$ to 1
0.6	1+ [1] 72	
	7+ [2] 72	
0.7	8+ [0] 72	
	0.3 0 60	

↓

(8+)

Address	Order	Notes
0.3	1 5 00	+ cue 03 } $-k_s$ to 5
	[0 0] 70 7	+ cue 03 } I_s to 1
0.4	1 4 00	I_s to 4
	4 3 05	i_0.000000$i_7$000..... to 3
0.5	(11) 3 41	3_f = 11 to print –
	0.7 6 63	Jump if z < 0 (to 0.7)
0.6	6 6 02	Negate z if $z \geqslant 0$ (i.e. $-\lvert z\rvert$ to 6)
	(3) 3 41	3_f = 14 to print Sp
0.7	6 5 21	$\zeta = \lvert kz\rvert$ (rounded). If ζ = +1.0, 6 = −1.0 and OVR
	(30) 2 40	2_c = +30 for printing CR

↓

(1+)

Address	Order	Notes
1.0	(19) 7 40	7_c = +19
	1.3 4 63	Jump if i_0 = 1 (i.e. no CR LF) (to 1.3)
1.1	16 2 10	Print CR
	16 7 12	Print LF
1.2	17 0 10	Print ϕ
	32 3 01	1.000000$i_7$000....
1.3	1.4 7 67	7_c = +18. Unconditional jump (to 1.4)
	16 7 12	Print (7 – S) Spaces.
1.4	1.3+ 1 66	(Print (7 – S) Spaces, continued: loops back to 1.3+)
	(1) 4 52	i_1. 0 i_3 i_4 i_5 i_6 i_7 etc. to 4
1.5	1.6 4 63	Jump if i_1 = 1 (sign delayed) (to 1.6)
	16 3 10	Print Sp if z positive, – if z negative.
1.6	(2) 4 52	i_3.i_4 i_5 i_6 i_7 etc. to 4
	32 2 00	i_3.000000....0 to 2
1.7	4 2 05	(i_3.000000....0 to 2, continued)
	32 5 00	−1.0 to 5

→ to 7+.0 (sheet 8)

from 1+.7+ (Sheet 7) → 2.0

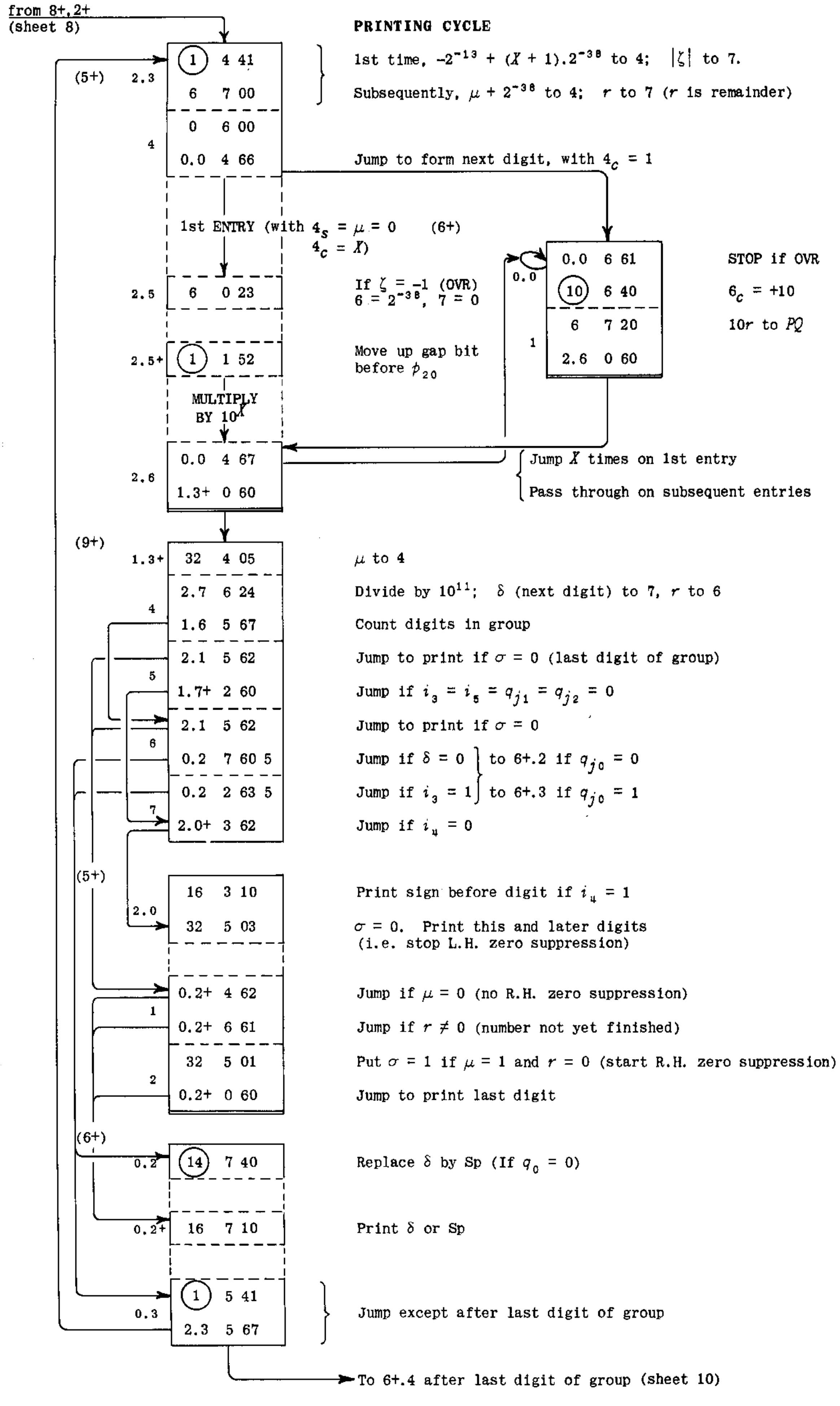
	Label	Instruction	Comment
	2.0	(1) 4 52	$i_4.i_5\ i_6\ i_7$ etc. to 4
		2.2 4 63	Jump if $i_4 = 1$
(7+)	1	2 5 06	$\sigma = 0$ in 5_s if $i_3 = 1$ and $i_4 = 0$; $\sigma = 1$ otherwise
		32 3 03	$i_4 = 0$; $i_4.000000i_7000......$ to 3
	2	(1) 4 52	$i_5.i_6\ i_7$ etc. to 4
		2.3+ 4 62	Jump if $i_5 = 0$
	3	(1) 2 41	$i_3.000000......000i_5$ to 2
		(1) 4 52	$i_6.i_7$ etc. to 4
	4	2.5 4 62	Jump if $i_6 = 0$
		35 3 01	$i_4.000000i_7 00000i_6 0.....$ to 3
	5	(34) 4 53	$0.000000....i_6\ i_7$ ←X→ to 4
		(7) 4 45	0.000000....0 0 ←X→ to 4
	6	35 4 03	1.111111111111100....0 ←X→ to 4
		6+ [0] 72	
	7	3+ [1] 72	
		1.0 0 60	

NEXT GROUP

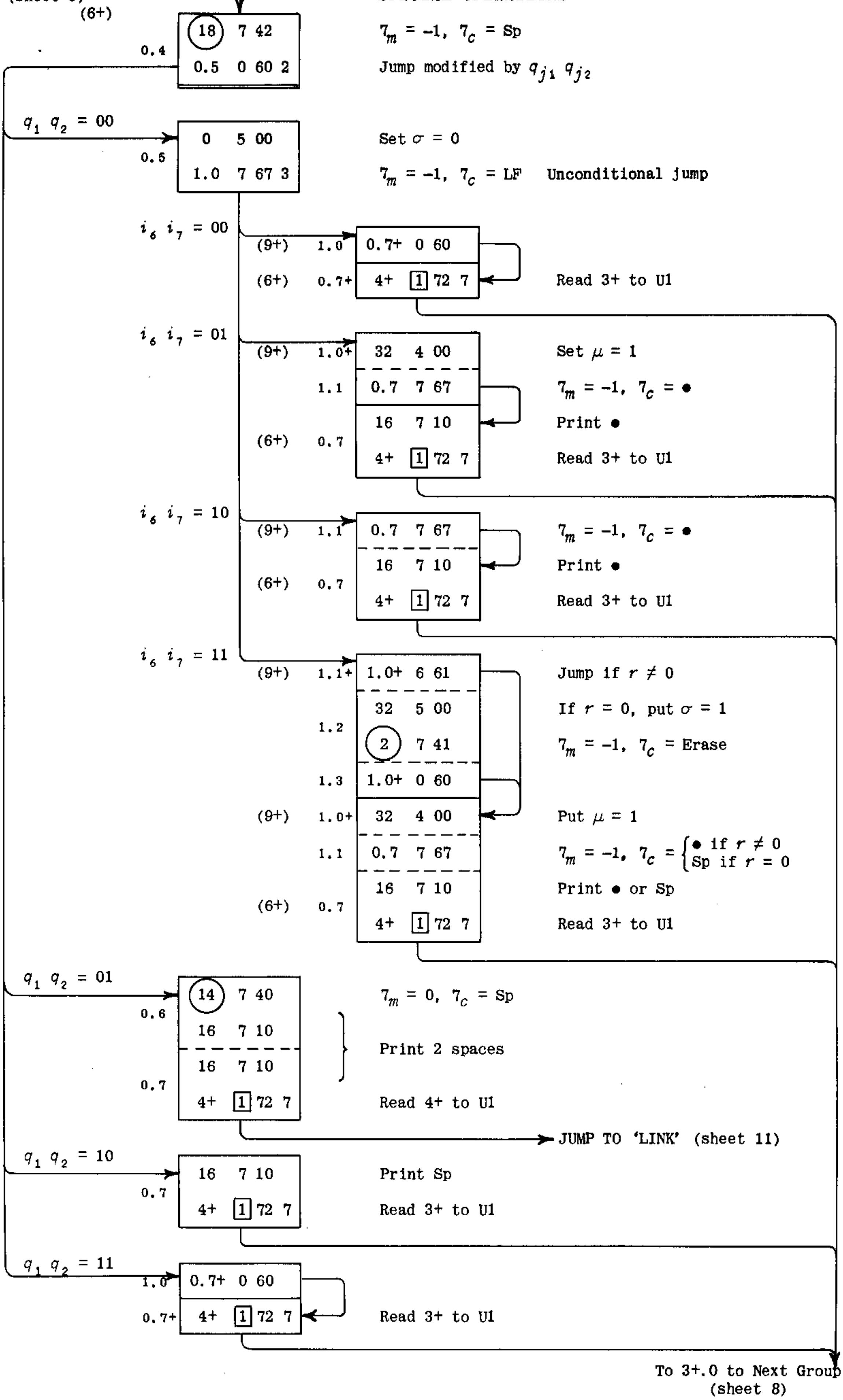
from 6+.7+ (sheet 10) → 1.0

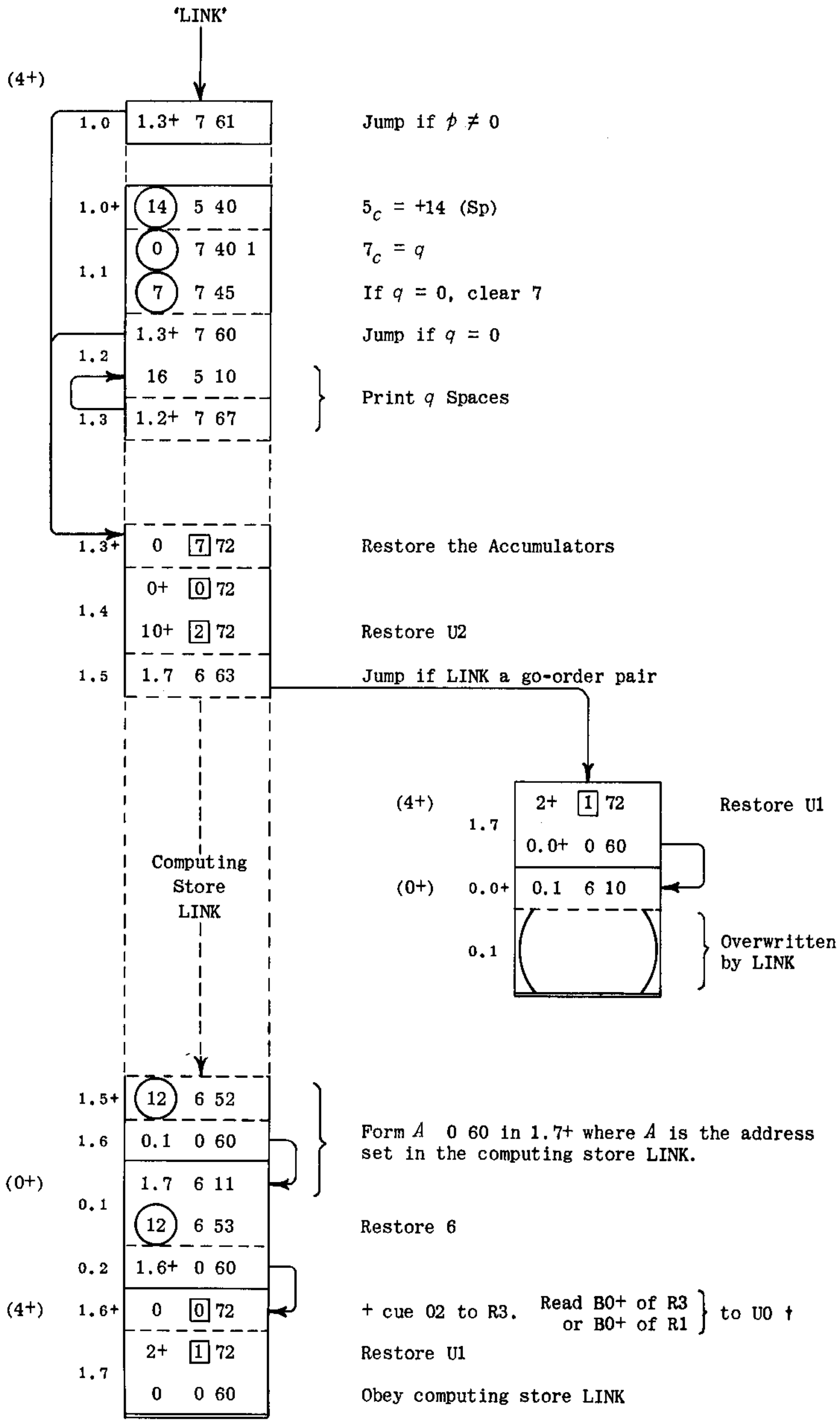
	Label	Instruction	Comment
	1.0	(6) 1 52	Next group markers ($p_j\ q_j$) to 1_m
		32 5 05	σ.000000..... to 5
(3+)	1	1 7 00	($p_j\ q_j$) to 7_m
		(28) 7 53	} $p_j.2^{-38}$ to 7
	2	(7) 7 45	}
		0.7+ 7 60	$p_j = 0$ → (6+) 0.7+ 4+ [1] 72 7 { Read B4+ to U1 if $p_j = 0$ → To 'LINK' (sheet 11)
		$p_j \neq 0$ ↓	
	1.3	7 5 01	$\sigma + p_j.2^{-38}$ to 5
		1 7 00	} q_{j0} to 7_m
	4	(2) 7 53	}
		35 7 05	}
	5	7 5 01	$\sigma + q_{j0}.2^{-13} + p_{j0}.2^{-38}$ to 5
		(0) 7 40 1	} $q_{j1}\ q_{j2}$ to 7_m
	6	(3) 7 45	}
		(25) 7 52	}
	7	8+ [2] 72	
		2.0+ 2 67	Jump if $i_5 = 0$
(8+)	2.0	(1) 7 41	$(q_{j1}\ q_{j2}).2^{-13} + i_5.2^{-38}$ to 7
		32 2 05	i_3 to 2_s
	1	7 2 01	$i_3 + (q_{j1}\ q_{j2}).2^{-13} + i_5.2^{-38}$ to 2
		9+ [1] 72	
	2	5+ [2] 72	
		2.3 0 60	

→ to 5+.3 to Printing Cycle (sheet 9)


from 8+.2+ (sheet 8)
PRINTING CYCLE
(5+) 2.3
1 4 41
6 7 00
1st time, $-2^{-13} + (X + 1).2^{-38}$ to 4; $|\zeta|$ to 7.
Subsequently, $\mu + 2^{-38}$ to 4; r to 7 (r is remainder)
4
0 6 00
0.0 4 66
Jump to form next digit, with $4_c = 1$
1st ENTRY (with $4_s = \mu = 0$ (6+) $4_c = X$)
2.5
6 0 23
If $\zeta = -1$ (OVR) $6 = 2^{-38}$, $7 = 0$
2.5+
1 1 52
Move up gap bit before p_{20}
MULTIPLY BY 10^X
0.0
0.0 6 61
10 6 40
1
6 7 20
2.6 0 60
STOP if OVR
$6_c = +10$
$10r$ to PQ
2.6
0.0 4 67
1.3+ 0 60
Jump X times on 1st entry
Pass through on subsequent entries
(9+) 1.3+
32 4 05
μ to 4
4
2.7 6 24
Divide by 10^{11}; δ (next digit) to 7, r to 6
1.6 5 67
Count digits in group
5
2.1 5 62
Jump to print if $\sigma = 0$ (last digit of group)
1.7+ 2 60
Jump if $i_3 = i_5 = q_{j1} = q_{j2} = 0$
6
2.1 5 62
Jump to print if $\sigma = 0$
0.2 7 60 5
Jump if $\delta = 0$ } to 6+.2 if $q_{j0} = 0$
0.2 2 63 5
Jump if $i_3 = 1$ } to 6+.3 if $q_{j0} = 1$
7
2.0+ 3 62
Jump if $i_4 = 0$
(5+) 2.0
16 3 10
Print sign before digit if $i_4 = 1$
32 5 03
$\sigma = 0$. Print this and later digits (i.e. stop L.H. zero suppression)
1
0.2+ 4 62
Jump if $\mu = 0$ (no R.H. zero suppression)
0.2+ 6 61
Jump if $r \neq 0$ (number not yet finished)
2
32 5 01
Put $\sigma = 1$ if $\mu = 1$ and $r = 0$ (start R.H. zero suppression)
0.2+ 0 60
Jump to print last digit
(6+) 0.2
14 7 40
Replace δ by Sp (If $q_0 = 0$)
0.2+
16 7 10
Print δ or Sp
0.3
1 5 41
2.3 5 67
Jump except after last digit of group
To 6+.4 after last digit of group (sheet 10)

END OF GROUP SPECIAL OPERATIONS





† See note on sheet 6.

	← N →							← X →			← F →						← M →		
I_S =	i_0	i_1 0	i_3	i_4	i_5	i_6	i_7	← X →			← S →			p_{00}	p_{01}	p_{02}	q_{00}	q_{01}	q_{02}
		0	p_{10}	p_{11}	p_{12}	q_{10}	q_{11} q_{12}	p_{20}	p_{21}	p_{22}	q_{20}	q_{21}	q_{22}	p_{30}	p_{31}	p_{32}	q_{30}	q_{31}	q_{32}

T 11+.0 **Optional Parameter List**

11+

01	1.2 0 71 0. 5.2 6 10	I_1
02	-100000000000	$-k_1$
03	1.2 0 71 5.2 6 10	I_2
04	-100000000000	$-k_2$
05	1.2 5 71 6.1 0 00	I_3
06	-1000000	$-k_3$
07	1.3 0 76 4 6.5 0 00	I_4
10	-1.0	$-k_4$

12+

11	1.0 0 77 7. 5.4 0 20	I_5
12	-1.0	$-k_5$
13	1.0 0 76 2 6.0 0 20	I_6
14	-1.0	$-k_6$
15	0.0 7 72 7 3.0 0 20	I_7
16	-1.0	$-k_7$
17	0.0 7 73 7 2.0 0 20	I_8
20	-1.0	$-k_8$

INTERLUDE

13+

Address	Order	Comment
0.0	0	Collating mask
	7.7 7 00 0.	
1	0	Cue 02 for R03
	0+ 0 00 0.	
2	0	
	0 2 00 0.	
3 (from Input J560.0+ − 13+.3)	1.5 1 10	Plant LINK
	33 4 00	(511.0, 0) to 4
4	34 4 03	
	[0 5] 70 4	Index address from 511.5 to 1
5	[1 0] 70 1	Index word to 1
	1 2 00	Index word to 2
6	(4) 2 45	Clear 2 if no parameter list
	1.4+ 2 60	Jump to obey LINK if no parameter list
7	5.5 5 00	Reference Address to 5
	(5) 1 52	Parameter List address to B.P position of b-order

14+

Address	Order	Comment
1.0	0.0 1 05	Form new cue 03 which is the Parameter
	0.2 1 03	List address less 2.
1	[1 0] 71 5	Store new cue 03 to R1
	34 7 00	Set Transfer Address equal to 11+.0 so that the next subroutine overwrites the optional Parameter List
2	34 7 01	
	5.7 7 13	
3	0.1 1 00	Plant cue 02 for R03 in 11+.1, since R03 will, if accepted, overwrite the unwanted optional Parameter List
	1.6 2 00	
4	[11 1] 71 2	
	1.5 1 00	Restore LINK
5 (to Input)	+0	Overwritten by LINK for return to Input
6	0+ - 00	
	0	

T 13+
J 560.0+ − 13+.3
L

Order	Comment
R 0 0 −0 2	Cue list title to satisfy the call for R3 made by R1.
03 - 28 -	

T 0+.2	Indicate to Assembly that R03 has 2 cues
L	Terminate cue list to R03
L	Terminate subroutine R03

R 1 NUMBER PRINT

NOTE: This subroutine is preceded on the Library Tape by R3

```
Ro 0-03
1 -28-
0+ 072
0.2+060
0+ 072
0.3+060
0
10+ 6000.
LN
Ø NUMBER PRINT
Ro 5-03
1 -01-
R4 6-02
3 -01-
R8 3-03
1 -01-
R11 0-20
1 -06-

1 000
0.1 610
1.7 611
12 653
1.6+060
0 773
1 600
26 752
2+173
10+273
0.0 300
0 1707
1+172
7+272
8+072
0.3 060

19 740
1.3 463
16 210
16 712
17 010
32 301
1.4 767
16 712
1.3+166
1 452
1.6 463
16 310
2 452
32 200
4 205
32 500
```

```
T 3+
6 152
32 505
1 700
28 753
7 745
0.7+760
7 501
1 700
2 753
35 705
7 501
0 7401
3 745
25 752
8+272
2.0+267

1.3+761
14 540
0 7401
7 745
1.3+760
16 510
1.2+767
0 772
0+072
10+272
1.7 663
12 652
0.1 060
0 072
2+172
0.0+060

16 310
32 503
0.2+462
0.2+661
32 501
0.2+060
1 441
6 700
0 600
0.0 466
6 023
1 152
0.0 467
1.3+060
+100000000000
```

```
0.0 661
10 640
6 720
2.6 060
14 740
16 710
1 541
2.3 567
18 742
0.5 0602
0 500
1.0 7673
14 740
16 710
16 710
4+1727

1 452
2.2 463
2 506
32 303
1 452
2.3+462
1 241
1 452
2.5 462
35 301
34 453
7 445
35 403
6+072
3+172
1.0 060

1 741
32 205
7 201
9+172
5+272
2.3 060
1 500
0 0707
1 400
4 305
11 341
0.7 663
6 602
3 341
6 521
30 240
```

```
0.7+060
32 400
0.7 767
1.0+661
32 500
2 741
1.0+060
32 405
2.7 624
1.6 567
2.1 562
1.7+260
2.1 562
0.2 7605
0.2 2635
2.0+362

T 11+

1.2 0710
5.2 6100.
-100000000000
1.2 0710
5.2 6100
-100000000000
1.2 5710
6.1 0000
-1000000
1.3 0764
6.5 0000
-1.0

1.0 0777
5.4 0200.
-1.0
1.0 0762
6.0 0200
-1.0
0.0 7727
3.0 0200
-1.0
0.0 7737
2.0 0200
-1.0
```

```
0
127 7000.
0
0+0000.
0
0 2000.
1.5 110
33 400
34 403
0 5704
1 0701
1 200
4 245
1.4+260
5.5 500
5 152

0.0 105
0.2 103
1 0715
34 700
34 701
5.7 713
0.1 100
1.6 200
11 1712
1.5 100
+0
0+ -00
0
T13+
J560.0+-13+.3
L
Ro 0-02
03 -28-
To+.2
L
L
```

PAGE LAYOUT SUBROUTINE

This subroutine uses R 1 to print a number from X1. R 3 itself arranges the numbers in rows, columns and blocks with all necessary CR and LF characters. R 1 is concerned only with printing the number in X1 in the manner directed by the layout parameter, the number of which, *s*, must be set in X7 on entry to R 3.

The preset parameters *L*, *B*, *N* determine the layout:

L No. of numbers in a line
B No. of lines in a block } If $B \neq 0$, then $N = 0$
N No. of numbers in a block } $B = 0$, then $N \neq 0$

	R 0	0 –0 2
	3	- 28 -
01	0+	0 00 0.
	0	
02	0	
	0+	0 00 0.

LN
ϕ BLOCK LAYOUT

R 1	1 –0 3	Title of Optional Parameter List
3	- 06 -	
R 3	7 –0 2	Call for cue 02 to R 1
1	- 01 -	
R 1	5 –0 1	Call for P.P.01
3	- 02 -	
R 1	6 –0 2	Call for P.P.02
3	- 02 -	
R 1	7 –0 3	Call for P.P.03
3	- 02 -	

Note: m is a marker set during R 3
$m = 0$ indicates that a line is unfinished
m positive ($= 13 \times 2^{-38}$) indicates start of new line
m negative ($= -1.0 + 13 \times 2^{-38}$) indicates start of new block.
At the end of a line or block, where $m \neq 0$, the five least significant bits of m contain LF for use by the master programme if required.

0+

Entry	Address	Instruction	Comment
Normal Entry	0.0	0.3 7 62	Jump if $7_c = +s$ (s is style number for R 1)
		7 7 02	$\lvert s \rvert$ to 7
New Line	1	0 [7] 73	Preserve Accumulators
		1+ [7] 72	Set counters etc. in accumulators
	2	(1) 1 41	m + 1 to 1 (to test for $m = 0$)
		0.7 0 60	
	3	0.6 6 62	Jump if LINK positive
		6 6 02	Negate LINK
New block	4	0 [7] 73	Preserve Accumulators
		1+ [7] 72	Set counters etc. in accumulators
	5	32 1 00	Set marker, $m = -1.0$ at start of new block
		0.7 0 60	
	6	0 [7] 73	Preserve Accumulators
		1+ [7] 72	Set counters etc. in accumulators
	7	2+ [0] 72	
to 2+.0		0.0 0 60	

1+

Address	Contents	Comment	
X0	+0		
X1	+4	Optional P.P.01; $L = 4$	becomes m
X2	+5	Optional P.P.02; $B = 5$	becomes L
X3	+0	Optional P.P.03; $N = 0$	becomes B
X4	+0		becomes N
X5	+0	+ P.P.01; L	
X6	+0	+ P.P.02; B	
X7	+0	+ P.P.03; N	

* The counter *B* counts the number of lines in a block. Normally *B* will be reduced by 1 in B3+.1+ before printing the last number in the line. If the Master Programme requires to start a new line before the previous line was completed, it may enter at 0+.1 (or at 0+.0 with −*s* in 7). On entry *m* will be zero, the a-order in 2+.3 will not then jump and the counter *B* will be reduced in 2+.3+.

2+

Word	Order	Notes
0 (from 0+.7+)	0.7 1 60	Jump if *m* = 0 i.e. middle of line
0.0	(30) 2 40	Print CRLF at end of line
1	16 2 10	
	(13) 2 40	
2	16 2 10	
	0.4 1 63	Jump if *m* negative i.e. end of block
3	0.5+ 1 67	Jump to start new line * but not new block
	0.5+ 3 67	
4	16 2 10	Print extra LF (New block)
	6 3 00	Reset *B* and *N* at end of block (New block)
5	7 4 00	
	5 2 00	Reset *L* at end of line
6	17 0 10	Print φ
	0	
7	3+ [0] 72	
	0.0 0 60	

3+

Word	Order	Notes
0.0	0 1 00	Set *m* = 0 (Overwritten by *m*)
	0.2+ 2 67	Count numbers on line (*L*) (Overwritten by *m*)
1	(13) 1 40	Set *m* = 13 x 2^{-38} at end of line (LF)
	0.2+ 3 67	Count lines in block (*B*)
2	32 1 01	Set *m* = −1.0 + 13 x 2^{-38} at end of block
	0.4 4 67	Count numbers in block (*N*)
3	(13) 1 40	Set *m* = −1.0 + 13 x 2^{-38}
	32 1 01	
4	1+ [7] 73	Store counters and marker in B1+
	0.0 1 10	*m* to 0.0
5	0 [7] 72	Restore Accumulators
	1 6 00	Number for printing to 6
6	0.0 1 00	Write *m* to B0.1 (R 1 leaves *m* in X1)
	[0 1] 71	
7	+0	+ cue 02 to R 1

L

NUMBER PRINT (SHORT)

This subroutine prints the contents of X7 as a signed or unsigned integer or fraction. It uses the Initial Orders number print routine.

	R 0 0 -0 9
	4 - 28 -
01	0+ [0] 72
	0.3 0 60
02	0+ [0] 72
	0.0+ 0 60
03	1+ [0] 72
	0.5 0 60
04	1+ [0] 72
	0.2 0 60
05	0+ [0] 72
	0.7 0 60
06	0+ [0] 72
	0.5 0 60
07	1+ [0] 72
	0.4 0 60
08	0+ [0] 72
	0.2 0 60
09	1+ [0] 72
	0.0+ 0 60

LN
I.O. NUMBER PUNCH

R 3 0 -0 4 4 - 06 -	Title of Optional Parameter List
R 0 1 -0 2 4 - 02 -	Call for P.P.02
R 0 3 -0 1 4 - 02 -	Call for P.P.01
R 1 2 -0 4 4 - 02 -	Call for P.P.04
R 1 5 -0 3 4 - 02 -	Call for P.P.03

0+

Entry	Address	Instruction	Notes
		449 - 00	= (449.0) Modifier for printing fraction
02	0.0	0 7 73	Store Accumulators
		(0) 2 40	+ P.P.02 { $2_c = n_2$
	1	0.5+ 0 60	Jump
08		0 7 73	Store Accumulators
	2	1+ 0 72	
01		0 7 73	+ P.P.01 { Store Accumulators
	3	(0) 2 40	$2_c = n_1$
		(0.7) 6 40	6_c = 0.7 To print sign
	4	0.6 0 60	
06		0 7 73	Store Accumulators
	5	(0.3+) 6 40	6_c = 0.3+ To omit sign
		0.0 4 00	4_m = 449.0 (Modifier for printing fraction)
	6	1+ 0 72	
05		0 7 73	Store Accumulators
	7	0.4 0 60	

1+

Entry	Address	Instruction	Notes
		448 - 00	= (448.0) Modifier for printing integer
09	0.0	0 7 73	Store Accumulators
		(0.3) 6 40	6_c = 0.3. To print 3 digit unsigned number
	1	0.6+ 0 60	
04		0 7 73	+ P.P.04 { Store Accumulators
	2	(0) 2 40	$2_c = n_4$
		(0.3+) 6 40	6_c = 0.3+ To omit sign
	3	0.6+ 0 60	
07		0 7 73	Store Accumulators
	4	0.6 0 60	
03		0 7 73	+ P.P.03 { Store Accumulators
	5	(0) 2 40	$2_c = n_3$
		(0.7) 6 40	6_c = 0.7 To print sign
	6	0.0 4 00	4_m = 448.0 (Modifier for printing integer)
		2+ 0 72	
to 2+.0	7	0.0 0 60	

2+

from 1+.7+

Address	Order	Notes
0.0	0.0 5 02	Prepare to print Space or CR LF
	(18) 0 42 2	Dummy. Constant used to print Sp Sp or CR LF
1	(25) 6 52	Form modifier for style of printing
	0.3+ 1 63	Jump if LINK set by 00 order
2	1 1 02	Negate 1 and Store LINK
	[0 1] 71	if LINK set by 02 order
3	(8) 5 53	Prepare to print CR LF
	16 5 10	Print CR or Sp
4	(4) 5 53	Prepare to print LF or Space
	16 5 10	Print LF or Space
5	0.7 1 00	Set Initial Orders LINK
	33 3 00	512.0 to 3_m
6	28 [0] 72 3	Enter Initial Orders number
	0 0 60 6	print routine in B540
7	77 [0] 72 4	LINK to enter I.O. L-sequence in B525.2: this restores the Accumulators from B 0
	0.2 0 60	and obeys the Main LINK

OPTIONAL PARAMETER LIST

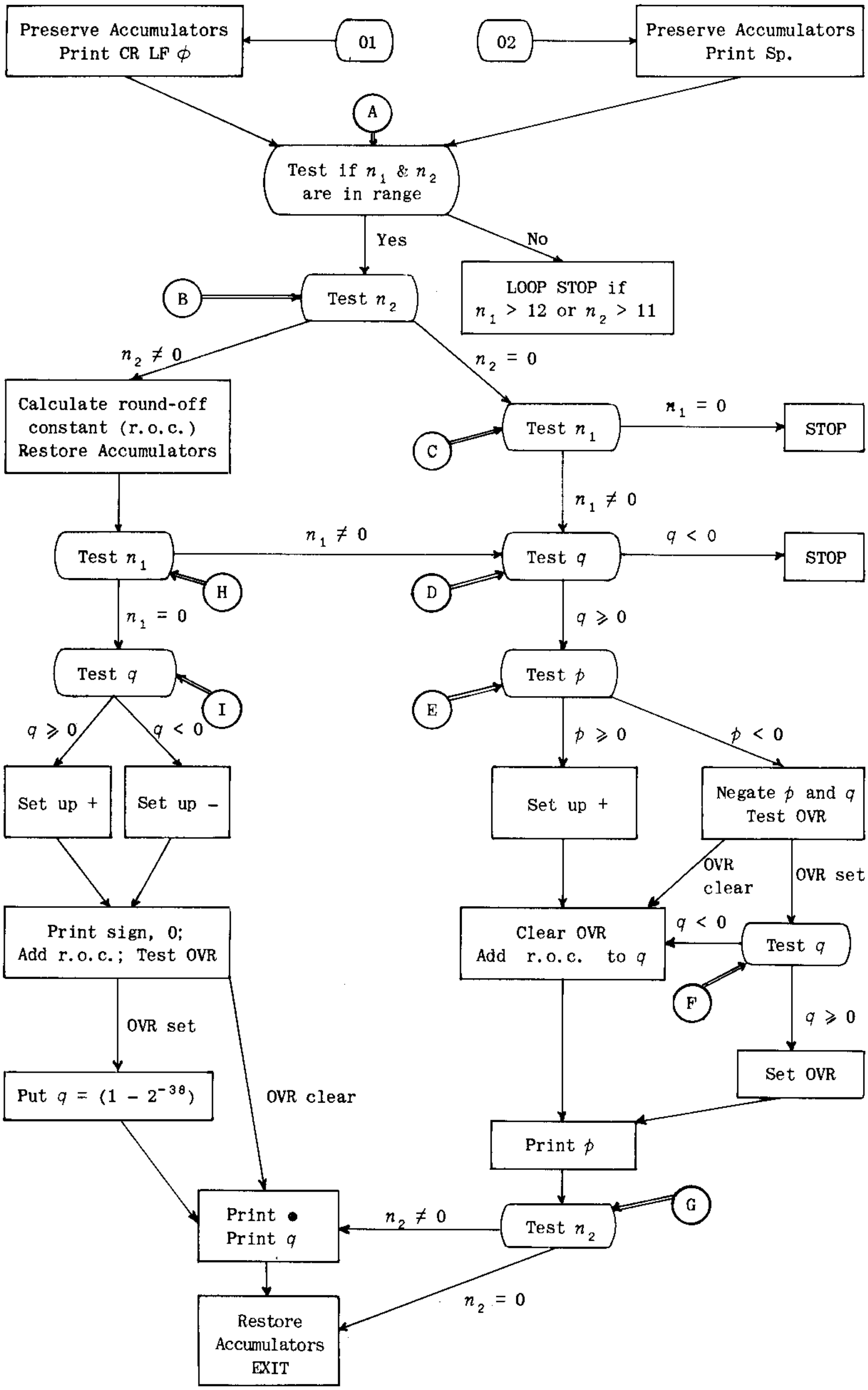
3+

Address	Word	Parameter
.0	0	P.P.01; $n_1 = 11$
	11 0 00 0.	
1	12 0 00 0.	P.P.02; $n_2 = 12$
	0	
2	0	P.P.03; $n_3 = 12$
	12 0 00 0.	
3	0	P.P.04; $n_4 = 12$
	12 0 00 0.	

L

DOUBLE LENGTH NUMBER PRINT

Flow Diagram



	R 0 0 -0 2
	5 - 28 -
01	0+ [0] 72
	0.0 0 60
02	0+ [0] 72
	0.2+ 0 60

LN
D.L. PRINT

Entry	Address	Order	Comment
		[0+]	
(01) →	0.0	0 [7] 73	
		(30) 3 40	Write Accumulators to B0
	1	16 3 10	Print CR
		(19) 3 42	Set up LF and set 3_p = 7
	2	0.3+ 0 60 (→ 3)	
(02) →		0 [7] 73	Write Accumulators to B0
	3	(14) 3 40	
		16 3 10	Print LF or Sp.
	4	10 0 10 3	Dummy order, or Print ϕ after CR LF
		(127) 4 45	Remove 32 m.s. bits of 4 and 5
	5	(127) 5 45	
		(12) 4 44	$12-n_1$ to 4_c
	6	(11) 5 44	$11-n_2$ to 5_c
(A) - - →		0.6+ 4 63	LOOP STOP if n_1 > 12
	7	1+ [0] 72	
		0.0 0 60	

from 0+.7+

Entry	Word	Instruction	Comment
		1+	**TEST n_1, n_2 and q**
A	0.0	0.0 5 63	LOOP STOP if n_2 > 11
		(12) 4 44	n_1 to 4_C
	1	(11) 5 44	n_2 to 5_C
B		0.3 5 60	Test n_2
	2	2+ [0] 72	
		0.1 0 60	Jump if $n_2 \neq 0$
C	3	0.3 4 60	LOOP STOP if $n_1 = n_2 = 0$
		0 7 00	Clear 7
	4	0 2 00	Clear 2, rounding off constant = 0
D		0.4+ 7 63	LOOP STOP if q < 0
	5	(11) 3 40	Set up −
E		0.7 6 62	Jump is $p \geqslant 0$
	6	6 6 02	Negate p, if p < 0
		3+ [0] 72	Jump if p < 0 (to 3+.7)
	7	3+ [0] 72	
		0.2 3 67	Set up + and jump (to 3+.2)

Entry	Word	Instruction	Comment
		2+	**FORM ROUND OFF CONSTANT**
	0.0	+0.1	
	0.1	33 6 00	Calculate round off constant (r.o.c.)
		0.0 6 21	
	2	0.1+ 5 67	
		0.2 6 10	r.o.c. to 0.2
	3	0 [7] 72	Restore Accumulators
		0.2 2 00	r.o.c. to 2
	4	(127) 4 45	Remove 32 m.s. bits of 4
H		0.7 4 60	Jump if $n_1 = 0$
	5	1+ [0] 72	
		0.4+ 0 60	Jump if $n_1 \neq 0$ to print fraction
	6	4+ [0] 72	
I		0.2 7 63	Jump to 4+.2+ if q < 0, to 4+.7 if $q \geqslant 0$ (to 4+.2)
	7	(11) 3 40	Set up −
		0.6 0 60	

Entry/exit	Line	Order	Comment
		3+	**PREPARE TO PRINT INTEGER**
	0.0	0.1 0 64	Test OVR
		0.3+ 7 62	Test q if OVR set, jump if $q \geqslant 0$
	1	6 0 23	Justify
		0.2 0 65	Clear OVR if p = −1.0 and $q \neq 0$ on entry
from 1+.7+	2	2 7 01	Add r.o.c. to q
		6 0 23	Justify
	3	0.4 0 60	
		32 0 02	Set OVR
	4	7 2 00	q to 2
		6 7 00	} p to PQ
	5	0 6 00	} p to PQ
		6 0 23	} p to PQ
	6	4+ [0] 72	
		0.0 0 60	Jump to 4+.0, to print p
from 1+.6+	7	7 7 02	Negate q if $p < 0$
		0.0 0 60	

Entry/exit	Line	Order	Comment
		4+	**PREPARE TO PRINT**
	0.0	(12) 4 44	Set $12-n_1$ in 4_c
		(12) 1 40	Set 12 in 1_c
	1	32 3 01	Make C(3) a GO order pair
to 5+.2		5+ [0] 72	Jump to print Integer
from 2+.6+	2	(11) 3 40	Set up −
		7 7 02	Negate q ($q < 0$)
	3	16 3 10	Print sign
		16 0 10	Print 0
	4	2 7 01	Add r.o.c. to q
		0.6 0 64	Test OVR
	5	(1) 7 42	} If OVR set, $(1 - 2^{-38})$ to 7
		32 7 03	} If OVR set, $(1 - 2^{-38})$ to 7
	6	6+ [0] 72	
to 6+.1		0.1 0 60	Jump to print Fraction
from 2+.6+	7	(10) 3 40	Set up +
		0.3 0 60	

PRINT INTEGER

Block 5+	Instruction	Comment
0.0	(14) 7 40	Print Sp for non-significant zeros after $(12-n_1)$ cycles
	16 7 10	
1	(10) 7 40	
	6 7 20	(Remainder) x 10 to PQ
2 (from 4+.1+)	0.7 6 24	Next Decimal Digit to 7
	(1) 4 43	Sets 4_m = -1 after $(12-n_1)$ cycles
3	0.4 1 67	Count 12 decimal digits
	32 7 01	Ensure printing of last integral digit, even if 0
4	0.1 7 60 4	Jump if digit is zero } becomes 0
	16 3 10	Print sign before first digit } becomes 0
5	0.4 3 10	Cancel sign printing
	16 7 10	Print digit of p
6	0.1 7 62	
	6+ [0] 72	
7	+100000000000	$= 10^{11} \times 2^{-38}$

PRINT FRACTION

Block 6+	Instruction	Comment
0.0	2 7 00	Restore q
	0.4 5 60	Jump if $n_2 = 0$
1 (from 4+.6+)	(12) 6 40	Print ●
	16 6 10	
2	(10) 6 40	Print fraction
	6 7 20	
3	16 6 10	
	0.2 5 67	
4	0 [7] 72	Restore Accumulators
	0.5 1 10	Plant LINK and EXIT
5	+0	Overwritten by LINK
6	+0	Unused
7	(127) 5 45	Remove 32 m.s. bits of 5
	0.0 0 60	

L

FAST D.L. PRINT FOR 7168 WORD STORE

This routine has the same specification as the original R 5 but it makes use of the full speeds of the Creed 3000 and Soroban punches. The extra space required to store such a routine is obtained by using some of the extra isolated blocks on the 7168 word store and the routine in this form can therefore only be used on that store.

The routine stores the accumulators, U1 and U2 in B1+, B2+ and B3+ respectively of R 5. This makes U1 and U2 available for use in the routine without altering the original specification of R 5. Four isolated store blocks are used for programme and two for rounding constants.

Most mixed numbers printed by R 5 are less than 10,000. The new routine separates such numbers in order to prevent going many times round a time-consuming zero omission routine.

Error Analysis

Representing the integer to be printed by $p = I.2^{-38}$ we form

$I^* = I.2^{-38}\left(\frac{2^{74}}{10^{11}} + \epsilon\right).2^{-38}$ where ϵ is the rounding error due to terminating $\frac{2^{74}}{10^{11}}$ after 38 bits and has the approximate value +.214 if we take $\left(\frac{2^{74}}{10^{11}}\right) = 188894659315$.

I^* is then multiplied by 4 and 1.2^{-38} added to the result, all succeeding digits being dropped. i.e. we form $\frac{I}{10^{11}} + 4\epsilon.I.2^{-76} + (1 - \delta).2^{-38}$ where δ is the error due to dropping digits ($0 \leqslant \delta < 1$).

During printing $I + 4\epsilon I.2^{-76}.10^{11} + (1 - \delta)2^{-38}.10^{11}$ is formed.

We require $0 \leqslant (4\epsilon I.2^{-38} + 1 - \delta)2^{-38}.10^{11} < 1$ if the process is not to fail.

Since $\delta < 1$ and I is made positive before printing, we ensure that the error term is positive by making $\epsilon \geqslant 0$.

The error will be greatest when $I = 2^{38}$ (after rounding) and we therefore require that

$$(4\epsilon + 1 - \delta) < 2^{38}.10^{-11}.$$

In the worst case $\delta = 0$ and $4\epsilon < 1$.

Thus $(4\epsilon + 1 - \delta) < 2$ and since $2^{38}.10^{-11} \doteq 2.74$, the required conditions are satisfied and the integer printed by this process will always be correct.

	R 0 0 -0 2
	5 - 28 -
01	0+ [0] 72
	0.0 0 60
02	0+ [0] 72
	0.2+ 0 60

LN
FAST D.L. PRINT

Entry	Address	Instruction	Comment
		0+	
01 →	0.0	1+ [7] 73	Preserve Accumulators in B1+
		(30) 3 40	
	1	(19) 1 42	Set up LF and set $1_p = 7$
		16 3 10	Print CR
	2	0.3+ 0 60	
02 →		1+ [7] 73	Preserve Accumulators in B1+
	3	(14) 1 40	
		2+ [1] 73	Preserve U1 in B2+
	4	(127) 4 45	Remove 32 m.s. bits of 4 and 5
		(127) 5 45	
	5	(12) 4 44	$12 - n_1$ to 4_c
		3+ [2] 73	Preserve U2 in B3+
	6	4+ [1] 72	
		0.6+ 4 63	LOOP STOP if $n_1 > 12$
	7	16 1 10	Print LF or Sp
to 4+.0 ←		(11) 5 44	$11 - n_2$ to 5_c

TEST n_1, n_2 and q

4+

Link	Addr	Instruction	Comment
from 0+.7+	1.0	5+ [0] 72	
		37 3 00	Set $3_m = 896.0$, $3_c = +10$
	1	(10) 3 41	
		0.0 0 60	
	2	1.0 6 10	Store p in 1.0
		1.1 7 10	Store q in 1.1
	3	35 5 20	Set $7_m = (896.0 + n_2)$
		3 7 01	
	4	[125 7] 70 7	Read round off constant from $B(1021.7 + n_2)$ $n_2 \geqslant 1$
		1.0 6 00	Restore p
	5	1.1 7 00	Restore q
		0.4+ 4 61	Jump if $n_1 \neq 0$
	6	122 [0] 72 3	B1018 to U0
to 1018.4+		0.4+ 7 62	Jump if $q \geqslant 0$ and $n_1 = 0$
	7	(1) 3 41	Set up – otherwise
to 1018.4		0.4 0 60	

5+

Link	Addr	Instruction	Comment
	0.0	0.0 5 63	LOOP STOP if $n_2 > 11$
		6+ [2] 72	
	1	(12) 4 44	n_1 to 4_c
		(11) 5 44	n_2 to 5_c
	2	10 0 10 1	Print figure shift if cue 01
		1.2 5 61	Jump if $n_2 \neq 0$
	3	0.3 4 60	STOP if $n_1 = n_2 = 0$
		0 7 00	Clear q for $n_2 = 0$
	4	0 1 00	Clear 1, round off constant = 0
		0.4+ 7 63	LOOP STOP if $q < 0$
	5	122 [1] 72 3	B1018 to U1
to 1018.1+		1.1+ 6 62	Jump if $p \geqslant 0$
	6	6 6 02	Negate p, if $p < 0$
		7 7 02	Negate q, if $p < 0$
	7	(1) 3 41	Set up –
to 1018.0+		1.0+ 0 64	Jump if OVR clear, $p \neq -2^{38}$

PREPARE TO PRINT

1018

Entry	Address	Instruction	Comment
from 5+.7+ if OVR set	1.0	1.3 7 62	Jump if q = 0 and OVR set to print -2^{38}
from 5+.7+ otherwise		6 0 23	Justify
	1	1.1+ 0 65	Clear OVR if p = −1.0 and $q \neq 0$
from 5+.5+		1 7 01	Add r.o.c. to q
	2	6 0 23	Justify
		1.3+ 0 60	
	3	32 0 02	Reset OVR if p = −1.0 and q = 0
		124 [1] 72 3	Read B1020 to U1 and jump
from 4+.7+	0.4	7 7 02	Negate q if $q < 0$ and $n_1 = 0$
from 4+.6+		16 3 10	Print sign
	5	(1 7 01)	Add r.o.c. to q } May be overwritten by LINK
		(0.7 0 64)	Jump if OVR clear } May be overwritten by LINK
	6	(1) 7 42	Form $(1 - 2^{-38})$ in 7
		32 7 03	
	7	16 0 10	Print 0
to 6+.0		2.0 0 60	Jump to print fraction

1020

Entry	Address	Instruction	Comment
	1.0	+188894659315	$\left(\frac{2^{74}}{10^{11}} + \epsilon\right).2^{-38} \doteq \frac{2^{36}}{10^{11}}$
	1	1.0 7 20	$(pq) = 2^2 \times 2^{-38} I \times \frac{2^{36}}{10^{11}}$
		(2) 0 54	
	2	(1) 6 41	$I' = \frac{I}{10^{11}}$ (rounded up) to 7 [see note on Sheet 5]
		6 7 00	
	3	111 [1] 72 3	Read B1007 to U1
to 1021.0		0.0 0 60	Jump to print integer
	4	7 2 00	q to 2
		6 7 00	p to 7
	5	1.0 6 00	Conversion constant to 6
		(11) 1 40	Set 11 in 1_c
	6	125 [0] 72 3	Read B1021 to U0
		(11) 4 44	$11 - n_1$ in 4_c
	7	1.1+ 0 65	Jump if OVR set. i.e. $(pq) = \pm 2^{38}$
		1.1 0 60	

Note: On entry to B1021

If	$I < 10^{11}$	$7_s = 0$ and OVR clear
If	$10^{11} \leqslant I < 2.10^{11}$	$7_s = 1$ and OVR set
If	$2.10^{11} \leqslant I \leqslant 2^{38}$	$7_s = 0$ and OVR set

PRINT INTEGER

1021 (from 1020.3+)

Addr.	Order	Notes
0.0	(5) 5 52	Set $5_c = n_2.2^5 + Sp$
	(14) 5 41	
1	0 6 00	Set $x_6 = 0$ if $I < 10^{11}$; $x_6 = 1$ if $10^{11} \leqslant I < 2.10^{11}$; $x_6 = -2$ if $I \geqslant 2.10^{11}$
	6 0 23	
2	7.7 6 12 6	Set $x_6 = +2$ if $I \geqslant 2.10^{11}$
	1.3 6 61	Jump if m.s. digit present
3	7 6 00	$\dfrac{I - 10^4}{10^{11}}$
	2.6 6 03	
4	1.0 6 62	Jump if $I \geqslant 10^4$
	(7) 4 43	$4 - n_1$ to 4_c (4.1–7.1: $I < 10^4$)
5	0.7 4 62	Jump if $n_1 \leqslant 4$ (5.0–5.1: May be overwritten by LINK)
	16 5 10	
6	(1) 4 41	Print Spaces if $n_1 > 4$ and $I < 10^4$
	0.5+ 4 61	
7	(4) 1 40	Set $1_c = 4$
	2.7 7 20	$I' = \left(\dfrac{I}{10^4}\right)$ to 7

1007

Addr.	Order	Notes
1.0	9 5 10 4	Print Space ($4_p = 7$) or dummy order ($4_p = 0$)
	2.5 7 20	$10I'$
1	1.2 1 67	Reduce counter in 1_c; initially $1_c = 11$ if $I \geqslant 10^4$, $1_c = 4$ if $I < 10^4$
	1.3 0 60	Jump to print sign if last digit
2	(1) 4 43	Count zero omissions
	1.0 6 60	Jump to print Sp if non-significant zero
3	16 3 10	Print sign on finding non-zero digit
	1.5+ 1 60	Jump if last digit
4	16 6 10	Print significant digits
	2.5 7 20	
5	1.4 1 67	
	(5) 5 53	Reset $5_c = n_2$
6	16 6 10	Print last digit
	2 7 00	Restore F (fractional part) to 7
7	2.2+ 5 60	Jump to EXIT if $n_2 = 0$ (to 6+.2+)
	0	

PRINT FRACTION

Block 6+

Entry/exit	Address	Instruction	Comment
from 1007.7+ →	2.0	(12) 6 40	Print •
		16 6 10	
loop ← (to 1)	1	2.5 7 20	Print fraction
		16 6 10	
		2.1 5 67	
from 1007.2+ →	2	1+ [7] 72	Restore Accumulators
	3	2+ [1] 72	Restore U1
		0.5 1 10	Plant LINK
	4	3+ [2] 72	Restore U2
to LINK ←		0.5 0 60	Jump to obey LINK
	5	+10	
	6	+27487	$+\frac{10^4}{10^{11}}$
	7	+10000000	$+10^7.2^{-38}$

ROUND OFF CONSTANTS

1022

0	+13743895347	+0.05 for $n_2 = 1$
1	+1374389535	+0.005 for $n_2 = 2$
2	+137438953	+0.0005 for $n_2 = 3$
3	+13743895	+0.00005 for $n_2 = 4$
4	+1374390	+0.000005 for $n_2 = 5$
5	+137439	+0.0000005 for $n_2 = 6$
6	+13744	+0.00000005 for $n_2 = 7$
7	+1374	+0.000000005 for $n_2 = 8$

1023

0	+137	+0.0000000005 for $n_2 = 9$
1	+14	+0.00000000005 for $n_2 = 10$
2	+1	+0.000000000005 for $n_2 = 11$
3	+0	Unused
4	+0	Unused
5	+0	Unused
6	+0	Unused

London Computer Centre,
68, Newman Street,
LONDON, W.1.

Issue 1
9th May, 1961.
D.M. M.J.M.

PRINT FLOATING POINT NUMBERS

	R 0 0 -0 6
	11 - 28 -
01	0+ [0] 72
	0.2 0 60
02	0+ [0] 72
	0.4+ 0 60
03	0+ [0] 72
	0.2 7 66
04	0+ [0] 72
	0.4+ 7 66
05	0+ 0 00 0.
	0
06	0
	0+ 0 00 0.

LN
F.P. PRINT MK 2

R 0 1 -0 1	Calls for P.P.01
11 - 02 -	
R 0 1 -0 1	
11 - 02 -	
R 2 7 -0 1	
11 - 02 -	

T1+

Optional Parameter list

R 0 0 -0 1	Title
11 - 06 -	
-256	P.P.01

NOTE: n_1 = number of digits to be printed before decimal point.
n_2 = number of digits to be printed after decimal point.
d = decimal exponent.

T0+ (0+)

Entry / exit	Address	Order	Comment
	0.0	0.0 6 10	Plant LINK
		0.0 0 60	Jump to plant and then to obey the LINK
	1	+0	+ (2 × P.P.01)
(01) (03) →	2	0 [7] 73	Preserve accumulators
		(30) 4 40	Print CR
	3	16 4 10	
		(19) 4 42	Set up LF and $4p$ = 7
	4	0.6 0 60 (→ 6)	
(02) (04) →		0 [7] 73	Preserve accumulators
	5	(14) 4 40	Print Sp
		16 4 10	
	6	1+ [1] 73	Preserve U1
		1 6 00	Number to 6
	7	0.1 6 05	Argument A to 6
		2+ [1] 72	

T2+ (2+)

BINARY/DECIMAL CONVERSION

Entry / exit	Address	Order	Comment
	1.0	6 1 03	Exponent $a + 2^{n-1}$ to 1
		7 3 00	n_2 to 3
	1	0 2 00	Zeroise decimal exponent
		3+ [0] 72	
	2	16 4 10	Print Sp or LF
to 3+.5+		0.5+ 6 60	Jump if argument zero
	3	1.7 1 01	Subtract 2^{n-1} to form a
to 3+.2		0.2 1 62	Jump if exponent ⩾ 0
	4	(10) 1 41	Convert negative binary exponent in 1 to decimal exponent in 2.
		(3) 2 43	
	5	0.0 6 20	
		(2) 1 56	
	6	1.4 1 63 (→ 4)	
to 3+.5+		0.5+ 0 60	
	7	+0	+ P.P.01

BINARY/DECIMAL CONVERSION

3+

Address	Order	Comment
0.0	+.9765625	$= \frac{10^3}{2^{10}}$
1	+.024	$= \frac{2^{10}}{10^3} - 1$
2	(10) 1 43 (from 2+.3+)	Convert positive binary exponent in 1 to decimal exponent in 2
	0.5 1 63	
3	(3) 2 41	
	0.1 6 22	
4	(2) 1 56	
	0.2 0 60	
5	(10) 1 41	
	4+ [1] 72 (from 2+.2+ and 2+.6+)	
6	1.2+ 1 60	Final conversion of binary exponent in 1 to decimal exponent in 2
	(3) 1 43	
7	1.7 6 20	
	(2) 1 56	

4+

Address	Order	Comment
1.0	(1) 2 41	
	0.6 1 62	
1	1 1 02	
	(25) 1 52	
2	(0) 6 51 1	
	10 0 10 4	Dummy order, or print φ after CR LF
3	(10) 4 40	Set up +
	1.5 6 62	
4	6 6 02	Negate 6, if negative
	(11) 4 40	Set up −
5	5+ [0] 72	
	6 7 00	Argument to 7
6	6+ [1] 72	
	1.0 0 60 (to 6+.0)	
7	+.8	$= \frac{2^3}{10}$

5+

Link	Address	Instruction	Notes
			ADD ROUND-OFF CONSTANT
	0.0	12 6 31 4.	
		4.6 3 14 7	$= +0.1 + 2^{-38}$
	1	(5 2 04)	$d' = n_1$ (floating) or $d' = d$ (fixed)
		(0.1 4 10)	Store sign in 0.1
	2	(3 4 00)	
		(0.2 7 10)	Store argument in 0.2
	3	(127) 4 45	n_2
		2 4 01	$d' + n_2$
	4	0.7 4 63	Jump if number effectively zero
		33 6 00	Form round-off constant
	5	0.6+ 4 60	Form round-off constant
		0.0 6 20	Form round-off constant
	6	0.5+ 4 67	Form round-off constant
		0.2 6 01	
	7	7+ [1] 72	
to 7+.2+		1.2+ 0 64	Jump unless round-off causes OVR

6+

Link	Address	Instruction	Notes
from 4+.6+	1.0	1.4 7 60	
		0.0 6 03	Subtract 0.1 from argument
	1	1.3 6 62	Jump if argument ⩾ 0.1
		(10) 6 40	
	2	6 7 20	Multiply by 10
		(1) 2 43	Reduce decimal exponent by 1 ($=d$)
	3	5 2 04	$(n_1 - d)$
		2 1 02	$(d - n_1)$ to 1 as exponent
	4	1.4+ 2 62 3	Jump if fixed point and $d \leqslant n_1$
		0 2 00	Clear 2
	5	32 3 06	Mark floating point
		1.6+ 7 60	
	6	0.1 0 60	Jump if argument not zero
		0.1 4 10	Write sign into 0.1 if argument zero
	7	7+ [1] 72	
to 7+.3		1.3 0 60	

7+

PRINT SPACES BEFORE NUMBER

Label	Order	Comment
	0.0 6 00	+0.1 to 6
1.0	1.2 3 63	Jump if floating point output
	(1) 2 41	$d + 1$
1	1.2+ 0 60	
	(1) 1 41	Increase exponent for floating point output
2 (from 5+.7+)	2 4 02	$- d'$
(from 6+.7+)	2 5 03 2	$n_1 - d'$ if $d' \geqslant 0$
3	(1) 5 43 4	$n_1 - 1$ if $d' \leqslant 0$
	1.6+ 5 63	Jump if no spaces before the number
4	1.6+ 5 60	
	(14) 7 40	Print spaces in place of non-significant zeros
5	16 7 10	
	1.5+ 5 67	
6	0.1 5 00	Sign to 5
	8+ [0] 72	
7	0.1 0 60	

8+

PRINT INTEGRAL PART

Label	Order	Comment
	448 - 00 0.	
0.0	14	
	6 7 00	Argument to 7
1	16 5 10	Print sign
	9+ [1] 72	
2	(10) 5 40	
	16 0 10 4	Print zero before decimal point if necessary
3	0.6 4 62	Jump if no digits before decimal point
	2 4 00	d' to 4
4	5 7 20	Print integral part of number
	16 6 10	
5	0.4+ 4 67	
	3 4 00	
6	(127) 4 45	n_2
(to 9+.6+)	1.6+ 4 60	Jump if no digits after decimal point
7	(12) 6 40	Set up decimal point

9+

PRINT FRACTIONAL PART

Word	Arrows	Instruction	Comment
0		16 6 10	Print decimal point
0	jump to 1.5 (labelled 1.0)	1.5 2 62	Jump if no zeros after decimal point
1		2 2 02	$- d'$
1		2 4 03	$n_2 + d'$
2	jump to 1.3	1.3 4 62	Max $(n_2, - d')$ to 2
2		3 2 00	(same as above)
3	from 1.2 and 1.3	16 0 10	Print zeros after decimal point
3	jump to 1.3	1.3 2 67	
4	jump to 1.6+	1.6+ 4 63	Jump if no significant digits in number
4	jump to 1.6+	1.6+ 4 60	(same as above)
5	from 1.0 and 1.6	5 7 20	
5		16 6 10	Print fractional digits
6	jump to 1.5	1.5 4 67	
6	from 8+.7; from 1.4	0.0 4 00	(448.0, 14) to 4
7		10+ [1] 72	
7	jump to 10+.1.0	1.0+ 0 60	

10+

PRINT EXPONENT AND EXIT

Word	Arrows	Instruction	Comment
0		1.3 0 60	LINK for return from I.O. print routine
0	from 9+.7 (labelled 1.0)	1 7 00	Exponent to 7
1		(3) 2 40	2_c = 3, to print 3 digits in exponent
1		92 [0] 72 4	Enter I.O. print routine if X3 < 0 (i.e. floating point), to print exponent
2		1.0 1 00	(same as above)
2	to I.Os	0.7 3 63	(same as above)
3	from I.Os	17 4 10 3	Print two spaces after exponent, but not in fixed point printing
3		17 4 10 3	(same as above)
4		0 [7] 72	Restore accumulators
4	jump to 1.6+	1.6+ 6 63	Jump if not computing store link
5		(12) 6 52	Add jump address to 1.7+
5		1.7 6 11	(same as above)
6		(12) 6 53	Restore X6
6	from 1.4	0+ [0] 72	Restore 0+ to U0
7		1+ [1] 72	Restore U1
7		0 0 60	Exit if computing store LINK; Jump to 0+.0+ for ordinary LINK.

L

SIGNED £.S.D. PRINT FROM PENCE

This subroutine prints the contents of X7 as £.s.d. in the form $a.b.c$ where

$$X7 = \pm(240a + 12b + c).2^{-38}$$

and a,b,c are integers such that

$0 \leqslant a$

$0 \leqslant b \leqslant 19$

$0 \leqslant c \leqslant 11$

On entry $X2 = n.2^{-38}$, where n is the required number of places of pounds.

	R 0 0 -0 4 42 - 28 -
01	0+ [0] 72 0.0 0 60
02	0+ [0] 72 0.2 0 60
03	0+ 0 00 0. 0
04	0 0+ 0 00 0.

LN
SIGNED £.S.D. PRINT

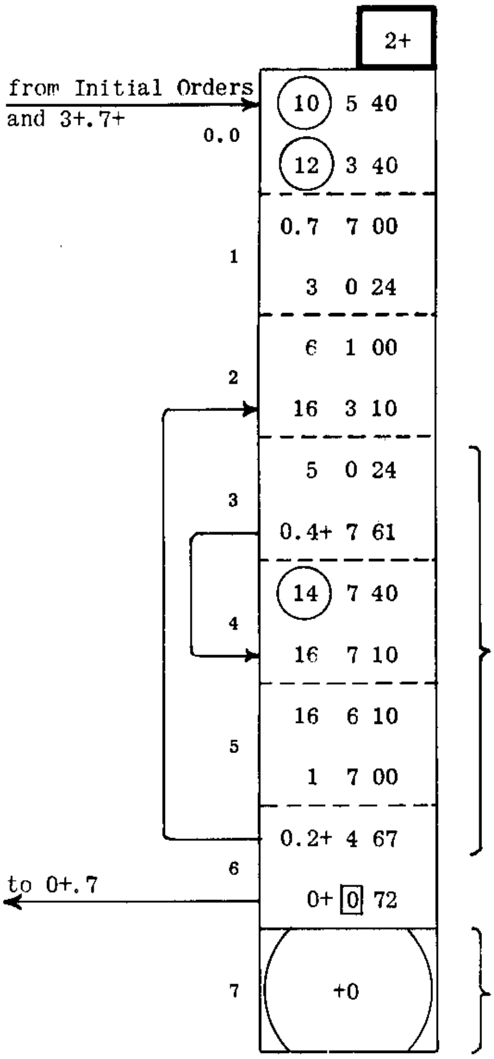
Entry	Address	Order	Comment
		0+	**Entry and Exit**
(01) →	0.0	(30) 6 40 16 6 10	Print CR
	1	(19) 6 42 0.2+ 0 60	Set up LF
(02) →	2	(14) 6 40 16 6 10	Set up Sp Print LF or Sp
to 1+.0 ←	3	1+ [0] 72 0.0 0 60	
	4	1 6 00 0.6 6 63	LINK to 6 Jump if LINK is a *go* order pair
	5	(12) 6 52 0.6 6 11	Add Computing Store link into N address of 0.6
	6	0.7 6 10 0 6 62	Plant LINK in 0.7 Jump to Computing Store link
from 2+.6+ →	7	0 [7] 72 0.4 0 60	Restore Accumulators } Overwritten by *go* order pair link

1+ **Form a**

Word	Order	Comment
0.0 (from 0+.6+)	0 [7] 73	Preserve Accumulators
	0.6 4 00	(448.0, 2) to 4
1	(14) 5 40	Set up Sp in 5
	10 0 10 6	Print ϕ after CR LF
2	0.3+ 7 62	Jump if X7 is positive
	7 7 02	Negate if negative
3	(21) 5 42	Set up – in 5_c and 7 in 5_p
	0.7 0 24	X7 = $\lvert a \rvert$; X6 = $\lvert 12b + c \rvert$
4	6 1 00	Store $\lvert 12b + c \rvert$ in B2+.7
	[2+ 7] 71	
5	3+ [0] 72	
	0.3+ 0 60	
6	448 - 00 0.	= (448.0, 2)
	2	
7	+240	

3+ **Print a**

Word	Order	Comment
0.0	(14) 7 40	Print Spaces
	16 7 10	
1	0.2 2 60	Jump if $n = 0$; i.e. print no Spaces
	0.0 2 67	Jump if $n \neq 1$ to print Spaces
2	16 5 10	Print –
	16 0 10	Print 0
3	0.7 0 60	Jump to B2+ through Initial Orders LINK
	0 7 12 5	Restore a to original sign
4	16 5 10 5	Print Sp if a is positive
	0.7 1 00	Plant LINK for Initial Orders
5	0.6 7 61	Jump to PRINT if $\lvert a \rvert \neq 0$
	0.1 5 63	Jump if $a = 0$ and number is negative
6	92 [0] 72 4	Enter 540.3+ if $x \geq 0$ (I.O. unsigned integer print)
	0.3+ 7 62	540.7 if $x < 0$ (I.O. signed integer print)
7	2+ [0] 72	LINK for Initial Orders
	0.0 0 60 (to 2+.0)	



Print *b.c*

+ 10 to 5_c

+ 12 to 3_c

$|12b + c|$ to 7

b to 7; *c* to 6

c to 1

Print ●

Print *b.c* with suppression of most significant zeros

Overwritten by $|12b + c|$

TEXT INPUT AND OUTPUT

	R 0 0 -0 1	
	52 - 28 -	
01	0+ [0] 72	
	0.0 0 60	

LN
TEXT IN/OUT

Method
See page 2.

0+ — Output Text

(01) → 0.0

	Order	Comment
0.0	0 [7] 73	Preserve Accumulators
	(13) 0 55	Shift text number, *r*, to 7_m } Look up index
1	[1+ 1] 70 7	Read text address to 1 } Look up index
	1 6 00	Text address to 6
2	[0 0] 70 6	Text word to 1
	1 7 00	Text word to 7
3	(7) 1 45	Character count, *n*, (in l.s. bits of text word) to 1
	0.3+ 1 60	LOOP STOP if no text (*n* = 0)
4	(3) 7 53	Remove *n* from 7
	17 7 10	Output one character from text word in 7
5	(5) 7 53	Shift next character into 5 l.s. bits of 7
	0.4+ 1 67	Count *n* characters
6	35 6 01	Step 6_m
	0.2 7 60	Jump if not last word
7	1+ [0] 72	
	0.0 0 60	

1+ — Search for first L during Input

	Order	Comment
0.0	0 [7] 72	Restore Accumulators
	0.1 1 10	Plant LINK and EXIT
1 (Interlude Entry)	((0) 0 74	Switch to Main Tape Reader } Overwritten by LINKS
	[0 0] 71)	Plant Initial Orders Link in B0.0 } Overwritten by LINKS
2	[0 1] 70	} Plant Assembly Link in B1+.1
	[1+ 1] 71	
3	0.6 1 00	} Set Link for first L in B0.1
	[0 1] 71	
4	0	
	0.5 1 00	Set 1_B = 475
5	59 [3] 72 1	} Enter Initial Orders at 534.1 to search for first L
	3.1 0 60 (to Initial Orders)	
6	1+ [0] 72	} LINK for first L
	0.7 0 60	
7 (from Initial Orders)	2+ [1] 72	
	3+ [2] 72	

No. Bits	1	5	5	5	5	5	5	5	3
Contents	m	a_7	a_6	a_5	a_4	a_3	a_2	a_1	n

n is the number of characters to be printed from the word
a_i $(i = 1,2,....7)$ is the i^{th} character to be printed from the word
$m = 0$ if there are further words in the text
$= 1$ if the word is the last word in the text

2+

	Instruction	Notes
1.0	5.7 3 00	Search for Letter Shift Set text address in 3_m
	4+ [3] 72	
1	5+ [4] 72	
	0.1 1 00	Restore Assembly Link to B0.1
2	[0 1] 71	
(from 5+.1+)	17 4 02	Read character negatively, $-c_1$, to 4_m
3	1.2+ 4 60	Ignore ϕ
	31 4 42 4	c_1 - 31 to 4_c
4	1.2+ 4 60	Ignore Er
	(1) 4 41	c_1 - 30
5	1.2+ 4 60	Ignore CR
	(17) 4 41	c_1 - 13
6	1.2+ 4 60	Ignore LF
	(14) 4 43	c_1 - 27
7	1.7 4 61	LOOP STOP if not λ
	16 4 00	Read warning character from tape, c_2, to 4_c

3+

	Instruction	Notes
2.0	(15) 4 43	Search for L, N or Z c_2 - 15
	1.7+ 4 60	Ignore Er
1	16 2 00	Read next character from tape c_3 to 2_c
	(15) 2 43	c_3 - 15
2	2.1 2 60	Ignore Er
	(1) 2 43	c_3 - 16
3	2.3 2 61	LOOP STOP if not ϕ
	(1) 4 41	c_2 - 14
4	2.7+ 4 60	Jump if warning character is N
	(14) 4 43	c_2 - 28
5 (to 5+.5+)	4.5+ 4 60	Jump if L
	(18) 4 41	c_2 - 10
6	2.6 4 61	LOOP STOP if not N, L, or Z
	0 0 77	77-stop for Z
7	1.2+ 0 60	Return to search for next λ
	3 1 00	Text address to 1

Address	Order (4+)	Comment
		Read and Store Text
3.0	[1+ 1] 71	Write text address in index
	34 1 00	Step index address
	3.0 1 11	
1	0 7 00	Clear 7
2	(7) 5 40	Set character count, n, in 5
	3.4 4 60 2	Jump to { 3.4 before first character; 4.2 after $\phi\phi$ (last character) }
3	2.7 2 00	6 to the last 7 bits of 2_m
	(0) 4 40 4	Last character to 4_c, clear 4_m
4	17 4 01	Add next character from tape, c_3, into 4_m
	4 6 00	Add last character from 4_c to 7
5	(5) 0 55	Build up text word in 7
	3.2+ 5 67	Count a maximum of 7 characters
6	3.6+ 4 60 2	Jump if last text word, to mark
	(7) 7 46	Set n = number of characters in word (Normally 7)
7	7 1 00	
	[0 0] 71 3	Write text word to store

Address	Order (5+)	Comment
		End of Text
4.0	35 3 01	Step text address by 0.1
	3.1+ 1 62	Jump if more words of text to be read
1	5.7 3 10	Reset Transfer Address in 5.7_m
	1.2+ 0 60	Return to search for next text (to 2+.2+)
2	5 6 00	Character count = 7-n to X6
	0	
3	(5) 7 53	Shift text down to correct position
	4.3 5 67	
4	6 7 06	Set 7-n at end of X7
	32 7 06	Mark last text word
5	3.6+ 0 60	
	(1) 0 74	Switch to 2nd Tape Reader (from 3+.5)
6	[0 0] 70	Plant Initial Orders Link
	4.7 1 10	
7		Overwritten by Link

T 2+.1
J 560.0+ − 1+.1
L

DOUBLE LENGTH INPUT

A double-length fraction may be regarded as being of the form $\pm 0.x$, where x represents a string of 23 decimal digits. If there are fewer than 23 digits, the remainder are treated as zeros; if more than 23 digits, they are ignored.

The sign of the fraction is stored and five is subtracted from its first decimal digit on input. The 23 digits are then read to form an integer, I, in the range

$$-5.10^{22} < I < +5.10^{22}$$

This integer may be converted to the required fraction, F, since

$$F = 10^{-23} \times 2^{76} (5.10^{22}.2^{-76} + I)$$

$$= \frac{1}{2} + 10^{-23} \times 2^{76} \times I$$

It would be possible to store $2^{76} \times 10^{-23}$ and use the above relation to form the required fraction. If this were done double-length there would be a loss of accuracy and the equation has therefore been re-arranged as follows:

$$F = \frac{1}{2} + I - I\ (1 - 10^{-23}.2^{76})$$

$$= \frac{1}{2} + I - \frac{1}{4} I\ \{4(1 - 10^{-23}.2^{76})\}$$

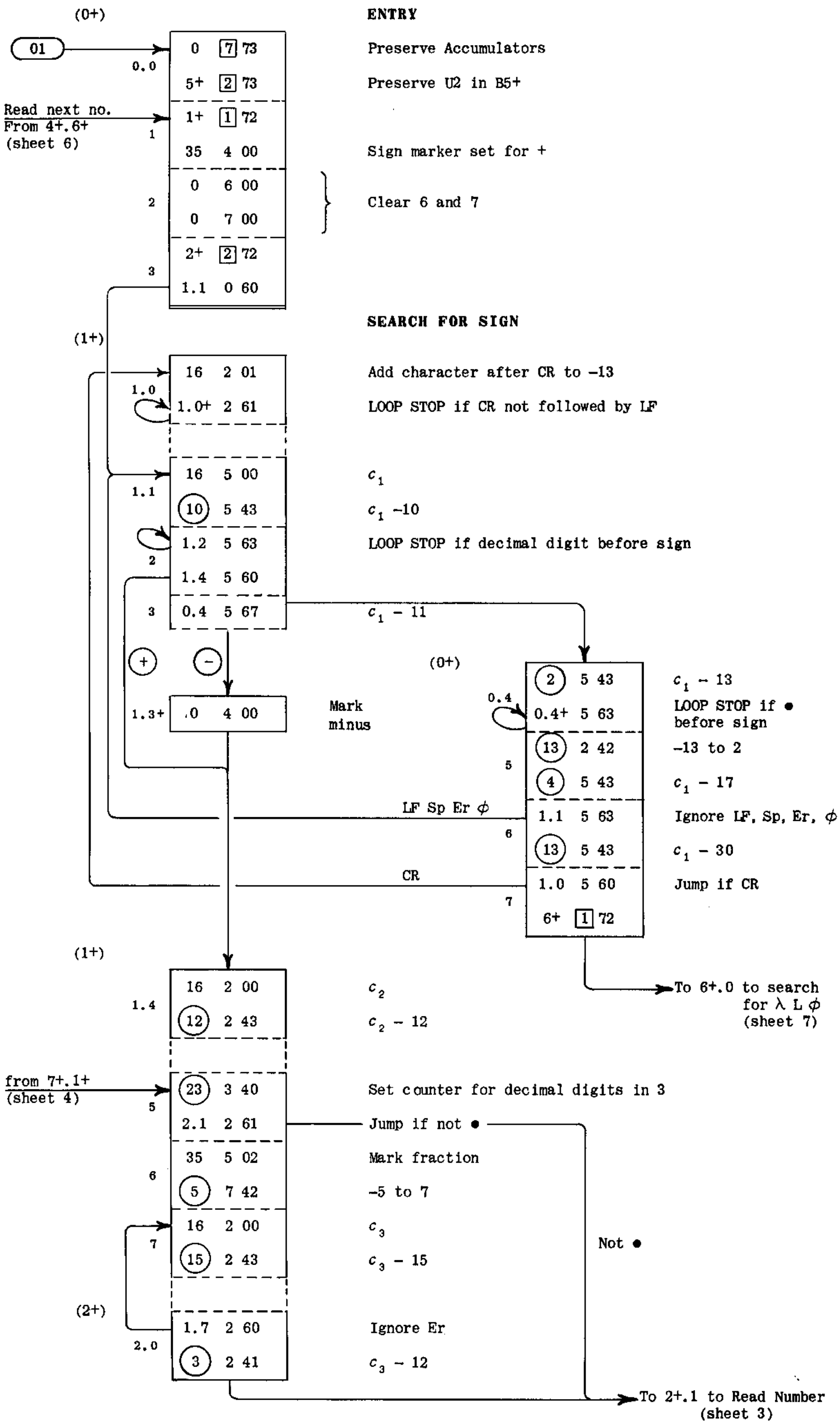
The double length fraction

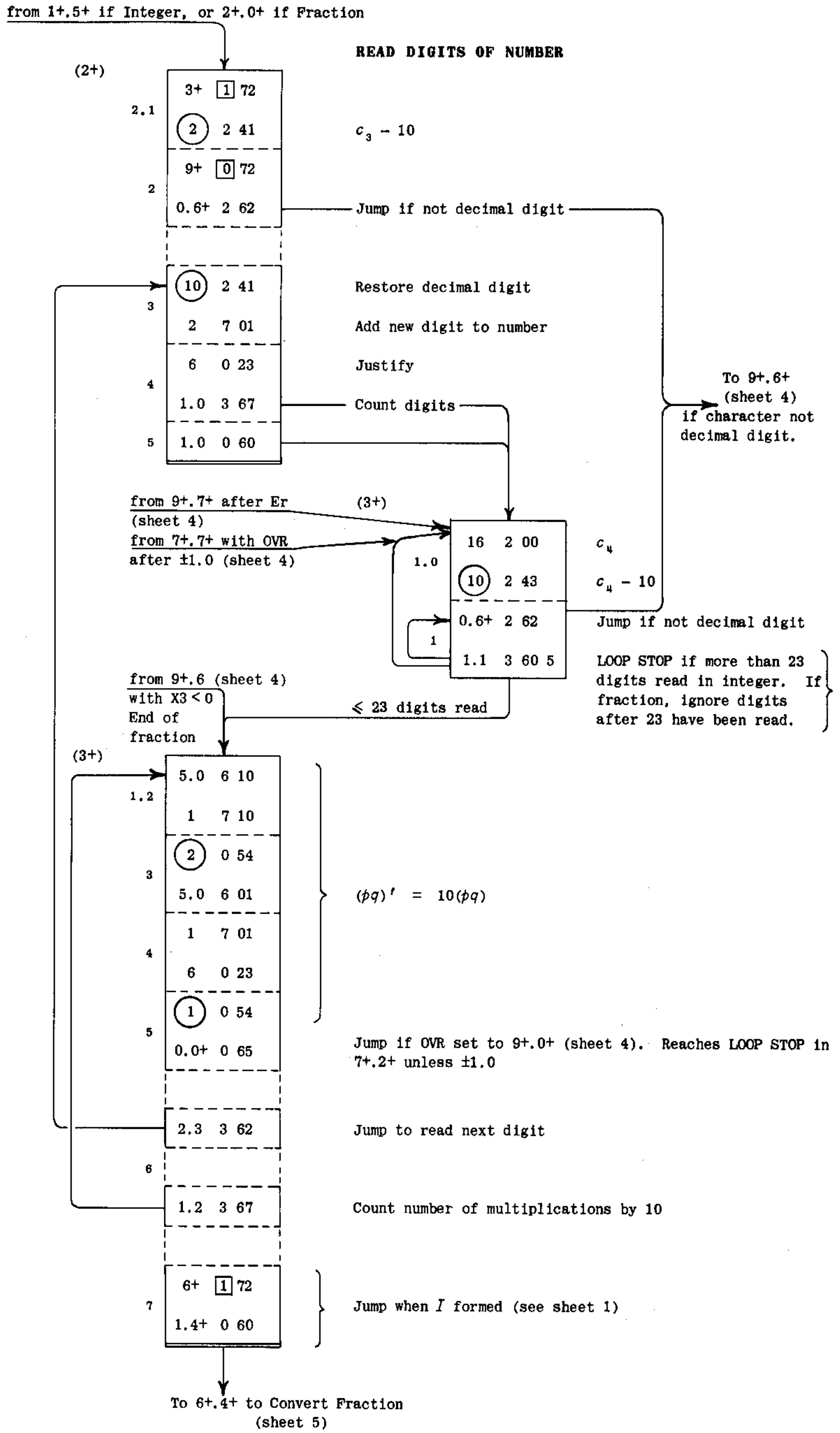
$$A + B.2^{-38} = 4(1 - 10^{-23}.2^{76})$$

is stored and this last relation is used to form F.

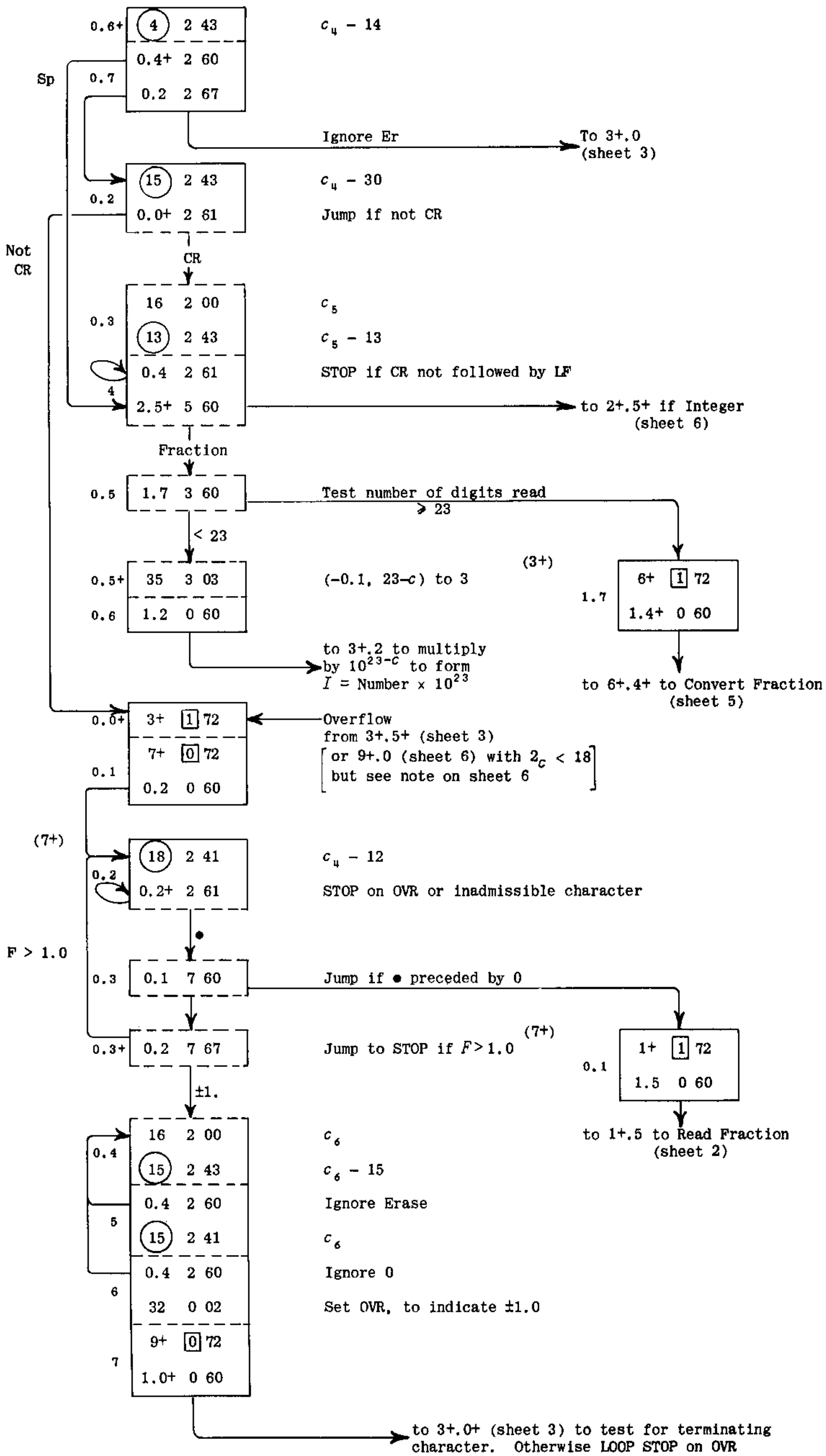
R 0	0 -0 3
100	- 28 -
0+	[0] 72
0.0	0 60
0+	0 00 0.
0	
0	
0+	0 00 0.

LN
D.L. INPUT A


ENTRY
(0+)
01
0.0
0 7 73
Preserve Accumulators
5+ 2 73
Preserve U2 in B5+
Read next no.
From 4+.6+
(sheet 6)
1
1+ 1 72
35 4 00
Sign marker set for +
2
0 6 00
0 7 00
Clear 6 and 7
3
2+ 2 72
1.1 0 60
SEARCH FOR SIGN
(1+)
1.0
16 2 01
Add character after CR to −13
1.0+ 2 61
LOOP STOP if CR not followed by LF
1.1
16 5 00
c1
10 5 43
c1 −10
2
1.2 5 63
LOOP STOP if decimal digit before sign
1.4 5 60
3
0.4 5 67
c1 − 11
+
−
1.3+
.0 4 00
Mark minus
(0+)
0.4
2 5 43
c1 − 13
0.4+ 5 63
LOOP STOP if ● before sign
5
13 2 42
−13 to 2
4 5 43
c1 − 17
LF Sp Er φ
6
1.1 5 63
Ignore LF, Sp, Er, φ
13 5 43
c1 − 30
CR
7
1.0 5 60
Jump if CR
6+ 1 72
To 6+.0 to search for λ L φ (sheet 7)
(1+)
1.4
16 2 00
c2
12 2 43
c2 − 12
from 7+.1+
(sheet 4)
5
23 3 40
Set counter for decimal digits in 3
2.1 2 61
Jump if not ●
6
35 5 02
Mark fraction
5 7 42
−5 to 7
7
16 2 00
c3
15 2 43
c3 − 15
Not ●
(2+)
2.0
1.7 2 60
Ignore Er
3 2 41
c3 − 12
To 2+.1 to Read Number (sheet 3)


from 1+.5+ if Integer, or 2+.0+ if Fraction
READ DIGITS OF NUMBER
(2+)
2.1
3+ [1] 72
(2) 2 41
c_3 – 10
2
9+ [0] 72
0.6+ 2 62
Jump if not decimal digit
3
(10) 2 41
Restore decimal digit
2 7 01
Add new digit to number
4
6 0 23
Justify
1.0 3 67
Count digits
5
1.0 0 60
To 9+.6+
(sheet 4)
if character not
decimal digit.
from 9+.7+ after Er
(sheet 4)
from 7+.7+ with OVR
after ±1.0 (sheet 4)
(3+)
1.0
16 2 00
c_4
(10) 2 43
c_4 – 10
0.6+ 2 62
Jump if not decimal digit
1
1.1 3 60 5
LOOP STOP if more than 23
digits read in integer. If
fraction, ignore digits
after 23 have been read.
from 9+.6 (sheet 4)
with X3 < 0
End of
fraction
≤ 23 digits read
(3+)
1.2
5.0 6 10
1 7 10
3
(2) 0 54
5.0 6 01
$(pq)' = 10(pq)$
4
1 7 01
6 0 23
5
(1) 0 54
0.0+ 0 65
Jump if OVR set to 9+.0+ (sheet 4). Reaches LOOP STOP in
7+.2+ unless ±1.0
2.3 3 62
Jump to read next digit
6
1.2 3 67
Count number of multiplications by 10
7
6+ [1] 72
1.4+ 0 60
Jump when I formed (see sheet 1)
To 6+.4+ to Convert Fraction
(sheet 5)

READ NON-DIGITAL CHARACTERS



(4+)

1.7	7.5 1 11 4. 95 7 55 2	= A { most significant half of double length number for multiplication of fractions

(7+)

0.0	54 5 65 6. 98 1 22 2	= B { least significant half of double length number for multiplication of fractions

from 3+.7+ (sheet 3)

CONVERT FRACTION

(6+)

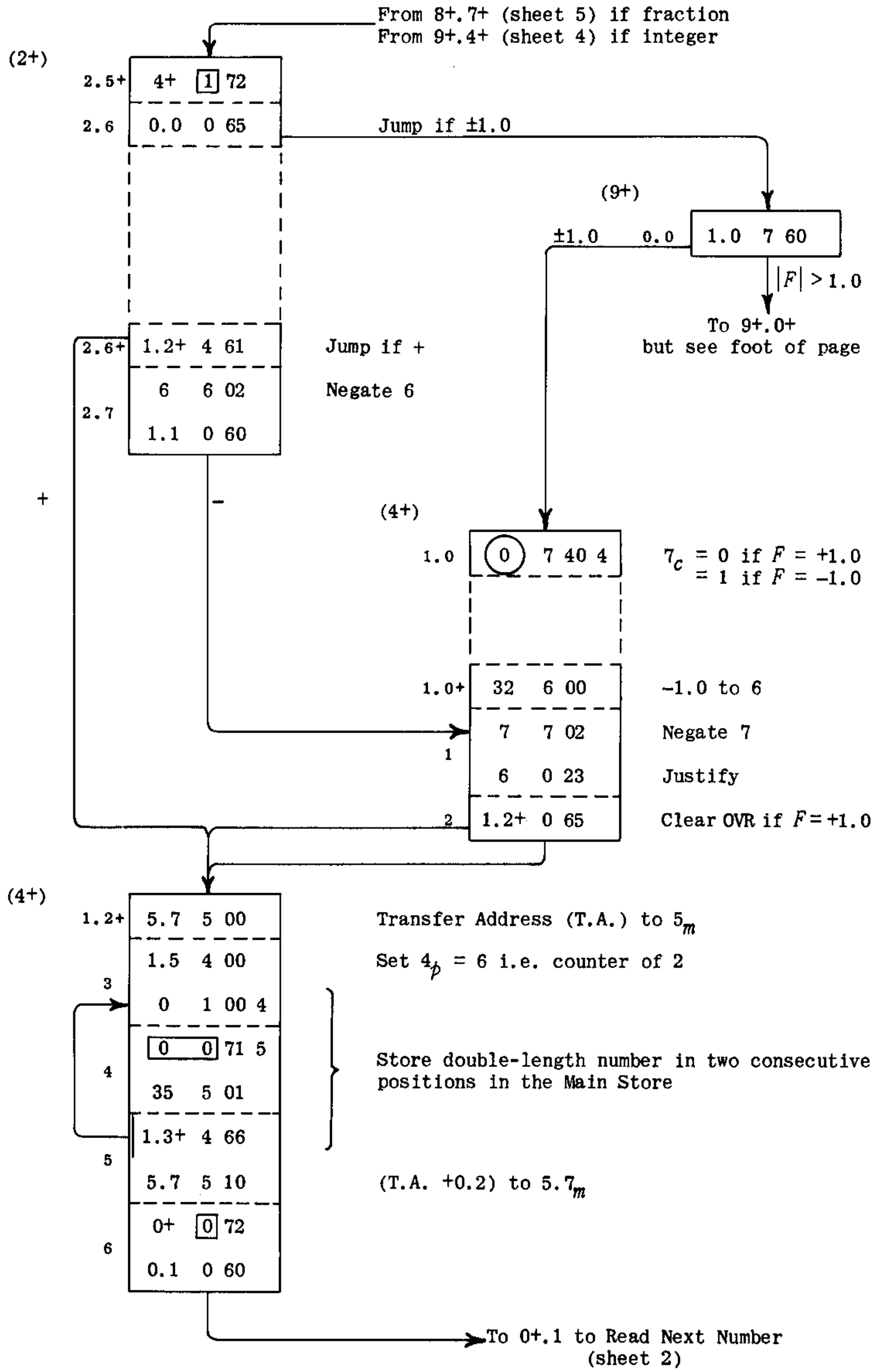
1.4+	1.0 6 10	Store double length number in 1.0 and 1.1
	1.1 7 10	
5	7+ 0 70	Read B to 1
	1.1 1 20	$B.q.2^{-38}$ to PQ
6	8+ 2 72	
	6 7 00	
7	0 6 00	

(8+)

	1.0 1 22	$B(p + q.2^{-38})$ to PQ
2.0	4+ 7 70	A to 1
	1.1 1 22	$[Aq + B(p + q.2^{-38})].2^{-38}$ to PQ
1	6 7 00	
	0 6 00	
2	6 0 23	
	1.0 1 22	$(A + B.2^{-38})\ (p + q.2^{-38})$ to PQ
3	(2) 7 41	$\frac{1}{4}\ (A + B.2^{-38})\ (p + q.2^{-38})$, rounded up, to PQ
	6 0 23	
4	(2) 0 55	
	1.0 6 04	$(p + q.2^{-38}) - \frac{1}{4}\ (A + B.2^{-38})(p + q.2^{-38}) + \frac{1}{2}$
5	1.1 7 04	$= (p + q.2^{-38}).2^{76}.10^{-23} + \frac{1}{2}$ to PQ
	6 0 23	
6	33 6 01	
	2+ 2 72	
7	2.5+ 0 60	

To 2+.5+ to Store Number (sheet 6)

STORE NUMBER



Note: The order in 9+.0 is intended to detect $|F| > 1.0$ but this should previously have been detected in 3+.5+

SEARCH FOR WARNING CHARACTER L

from 0+.7+ (sheet 2)

(6+)

Block	Instruction	Comment
	(3) 5 41	c_1 − 27
1.0	1.0+ 5 61	LOOP STOP if not λ
1	16 5 00	c_5
	(28) 5 43	c_5 − 28
2	17 5 01	(c_6, c_5 − 28)
	1.2+ 5 61	LOOP STOP if λ not followed by L φ
3	0 [7] 72	Restore Accumulators
	5+ [2] 72	Restore U2
4	1.5 1 10	Plant LINK
	1.0 6 10	Does no harm, used in conversion
5		Overwritten by LINK. Obey LINK

EXIT

FLOATING POINT INPUT

	R 0 0 -0 2	
	101 - 28 -	
01	0+ [0] 72	
	0.0 0 60	
02	0+ [0] 72	
	0.0+ 0 60	

LN
F.P. READ

R 8 2 -0 1	Calls for P.P.01
101 - 02 -	
R 8 3 -0 1	
101 - 02 -	
R 8 3 -0 1	
101 - 02 -	

Entry / exit	Addr	0+	Comment
(01) →	0.0	0 5 00	**Search for sign** Mark to read only one number
(02) →		[0 0] 71	Store LINK
		0 2 00	Clear 2
from (9+.7+ / 5+.7+) →	1	0 0 20	Clear 6 and 7
from 2+.1+ →		1+ [1] 72	
from 1+.2 →	2	16 1 00	Character from tape, c_1, to 1
		(17) 1 43	Ignore ϕ, Sp, Er, LF
to 1+.0 ←	3	1.0 1 62	
		(4) 1 41	
	4	0.2+ 1 62	
		(3) 1 41	$c_1 - 10$
	5	0.7 1 60	Jump if +
to 1+.2+ ←		1.2+ 1 67	Jump if not −
	6	32 1 00	Mark −
		3+ [0] 72 7	To 3+.0 to read number
to 3+.0 or 4+.0 ←	7	0.0 0 60	or 4+.0 to read exponent (after Q)

Search for Warning Character

Block 1+ (from 0+.3+ to 1.0)

Word	Instruction	Comment
1.0	(13) 1 43	c_1 − 30
	1.3 1 61	Jump if not CR
1	16 1 00	Character from tape, c_2, to 1
	(13) 1 43	c_2 − 13
2	0.2+ 1 60	Jump if LF (to 0+.2+)
	1.2+ 0 60	LOOP STOP for punching error.
3	(3) 1 41	c_1 − 27
	1.2+ 1 61	Jump if not λ
4	2+ [0] 72	
	16 1 00	Warning character from tape, c_3, to 1
5	(15) 1 43	Ignore Er
	1.4+ 1 60	
6	16 3 00	Search for ϕ after warning character
	(15) 3 43	Ignore Er
7	1.6 3 60	
	0.2 0 60	

Interpret Warning Character

Block 2+

Word	Instruction	Comment
0.0	[0 0] 70	Read LINK } After warning character L
	0.1 1 10	Plant LINK } After warning character L
1	0 0 77	77 stop after warning character Z } Over-written by LINK
	0+ [0] 72	Return to Number Read } Over-written by LINK
2	1.2+ 3 67	Jump to LOOP STOP if not ϕ after warning character
	(13) 1 43	c_3 − 28
3	0.0 1 60	Jump if L
	(18) 1 41	c_3 − 10
4	0.1 1 60	Jump if Z
	(7) 1 43	c_3 − 17
5	0.6+ 1 61	Jump if not Q
	34 7 00	Set marker in 7
6	0.1+ 0 60	Jump if Q
	(3) 1 41	c_3 − 14
7	0.7 1 61	LOOP STOP if not L, Z, Q or N
	5+ [1] 72	to 5+.0 after N

Read Number

3+

from 2+.7+

Address	Order	Comment
0.0	0 4 00	Clear 4 (Used to store argument)
	6+ [1] 72	
1	0.0 0 10	Clear 0.0 (Used to store exponent)
	0.1 2 10	Block exponent to 0.1
2	16 3 00	Character from tape, c_3, to 3
	3 7 00	c_3 to 7
3	(10) 3 43	c_3 − 10
	1.0+ 3 62	Jump if not a decimal digit
4	(10) 3 40	+ 10 to 3_c
	0.0 3 22	Multiply by 10 and add next digit
5	0.7 6 60	Jump if number does not OVR
	0.2 4 63	Jump to ignore overflow digits after ●
6	(1) 4 40	Prepare to increase exponent if overflow digits before ●
	0.7+ 0 60	
7	0.0 7 10	Store number
	0.1 4 11	Reduce exponent by 1 for each digit after ●

6+

Address	Order	Comment
	0.2 0 60	
1.0	(5) 3 43	c_3 − 15
	0.2 3 60	Ignore Er
1	(3) 3 41	c_3 − 12
	1.3+ 3 61	Jump if not decimal point
2	(1) 4 42	Mark decimal point in 4
	0.2 0 60	
3	(2) 3 43	c_3 − 14
	1.7 3 60	Jump if Sp
4	(16) 3 43	c_3 − 30
	1.5 3 61	LOOP STOP if not CR LF.
5	16 3 00	Character from tape, c_4, to 3
	(13) 3 43	c_4 − 13
6	1.5 3 61	Jump to LOOP STOP if not LF after CR
	7+ [1] 72	
7	1.0 0 60	

to 7+.0

4+ Read Decimal Exponent

from 0+.7+

Address	Instruction	Comment
0.0	0 2 00	Clear 2
	16 3 00	Character from tape, c_5, to 3
1	3 7 00	c_5 to 7
	(10) 3 43	c_5 – 10
2	0.4+ 3 62	Jump if not a decimal digit
	(10) 3 40	+ 10 to 3_c
3	2 3 22	Multiply by 10 and add next digit
	7 2 00	Copy exponent to 2
	0.0+ 0 60	
4	(5) 3 43	c_5 – 15
	0.0+ 3 60	Ignore Er
5	(1) 3 41	c_5 – 14
	5+ [1] 72	
6	1.6 3 60	Jump if Sp
	(16) 3 43	c_5 – 30
7	1.4 0 60	

5+ Print Name; Set Exponent

from 2+.7+

Address	Instruction	Comment
1.0	(0) 3 40 3	Last character to 3_c; clear 3_m
	17 3 10	Print ϕ on entry, then last character.
1	17 3 01	Next character to 3_m, leave 3_c
	1.0 3 60 1	Jump if last 2 characters were $\phi\phi$
2	1.0 1 61	Jump except at first entry
	34 1 00	Set 7 in 1_m
3	35 1 03	(Set 7 in 1_m)
	1.0 0 60	
4	1.4 3 61	LOOP STOP if not CR or CR LF
	16 3 00	Character from tape, c_6, to 3
5	(13) 3 43	c_6 – 13
	1.4 3 61	Jump if not LF after CR
6	1.7 1 62	Jump if marker is positive
	2 2 02	Negate block exponent
7	0+ [0] 72	
	0.1+ 0 60	

to 0+.1+

Convert decimal exponent in 0.1 to binary exponent in X3

7+

from 6+.7+

Word	Instruction	Comment
0	0.0 6 00	Argument to 6
	0.1 4 00	Exponent to 4
1	8+ [0] 72	
	0.7 6 60	Jump if argument = 0
2	(38) 3 40	Set binary exponent
	(38) 3 56	NORMALIZE
3	0.4 4 60	Jump if decimal exponent = 0
	1.6 4 63	Jump if decimal exponent < 0
4	(1) 4 43	Subtract 1 from decimal exponent
	(4) 3 41	Add 4 to binary exponent
5	0.0 6 20	Multiply argument by (10.2^{-4})
	1.2+ 0 60	Jump
6	(1) 4 41	Add 1 to decimal exponent
	(3) 3 43	Subtract 3 from binary exponent
7	0.1 6 20	Multiply argument by $(2^3.10^{-1})$
	1.2+ 0 60	Jump

8+

Word	Instruction	Comment
0.0	+.625	$= 10.2^{-4}$
1	+.8	$= 2^3.10^{-1}$
2	+0	+ P.P.01 = -2^{n-1}
3	+0	+ 2(P.P.01) = -2^n
4	0.5+ 1 62	Jump if marker is positive
	6 6 02	Negate argument
5	(2) 3 56	Re-normalize
	0.2 6 03	Add in round-off constant = 2^{n-1}
6	(2) 3 56	Re-normalize
	3 7 00	Binary exponent to 7
7	9+ [1] 72	
	1.3+ 6 60	Jump if argument is zero

to 9+.3+

Pack number, store and count

9+

Entry	Block	Instruction	Description
from 8+.7+ argument ≠ 0		1.1 7 62	Jump if binary exponent ⩾ 0
		7 7 02	Negate binary exponent
1.0 →	1	0.2 7 01	Subtract 2^{n-1} from X7
		1.1+ 7 62	LOOP STOP if C(7) ⩾ 0, i.e. exponent overflows
	2	0.3 6 05	Clear last n digits of 6
		0.2 3 03	Add 2^{n-1} to exponent
	3	3 6 01	Pack
from 8+.7+ argument = 0		1.6 5 60	Jump if C(5) = 0, i.e. Cue 01
	4	6 1 00	
		[0 0] 71 5	Write number into Main Store
	5	35 5 01	Step 5_m
		1.7 5 67	Count numbers
	6	[0 0] 70	Plant LINK
		1.7 1 10	
	7	0+ [0] 72	Overwritten by LINK
to 0+.1+		0.1+ 0 60	Jump to read next number (overwritten by LINK)

Optional Parameter List

	R 0 0 -0 1	Title
	101 - 06 -	
01	-256	P.P.01 = -2^8 (i.e. $n = 9$)

L

READ IN TABLES OF NUMBERS

Label	Order
	R 0 0 -0 2
	102 - 28 -
01	0+ [0] 72
	0.0 0 60
02	0+ [0] 72
	0.5 0 60

LN
INPUT TABLES

Order	Comment
R 0 2 -0 1	Call for P.P.01
102 - 02 -	

Entry	Address	Order	Comment
		[0+]	
01 →	0.0	0.2 7 00	7_m = *B.P* = Address of 1st word of Index
		0 [4] 72 7	Block *B* to U4
	1	4.0 7 01 7	(*B.P*, *t*) to 7
		0.6+ 0 60	
	2	+0	+P.P.01 = (*B.P*, 0)
from I.0s Input →	3	0+ [0] 72	LINK for Initial Orders, obeyed in 0.3
		0.7 7 67	Count Tables
	4	6 1 00	Plant LINK and exit
		0.5 1 10	
02 →	5	(0.2 7 00	Overwritten by LINK
		(1) 7 41)	(*B.P*–1, 1) to 7
	6	35 7 03	
		1 6 00	Store LINK in 6
	7	35 7 01	Step address
		1+ [1] 72	

	1+	
	0 [4] 72 7	Block to contain Index to U4
1.0	5.7 1 00	Bring Transfer Address to start of next available block and set X5 = (512.0, 0)
	35 1 03	
1	33 5 00	
	1.2 1 66	
2	5.7 1 10	
	4.0 1 11 7	Store Table Address in Index
3	0 [4] 73 7	Write up Index into correct block
	0.3 1 00	Set LINK for Initial Orders
4	0	
	10 [0] 72 5.	Optional Stop before Reading Table
5	0.0 0 60	Enter Initial Orders Input
	R 0 0 -0 1	**Optional parameter list**
	102 - 06 -	
	2 - 00 0.	
6	0	P.P.01

L

DOUBLE LENGTH INPUT

	R 0 0 -0 1
	105 - 28 -
	0+ [0] 72
01	0.0 0 60

LN
DL INPUT

Address	Instruction	Comment
	0+	**Search for start of number**
(01) →	1+ [1] 72	
0.0	16 7 00	1st character, c_1, from tape to 7_c
1	(10) 7 43	$c_1 - 10$
	0.1+ 7 63	LOOP STOP if a decimal digit
2	(2) 7 43	$c_1 - 12$
	0.7+ 7 63	Jump if + or −
3	0.3 7 60	LOOP STOP if decimal point
	(5) 7 43	$c_1 - 17$
4	0.0+ 7 63	Ignore Er, Sp, LF or ɸ before sign
	(13) 7 43	$c_1 - 30$
5	0.0+ 7 60	Ignore CR before sign
	0.5+ 0 60	LOOP STOP for punching error
6	+0	Unused
7	0	
	0 6 00	Clear 6

1+

Read integral part of number to 6

Word	Instruction	Comment	Note
1.0	(2) 7 41	c_1 − 10	Overwritten by sign marker
	1.0 7 10		
1	16 7 00	Read next character, c_2	Overwritten by integral part
	(10) 7 43	c_2 − 10	
2	1.6 7 62	Jump if not decimal digit	Overwritten by LINK
	(10) 7 41	c_2	
3	(1) 6 50	Multiply by 10 and add new digit	Overwritten by fractional part
	6 7 01		
4	(2) 6 50		Overwritten by +0.1
	7 6 01		
5	1.1 0 64		
	1.5+ 0 60	LOOP STOP if OVR	
6	(5) 7 43	c_2 − 15	
	1.1 7 60	Ignore Er	
7	2+ [0] 72		
	0.0 0 60		

2+

Prepare to read fractional part

Word	Instruction	Comment
0.0	1.1 6 10	Integral part to 1.1
	1.2 1 10	Plant LINK
1	1.3 0 10	Clear 1.3
	(3) 7 41	c_2 − 12
2	0.4 7 60	Jump if decimal point
	0	
3	4+ [0] 72	
	0.0 0 60	to 4+.0
4	0.7 1 00	+0.1 to 1 and 1.4
	1.4 1 10	
5	3+ [0] 72	
	0.0 0 60	to 3+.0
6	+0	Unused
7	+0.1	

3+ (from 2+, 5+ to 0.0)

Read fractional part of number to 1.3

Address	Instruction	Comment
0	16 7 00	Read next character, c_3
	10 7 43	$c_3 - 10$
1	0.4+ 7 62	Jump if not decimal digit
	10 7 41	c_3
2	1 7 20	Multiply digit by $(0.1)^n$
	1.3 7 11	Accumulate number in 1.3
3	1.4 1 21	$(0.1)^{n+1}$ to 1
	1 6 10	
	0.0 0 60	
4	5 7 43	$c_3 - 15$
	0.0 7 60	Ignore Er
5	15 7 43	$c_3 - 30$
	4+ 0 72	
6	0.0+ 0 60	
7	+0	Unused

4+ (from 2+, 3+ to 0.0)

Complete number and exit

Address	Instruction	Comment
0	18 7 43	$c_2 - 30$
	0.0+ 7 61	LOOP STOP if not CR after last decimal digit
1	16 7 00	Read character after CR, c_4
	13 7 43	$c_4 - 13$
2	0.2 7 61	LOOP STOP if not LF after CR
	1.0 1 00	Sign marker to 1
3	0.5+ 1 60	Jump if +
	1.1 6 02	Negate Integral part in 6
4	1.3 7 02	Negate Fractional part in 7
	6 0 23	Justify
	1.2 0 60	Jump to obey LINK
5	1.1 6 00	Restore Integral part to 6
6	1.3 7 00	Restore Fractional part to 7
	1.2 0 60	Jump to obey LINK

L

SHORT NUMBER READ

This subroutine reads a single length mixed number, N, from tape and leaves it in the form

$$p' = N.10^m.2^{-38} \qquad q' = 10^m.2^{-38}$$

where m is the number of figures after the decimal point.

	R 0 0 -0 3
	116 - 28 -
01	0+ [0] 72
	0.4 0 60
02	0+ 0 00 0.
	0
03	0
	0+ 0 00 0.

LN
SHORT NUMBER READ

0+

Link	Word	Order	Comment
from 2+.0+		0.1 5 67	Jump if not LF
	0.0	0.6+ 7 63	Jump if in α or β-search
to 1+.4		1.4 5 67	Jump if not Sp
from 2+.7	1	0.6+ 7 63	Jump if in α or β-search
		5 6 12 1	Negate 6 if N negative
	2	0.3+ 7 61	Jump if $m \neq 0$
		(1) 7 40	Put 7 = 1.2^{-38} for $m = 0$
to 2+.7+	3	2.7+ 0 65	Jump to LOOP STOP if OVR set
01		1+ [1] 72	Overwritten by LINK
	4	2+ [2] 72	
		0.4 1 10	Plant LINK in 0.4
	5	32 7 00	-1.0 to 7
		0 6 00	Clear 6
	6	0 1 00	Clear 1
to 1+.4+		1.4+ 0 60	Jump to read character from tape
from 1+.5+	7	(10) 5 41	c_1 to 5

Note that q'(= 2.10^r) will not overflow into p' on the order 1 0 54 in 1+.0 unless OVR has already been set by the order 2 5 50 in 1+.2+.

1+

Link	Address	Order	Notes
	1.0	1 0 54	$p' = 2p$ and $q' = 2q$. Clear sign of 7
		6 5 01	Form $N.10^m.2^{-38}$ in 6
	1	2 6 50	
		5 6 01	
	2	7 5 00	Form $10^m.2^{-38}$ in 7
		2 5 50	
		5 7 01	
	3	1.4+ 1 62	Jump unless digit read in α-search
from 0+.1		2.3+ 5 67	Jump if not Erase
from 0+.7	4	16 5 00	Read character from tape, c_1, to 5
		10 5 43	c_1-10
to 0+.7+	5	0.7+ 5 63	Jump if decimal digit
		1.7+ 5 60	Jump if +
	6	2.0+ 5 67	Jump if not –
		35 1 00	Set 1_p = 1 to mark negative number
	7	32 7 06	Change sign digit of 7 for + or –

2+

Link	Address	Order	Notes
		2.2+ 7 62	Jump unless sign read in middle of number
to 0+.0	2.0	0.0 5 67	Jump if not •
		2.2 7 63	Jump if in α or β-search
	1	2.1+ 7 61	LOOP STOP if two • in one number
		1 7 40	2^{-38} to 7 after •
	2	35 1 05	Retain 1_p (= 1 if – read, otherwise 0)
		1.4+ 0 60	Jump to read next character
	3	2.5 5 67	Jump if not ϕ
		32 1 00	-1.0 to 1 to mark α-search
	4	1.4+ 7 63	Jump unless ϕ read in middle of number
		14 5 43	c_1 – 30
	5	2.5+ 5 61	LOOP STOP on inadmissible character 0.1+ 5 6
		16 5 00	Read character from tape, c_2, to 5
	6	13 5 43	c_2 – 13
to 0+.1+		0.1+ 5 60	Jump if CR followed by LF
from 0+.3+	7	2.7+ 0 60	LOOP STOP

GENERAL SINGLE-LENGTH NUMBER READ

This subroutine reads a single-length number, N, from tape:-

$p' = N.10^m.2^{-38}$ and $q' = m.2^{-38}$ where m is the number of figures punched after the decimal point.

	R 0 0 -0 3
	121 - 28 -
01	0+ [0] 72
	0.5+ 0 60
02	0+ 0 00 0.
	0
03	0
	0+ 0 00 0.

LN
NUMBER READ A

		0+	Exit and Entry
from 3+.6+ →	0.0	6 7 12 3	Negate 7 if $3_m = 1$
		1.0 7 10	$N.10^m.2^{-38}$ to 1.0
	1	1.1 4 10	$m.2^{-38}$ to 1.1
		0 [7] 72	Restore Accumulators
	2	1.0 6 00	$N.10^m.2^{-38}$ to 6
		1.1 7 00	$m.2^{-38}$ to 7
	3	0.4+ 1 63	Jump if LINK is *go* order pair
		(25) 1 50	EXIT by Computing Store Link
←		0 0 60 1	
	4	1.7 1 10	Plant LINK
←		1.7 0 60	Jump to LINK
(01) →	5	0 [7] 73	Preserve Accumulators
	6	1+ [1] 72	
		1.3 5 00	Set 0.4 in 5_m
	7	7.5 0 10 5	Clear 1, 2, 3 and 4
		0.7 5 66	

Search for First Character of Number

Block 1+

Address	Instruction	Comment
	4+ [0] 72	
1.0	0 5 00	Set β – search indicator
1	16 7 00	Read character from tape, c_1, to 7
	(10) 7 43	c_1 – 10
2	1.4 7 62	Jump if not a decimal digit
	1.2+ 5 61	LOOP STOP on decimal digit in α – search
3	(10) 7 41	Decimal Digit in β – search
	1.6+ 0 60	Jump to MAIN INPUT with c_1 in 7
4	1.6 7 60	Jump if +
	(1) 7 43	c_1 – 11
5	0.0 7 61	Jump if not –
	35 3 00	Set 3_m = 1 if – read
6	32 2 00	Set –1.0 in 2 if + or – read
	2+ [0] 72	
7	3+ [1] 72	
	0.3+ 0 60	To MAIN INPUT (to 2+.3+)

Block 4+

Address	Instruction	Comment
0.0	(1) 7 43	c_1 – 12
	0.2 7 61	Jump if not •
1	35 1 00	Set 1_m = 1 if • read
	1.6+ 0 60	Jump to MAIN INPUT
2	(3) 7 43	c_1 – 15 ($c_1 \geqslant 13$)
	1.0+ 7 63	Return to β – search if LF or Sp
3	1.1 7 60	Ignore Er in α or β – search
	(1) 7 43	c_1 – 16
4	(1) 5 40	Set α – search indicator
	1.1 7 60	Jump to α – search on ϕ
5	(14) 7 43	c_1 – 30
	0.7+ 7 61	Jump on inadmissible character
6	16 7 00	Read character from tape, c_2, to 7
	(13) 7 43	c_2 – 13
7	1.0+ 7 60	Return to β – search on CR LF
	0.7+ 0 60	LOOP STOP on inadmissible characters

PEGASUS LIBRARY PROGRAMME

Main Input

2+

Pair	Instruction	Comment
0.0	(10) 5 41	c_3 to 5
	1.7 7 20	Multiply number by 10
1	5 7 01	Add new digit c_3
	6 0 23	Justify
2	0.2 6 61	LOOP STOP if 6 ≠ 0
	(0) 4 41 1	Add 1 to 4 for each digit after :●
3	0 2 00	Clear 2 if last character not Sp or Er
	16 5 00	Read character from tape, c_3, to 5 (from 1+.7+)
4	(10) 5 43	c_3 – 10
	0.0 5 63	Jump if decimal digit
5	(4) 5 43	c_3 – 14
	1.0 5 61	Jump if not Space
6	0.3+ 2 63	Ignore Spaces after sign
	1.6 2 61	Jump to EXIT on 2nd internal Sp.
7	(1) 2 40	Set 1 in 2 on 1st internal Sp.
	0.3+ 0 60	Jump to read next character

3+

Pair	Instruction	Comment
1.0	(2) 5 41	c_3 – 12
	1.2+ 5 61	Jump if not :●
1	1.1 1 61	LOOP STOP if 2nd :●
	35 1 00	Set 1_m = 1 if :● read
	0.3 0 60	Jump to read next character
2	(3) 5 43	c_3 – 15
	0.3+ 5 60	Ignore Er
3	(15) 5 43	c_3 – 30
	1.5+ 5 61	Jump to LOOP STOP on inadmissible character
4	16 5 00	Read character from tape, c_4, to 5
	(13) 5 43	c_4 – 13
5	1.5+ 5 61	LOOP STOP on inadmissible character
	0+ [0] 72	
6	0.0 0 60	Jump to EXIT (to 0+.0)
7	+10	Overwritten by LINK

SIGNED £.S.D. READ TO PENCE

This subroutine reads from tape a sterling sum $a.b.c$ and leaves it in X1 as $(240a + 12b + c)$. The sum N must not exceed £1,145,324,612.5.3d in modulus. i.e. $| 240a + 12b + c | < 2^{38}$

	R 0 0 -0 3
	142 - 28 -
01	0+ [0] 72
	0.5+ 0 60
02	0+ 0 00 0.
	0
03	0
	0+ 0 00 0.

LN
SIGNED £.S.D. READ

Note	Block	Order	Comment
		0+	**Exit and Entry**
from 5+.6+ →	0.0	6 7 12 4	Negate X7 if number is negative
		1.7 7 10	Store number in 1.7
	1	0 [7] 72	Restore Accumulators
jump to 3.1		0.3+ 6 63	Jump if *go* order pair LINK
	2	6 1 00	Add Computing Store link into N address of 0.4+
		(12) 1 52	
	3	0.4 1 11	
→		1.7 1 00	Number to 1
	4	1.7 6 10	Plant LINK in 1.7
		0 6 62	Jump if Computing Store Link
	5	1.7 0 60	Jump to LINK (*go* order pair)
01 →		0 [7] 73	Preserve Accumulators
	6	1+ [1] 72	
		0 2 00	Clear X2, X3 and X4
	7	0 3 00	
		0 4 00	

Search for first significant character

1+

Word	Order	Comment
1.0	2+ [0] 72	
	0 5 00	Set β- search indicator
1	16 7 00	Read character from tape, c_1, to 7
	(10) 7 43	c_1 − 10
2	1.4+ 7 62	Jump if not a decimal digit
	1.2+ 5 61	LOOP STOP on decimal digit in α-search
3	(10) 7 41	c_1
	(1) 2 40	+ 1 to 2_c
4	1.6+ 0 60	
	1.6+ 7 60	Jump if +
5	(1) 7 43	c_1 − 11
	0.0 7 61	Jump if not −
6	35 4 00	Marker in 4 for negative number
	3+ [0] 72	To read number
7	4+ [1] 72	To read number
	0.5 0 60	(to 3+.5)

2+

Word	Order	Comment
0.0	(1) 7 43	c_1 − 12
	0.2 7 61	Jump if not ●
1	32 3 00	Set X3 = −1.0 after first point
	1.6+ 0 60	
2	(3) 7 43	c_1 − 15
	1.0+ 7 63	Jump to β- search if LF or Sp
3	1.1 7 60	Ignore Er
	(1) 7 43	c_1 − 16
4	(1) 5 40	Set α- search indicator
	1.1 7 60	Jump to α- search if ϕ
5	(14) 7 43	c_1 − 30
	0.7+ 7 61	Jump to LOOP STOP for punching error
6	16 7 00	Read character from tape, c_2, to 7
	(13) 7 43	c_2 − 13
7	1.0+ 7 60	Jump to β- search after CR LF
	0.7+ 0 60	LOOP STOP for punching error

Read the digits of the Number

3+

	Instruction	Comment
0.0	(1) 2 40	+ 1 to 2_c after decimal digit
	(10) 5 41	c_3
1	(10) 6 40	+ 10 to 6_c
	6 7 20	$N' = x_6.N$ (x_6 = radix)
2	5 7 01	Add new digit to N
	6 0 23	
3	0.3 6 61	LOOP STOP if $N \geqslant 2^{38}$
	0.5 2 61	Jump after decimal digit
4	1.7 7 10	After ● store $20a$ or ($240a + 12b$)
	0 7 00	Clear X7
5 (from 1+.7+ and 5+.7+)	16 5 00	Read character from tape, c_3, to 5
	(10) 5 43	c_3 − 10
6	0.0 5 63	Jump if decimal digit
	(2) 5 43	c_3 − 12
7	1.4+ 5 61	Jump if not ●
	3 1 00	Point indicator to 1

4+

	Instruction	Comment
1.0	0 2 00	Clear 2 after point
	32 3 00	Set X3 = −1.0
1	(20) 6 40	+ 20 to 6_c
	0.1+ 1 60	Jump if first ●
2	35 3 00	(0.1, 0) to 3
	(12) 6 40	+ 12 to 6_c
3	1.7 7 01	$20a + b$ to 7
	0.1+ 1 63	Jump if second ●
4	1.4 0 60	LOOP STOP if third ●
	(2) 5 43	c_3 − 14
5	1.6 5 61	Jump if not Sp
	0.5 2 60	Ignore Sp before decimal digit
6	5+ [0] 72	
	0.0 0 60 (to 5+.0)	
7	+0	Number store } Overwritten by LINK

5+

Read Terminating Character

Block	Instruction	Description
0	0.3+ 5 60 3	Jump if Sp
	(1) 5 43	$c_3 - 15$
1	0.7 5 60	Jump if Er to ignore Er
	(15) 5 43	$c_3 - 30$
2	0.2 5 60 3	LOOP STOP after CR in a or b
	0.2+ 0 60	LOOP STOP on other punching errors
3	16 5 00	Read character from tape, c_4, to 5
	(13) 5 43	$c_4 - 13$
4	0.2+ 5 61	Jump if CR without LF or Sp in number
	1.7 7 01	$\mid 240a + 12b + c \mid$ to 7
5	0.6 0 64	Jump if OVR clear
	0.5+ 0 60	LOOP STOP on overflow
6	0+ [0] 72	To B 0+ to complete formation of number
	0.0 0 60	
7	3+ [0] 72	Return to read rest of number after reading Er
	0.5 0 60	

from 4+.6+ (entry at 0.0)

to 0+.0 (from 6)

to 3+.5 (from 7)

L

SQUARE ROOT

Method

An iterative Newton-Raphson process is used to find

$$y = \sqrt{x}, \text{ where } x = p + 2^{-38}q.$$

$$y_{n+1} = y_n + d_n \quad \text{where} \quad d_n = \frac{1}{2}\left(\frac{x}{y_n} - y_n\right)$$

$$\text{with } y_0 = \frac{1}{2}x + \frac{1}{2}.$$

In fact, $y_0 = \frac{1}{2}p + \frac{1}{2}$, and the process terminates when $d_n \geqslant 0$. y_n is then taken to be the answer. There is special treatment of the case $p = (1 - 2^{-38})$, when the root is also $(1 - 2^{-38})$.

	R 0 0 -0 1
	200 - 28 -
01	0+ [0] 72
	0.0 0 60

LN
SQUARE ROOT

0+

Entry	Address	Order	Notes
01 →	0.0	0.0 6 63	LOOP STOP if $p < 0$ } Overwritten by q
		6 5 00	p to 5 } Overwritten by q
	1	0.0 7 10	q to 0.0
		6 7 00	p to 7
	2	(1) 7 51	$\frac{1}{2}p$ to 7
→		0.7 7 11	$y_0 = (\frac{1}{2}p + \frac{1}{2})$ or $y_{n+1} = (y_n + d_n)$ to 0.7
	3	0.6+ 0 65	Test OVR and exit if $p = (1 - 2^{-38})$
		0.0 7 00	q to 7
	4	0.7 5 24	x/y_n to 7
		0.7 7 03	$(x/y_n - y_n) = 2d_n$ to 7
	5	(1) 7 51	d_n to 7
		0.2+ 7 63	Test d_n
	6	0.7 6 00	$\sqrt{x}$ to 6
→		0.7 1 10	Plant LINK and exit
	7	+0.5	Initially $\frac{1}{2}$ } Overwritten by y_n } Overwritten by LINK

L

FERRANTI LTD

PEGASUS LIBRARY PROGRAMME

Issue 2
8.7.57.

EXPONENTIAL

Method: Let $y = \dfrac{1 + ax + bx^2 + cx^3}{1 - ax + bx^2 - cx^3}$ be an approximation to e^{2x} in the range

$-k \leqslant x \leqslant k$.

Expanding log y, it can be shown that

$$x = \tfrac{1}{2} \log y = ax + (c - ab + \tfrac{1}{3} a^3)x^3 + \{a^2c + ab^2 - bc - a^3b + \tfrac{1}{5} a^5\} x^5 + \ldots,$$

and, equating coefficients:

$$a = 1; \quad b = \tfrac{2}{5}; \quad c = \tfrac{1}{15}; \quad \text{coefficient of } x^7 = \tfrac{1}{1575}.$$

If we set $a = 1 + \alpha$, $b = \frac{2}{5} + \beta$, $c = \frac{1}{15} + \gamma$, where α, β and γ are small, then the logarithmic error is

$$\delta = \tfrac{1}{2} \log y - x = \alpha x + [\gamma - (\beta + \tfrac{2}{5}\alpha + \alpha\beta) + (\alpha + \alpha^2 + \tfrac{1}{3}\alpha^3)] x^3 + \ldots$$

Now we can make $|\delta|$ as small as possible in the range $-k \leqslant x \leqslant k$ by choosing α, β and γ so that δ is a multiple of the Chebycheff polynomial

$$T_7(\theta) = 64\theta^7 - 112\theta^5 + 56\theta^3 - 7\theta,$$

where $\theta = x/k$. We find, in fact,

$$a = 1 - \frac{1}{14400} k^6,$$

$$b = \frac{2}{5} - \frac{1}{300} k^2 - \frac{1}{1000} k^4 - \frac{17}{200,000} k^6 - \ldots,$$

$$c = \frac{1}{15} - \frac{1}{300} k^2 - \frac{1}{2250} k^4 - \frac{3}{200,000} k^6 - \ldots.$$

These approximations are satisfactory only when k is considerably less than 1, but the following method allows the range to be extended.

Let X be the operand and define $x = X/16$, so that $k = \frac{1}{16}$, and $|\delta| < 0.65 \times 2^{-44}$ approximately. Now

$$e^X = e^{16x} = (ye^{-2\delta})^8 = y^8 e^{-16\delta}$$

where $16|\delta| < 0.65 \times 2^{-40}$ and can be neglected, so that we can write

$$Y = e^X = e^{16x} = y^{2^3}$$

within the required accuracy. If we substitute X/16 for x in the original definition of y we find

$$y = \frac{1 + a(X/16) + b(X/16)^2 + c(X/16)^3}{1 - a(X/16) + b(X/16)^2 - c(X/16)^3}.$$

To preserve significant figures of X when evaluating y, form

$$\eta_0 = 2^3(y-1) = \frac{aX + cX(X/16)^2}{1 - a(X/16) + b(X/16)^2 - c(X/16)^3}$$

which gives $y = 1 + 2^{-3}\eta_0$.

If we define η_r so that $y^{2^r} = 1 + 2^{-(3-r)}\eta_r$ then

$$\eta_{r+1} = \eta_r + 2^{-(4-r)}\eta_r^2$$

and $Y = y^{2^3} = 1 + \eta_3$

In actual operation η_r lies outside the range ± 1 so $\zeta_r = \frac{1}{2}\eta_r$ is used instead.

This gives $\zeta_0 = \dfrac{2^{-2}aX + 2^{-10}cX^3}{\frac{1}{2} - 2^{-5}aX + 2^{-9}bX^2 - 2^{-13}cX^3}$,

$$\zeta_1 = \zeta_0 + \tfrac{1}{8}\zeta_0^2.$$

$$\zeta_2 = \zeta_1 + \tfrac{1}{4}\zeta_1^2.$$

$$\zeta_3 = \zeta_2 + \tfrac{1}{2}\zeta_2^2.$$

so that $\frac{1}{4}e^X = \frac{1}{4}y^{2^3} = \frac{1}{4} + \frac{1}{2}\zeta_3$.

Since $k = 2^{-4}$, the last term in a, b and c is negligible and may be omitted.

Then $a = 1$,

$$b = \frac{2}{5} - \frac{1}{300}2^{-8} - \frac{1}{1000}2^{-16},$$

$$c = \frac{1}{15} - \frac{1}{300}2^{-8} - \frac{1}{2250}2^{-16}.$$

Putting $a = 1$ and substituting $z = 2^{-2}X$, the formula for ζ_0 may be written

$$\zeta_0 = \frac{z + 2^{-4}cz^3}{\frac{1}{2} + 2^{-3}(-z + 2^{-2}bz^2 - 2^{-4}cz^3)}.$$

	R 0 0 -0 1
	220 - 28 -
01	1+ [0] 72
	0.0 0 60

LN
EXP

0+

Address	Instruction	Notes
0.0	7 6 03	$\frac{1}{2} + 2^{-3}\ (-z + Bz^2 - Cz^3) \longrightarrow 6$
	(3) 6 51	
1	33 6 01	
	6 7 26	$\zeta_0 \longrightarrow 7$
2	7 6 00	$\zeta_1 \longrightarrow 6$
	(3) 7 51	
3	6 7 22	
	6 7 00	$\zeta_2 \longrightarrow 6$
4	(2) 7 51	
	6 7 22	
5	(1) 6 51	$\frac{1}{4} + \frac{1}{2}\zeta_3 = \frac{1}{4} \exp p \longrightarrow 6$
	6 6 22	
6	0.7 6 01	
	0.7 1 10	Plant LINK and exit
7	+.25	Overwritten by LINK

1+

(01) →

Address	Instruction	Notes
0.0	(2) 6 51	$\frac{1}{4}p = z \longrightarrow 6$ and 0.0
	0.0 6 10	
1	6 6 21	$z^2 \longrightarrow 6$ and 0.1
	0.1 6 10	
2	0.7 6 21	$Cz^2 \longrightarrow 6$
	0.0 6 21	$Cz^3 \longrightarrow 6$
3	0.0 6 11	$z + Cz^3 \longrightarrow 6$
	0.1 6 00	
4	0.6 6 21	$Bz^2 \longrightarrow 6$
	0.0 7 00	$z + Cz^3 \longrightarrow 7$
5	0+ [0] 72	
	0.0 0 60	
6	12 6 31 3.	$B = 2^{-2}b\ (\simeq 2^{-2} . \frac{2}{5})$
	11 5 41 5	
7	0 4 21 0.	$C = 2^{-4}c\ (\simeq 2^{-4} . \frac{1}{15})$
	13 5 00	

L

LOGARITHM, WIDE RANGE

This subroutine has been superseded by R 224 for most applications, but it is still used by certain other subroutines.

The subroutine evaluates

$$p' = \frac{1}{32} \log_e (pq)$$

in the range $2^{-46} \leqslant (pq) < 1$.

The method used is the same as that described for the programme of R 224, but only four terms of equation 11 are used in R 221.

	R 0 0 -0 1
	221 - 28 -
01	1+ [0] 72
	0.0 0 60

LN

LOG

1+

Block	Instruction	Comment
(01 →)	(2) 5 42	-2 to 5
(0.0 ↺)	0.0+ 6 63	LOOP STOP if number negative
1	(46) 5 56	y to 6, $a - 2$ to 5
	0.5 6 14	$(y - \frac{1}{4}\sqrt{2})$ to 0.5
2	0.6 6 21	$(3 - 2\sqrt{2})y$ to 6
	0.7 6 11	$(3 - 2\sqrt{2})y + (\frac{3}{4}\sqrt{2} - 1)$ to 0.7
3	0.5 6 00	$(y - \frac{1}{4}\sqrt{2})$ to 6
	0 7 00	Clear 7
4	0.7 6 25	z to 7
(to 0+.5 ←)	0+ [0] 72	
5	+97184015999	$= \frac{1}{4}\sqrt{2}$ } Overwritten by $(y - \frac{1}{4}\sqrt{2})$
6	+47161592838	$= 3 - 2\sqrt{2}$
7	+16674141054	$= \frac{3}{4}\sqrt{2} - 1$ } Overwritten by $(3 - 2\sqrt{2})y + (\frac{3}{4}\sqrt{2} - 1)$

0+

Address	Contents	Notes	
0.0	−95265423098	$= -\frac{1}{2}\log_e 2 = -0.34657359027$	
1	+362106	$T_7 = +0.00000131733$	
2	+16333229	$T_5 = +0.00005941994$	
3	+925542347	$T_3 = +0.00336710344$	
4	+94323185383	$T_1 = +0.34314574944$	
5 (from 1+.4+)	0.5 7 10	z to 0.5	Overwritten by z
	7 7 21	z^2 to 0.6	
6	0.6 6 10		Overwritten by z^2
	2+ [1] 72		
7	0.0 5 20	$-\frac{1}{32}(a - 2)\log_e 2$ to 7	Overwritten by $-\frac{1}{32}(a - 2)\log_e 2$
	(4) 0 55		

2+

Address	Order	Notes	
0	0.7 7 10	$-\frac{1}{32}(a - 2)\log_e 2$ to 0.7	
1.0	1.0+ 6 61	LOOP STOP if number too small	
1	1.6+ 0 65	Dummy order (Due to rounding OVR is never set)	
	0.1 6 00	Sum series in z^2	
2	0.6 6 21		
	0.2 6 01		
3	0.6 6 21		
	0.3 6 01		
4	0.6 6 21		
	0.4 6 01		
5	0.5 6 21	Multiply by z	
	0.0 6 03	$\frac{1}{32}(\log_e y + 2\log_e 2)$ to 6	
6	(5) 6 51		
	0.7 6 03	$\frac{1}{32}\log_e(pq)$ to 6	
7	1.7 1 10	Plant LINK	Overwritten by LINK
	1.7 0 60	Jump to obey LINK	

L

Issue 2
5th November, 195
C.S.

FLOATING-POINT LOGARITHM

Location	Word	Comment
	R 0 0 -0 1	Given $p = A.2^a$
	223 - 28 -	(Standard floating-point form)
01	0+ [0] 72	This subroutine evaluates $p' = \log_e p$
	0.2 0 60	$= \log_e A + a \log_e 2$

LN
FP LOG

Location	Word	Comment
	R 0 5 -0 1	Tag calling for R 221
	221 - 01 -	

0+

Location	Word	Comment
0.0	127 7 77 7	Collating Mask for Argument A
	127 7 00 0	
1	+256	Constant added to exponent and round-off constant for argument
2 (← 01)	0 7 00	Clear 7
	6 4 00	$A.2^a$ to 4
3	0.0 6 05	A to 6
	6 4 03	a + 256 to 4
4	[1+ 5] 71	Plant the LINK
	0.6 1 00	Set link for return from R 221
5	+0	Cue to R 221; $\frac{1}{32} \log_e A$ to 6
6	0+ [0] 72	Link for return from R 221
	0.7 0 60	
7	0.1 4 03	a to 4
	1+ [1] 72	

1+

Location	Word	Comment
1.0	(32) 7 40	7_c = +32
	6 7 20	$2^{-38} \log_e A$ to PQ
1	1.7 4 22	$2^{-38}[\log_e A + a \log_e 2]$ to PQ
	(38) 5 40	5_c = +38
2	(79) 5 56	Normalize
	0.1 6 01	Round argument
3	(2) 5 56	Re-normalize
	0.0 6 05	Pack result in 6 in standard floating point form.
4	0.1 5 01	
	5 6 01	
5	0	Overwritten by LINK
	0	
6	0	
	0	
7	+190530846196	$= \log_e 2$

L

London Computer Centre,
21, Portland Place,
LONDON W.1.

Issue 2
6th November, 1958.
M.E.H.

LOGARITHM, VARIABLE RANGE

This subroutine evaluates

$$p' = \frac{1}{2^n} \log_e(pq)$$

(pq) is first normalized to give an argument y and an exponent a. Only the most significant 39 bits of y are used in the calculation. The required logarithm may be written

$$\log_e y + a \log_e 2 \quad \text{where} \quad \tfrac{1}{4} \leqslant y < \tfrac{1}{2}$$

We have*:

$$\log_e (1-2t \cos\theta + t^2) = -2\sum_{r=1}^{\infty} \frac{t^r}{r} \cos r\theta \qquad \ldots(1)$$

If we put $z = \cos\theta$, then

$$\log_e (1+t^2-2tz) = -2\sum_{r=1}^{\infty} \frac{t^r}{r} T_r(z) \qquad (-1 \leqslant z \leqslant 1) \qquad \ldots(2)$$

where the T_r are Chebycheff polynomials.

Changing the sign of z and using $T_r(-z) = (-1)^r T_r(z)$, we have

$$\log_e (1 + t^2 + 2tz) = -2\sum_{r=1}^{\infty} (-1)^r \frac{t^r}{r} T_r(z) \qquad \ldots(3)$$

Taking the difference of (2) and (3):

$$\log_e \left[\frac{(1+t^2) + 2tz}{(1+t^2) - 2tz}\right] = 4\sum_{r=1}^{\infty} \frac{t^{2r+1}}{2r+1} T_{2r+1}(z) \qquad \ldots(4)$$

If we now put $\frac{2t}{1+t^2} = k$, then

$$\log_e \left(\frac{1+kz}{1-kz}\right) = 4\sum_{r=0}^{\infty} \frac{t^{2r+1}}{2r+1} T_{2r+1}(z) \qquad \ldots(5)$$

where $k^2 < 1$ and $t = \frac{1}{k}(1 - \sqrt{1-k^2})$ ...(6)

If A is any constant

$$\log_e \left(A.\frac{1+kz}{1-kz}\right) = \log_e A + 4\sum_{r=0}^{\infty} \frac{t^{2r+1}}{2r+1} T_{2r+1}(z) \qquad \ldots(7)$$

It is required to choose k and A such that this series will give $\log_e y$.

We require

$$y = A.\frac{1+kz}{1-kz} \quad \text{which gives}$$

$$z = \frac{1}{k} \cdot \frac{y-A}{y+A} \qquad \ldots(8)$$

where $\tfrac{1}{4} \leqslant y < \tfrac{1}{2}$ and $-1 \leqslant z \leqslant 1$

For $y = \frac{1}{4},\ z = -1$ $\qquad \frac{1}{4} - A = -k(\frac{1}{4} + A)$

$y = \frac{1}{2},\ z = 1$ $\qquad \frac{1}{2} - A = k(\frac{1}{2} + A)$

Solving these equations, we obtain

$$A = \tfrac{1}{4}\sqrt{2} \quad \text{and} \quad k = \frac{\sqrt{2} - 1}{\sqrt{2} + 1}$$

Substituting in (6) gives $t = 3 + 2\sqrt{2} - 2\sqrt{4 + 3\sqrt{2}}$ ...(9)

Substituting in (8) gives $z = \dfrac{y - \frac{1}{4}\sqrt{2}}{(3 - 2\sqrt{2})y + (\frac{3}{4}\sqrt{2} - 1)}$ ...(10)

Substituting in (7)

$$\log_e y = -\frac{3}{2}\log_e 2 + 4\sum_{r=0}^{\infty} \frac{t^{2r+1}}{2r+1} T_{2r+1}(z) \qquad \text{...(11)}$$

The first five terms in equation (11) are used by the subroutine to evaluate $\log_e y$. The truncation error introduced by taking only five terms is about 2^{-37}. $a \log_e 2$ is added to $\log_e y$ to give the required $\log_e(pq)$.

* "Tables of Integrals and Other Mathematical Data" - H.B. Dwight (Macmillan, New York, 1947).

	R 0 0 -0 1
	224 - 28 -
01	1+ [0] 72
	0.1 0 60

LN
LOG. MK. 2

R 2 7 -0 1	Title of Optional Parameter List
224 - 06 -	
R 2 0 -0 1	Calls for P.P.01
224 - 02 -	
R 2 6 -0 1	
224 - 02 -	

1+

Address	Instruction	Comment
0.0	+97184015999	$A = \frac{1}{4}\sqrt{2}$ } Overwritten by $(y - \frac{1}{4}\sqrt{2})$
1 (01 →)	(2) 5 42	-2 to 5
	0.1+ 6 63 (loop 1)	LOOP STOP if number negative
2	(100) 5 56	y to 6, $(a - 2)$ to 5
	0.0 6 14	$(y - \frac{1}{4}\sqrt{2})$ to 0.0
3	0.6 6 21	$(3 - 2\sqrt{2})\,y$ to 6
	0.7 6 11	$(3 - 2\sqrt{2})\,y + (\frac{3}{4}\sqrt{2} - 1)$ to 0.7
4	0.0 6 00	$(y - \frac{1}{4}\sqrt{2})$ to 6
	0.7 6 26	z to 7
5	0+ [0] 72	
	2+ [1] 72	→ to 0+.6
6	+47161592838	$= 3 - 2\sqrt{2}$
7	+16674141054	$= \frac{3}{4}\sqrt{2} - 1$ } Overwritten by $(3 - 2\sqrt{2})\,y + (\frac{3}{4}\sqrt{2} - 1)$

0+

Address	Contents	Notes	
0.0	-95265423098	$-\frac{1}{2}\log_e 2 = -0.34657359027$	
1	+8415	$T_9 = +0.00000003061$	
2	+343173	$T_7 = +0.00000124845$	
3	+16347428	$T_5 = +0.00005947159$	
4	+925538404	$T_3 = +0.00336708909$	
5	+94323185678	$T_1 = +0.34314575051$	
6 (from 1+.5+ →)	0.6 7 10	z to 0.6	Overwritten by z
	7 7 21	z^2 to 6	
7	0.7 6 10	z^2 to 0.7	Overwritten by z^2
	0.0 5 20	$-\frac{1}{2}(a-2)\log_e 2$ to PQ	

2+

Address	Contents	Notes	
1.0	(127) 0 55 0.	+ P.P.01 Shifts down n-1 places	Overwritten by $-2^{-n}(a-2)\log_e 2$
	1.7 1 10	Plant LINK in 1.7	
1	1.0 7 10	$-2^{-n}(a-2)\log_e 2$ to 1.0	
	1.1+ 6 61	LOOP STOP if number too small	
2	1.6+ 0 65	Dummy Order (Due to rounding, OVR is never set)	
	0.4 5 02	4 to 5_p	
3	7.5+ 6 01 5	Sum series in z^2	
	0.7 6 21		
4	1.3 5 66		
	0.5 6 01		
5	0.6 6 20	Multiply by z	
	0.0 6 03	$\log_e y + 2\log_e 2$ to 6	
6	(0) 6 51	+ P.P.01. Divide C(6) by 2^n	
	1.0 6 03	$2^{-n}\log_e(pq)$ to 6	
7	5 0 00 0.	= Optional P.P.01 (n = 5)	Overwritten by LINK
	0		

FLOATING-POINT EXPONENTIAL

The method is explained in Section 2 of the Library Specification.

R 0	0 -0 1
225	- 28 -
0+	[1] 72
1.0	0 60

LN
FP EXP MK 4

R 0	5 -0 1	Calls for P.P.01
225	- 02 -	
R 0	5 -0 1	
225	- 02 -	
R 0	6 -0 1	
225	- 02 -	
R 0	7 -0 1	Title of optional parameter-list
225	- 06 -	
R 1	4 -0 1	Call for cue 01 to R 220
220	- 01 -	

0+

(01) →	1.0	1.7	1 10	Plant LINK in 1.7 } Overwritten by $-\mathrm{I}\left(x.\frac{1}{\log_e 2}\right)$ [= $-(b-2)$]
		6	1 00	$x = A.2^a$ to 1
	1	1.5	6 05	A to 6
		1+	[0] 72	
	2	6	1 06	$a + 2^{n-1}$ to 1_c
		1.6	1 01	a to 1_c
	3	(38)	1 41	a + 38 to 1_c
to 1+.0 ←		0.0	1 62	Jump if $a \geqslant -38$
	4	0	0 20	If $a < -38$, clear PQ and jump to find exp (0)
to 1+.2+ ←		0.2+	0 60	
	5	+0		$+ 2 \times$ P.P.01 $= -2^n.2^{-38}$
	6	+0		+ P.P.01 $= -2^{n-1}.2^{-38}$
from 2+.5+ or 2+.7+ →	7	−256		Optional P.P.01 $= -2^{n-1}.2^{-38}$ where $n = 9$ } Overwritten by LINK

1+

from 0+.3+	0.0	(75) 1 44	$(37 - a)$ to 1_c
		0.5 1 63	Jump if $a > 37$
	1	(25) 1 52	$(37 - a)$ to 1_m
		0.7 6 20	$x.\frac{1}{\log_e 2}$ (fixed-point form) to PQ
	2	(0) 0 55 1	
from 0+.4+		1.0 6 12	−I = − Integral part of $x/\log_e 2$ to 1.0
	3	0.6 7 21	F $\log_e 2$ to 6
		0.5 1 00	Set link in 1 for return from R 220
	4	+0	+ cue 01 to R 220; Forms exp (F $\log_e 2$)
	5	2+ [0] 72	Link
		0.0 0 60	
	6	+190530846196	= $\log_e 2$
	7	+198282496599	= $\frac{1}{2} \cdot \frac{1}{\log_e 2}$

2+

	0.0	0.7 6 63	Jump if floating-point underflow
		1.0 1 02	$b = I\left(x \cdot \frac{1}{\log_e 2}\right) + 2$ to 1
	1	(2) 1 41	
		(2) 1 56	Normalize
	2	1.6 6 03	Round argument, B, in 6
		(2) 1 56	Re-normalize
	3	1.6 1 03	$b + 2^{n-1}$ to 1
		0.7 1 63	Jump if floating-point underflow
	4	1.5 6 05	Pack result in 6 in standard floating-point form, $B.2^b$
		1 6 01	
	5	1.5 1 05	
to LINK		1.7 1 60	Jump to obey LINK if in range
	6	0.6 0 60	LOOP STOP if floating-point OVR
		0	
	7	0 6 00	Clear 6 if floating-point OVR
to LINK		1.7 0 60	Jump to obey LINK

L

London Computer Centre,
21, Portland Place,
LONDON, W.1.

Issue 1
4th August, 1960.
B. C.

SIN/COS

This subroutine evaluates $\sin \pi x$ or $\cos \pi x$ for $-1.0 \leqslant x < 1.0$.

Method

The method is based on the Chebycheff expansion for

$\cos \frac{\pi}{2} y$ in the range $-1 \leqslant y < 1$.

$$\cos \frac{\pi}{2} y = J_0\left(\frac{\pi}{2}\right) + 2 \sum_{n=1}^{\infty} J_{2n}\left(\frac{\pi}{2}\right) T_{2n}(y)$$

The truncation error in taking the first seven terms of this series is about $7.5 \times 10^{-13} (< 2^{-40})$. Re-arranging the first seven terms as a power series in y, we have

$$\cos \frac{\pi}{2} y = \sum_{r=0}^{6} c_r y^{2r}$$

In this series $c_1 < -1$ and it must therefore be re-arranged to read

$$\cos \frac{\pi}{2} y = \left[\sum_{r=0}^{6} b_r y^{2r}\right] - y^2$$

where $b_1 = c_1 + 1, \quad b_r = c_r$ for $r \neq 1$.

To reduce rounding errors in evaluating the series, it has been re-arranged to read

$$\cos \frac{\pi}{2} y = \left[\frac{1}{2} \sum_{r=0}^{6} a_r y^{2r}\right] - y^2 + a_0$$

where $a_0 = \frac{2}{3} b_0, \quad a_r = 2b_r$ for $r \neq 0$.

Overflow

If $|x| > \frac{1}{2}$ $\quad 2x$ will overflow,

leaving $y = 2x \pm 2$ and OVR set

but $\cos \theta = -\cos(\theta \pm \pi)$

so that $\cos \pi x = -\cos \frac{\pi}{2}(2x \pm 2)$

$$= -\cos \frac{\pi}{2} y .$$

Thus, if OVR is set in forming y, the series for $\cos \frac{\pi}{2} y$ is still used but the result is negated and OVR cleared at the end of the routine.

To find $\sin \pi x$

$$\sin \theta = \cos\left(\theta - \frac{\pi}{2}\right) = \cos\left(\theta + \frac{3\pi}{2}\right)$$

so that $\sin \pi x = \cos \pi\left(x - \frac{1}{2}\right) = \cos \pi\left(x + \frac{3}{2}\right).$

The cue for sin πx therefore subtracts $\frac{1}{2}$ from x and then uses the cosine series as before, but with $y = 2x - 1$.

Note that $(x - \frac{1}{2})$ will overflow if $-1 \leqslant x < -\frac{1}{2}$, leaving $(x + \frac{3}{2})$ in X6 and OVR set. The fact that OVR is set does not matter because

$$(x + \tfrac{3}{2}) > \tfrac{1}{2}$$

and overflow will occur when $x + \frac{3}{2}$ is doubled to form y; thus the rule given in the previous paragraph will still apply.

Special Cases

If $y = -1.0$, y^2 will overflow and become negative ($y^2 = -1.0$) and the series will give the wrong result. For $y = -1.0$, $\cos \frac{\pi}{2} y = 0$. The routine therefore tests y^2 and sets the result equal to zero if y^2 is negative.

If $y = 0$ the series will give the value $\cos \frac{\pi}{2} y = 1 - 2^{-38}$.

	R 0 0 -0 2
	240 - 28 -
01	0+ [0] 72
	0.0 0 60
02	0+ [0] 72
	0.0+ 0 60

LN
SIN/COS

0+

Entry	Block	Instruction	Comment
01 →	0.0	33 6 03	$x - \frac{1}{2}$ to 6
02 →		(1) 6 50	$y = 2x$ or $(2x - 1)$ to 6
	1	1+[1] 72	
		6 6 21	y^2 to 6
	2	6 5 00	y^2 to 5
		1.1 6 00	a_6 to 6
	3	1.1 1 10	Plant LINK in 1.1
		0.4 1 00	$1_p = 2$
	4	5 6 21	Form $\sum_{r=0}^{6} a_r y^{2r}$ in 6
		1.0 6 01 1	
	5	0.4 1 66 (jump to 4)	
		(1) 6 51	$\left[\frac{1}{2}\sum_{r=0}^{6} a_r y^{2r}\right] - y^2 + a_0$ in 6
	6	5 6 03	
		1.7 6 01	
	7	1.0 5 62	If $y^2 = -1$ (OVR when $y = -1$) clear 6
		0 6 00	

1+

Block	Instruction		Comment
1.0	1.1 0 64		EXIT if OVR clear
	6 6 02		Negate 6 if OVR set
1	(+246961)	=	a_6 = +0.00000089843 } Overwritten by LINK
2	−13838561	=	a_5 = −0.00005034439
3	+505357534	=	a_4 = +0.00183847999
4	−11469815955	=	a_3 = −0.04172694736
5	+139456286127	=	a_2 = +0.50733901344
6	−128478236105	=	a_1 = −0.46740110012
7	+183251937962	=	a_0 = +0.66666666666

L

ARCTAN

We have[1]:-

$$\frac{1 - t^2}{1 - 2t\cos\theta + t^2} = 1 + 2\sum_{r=1}^{\infty} t^r \cos r\theta \qquad (t^2 < 1) \tag{1}$$

For $\cos\theta$ write z, then $\cos r\theta = T_r(z)$ where T_r are the Chebycheff polynomials, and we get, for $-1 \leqslant z \leqslant 1$.

$$\frac{1 - t^2}{(1 + t^2) - 2tz} = 1 + 2\sum_{r=1}^{\infty} t^r\, T_r(z) \tag{2}$$

Change the sign of z and use $T_r(-z) = (-1)^r\, T_r(z)$. Then

$$\frac{1 - t^2}{(1 + t^2) + 2tz} = 1 + 2\sum_{r=1}^{\infty} (-1)^r\, t^r\, T_r(z) \tag{3}$$

Now take the difference of equations (2) and (3):-

$$\frac{4(1 - t^2)tz}{(1 + t^2)^2 - 4t^2z^2} = 4\sum_{r=0}^{\infty} t^{2r+1}\, T_{2r+1}(z)$$

or

$$\frac{(1 - t^2)z}{(1 + t^2)^2 - 4t^2z^2} = \sum_{r=0}^{\infty} t^{2r}\, T_{2r+1}(z) \tag{4}$$

Change the sign of t^2:-

$$\frac{(1 + t^2)z}{(1 - t^2)^2 + 4t^2z^2} = \sum_{r=0}^{\infty} (-1)^r t^{2r} T_{2r+1}(z) \tag{5}$$

Now integrate with respect to t from 0 to t:-

$$\int_0^t \frac{1 + t^2}{(1 - t^2)^2} \cdot \frac{z}{1 + \left(\frac{2t}{1-t^2}\right)^2 z^2}\, dt = \sum_{r=0}^{\infty} (-1)^r \frac{t^{2r+1}}{2r+1}\, T_{2r+1}(z) \tag{6}$$

Substitute $t = \tan\frac{\alpha}{2}$ on the left-hand side, so that $\tan\alpha = \frac{2t}{1 - t^2}$ and

$\frac{d(\tan\alpha)}{dt} = \frac{2(1 + t^2)}{(1 - t^2)^2}$:-

$$\frac{1}{2}\int_0^{\alpha} \frac{z}{1 + z^2\tan^2\alpha}\, d(\tan\alpha) = \frac{1}{2}\arctan(z\tan\alpha) \tag{7}$$

Thus

$$\arctan(z\tan\alpha) = 2\sum_{r=0}^{\infty} (-1)^r \frac{t^{2r+1}}{2r+1} T_{2r+1}(z) \tag{8}$$

where $t = \tan\frac{\alpha}{2}$. If we put $k = \tan\alpha$, then:-

$$\arctan(kz) = 2\sum_{r=0}^{\infty} (-1)^r \frac{t^{2r+1}}{2r+1} T_{2r+1}(z) \tag{9}$$

where $t = \frac{1}{k}(\sqrt{1 + k^2} - 1)$.

To find arctan of a given number, reduce it to the range $0 \leqslant x \leqslant 1$, the mid-point of which, in terms of angle, is $\pi/8$.

Put $\qquad \arctan x = \frac{\pi}{8} + \theta$ so that $-\frac{\pi}{8} \leqslant \theta < \frac{\pi}{8}$

and put $\qquad \arctan ky = \theta$, choosing k so that $-1 \leqslant y < 1$

i.e. $k = \tan\frac{\pi}{8} = \sqrt{2} - 1$.

Now,

$$\arctan ky = -\frac{\pi}{8} + \arctan x$$

$$= -\arctan(\sqrt{2} - 1) + \arctan x$$

$$= \arctan\left\{\frac{x - (\sqrt{2} - 1)}{1 + x(\sqrt{2} - 1)}\right\}$$

and

$$y = \frac{1}{\sqrt{2} - 1} \cdot \frac{x - (\sqrt{2} - 1)}{1 + x(\sqrt{2} - 1)} = \left(\frac{x - (\sqrt{2} - 1)}{(\sqrt{2} - 1) + (3 - 2\sqrt{2})\,x}\right) \text{ and } -1 \leqslant y < 1 \qquad (10)$$

The quantity y can thus be found from x by means of a multiplication, an addition, a subtraction and a division and we then use:-

$$\arctan x = \frac{\pi}{8} + 2\sum_{r=0}^{\infty} (-1)^r \frac{t^{2r+1}}{2r+1} T_{2r+1}(y), \qquad (11)$$

where

$$t = \tan\frac{\pi}{16} = \sqrt{4 + 2\sqrt{2}} - (\sqrt{2} + 1)$$

We can now write:-

$$\frac{1}{\pi}\arctan x = \frac{1}{8} + \sum_{r=0}^{\infty} (-1)^r \frac{2t^{2r+1}}{\pi(2r+1)} T_{2r+1}(y) \qquad (12)$$

N.B. $\qquad (3 - 2\sqrt{2}) \doteqdot 0.172$ and $(\sqrt{2} - 1) \doteqdot 0.414$ so that there is no danger of exceeding capacity during the formation of y from x $(0 \leqslant x < 1)$.

The truncation error in taking the first six terms of the series is about 1.2×10^{-12} $(\doteqdot \frac{1}{3} \times 2^{-38})$.

1. *"Tables of Integrals and Other Mathematical Data" – H.B. Dwight (Macmillan, New York, 1947).*

$$\phi' = \frac{1}{\pi}\arctan(\phi)$$

$$\text{or } \phi' = \frac{1}{\pi}\arctan\left(\frac{\phi}{q}\right)$$

$$\text{where } -\tfrac{1}{2} \leqslant \phi' \leqslant \tfrac{1}{2}$$

	R 0 0 -0 4
	241 - 28 -
01	1+ [0] 72
	0.0 0 60
02	1+ [0] 72
	0.1+ 0 60
03	0+ 0 00 0.
	0
04	0
	0+ 0 00 0.

LN
ARCTAN

[1+]

Entry / jump	Address	Instruction	Comment
(01) →	0.0	6 4 00	ϕ to 4 [Entry for $\frac{1}{\pi}$ arctan (ϕ)]
		32 7 00	- y = -1 to 7
		0.2+ 0 60	
(02) →	1	6 4 00	ϕ to 4 [Entry for $\frac{1}{\pi}$ arctan $\left(\frac{\phi}{q}\right)$]
		7 4 06	Sign 4 = sign (ϕ/q)
	2	33 2 02	- ½ to 2
		0.4 6 63	- $\lvert\phi\rvert = -u$ to 6
	3	6 6 02	
		0.5 7 63	- $\lvert q\rvert = -v$ to 7
	4	7 7 02	
		6 7 03	$u-v$ to 7. If $u = 1$, $v = 0$, 7 = -1 & OVR set
	5	3+ [1] 72	
		(2) 2 53	$a_0 = \frac{3}{8}$ to 2
to 3+.2 ←	6	1.2 7 62	Jump if $u \geqslant v$ (except u=1, v=0) with { X6 = $-u$; X7 = $u-v$ }
		7 6 01	$-v$ to 6
to 3+.1 ←	7	1.1 6 60	Jump if $u = 1$, $v = 0$

$A = 1 - \tan\frac{\pi}{8} = (2 - \sqrt{2})$

$B = k\tan^2\frac{\pi}{8} = k(3 - 2\sqrt{2})$

$C = k\left(1 - \tan\frac{\pi}{8}\right) = k(2 - \sqrt{2})$

3+

Entry	Addr	Instruction	Comment
from 1+.7+	1.0	33 2 03	Mark for interchange by putting $a_0 = -\frac{1}{8}$ in 2
		32 4 06	and reversing the sign of 4
from 1+.7+	1	7 7 02	$v - u$ to 7 If $\begin{cases} u = 0, & v = 1 \\ u = 1, & v = 0 \end{cases}$ 7=-1 & OVR set
		1.3+ 0 65	Jump if $u = 1$, $v = 0$ or $u = 0$, $v = 1$
from 1+.6+	2	7 3 00	
		1.2+ 6 60	LOOP STOP if $u = v = 0$
	3	6 3 26	$(\xi - 1)$ to 7 where $\xi = \frac{u}{v}$ or $\frac{v}{u}$, $0 \leqslant \xi \leqslant 1$
		7 3 00	$\xi - 1$ to 3
	4	1.6 3 01	$\xi - 1 + A$ to 3
		1.7 7 21	$B(\xi - 1)$ to 6
	5	2+ [0] 72	
		0.1 0 60	
	6	+161019749891	$A = +0.58578643762$
	7	+47161592863	$B = +0.17157287534$

2+

Entry	Addr	Instruction	Comment
	0.0	(+161019749976)	$C = +0.58578643793$ — Overwritten by z
	1	0.0 6 01	$B(\xi-1) + C$ to 6
		6 3 26	$y = \frac{\xi - 1 + A}{B(\xi-1) + C} = \frac{\xi - \tan\pi/8}{k(\tan\pi/8)(1 + \xi\tan\pi/8)}$ to 7
	2	0.0 7 10	Plant z in 0.0
		7 7 21	y^2 to 6
	3	0+ [1] 72	
		6 3 00	y^2 to 3
	4	1.1 6 00	a_{13} to 6
		0.5 5 00	Set counter of 6 i.e. $5_m = 2$
	5	3 6 21	Form $\frac{2}{\pi}\sum_{r=0}^{6} a_{2r+1}\, y^{2r+1}$ in 6
		1.0 6 01 5	
	6	0.5 5 66	
		0.0 6 21	
	7	2 6 04	
to 0+.0		1.1 1 10	Plant LINK

A, B and C as above (3+.6, 3+.7, 2+.0.0). Jumps: 3+.1.3+ 0 65 jumps to 3+.3 (7 3 00); 3+.0.1 0 60 jumps to 2+.0.0 6 01; 2+.0.5 5 66 loops back to 2+.5 (3 6 21).

0+

Address	Content	Comment
from 2+.7+ →	1.1 4 62	Jump to LINK if 4 is positive i.e. $p' = (1/\pi)$ arctan (p)
1.0	6 6 02	$p' = (1/\pi)$ arctan (p/q)
1	+42077	$a_{13} = +\ 0.00000015307$ } Overwritten by LINK
2	-450949	$a_{11} = -\ 0.00000164054$
3	+3461449	$a_9 = +\ 0.00001259267$
4	-26138321	$a_7 = -\ 0.00009509065$
5	+213371737	$a_5 = +\ 0.00077624185$
6	-2072724634	$a_3 = -0.00754052829$
7	+36242177026	$a_1 = +\ 0.13184827194$

ARC SIN/ARC COS

The subroutine evaluates, using R 241 and R 200 as subroutines,

$$y = \frac{1}{\pi} \text{arc sin } x \qquad -\tfrac{1}{2} \leqslant y \leqslant \tfrac{1}{2}$$

$$\text{or } y = \frac{1}{\pi} \text{arc cos } x \qquad 0 \leqslant y < 1$$

$$\text{where } \frac{1}{\pi} \text{arc sin } x = \frac{1}{\pi} \text{arc tan}\left(\frac{x}{\sqrt{1-x^2}}\right)$$

$$\text{and } \frac{1}{\pi} \text{arc cos } x = \tfrac{1}{2} - \frac{1}{\pi} \text{arc tan}\left(\frac{x}{\sqrt{1-x^2}}\right)$$

Initially $p = x$ and the result, y, is left in X6.

	R 0 0 -0 4 242 - 28 -	
01	0+ [0] 72 0.1+ 0 60	arc sin
02	0+ [0] 72 0.1 0 60	arc cos
03	0+ 0 00 0. 0	
04	0 0+ 0 00 0.	

LN
ARC SIN/ARC COS

R 0 5 -0 1 200 - 01 -	Call for cue 01 to R 200
R 1 1 -0 2 241 - 01 -	Call for cue 02 to R 241

0+

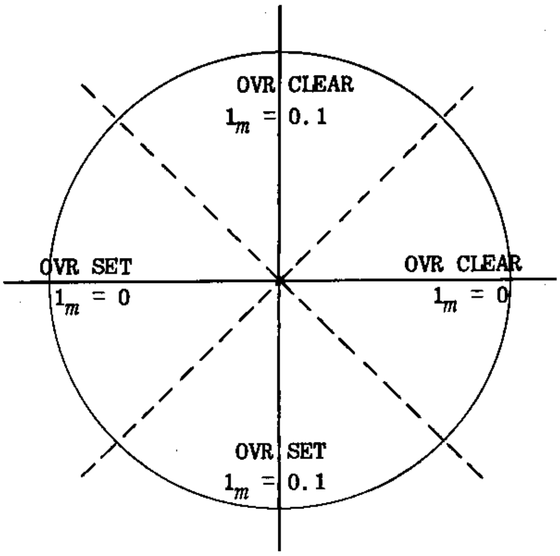
Address	Order	Comment
0.0	1 0 20 / 0	≡ (8.2, 0) { Modifier 0.2 for 00 order in 1.0; Modifier 0.1+ for 60 order in 1.3 }
1 (entry 02)	0.0 0 10	
(entry 01)	0.0 3 00	0 or (8.2, 0) to 3
2	1+ 7 71	Plant LINK
	6 4 02	$-x$ to 4
3	4 6 20	$(1-x^2)$ to PQ
	32 6 03	
4	0.6 0 65	Jump if $x = 0$ or -1.0
	0.7 1 00	Set LINK
5	+0	Cue to R 200; $\sqrt{1-x^2}$ to 6
6	1+ 1 72	
	1.2+ 0 60	
7	1+ 1 72	LINK obeyed in 0.7 of R 200
	6 7 00	$\sqrt{1-x^2}$ to 7

1+

Address	Order	Comment
1.0	4 6 02	x to 6
	1.5 1 00 3	Set LINK from 1.5 or 1.7
1	+0	Cue to R 241; $\frac{1}{\pi}$ arc sin x to 6
2	0	
	4 6 00	$x = 0$ or -1.0 to 6
3	(1) 6 51	$\frac{1}{\pi}$ arc sin $x = 0$ or $-\frac{1}{2}$ to 6
	1.5+ 0 60 3	Jump to 1.5+ (arc cos) or 1.7 (arc sin)
4	(1) 6 43	$+(1-2^{-38})$ to 6 and set OVR
	1.7 0 65	EXIT
5	1+ 1 72	LINK for arc cos
	1.6 0 60	
6	33 6 04	$\frac{1}{\pi}$ arc cos x to 6
	1.4 0 65	Jump if $x = -1.0$ and clear OVR
7	( )	LINK

L

COSINE AND SINE



This subroutine evaluates

$$p' = \cos \pi p$$

and $$q' = \sin \pi p$$

where $-1 \leqslant p \leqslant 1$

and $\pi p = (\eta + \xi)\, \pi/2$

$\eta = -2, -1, 0, 1, 2$ and $-\tfrac{1}{2} \leqslant \xi < \tfrac{1}{2}$

Let $$t = \tan\left(\frac{1}{2}\xi \cdot \frac{\pi}{2}\right) \doteqdot \frac{\xi}{1 - \left(\xi^2/4\right)} \cdot P(\xi^2)$$

where $P(\xi^2) = a_0 + a_2\,\xi^2 + a_4\,\xi^4 + a_6\,\xi^6$

Then $$\cos \xi = c = \frac{1 - t^2}{1 + t^2}$$

and $$\sin \xi = s = \frac{2t}{1 + t^2}.$$

$\cos p\pi$ and $\sin p\pi$ are then evaluated using the following relations:

	$\eta = -2,\ 2$	$\eta = -1$	$\eta = 0$	$\eta = 1$
$\cos p\pi$	$-c$	s	c	$-s$
$\sin p\pi$	$-s$	$-c$	s	c
OVR	Set	Set	Clear	Clear
1_m	0	0.1	0	0.1

Address	Orders	Notes
	R 0 0 -0 2	
	243 - 28 -	
01	0+ 0 00 0.	a – order partial cue
	0	
02	0	b – order partial cue
	0+ 0 00 0.	

LN
COS + SIN

0+

ENTRY → 3.0

Address	Order	Notes
3.0	33 7 00	$\frac{1}{2}$ to 7
	33 7 22	$p + \frac{1}{4}$ to 6; OVR set if $p \geqslant +\frac{3}{4}$
1	32 6 01	$p - \frac{3}{4}$ to 6; OVR set if $p < -\frac{1}{4}$
	(1) 6 52	$2p - \frac{3}{2}$ modulo 2
2	3.3+ 6 62	Jump if $p < -\frac{3}{4}$ or $p \geqslant +\frac{3}{4}$ or $-\frac{1}{4} \leqslant p < \frac{1}{4}$
	35 1 01	Add 0.1 to 1_m
3	32 6 06	Change sign bit of 6
	→ 33 6 03	ξ to 6
4	5.3 6 10	ξ to 5.3
	6 6 21	ξ^2 to 6
5	6 5 00	ξ^2 to 5
	4.0 4 02	Set $4_p = 5$
6	5.4 6 00	a_6 to 6
	→ 5 6 21	Form $P(\xi^2) = a_0 + a_2\xi^2 + a_4\xi^4 + a_6\xi^6$ in 6
7	5.0 6 01 4	
	3.6+ 4 66	

1+

4.0	5.3 6 21	$\xi P(\xi^2)$ to 6
	(2) 5 51	¼ (ξ^2) to 5
1	32 5 01	¼ $(\xi^2) - 1.0$ to 5
	5 6 26	$-t$ to 7
2	7 4 02	t to 4
	(1) 7 51	$-\frac{1}{2}t$ to 7
3	4 7 21	$-\frac{1}{2}(t^2 + 1)$ to 6
	33 6 03	
4	6 4 26	$-s = -\dfrac{2t}{(t^2 + 1)}$ to 7 and 5
	7 5 00	
5	4 5 21	$-\left(\dfrac{2t^2}{t^2 + 1}\right)$ to 6
	4.6+ 6 61 (jump to 6.1)	If $C(6) = 0$ replace it by -2^{-38} to avoid OVR
6	(1) 6 42	
	32 6 03	$c = \left(\dfrac{1 - t^2}{1 + t^2}\right)$ to 6
7	5 7 02	s to 7
	6 6 10 1	c to 6 or 7

2+

5.0	5 5 10 1	s to 5 or 6
	5.2 0 64 (jump to 2.0)	Negate s and c if OVR set (i.e. if original angle is ⩾ ¾ or < −¼)
1	6 6 02	
	7 7 02	
2	(25) 1 52	Set Computing Store Link modifier
	0 0 60 1	Obey LINK
3	+0	Overwritten by ξ
4	−3585500	$= a_6 = -0.00001304397$
5	−144697583	$= a_4 = -0.00052640674$
6	−9581836739	$= a_2 = -0.03485851898$
7	+215888603285	$= a_0 = +0.78539816344$

L

Issue 2
9th August, 1960.
D.R. W.F.M.P.

FLOATING-POINT ARCTAN

An accurate series cannot be obtained for arctan (p/q) over the full range of $-\infty \leqslant p/q \leqslant \infty$. It is possible, however, to evaluate arctan for some reduced range of p/q and then transform the result to the required value.

R 251 uses the following rules to reduce its operand to the range

$$-\tan \pi/12 \leqslant x \leqslant \tan \pi/12.$$

The original range is

$$-\infty \leqslant p/q \leqslant \infty.$$

(i) If $p > 0$, arctan $(p/q) = -$ arctan $(-p/q)$.

If $p > 0$ the routine negates p and sets a marker to indicate that the arctan then obtained must be negated. $p \leqslant 0$ throughout.

(ii) If $q < 0$ and $p \leqslant 0$, arctan $(p/q) = -\pi -$ arctan $(p/-q)$.

The routine makes $q \geqslant 0$ and sets a marker to show that the resulting arctan must be transformed.

The range is then $-\infty \leqslant p/q \leqslant 0$.

(iii) If $p \leqslant 0$, $q \geqslant 0$, arctan $(p/q) = -\pi/2 -$ arctan (q/p).

If $|p| > |q|$, the routine interchanges p and q and sets a marker to indicate that the above transformation must be made to the result. The range is then $-1.0 \leqslant x \leqslant 0$.

(iv) If $x (= \tan\theta)$ is less than $-\tan(\pi/12)$ the routine forms a new variable X using the formula

$$X = \tan(\theta + \pi/6) = \frac{x + \tan \pi/6}{1 - x \tan \pi/6}$$

$$= \frac{x + 1/\sqrt{3}}{1 - x/\sqrt{3}}$$

and sets a marker to indicate that it must later subtract $\pi/6$ from the $\tan^{-1} X$ thus obtained to give θ.

The range of the operand is then

$$-\tan \pi/12 \leqslant X \leqslant \tan \pi/12.$$

Six terms of a Chebycheff expansion are used to evaluate $\tan^{-1} X$ in this range and the appropriate transformations are then made to form $\tan^{-1}(p/q)$.

	R 0 0 -0 4
	251 - 28 -
01	0+ [0] 72
	0.2 0 60
02	0+ [0] 72
	0.2+ 0 60
03	0+ 0 00 0.
	0
04	0
	0+ 0 00 0.

LN
FP ARCTAN

R 0 0 -0 1	Tags calling for parameters
251 - 02 -	
R 0 1 -0 1	
251 - 02 -	
R 0 1 -0 1	
251 - 02 -	
R 4 1 -0 1	
251 - 02 -	
R 4 1 -0 1	
251 - 02 -	
R 4 2 -0 1	
251 - 02 -	
R 5 7 -0 1	Title of optional parameter list
251 - 06 -	

The operands are in packed floating-point form,

$$p = A.2^a \qquad q = B.2^b$$

0+

Entry / jump	Address	Instruction	Comment
	0.0	(3.7 7 77 7)	+ P.P.01 = -1 (f.p.) } Overwritten by $A.2^a$
		(7.7 7 77 6)	
	1	+0	+ 2 × P.P.01 = $-2^n.2^{-38}$
01 →	2	0.0 7 02	+ 1 to 7 [Entry for arctan (p)]
02 →		7 5 00	$q = B.2^b$ to 5 [Entry for arctan (p/q)]
	3	0.1 7 05	B to 7
		1+ [1] 72	
	4	7 5 06	b to 5
		0.0 6 10	$p = A.2^a$ to 0.0
	5	0.1 6 05	A to 6
		0.0 6 16	a to 0.0
	6	0.0 5 04	$(a - b)$ to 5
		(2) 4 40	Set marker; $4_m = 0$, $4_c = +2$
	7	3+ [0] 72	
→ 1.1		1.1 6 63	Jump if $p < 0$

1+

Entry / jump	Address	Instruction	Comment
	1.0	35 4 06	4_m = 0.1 as marker to indicate $p \geqslant 0$
		6 6 02	$- \lvert A \rvert$
	1	1.3 7 62	Jump if $q \geqslant 0$
		33 4 06	} Set marker for $q < 0$; set OVR for $q < 0$
	2	32 0 02	
		7 7 02	$+ \lvert B \rvert$
to 3+.6+	3	0.6+ 5 63	Jump if $a < b$ (i.e. $\lvert p \rvert < \lvert q \rvert$)
		6 3 00	$- \lvert A \rvert$ to 3
	4	7 3 01	$\lvert B \rvert - \lvert A \rvert$ to 3
→ 1.5		1.5+ 5 61	Jump if $a > b$ (i.e. $\lvert p \rvert > \lvert q \rvert$)
to 3+.6+	5	0.6+ 3 62	Jump if $\lvert B \rvert \geqslant \lvert A \rvert$ and $a = b$ (i.e. $\lvert p \rvert < \lvert q \rvert$)
		5 5 02	$(b - a)$ to 5
	6	32 0 02	Set OVR if interchange
		32 4 06	$4_s = 1$
	7	3 6 04	$+ \lvert B \rvert$
to 3+.6		0.6 0 60	

3+

Address		Instruction	Notes	
0.0		−6660229215		
1		+9621296443		
2		−12495809411		
3		+17499210016		
4		−29165451385		
5		+87496355273		
6	from 1+.7+	3 7 04	$-\lvert A\rvert$	Overwritten by (A/B)
	from 1+.3 or 1+.5	0.6+ 7 60	LOOP STOP if $p = q = 0$	
7		(1) 6 51	$A/2$	Overwritten by $(\lvert a - b\rvert + 2)$
		6+ [1] 72		

6+

Address		Instruction	Notes
1.0		7 6 26	$A/2B$ to PQ
		7 6 00	
1		0 7 00	
		(2) 5 56	Normalize
2		6 6 01	A/B
		(2) 5 41	$(-\lvert a - b\rvert + 2)$ to 5_c
3		0.6 6 10	(A/B) to 0.6
		0.7 5 10	$(-\lvert a - b\rvert + 2)$ to 0.7
4		5 5 02	$\lvert a - b\rvert - 2$
		(37) 5 43	$\lvert a - b\rvert - 39$
5		1.6 5 62	Jump if $\lvert a - b\rvert \geqslant 39$
		(25) 5 50	$5_m = \lvert a - b\rvert - 39$
6		(39) 6 51 5	Form fixed point x between -1.0 and 0
		6 5 00	x to 5
7		2+ [1] 72	
	to 2+.2	1.2 0 60	

2+

1.0	+73653313182	$2 - \sqrt{3}$ $\left[= \tan \frac{\pi}{12} \right]$
1	−79350416784	$-\frac{1}{2\sqrt{3}}$
2	1.0 6 01	$x + \tan(\pi/12)$
	1.7 6 62	Jump if $x < -\tan(\pi/12)$
3	1.1 5 21	Form $X = \frac{x + (1/\sqrt{3})}{1 - (x/\sqrt{3})}$ in 5
	33 6 01	
4	(1) 5 51	
	1.1 5 03	
5	6 5 26	
	7 5 00	
6	32 0 02	Set OVR (as marker)
	(1) 4 43	$4_c = +1$
7	7+ [1] 72	
	1.0 0 60	

from 6+.7+ (enters at 2)

7+

1.0	1.2 0 64	Jump if $x \geqslant -\tan \pi/12$
	0.6 5 10	Replace x by X in 0.6
1	(2) 6 40	Exponent of X = 2 to 0.7
	0.7 6 10	
2	5 5 21	x^2 (or X^2) to 6
	1.0 6 10	to 1.0
3	1.1 5 02	Set $5_p = 3$
	0.0 6 00	First coefficient of series to 6
4	1.0 6 21	Sum Chebycheff series to form $\frac{1}{\pi}$ arctan x (or $\frac{1}{\pi}$ arctan X)
	7.6+ 6 01 5	
5	1.4 5 66	
	0.6 6 21	
6	0.7 5 00	Exponent to 5
	4+ [0] 72	
7	0.4+ 4 67	Jump if $x \geqslant -\tan(\pi/12)$
	0.4 0 60	

to 4+.4+

to 4+.4

4+

	0.0	21 2 52 5. 42 5 25 3	= 1/6
	1	+0	+ 2 × P.P.01 = $- 2^n . 2^{-38}$
	2	+0	+ P.P.01 = $- 2^{n-1} . 2^{-38}$
	3	+215888603272	= $\pi/4$
from 7+.7+ from 7+.7	4	0.0 6 03 0	$\frac{1}{\pi}$ arctan x = $\frac{1}{\pi}$ arctan $X - \frac{1}{6}$
	5	5+ [1] 72 0.7 4 62	 Jump if $\lvert p \rvert > \lvert q \rvert$
	6	6 6 02 33 6 03	$\frac{1}{\pi}$ arctan $\frac{p}{q} = -\frac{1}{2} - \frac{1}{\pi}$ arctan $\frac{q}{p}$
	7	(1) 4 52 1.1 4 62	Shift marker Jump if $q \geqslant 0$

5+

1.0	6 6 02 32 6 06	$\frac{1}{\pi}$ arctan $\frac{p}{q} = -1.0 - \frac{1}{\pi}$ arctan $\frac{q}{p}$
1	1.6+ 6 60 0.3 6 20	Jump if arctan (p/q) = 0 Multiply by $\pi/4$ to form $\frac{1}{4}$ arctan (p/q)
2	(78) 5 56 4 6 12 4	Normalize If $p < 0$, $4_p = 0$, No effect If $p \geqslant 0$, $4_p = 2$, arctan (p/q) = −arctan ($-p/q$)
3	(2) 5 56 0.2 6 03	Re-normalize Round
4	(2) 5 56 0.2 5 03	Re-normalize Pack result in 6 in standard floating-point form
5	0.1 6 05 5 6 01	(Pack result in 6 in standard floating-point form, continued)
6	6 6 00 5 1.7 1 10	Set result zero if floating-point underflow Plant LINK
7	−256	Optional P.P.01 = $-2^{n-1} . 2^{-38}$ where $n = 9$ — Overwritten by LINK

GAUSSIAN QUADRATURE

This subroutine evaluates the integral

$$\int_{X-h}^{X+h} f(x)\ dx$$

where $|h| < 1$. This integral may be expressed † in the form

$$\int_{X-h}^{X+h} f(x)\ dx \ \doteqdot\ \sum_{r=1}^{n} H_r \phi_r$$

where the H_r are the weights of the Gaussian quadrature formula and

$$\phi_r = \tfrac{1}{2}\left\{f(X + ht_r) + f(X - ht_r)\right\}$$

where the t_r are the roots of the Legendre polynomials, $P_n(x)$.

In this subroutine, the roots and weights are stored as positive integers to preserve greater accuracy.

† *"Numerical Analysis" (Chapter VII) – Zdenek Kopal (Chapman and Hall, London, 1955).*

	R 0	0 –0 1
	300	– 28 –
	0+	[0] 72
01	0.1	0 60

LN
GAUSS. QUAD.

R 1	5 –0 2
300	– 02 –

Call for P.P.02

0+

Address	Instruction	Comment
0.0	+ 0	Overwritten by $\int_{X-h}^{X+h} f(x)\,dx$
1 (entry 01)	0 [7] 73	Preserve Accumulators in B0
	6 3 00	h to 3
2	0 2 00	Clear 2
	(0) 4 40	Becomes (n') 4 40. Set by Interlude
3	1+ [0] 72	
	2+ [1] 72	Read first block of roots and weights
4	(1) 6 50	$2h\sum_{r=1}^{n} H_r\phi_r = \int_{X-h}^{X+h} f(x)\,dx$ to 6
	0.5+ 0 64	
5	0.5 0 60	LOOP STOP on OVR
	0.0 6 10	
6	0 [7] 72	Restore Accumulators
	0.0 6 00	$\int_{X-h}^{X+h} f(x)\,dx$ to 6
7	0.7 1 10	Plant LINK and exit
	0.7 0 60	

1+

Address	Instruction	Comment
0.0	1.1 6 21 4	$H_r\phi_r = \frac{1}{2}H_r\left\{f(X + ht_r) + f(X - ht_r)\right\}$ to 6
	6 2 01	$\Sigma\,(H_r\phi_r)$ to 2
1	35 4 01	Step modifier in 4
	0.2+ 4 66	
2	2+ [1] 72 4	Read next block of roots and weights
	0.4 4 67	Count n' points
3	2 3 21	$h\sum_{r=1}^{n} H_r\phi_r$ to 6
	0+ [0] 72	
4	1.0 3 21 4	ht_r (rounded) to 6
	0.6 1 00	Set LINK for return from Auxiliary
5	+ 0	+P.P.02 = Cue to Auxiliary
6	1+ [0] 72	LINK for return from Auxiliary
	0.7 0 60	
7	2+ [1] 72 4	Read roots and weights
	0.0 0 60	

INTERLUDE

T 6+.0

6+

Entry / exit	Address	Instruction	Comment
from Assembly after last list	0.0	35 5 00	
		0.1+ 0 60	
from Assembly	1	0 5 00	Clear 5
		0.5 1 10	Set LINK for return to Assembly in 0.5
	2	0.6 2 00	
(to 7+.1.0)		1.0 2 63	Jump if required list not yet found
from 9+.7+	3	0.4 5 61	Jump if last list
		0.7 2 00	(7+.0, 0) to 2
from 9+.7+ or 10+.5	4	5.7 2 10	Plant new T.A. in 5.7_m
		0.5 1 00	Replace LINK in 1
to Assembly	5	+ 0	Overwritten by LINK for return to Assembly
	6	− 1.0	Overwritten by new T.A.
	7	7+ − 00 0. / 0	= (7+.0, 0)

7+

Entry / exit	Address	Instruction	Comment
(from 6+.2)	1.0	33 4 00	511.0 to 4_m
		34 4 03	
	1	[0 5] 70 4	Index Address to 1
		[1 0] 70 1	Index word to 1
	2	1 2 00	Index word to 2
		(4) 2 45	
(loop)	3	1.3 2 60	LOOP STOP if no parameter list
		33 4 00	Collating mask for last 12 bits of modifier
	4	35 4 03	
		(21) 1 52	Form parameter list address in 1_m and 2_m
	5	4 1 05	
		1 2 00	
	6	[0 0] 70 1	1st word of parameter list to 1 and 3
		1 3 00	
	7	0.7 7 00	7+.0 to 7_m
		3 [4] 72 7	Read B10+ to U4

INTERLUDE (continued)

8+

		Order	Comment
to 10+.1 ←	2.0	4.1 1 62	Jump if a 6, 8, 10, 12 or 16 point formula
		(1) 1 45	+1 to 1 if n odd; 0 to 1 if n even
	1	1 3 04	n' = (n + 1) to 3 if n odd; n' = n to 3 if n even
		3 1 00	n' to 1
	2	35 2 01	Form address of 3rd parameter in 2
		35 2 01	
	3	34 3 01	(8, n') to 3
		0 [1] 72 2	Read first block containing roots and weights
	4	1.0 6 00 2	Copy roots and weights from parameter list into B2+ onwards
		4.0 6 10 3	
	5	2.6 2 66	
		0 [1] 72 2	
	6	2.7+ 3 66	
		0+ [4] 73 3	
	7	1+ [4] 72 3	
		2.4 3 67	

9+

		Order	Comment
	3.0	1+ [4] 73 3	Write last block of roots and weights
		3 [4] 72 7	Read B10+ to U4
	1	34 3 01	(2+.0 + n') to 2_m
		4.0 3 01	
	2	3 2 00	
from 10+.3 →		1 3 00	n' to 3
	3	(1) 1 45	1 to 1 if n' odd; 0 to 1 if n' even
		1 3 01	n' to 3
	4	[0+ 2] 70	Read order-pair in 0+.2 to 1
		(11) 3 52	
	5	3 1 01	Add n' into N position of b-order
		[0+ 2] 71	Restore adjusted order-pair in 0+.2
	6	2 1 00	New T.A. to 1
		4.0 7 00	0+.0 to 7_m
	7	[6 6] 71 7	Plant new T.A. in 6+.6
to 6+.3 or 6+.4 ←		0.3 0 60 5	

INTERLUDE (continued)

10+

Jump	Address	Order	Comment
	4.0	0+ - 00 0.	= (0+.0, 0)
		0	
from 8+.0 →	1	5.7 2 00	Transfer Address to 2
		35 2 03	Reduce T.A. by 0.1
	2	[0 0] 70 2	Read last word of list to 1
		1 3 03	Test list of roots and weights and jump if required
← to 9+.2+	3	3.2+ 3 60	
		4.0 2 00	Set T.A. for next list as 2+.0
	4	34 2 01	
		34 2 01	
← to 6+.4	5	0.4 5 60	Jump if list rejected, unless last list
		4 [1] 72 7	Read B11+ to U1
	6	3 1 01	n' to 1
		1 7 00	n' to 7
	7	(10) 6 40	+10 to 6_c
to 11+.1.0		1.0 0 60	

11+

Jump	Address	Order	Comment
from 10+.7 →	1.0	6 0 24	Arrange to print n'
		1.7 5 00	Printing constant to 5
→	1	16 5 10	Print CRLF * 300 Sp + n
loop to 1.1		(5) 5 53	
	2	1.1 5 61	
to 1.3+		1.3+ 7 60	
	3	16 7 10	
→		16 6 10	
	4	5.7 2 10	Plant T.A. in 5.7_m
		1.6 1 00	LINK to 1
	5	0 0 77	STOP
		0	
	6	59 [3] 72 1	Self modified LINK for return to Initial Orders input
		3.1 0 60	
	7	39 0 00	Constant for printing if no parameter list supplied
		3.4+ 6 67 6	

LISTS OF ROOTS AND WEIGHTS

J560.0+ – 6+.1
+256315268353
+47093317874
+181751852204
+99165386105
+65591142427
+128619202965
+6

J560.0+ – 6+.1
+263962465814
+27825488178
+218986013844
+61127633295
+144457248869
+86231026213
+50422130590
+99693759258
+8

J560.0+ – 6+.1
+267705388118
+18326479577
+237786807609
+41080874044
+186754680092
+60222000768
+119130818817
+74015472214
+40922266697
+81233080341
+10

J560.0+ – 6+.1
+269809332680
+12967457725
+248521859063
+29395258100
+211629235633
+44001995897
+161440729985
+55846237017
+101108752553
+64181939731
+34423897211
+68485018474
+12

J560.0+ – 6+.1
+271964458139
+7463611212
+259642805294
+17112118360
+237942893098
+26156972519
+207643982665
+34257750763
+169840528812
+41120632293
+125898693188
+46497389997
+77406594626
+50193644529
+26116839838
+52075787271
+16

J560.0+ – 6+.0

LINEAR INTERPOLATION

Method

See Section 5 of the Library Specification.

	R 0 0 -0 3
	320 - 28 -
01	0+ [0] 72
	0.2 0 60
02	0+ [0] 72
	0.1 0 60
03	0+ [0] 72
	0.4 0 60

LN
LINEAR INTERPOLATION

R 2 6 -0 2 320 - 06 -	Title of optional parameter list
R 0 0 -0 1 320 - 02 -	Call for P.P. 01
R 1 4 -0 2 320 - 02 -	Call for P.P. 02

0+

Entry	Location	Instruction	Comment
	0.0	+0	+ P.P.01 = Address *B.P.* of index
02 →	1	0.0 7 00	(*B.P*, *0*) to 7
		0.3 0 60	
01 →	2	(25) 7 52	Table number, q, to 7_m
		0.0 7 01	$(B.P + q)$ = address of q^{th} word in index to 7_m
(from 1)	3	0 [1] 72 7	Block containing index to U1
		1.0 7 00 7	Address of table to 7
03 →	4	0 [1] 72 7	1st block of table to U1
		1.7 7 10	Address of table to 1.7
	5	6 7 00 0 6 00	x to PQ
	6	1.2 7 03 6 0 23	$(x - x_0)$ to PQ
	7	1+ [0] 72	
to 1+.0 ←		0.0 0 60	

1+

Address	Jump	Order	Notes
0.0		1.3 6 24	$r = \frac{x - x_o}{h}$ to 5 and 7; $(x - x_r)$ to 6
		7 5 00	
1	$x<x_o$	0.4 7 63	Jump to P.P.02 if x is out of range
		1.1 7 03	$(r - m)$
2	$x>x_m$	0.4 7 62	
		(1) 7 41	$(r - m + 1)$
3	$x = x_r$	0.5 6 60	
	$r \neq m-1$	0.5 7 61	
4		+0	+ P.P.02 (May be loopstop or cue to extrapolation routine)
5		(4) 5 41	
		(25) 5 52	$(A + r + 4)$ to 5_m. (Address of $f(x_r)$)
6		1.7 5 01	
		1.3 6 26	$\delta x = \frac{x - x_r}{h}$ to 7
7		2+ [0] 72	
		0.0 0 60	

2+

Address	Order	Notes
0.0	(0.0 7 12	$-\delta x$ to 0.0 — Overwritten by $-\delta x$
	0)	
1	(0.1 7 10	δx to 0.1 — Overwritten by δx
	0 7 00)	Clear 7
2	0 [1] 72 5	Table containing $f(x_r)$ to U1
	1.0 6 00 5	$f(x_r)$ to 6
3	0.4 5 66	Bring down next block of table if necessary
	0 [1] 72 5	
4	0.0 6 22	$(1 - \delta x)\ f(x_r)$ to 6
	1.0 5 00 5	$f(x_{r+1})$ to 5
5	0.1 5 22	$(1 - \delta x)\ f(x_r) + \delta x\ f(x_{r+1}) = f(x)$ to 6
	0.6 1 10	Plant LINK
6	(2 - 00 0.	Optional P.P.01 = (2.0, 0) Overwritten by LINK
	0)	
7	0.4 0 60	Optional P.P.02 LOOP STOP if x out of range
	0	

L

POLYNOMIAL INTERPOLATION

Method: See Section 6 of the Library Specification.

	R 0 0 -0 3
	321 - 28 -
01	0+ [0] 72
	0.1 0 60
02	0+ [0] 72
	0.2+ 0 60
03	0+ [0] 72
	0.4 0 60

LN
POLYNOMIAL INTERPOLATION

Instruction	Comment
R 0 0 -0 1	Call for P.P.01
321 - 02 -	
R 5 2 -0 2	Call for P.P.02
321 - 02 -	

Entry	Address	Instruction	Comment
		[0+]	**Entry**
	0.0	+0	+ P.P.01 = Address $B.P$ of index
01 →	1	(25) 7 52	Table number, q, to 7_m
		0.0 7 01	$7_m = B.P + q$ (Address of qth word in index)
	2	0.3 0 60 (→ 3)	
02 →		0.0 7 00	$7_m = B.P$
	3	0 [1] 72 7	Read address of table (A, n) from the index
		1.0 7 00 7	
03 →	4	0 [7] 73	Preserve accumulators
		0 [2] 72 7	Read first block of table to U2
	5	5+ [1] 72	
		7 2 00	(A, n)
	6	6 7 00	x to 7
		0 6 00	
	7	6+ [0] 72	
		2.2 7 03	$x - x_0$

Block 5+ — **Check that x is within range**

Label	Order	Comment
1.0	6 0 23	$x - x_0$ to PQ
	2.3 6 24	$r = \frac{x-x_0}{h}$ to 3, $x-x_r$ to 6
1	7 3 00	
	1.3 7 62	Jump if $r \geqslant 0$
2	+0	+ P.P.02 = { LOOP STOP or cue to extrapolation routine
3	2.1 7 03	$r-m$
	1.2 7 62	Jump if $r \geqslant m$
4	(1) 7 41	$r - m + 1$
	1.5+ 6 60	Jump if $x = x_r$
5	1.2 7 60	Jump if $r = m - 1$
	6 5 00	$x-x_r$ to 5
6	2 6 00	(A, n)
	(127) 6 45	n
7	2.0 6 04	$N-n$
	0.1 0 60	

Block 6+ — **Form n', where n' = number of points to be used for interpolation**

Label	Order	Comment
0.0	0 0 40 0.	
	0	
1	0.0 2 01	(A_0, n) to 2
	0.2+ 6 63	Jump if $N < n$
2	2.0 6 13	n to 2.0
	2.0 6 00	$n' = \min(n, N)$ to 6
3	6 1 00	n'
	(9) 6 43	$n' - 9$
4	0.4 6 62	LOOP STOP if $n' > 8$
	(7) 6 41	$n' - 2$
5	0.5 6 63	LOOP STOP if $n' \leqslant 1$
	0 7 00	
6	(1) 0 55	$\frac{1}{2}(n'-2)$ to 6 { If n' even X7 = 0; If n' odd X7 ≠ 0
	3 6 04	$r - \frac{1}{2}(n'-2)$ to 6
7	1+ [0] 72	
	0.0 0 60	→ to 1+.0

Optional Parameter List
(Punched after Block 6+)

Label	Order	Comment
	R 0 0 -0 2	Title
	321 - 06 -	
01	2 - 00 0.	= Address $B.P$ of index
	0	
02	1.2 0 60	LOOP STOP in 1.2 if given x is out of range
	0	

Form s

$s = 0$ if $s_1 < 0$

$s = m-n'$ if $s_1 > m-n'$

$s = s_1$ otherwise

$s_1 = r - \frac{1}{2}(n'-1) + \delta$

$\delta = 1$ if n is odd and $x-x_r \geqslant \frac{1}{2}h$

$\delta = 0$ otherwise.

1+

from 6+.7+ → 0.0

Word	Instruction	Comment
0.0	2+ [1] 72	
	0.3+ 7 60	Jump if n' is even
1	5 7 00	$x-x_r$
	2.3 7 03	$x-x_r-h$
2	5 7 01	$2(x-x_r)-h$
	0.3+ 7 62	Jump if $x-x_r \geqslant \frac{1}{2}h$
3	(1) 6 43	$s_1 = r - \frac{1}{2}(n'-1)$
	1.0 6 63	Jump if $s_1 < 0$, leaving $s = 0$.
4	2.1 1 13	$(m-n')$ to 2.1
	2.1 7 00	$(m-n')$ to 7
5	6 7 03	$(m-n'-s_1)$ to 7
	0.6+ 7 62	Jump if $s_1 \leqslant m-n'$
6	2.1 6 00	$s = m-n'$
	6 3 03	$r-s$
7	35 6 20	$7_m = s$
	7 2 01	(A_s, n) to 2

2+

Word	Instruction	Comment
1.0	2.3 4 00	h
	1 7 00	n'
1	0 [0] 72 2	Read block of table to U0
	0.0 6 00 2	
2	2.0 6 10 7	Transfer f_{s+i} to $2.i$, $i = 0, 1, \ldots, 7$
	1.3+ 2 66	
3	0 [0] 72 2	
	1.1+ 7 66	
4	(1) 1 43	$n'-1$
	1 4 20	$(n'-1)h$
5	3+ [0] 72	
	35 2 00	$2_m = 1$
6	0.0 6 60	Jump if $(n'-1)h$ is single length
	(1) 4 51	$\frac{1}{2}h$
7	(1) 5 51	$\frac{1}{2}(x-x_r)$
	1.4+ 0 60	

← to 3+.0

Inner Loop: Neville's Iteration Process

3+ (from 2+.6)

Address	Instruction	Comment	Note
0.0	0.0 0 10	Clear shift marker	0.0 is used to store shift marker
	4+ [1] 72		
1	3 4 20	$h(r-s)$ to 7	0.1 is used to store $x-x_s$
	7 5 01	$x-x_s$ to 5	
2	0.1 5 10	$x-x_s$ to 0.1	0.2 is used to store $(t+1)h$
	0.2 0 10		
3	1 3 00	$(n'-1-t)$ to 3	
	0.1 5 00	$x-x_s$ to 5	
4	0.2 4 11	$(t+1)h$ to 0.2	
	2.0 6 02 3	$-f^{(t)}_{s+i}$ to 6	
5	2.1 6 01 3	$f^{(t)}_{s+i+1} - f^{(t)}_{s+i}$	
	5 6 20	$\left(f^{(t)}_{s+i+1} - f^{(t)}_{s+i}\right)\left(x-x_{s+i}\right)$	
6	0.2 6 25	$\left(f^{(t)}_{s+i+1} - f^{(t)}_{s+i}\right)\left(x-x_{s+i}\right)/(t+1)h$	
	2.0 7 01 3	$f^{(t+1)}_{s+i}$	
7	1.3 0 64	Jump if OVR clear.	
	0 6 00		

4+

Address	Instruction	Comment
1.0	2.0 7 00 6	Halve all the values of f in U2
	(1) 7 51	
1	2.0 7 10 6	
	1.0 6 66	
2	0.0 2 11	Add 1 to the shift marker
	0.4+ 0 60	
3	2.0 7 10 3	$f^{(t+1)}_{s+i}$ to 2.i
	35 3 01	Add 1 to i
4	4 5 03	$(x-x_{s+i+1})$ to 5
	0.4+ 3 67	Count on i
5	0.3 1 67	Count on t
	0 [7] 72	Restore accumulators
6	2.0 6 00	$f^{(n'+1)}_s$ to 6
	0.0 7 00	Shift marker to 7
7	(0) 6 50 7	Answer, $f(x)$, to 6
	2.0 1 10	Plant LINK and obey it in 2.0

TWO WAY LINEAR INTERPOLATION

Method

See Section 5 of the Library Specification.

	R 0 0 -0 3
	322 - 28 -
01	0+ [0] 72
	0.1+ 0 60
02	0+ [0] 72
	0.1 0 60
03	0+ [0] 72
	0.3+ 0 60

LN
TWO WAY LINEAR INTERPOLATION

R 0 0 -0 1	Call for P.P.01
322 - 02 -	
R 1 2 -0 2	Call for P.P.02
322 - 02 -	

[0+]

Entry	Address	Instruction	Comment	
	0.0	+0	+ P.P.01	Overwritten by LINK
02 →	1	0 5 00		Overwritten by x
01 →		(25) 5 52	k to 5_m	
	2	0.0 5 01	$B.P + k$	Overwritten by y
		0 [1] 72 5	Index to U1	
	3	1.0 5 00 5	A to 5_m	Overwritten by m_y
03 →		0 [2] 72 5	Table to U2	
	4	1+ [1] 72		Overwritten by $f(x_r\ y)$
		0.0 1 10	Plant LINK in 0.0	
	5	0.1 6 10	x to 0.1	Overwritten by $(x-x_r)/h_x$
		0.2 7 10	y to 0.2	
	6	1.5 1 00	Put 6 in 1_p	Overwritten by $(y-y_s)/h_y$
from 1+.6+ →		7.3+ 7 00 1	x or y to 7	
	7	0 6 00		Overwritten by loop marker
		1.6 7 03 1	$x-x_0$ or $y-y_0$ to 7	

1+

Address	Order	Notes
1.0	6 0 23	(1) $x-x_0$ to PQ (2) $y-y_0$ to PQ
	2.0 6 24 1	(1) r to 7; $x-x_r$ to 6 (2) s to 7; $y-y_s$ to 6
1	7.5+ 7 10 1	(1) r to 0.3 (2) s to 0.4
	1.3 7 62	Jump if $r,\ s \geqslant 0$
2	+0	+ P.P.02 } cue to extrapolation routine
3	1.4 7 03 1	(1) $r-m_x$ (2) $s-m_y$
	1.2 7 62	
4	(1) 7 41	(1) $r-m_x+1$ (2) $s-m_y+1$
	1.5+ 6 60	Jump if $x = x_r$ or $y = y_s$
5	1.2 7 60	
	2.0 6 26 1	} (1) $(x-x_r)/h_x$ to 0.5 (2) $(y-y_s)/h_y$ to 0.6
6	7.7+ 7 10 1	
	0.6+ 1 66	Jump if first time round loop (to 0+.6+)
7	2+ [1] 72	
	1.2+ 0 60	(to 2+.2+)

3+

Address	Order	Notes
1.0	0.7 0 10	Indicate second time round loop
	0.4 6 10	$f(x_r\ y)$ to 0.4
1	0 [2] 72 5	(from 2+.7+)
	0.6 6 02	$-(y-y_s)/h_y$
2	2.0 6 20 5	
	2.0 6 01 5	$\{1-(y-y_s)/h_y\}f(x_i\ y_s)$
3	1.4 5 66	
	0 [2] 72 5	
4	2.0 1 00 5	$f(x_i\ y_{s+1})$
	0.6 1 22	$f(x_i\ y)$
5	0.3 5 01	} $A_{r+1,s}$ to 5_m
	35 5 03	
6	0.7 1 00	
	1.0 1 61	Jump if first time round loop
7	6 5 00	$f(x_{r+1}\ y)$ to 5
	2+ [2] 72	

2+

Entry/exit	Word	Order	Notes
from 3+.7+ →	2.0	0.5 6 02	$-(x-x_r)/h_x$ to 6
		0.4 6 20	$\{1-(x-x_r)/h_x\}f(x_r\ y)$
	1	0.4 6 01	
		0.5 5 22	$f(x\ y)$ to 6
	2	0.0 0 60	Jump to obey the LINK
from 1+.7+ → / to 1.2+		0.3 7 00	r to 7
	1.3	34 5 01	A_{00} to 5_m
		2.3 7 20	$m_y\ r$ to 7
	4	0.4 7 01	$m_y\ r+s$ to 7_c
		35 7 20	$m_y\ r+s$ to 7_m
	5	7 5 01	A_{rs} to 5_m
		2.3 7 00	m_y to 7_c
	6	35 7 20	m_y to 7_m
		0.3 7 10	m_y to 0.3_m
	7	3+ [1] 72	
← to 3+.1		1.1 0 60	

Optional Parameter List
(Punched after Block 3+)

Tape	Notes
R 0 0 -0 2 322 - 06 -	Title
2 - 00 0. 0	P.P.01
1.2 0 60 0	P.P.02

TWO WAY POLYNOMIAL INTERPOLATION

Method

See Section 6 of the Library Specification.

	R 0 0 -0 3
	323 - 28 -
01	0+ [0] 72
	0.2+ 0 60
02	0+ [0] 72
	0.2 0 60
03	0+ [0] 72
	0.4+ 0 60

LN
2 WAY POL. INT

R 0 0 -0 2	Call for P.P.02
323 - 02 -	
R 0 1 -0 1	Call for P.P.01
323 - 02 -	

[0+]

Entry	Address	Instruction	Comment
	0.0	+0	+ P.P.02 Extrapolation Cue
	1	+0	+ P.P.01 Address of Index
(02) →	2	0 5 00	
(01) →		(25) 5 52	Table number k, to 5_m
	3	0.1 5 01	$5_m = B.P + k$
		0 [1] 72 5	Read Index to U1
	4	1.0 5 00 5	Address of Table to 5_m
(03) →		0 [7] 73	Write Accumulators to B0
	5	0 [3] 72 5	Read 1st Block of Table to U3
		1+ [1] 72	
	6	0.1 6 10	x to 0.1
		5 3 00	(A, n) to 3
	7	(127) 3 45	n to 3_c
		3.0 3 03	$n - N$

1+

Form n' = number of points to be used for interpolation

	Order	Comment
1.0	3 3 06 3	$n - N$ if $n < N$; 0 if $n \geqslant N$
	3.0 3 01	$n' = \min(N, n)$ to 3
1	3 1 00	n' to 1
	(9) 1 43	$n' - 9$
2	1.2 1 62	STOP if $n' > 8$
	(7) 1 41	$n' - 2$
3	1.3 1 63	STOP if $n' < 2$
	1.2 2 00	6 to 2_p
4	0.2 7 10	y to 0.2
	7.3+ 7 00 2	(1) x to 7 (2) y to 7
5	0 6 00	
	2+ [2] 72	
6	2.6 7 03 2	(1) $x-x_0$ (2) $y-y_0$
	6 0 23	
7	3.0 6 24 2	(1) $(x-x_0)/h_x$ (2) $(y-y_0)/h_y$
	7.5+ 7 10 2	(1) r_x to 0.3 (2) r_y to 0.4

2+

Check that x is within range

Jump		Order	Comment
to 0+.0	2.0	0.0 7 63	Extrapolate if $r < 0$
		2.4 7 03 2	$r - m$
to 0+.0	1	0.0 7 62	Extrapolate if $r > m$
		(1) 7 41	$r - m + 1$
	2	2.3 6 60	Jump if $(x-x_r)$ or $(y-y_r) = 0$
to 0+.0		0.0 7 60	Extrapolate if $r = m$
	3	7.7+ 6 10 2	(1) $x-x_r$ to 0.5 (2) $y-y_r$ to 0.6
		0 7 00	
	4	1 6 00	$n' - 2$
		(1) 0 55	$\frac{1}{2}(n'-2)$ to 6; 0 to 7 if n' even
	5	6+ [1] 72	
		7.5+ 6 04 2	$r - \frac{1}{2}(n'-2)$
	6	7.7+ 4 00 2	(1) $x-x_r$ to 4 (2) $y-y_r$ to 4
		7.3+ 0 10 2	Set s initially zero
	7	7+ [2] 72	
to 6+.0		1.0 0 60	

Form s, where $s = 0$ if $s_1 < 0$
$s = m - n'$ if $s_1 > m - n'$
$s = s_1$ otherwise.

$$s_1 = r - \tfrac{1}{2}(n' - 1) + \delta$$

where $\delta = 1$ if n is odd and $x - x_r \geqslant \tfrac{1}{2} h$; $\delta = 0$ otherwise.

6+ (from 2+.7+)

Addr	Instruction	Comment
1.0	1.2+ 7 60	Jump if n' even
	3.0 4 03 2	(1) $x-x_r-h_x$ (2) $y-y_r-h_y$
1	7.7+ 4 01 2	(1) $2(x-x_r)-h_x$ (2) $2(y-y_r)-h_y$
	1.2+ 4 62	Jump if $x-x_r > \frac{1}{2} h_x$; $y-y_r > \frac{1}{2} h_y$
2	(1) 6 43	$s' = r - \frac{1}{2}(n' - 1)$
	1.6+ 6 63	Jump if $s' < 0$, leaving s zero
3	3 7 00	n'
	2.4 7 04 2	$m - n'$
4	6 7 03	$m - n' - s'$
	1.5+ 7 62	Jump if $m \geqslant n' + s'$
5	7 6 01	$m - n'$
	7.3+ 6 10 2	(1) s_x to 0.1 (2) s_y to 0.2
6	7.5+ 6 13 2	(1) $r_x - s_x$ to 0.3 (2) $r_y - s_y$ to 0.4
	3.0 3 20 2	(1) $n'h_x$ (2) $n'h_y$
7	3.0 4 00 2	(1) h_x (2) h_y
	7.7+ 7 00 2	(1) $x-x_r$ to 7 (2) $y-y_r$ to 7

7+

Addr	Instruction	Comment
2.0	2.1+ 6 60	Jump if $n'h$ is single length
	(1) 4 51	(1) $\frac{1}{2} h_x$ (2) $\frac{1}{2} h_y$
1	(1) 7 51	(1) $\frac{1}{2}(x-x_r)$ (2) $\frac{1}{2}(y-y_r)$
	7.7+ 4 10 2	(1) h_x to 0.5 (2) h_y to 0.6
2	7.5+ 4 22 2	(1) $x-x_r + (r-s)h$ (2) $y-y_r + (r-s)h$
	7.5+ 7 10 2	(1) $x-x_s$ to 0.3 (2) $y-y_s$ to 0.4
3	2.7 2 66	Jump to form $y-y_s$
	3.3 2 00	m_y to 2
4	0.1 2 20	$m_y s_x$
	0.2 7 01	$m_y s_x + s_y$
5	35 7 20	$(A_{s_x s_y} - 1.0)$ to 5_m
	7 5 01	
6	4+ [2] 72	
	2.0 0 60	to 4+.0
7	1+ [1] 72	Return to B1+ to form s_y, $y-y_s$ etc.
	1.4+ 0 60	

from 9+.7 → 1.0

	3+	
1.0	4+ [2] 72	
	0.7 6 00	Stage Modifier to 6
1	1.6 3 67	Count n'
	1.4 3 61	Jump to interpolate in x
2	3.0 1 10	Answer to 3.0
	0 [7] 72	Restore Accumulators
3	3.0 6 00	Answer to 6
	1.4 1 10	Plant LINK
4	(5+ [3] 72	} Overwritten by LINK
	(1) 3 40)	(0, 1) to 3 } Overwritten by LINK
5	3.0 1 10 6	$f(x_{s+n}\ y)$ to 3.n
	2.6+ 0 60	
6	[5+ 0] 71 6	$f(x_{s+i}\ y)$ to 5+.i
	0.1 5 00	} Add m_y to form address of $f(x_{s+i+1}y_s)$
7	0.0 5 01	}
	2.2 6 66	Increase Stage Modifier

from 7+.6+ → 2.0

	4+	
2.0	35 2 20	} m_y to 0.0_m
	0.0 7 10	}
1	3 6 00	n' to 6_c
	35 3 01	(0.1, n') to 3
2	0.7 6 10	(Stage Modifier, n') to 0.7
	0.1 5 10	Table Address to 0.1
3	1 [1] 72 5	} Transfer n' consecutive values from Table to U3
	1.0 7 00 5	}
4	3.0 7 10 2	}
	2.5+ 5 66	}
5	1 [1] 72 5	}
	35 2 01	}
6	2.3+ 6 67	}
	0.7 2 00	n' to 2_c
7	8+ [1] 72	
	1.0 2 67	$n' - 1$ to 2_c

to 8+.0

Optional Parameter List
(Punched after Block 4+)

	R 0 0 -0 1	} Title
	323 - 06 -	}
01	2 - 00 0.	= (2.0, 0) Address of the index
	0	
02	0.0 0 60	LOOP STOP
	0	

Inner loop: Neville's Iteration Process

Address	Instruction	Comment
	8+	
1.0	(127) 2 45	(0, $n' - 1$) to 2
	0 4 00	
1	0.2 0 10	Make Shift Marker zero
	9+ [2] 72	
2	2 1 00	$n' - 1 - t$
	0.3 5 00 3	$y - y_s$ or $x - x_s$ to 5
3	0.5 4 01 3	$(t + 1)h$
	3.0 6 02 1	$- z_{s+i}^{(t)}$
4	3.1 6 01 1	$z_{s+i+1}^{(t)} - z_{s+i}^{(t)}$
	5 6 20	$(z_{s+i+1}^{(t)} - z_{s+i}^{(t)})\ (y - y_{s+i})$
5	4 6 25	$(z_{s+i+1}^{(t)} - z_{s+i}^{(t)})\ (y - y_{s+i})/(t + 1)h$
	3.0 7 01 1	$z_{s+i}^{(t+1)}$
6	2.2+ 0 64	Jump unless OVR set
	0 6 00	
7	3.0 7 00 6	
	(1) 7 51	
	9+	Halve all values of z in U3
2.0	3.0 7 10 6	
	1.7 6 66	
1	35 6 00	
	0.2 6 11	Add 1 to Shift Marker
2	1.3+ 0 60	
	3.0 7 10 1	$z_{s+i}^{(t+1)}$
3	35 1 01	
	0.5 5 03 3	$y - y_{s+i+1}$ or $x - x_{s+i+1}$
4	1.3+ 1 67	Count on i
	1.2 2 67	Count on t
5	7 1 00	Interpolate z to 1
	0.2 7 00	Shift Marker to 7
6	3+ [1] 72	
	(0) 1 50 7	Shift up z if necessary
7	1.0 0 64	
	2.7+ 0 60	Stop if $\lvert z \rvert > 1.0$

L

POLYNOMIAL INTERPOLATION (UNEQUAL INTERVALS)

This subroutine estimates the value of $y = f(x)$, for given x, by interpolating in a table of values of $f(x)$ given at values of x which are not necessarily equidistant.

The method used is Neville's iterative process which is described in Section 6 of the Pegasus Library Specification of R 327.

	R 0 0 -0 2
	327 - 28 -
01	0+ [0] 72
	0.1 0 60
02	0+ [0] 72
	0.3 0 60

LN
POLYNOMIAL INTERPOLATION (UNEQUAL INTERVALS) MK 3

R 0 0 -0 1 327 - 02 -	Call for P.P.01
R 4 7 -0 2 327 - 02 -	Call for P.P.02

0+

Link	Address	Instruction	Comment
			Entry
	0.0	+0	+ P.P.01 = (*B.P*, 0)
01 →	1	35 7 22	q to 7_m
		0.0 7 01	Address of qth entry in index to 7_m
	2	0 [1] 72 7	$7_m = A_q$ (address of qth table);
		1.0 7 00 7	$7_c = n_q$ (no. of points for interpolation)
02 →	3	0 [7] 73	Store Accumulators
		0 [3] 72 7	1st block of table to U3
	4	3.0 5 00	m to 5_c (m = no. of values tabulated)
		(3) 4 40	$7_m = A_q + 3$
	5	35 4 22	
		7 2 00	$2_m = A_q + 3 \quad = A(x_0)$
	6	35 5 22	$7_m = A_q + m + 3 = A(y_0)$
		1+ [1] 72	
	7	7+ [0] 72	
to 7+.0 (sheet 4) ←		0.0 0 60	

1+

Form n', where n' = number of points to be used for interpolation

Link	Address	Instruction	Comment
from 7+.0+ or 7+.2 (sheet 4) →	1.0	3+ [2] 72	Overwritten by n'
		3.1 1 00 4	N to 1_c
	1	(127) 7 45	n to 7_c (Overwritten by m)
		1 7 04	$(N - n)$ to 7_c (Overwritten by m)
	2	4+ [3] 72	
		1.3+ 7 63	Jump if $N < n$
	3	7 1 03	Replace N in 1_c by n
		1.0 1 10	Store n' in 1.0
	4	1.1 5 10	Store m in 1.1
		(9) 1 43	$n' - 9$
	5	1.5 1 62	LOOP STOP if $n' \geqslant 9$
		(7) 1 41	$n' - 2$
	6	1.6 1 63	LOOP STOP if $n' < 2$
		(3) 1 41	$n' + 1$ to 1_c
	7	(1) 1 53	$1_m = \frac{n' + 1}{2} = r$ (say)
		(25) 1 52	

Check that x is within range

3+

Label	Word	Order	Comment
2.0	0	(1) 6 51	$x/2$ to 6
		0 [0]72 2	Block containing x_0 to U0
	1	2 4 00	$4_m = A(x_0)$
		0.0 7 00 4	x_i to 7
	2	(1) 7 51	$(x_i/2)$ to 7
		6 7 04	$(x/2) - (x_i/2)$ to 7
	3	2.5+ 7 63	Jump if $x < x_i$
		2.4+ 4 66	Step modifier in 4
	4	0 [0]72 4	Read new block of x where necessary
		2.1+ 5 67	Count on m
	5	3.7 7 61	Jump if $x > x_{m-1}$ (out of range)
		2 4 03	$i = A(x) - A(x_0)$ to 4
	6	3.7 4 60	Jump if $x < x_0$ (out of range)
		1 4 03	$i - r$ to 4
	7	3.0 4 62	Jump if $i \geqslant r$
		0 4 00	Set $s = 0$ if $i < r$

Form s

4+

Label	Word	Order	Comment
3.0	0	1.0 5 03	$m - i - n$ to 5_c
		(0) 5 41 1	$m - i - n + r$ to 5_c
	1	3.3 5 62	Jump if $i - r \leqslant m - n$
		1.1 4 00	Set $s = m - n$ if $i - r > m - n$
	2	1.0 4 03	(as above)
		(25) 4 52	(as above)
	3	4 2 01	$2_m = A(x_s)$
		4 3 01	$3_m = A(y_s)$
	4	1.0 1 00	n' to 1
		2+ [1]72	
	5	35 4 02	Set marker for transfer; $4_m = -0.1$
		1 5 00	n' to 5 (from 2+.4+ (sheet 4))
	6	0 [0]72 3	Block containing y_s (or x_s) to U0
		1.0 0 60	(to 2+.0 (sheet 4))
	7	+0	+ P.P.02 { Loop stop or cue to extrapolation routine if out of range

2+

Connection	Label	Order	Comment
from 4+.6+ (sheet 3)	1.0	0.0 7 00 3	x_{s+i} (y_{s+i}) to 7
		1.5 4 60	Jump if x
	1	2.0 7 10 5	y_{s+i} to 2.i
		1.2+ 3 66	Step modifier in 3
	2	0 [0] 72 3	Next block containing y_s (or x_s) to U0
		35 5 01	Step 5_m
	3	1.0 5 67	Count on n'
		1.7 4 66	Jump and set shift unit second time
	4	2 3 00	$A(x_s)$ to 3
to 4+.5+ (sheet 3)		3.5+ 0 60	Jump to read y_s
	5	(1) 7 51	$1/2(x_{s+i} - x)$ to 3.i
		6 7 03	
	6	3.0 7 10 5	
		1.1+ 0 60	
	7	5+ [0] 72	
to 5+.0 (sheet 5)		0.0 1 67	Jump and set $(n' - 1)$ in 1

7+

Connection	Label	Order	Comment
from 0+.7+ (sheet 2)	0.0	7 3 00	$3_m = A_q + m + 3 = A(y_0)$
		1.0 1 63	Jump if not inverse interpolation
to 1+.0 (sheet 2)	1	2 3 00	Interchange addresses of x_0 and y_0 for inverse interpolation
		7 2 00	
	2	1.0 4 66	$4_m = 0.1$ for inverse interpolation
		0	
		R 0 0 -0 2	Title of Optional Parameter List
		327 - 06 -	
	0.3	2 - 00 0.	Optional P.P.01 { Address of Index B.P = 2.0
		0	
	0.4	3.7 0 60	Optional P.P.02 { LOOP STOP if out of range
		0	

Inner Loop: Neville's Iteration Process

5+ (from 2+.7+)

Label	Order	Comment
0.0	(0.0 0 10	Set initial shift marker in 0.0 } Overwritten by shift marker
	6+ [1] 72)	
1	1 2 00	
	0 3 00	Set $3_m = 0$
2	3.0 6 02 3	}
	2.1 6 20 3	}
3	3.1 5 00 2	}
	2.0 5 22 3	} y
4	3.0 5 03 3	}
	5 6 25	}
5	1.1 0 64	Jump if OVR clear
	0 5 00	Set $5_m = 0$
6	2.0 6 00 5	}
	(1) 6 51	} Divide each value of y by 2
7	2.0 6 10 5	}
	0.6 5 66	}

6+

Label	Order	Comment
1.0	0.0 4 11	Add 1 to shift marker in 0.0_m
	0.2 0 60	
1	2.0 7 10 3	$y_{s+i}^{(t+1)}$ to $2.i$
	35 3 01	Step y modifier
2	35 2 01	Step x modifier
	0.2 2 67	Count on i
3	35 1 01	Step t
	0.1 1 67	Count on t
4	0 [7] 72	Restore Accumulators
	2.0 6 00	$y_s^{(n'-1)}$ to 6
5	0.0 7 00	Shift marker to 7
	(0) 6 50 7	Scale up $y_s^{(n'-1)}$ if necessary
6	1.7 1 63	Jump if LINK is a go order-pair
	1.7 1 12	Plant LINK negatively otherwise
7	(1.7 1 10	} Overwritten by LINK
	1.7 0 60)	}

London Computer Centre,
21, Portland Place,
LONDON, W.1.

Issue 1
3rd August, 1960.
G.H.L.B. D.M.

DETERMINATION OF THE ZERO OF A FUNCTION

Method

See Section 4 of the Library Specification.

	R 0 0 -0 1	
	340 - 28 -	
01	3+ [0] 72	
	0.0 0 60	

LN
ZERO OF F(x)

R 4 3 -0 2		Title of Optional Parameter List
340 - 06 -		
R 4 1 -0 1		Call for P.P.01
340 - 02 -		
R 3 3 -0 2		Call for P.P.02
340 - 02 -		

3+

Entry	Address	Instruction	Comment
(01) →	0.0	4+ [1] 72	
		1 [7] 73	Preserve Accumulators in B1
	1	1.4 6 10	x_a to 1.4
		1.5 7 10	x_b to 1.5
	2	0.7 1 00	Set LINK 1 for Auxiliary
		4+ [1] 73	Preserve U1
to Auxiliary ←	3	+0	+ P.P.02 = cue to Auxiliary
from Auxiliary (1st time) →	4	4+ [1] 72	Restore U1
		1.6 6 10	F_a to 1.6
	5	1.2 1 00	Set LINK 2
		1.5 6 00	x_b to 6
from 2+.5+ →	6	0.2+ 0 60	Enter Auxiliary
		0	
	7	3+ [0] 72	LINK 1 for Auxiliary
		0.4 0 60	

Block 4+

Link	Word	Orders	Notes	Overwritten by
from 1+.7 →	0 (1.0)	2+ [0] 72	Start next cycle	
← to 2+.1		0.1 0 60		
	1	+0	+ P.P.01 = ϵ	
	2	2+ [0] 72	LINK 2 for Auxiliary	
		0.0 0 60		
	3	+4	Optional P.P.01 = ϵ	x_c
	4	0.7 2 10	Optional P.P.0.2 = cue to Auxiliary	x_a
		0.7 0 60		
	5			x_b
	6			F_a
	7			F_b

Block 2+

Link	Word	Orders	Notes
Auxiliary (2nd Time) →	0 (0.0)	4+ [1] 72	Restore U1
		1.7 6 10	F_b to 1.7
Next cycle from 4+.0+ →	1	1.7 6 00	F_b to 6
		1.4 6 20	x_aF_b
	2	1.6 5 02	$-F_a$ to 5
		1.5 5 22	$x_aF_b - x_bF_a$
	3	1.7 5 01	$F_b - F_a$
		5 6 25	$x_c = (x_aF_b - x_bF_a)/(F_b - F_a)$ to 7
	4	1.3 7 10	x_c to 1.3 and 6
		7 6 00	
	5	0.6 1 00	Set LINK 3 for Auxiliary
← to 3+.6		3+ [0] 72	Enter Auxiliary
	6	1+ [0] 72	LINK 3 for Auxiliary
		0.0 0 60	
	7	+0	Not used

1+

Auxiliary (3rd time) → 0.0

Address	Instruction	Notes	
0.0	4+ [1] 72	Restore U1	
	0.3 6 60	Exit if $F_c = 0$	
1	6 5 00	F_c to 5	
	1.6 7 00	F_a to 7	
2	1.6 6 06	C(6) < 0 if F_a and F_c have opposite signs	
	0+ [0] 72		
3	1 [7] 72	Restore Accumulators	EXIT
	1.3 6 00	x_c to 6	EXIT
4	0.7 1 10	Plant and obey Main LINK	EXIT
	0.7 0 60		EXIT
5	1.0 0 65	Next cycle if OVR (Next cycle to 4+.0)	
	0.6+ 6 63	Form $\epsilon - \lvert x_c - x_b \rvert$	
6	6 6 02		
	1.1 6 01		
7	(1.0 6 63	Next cycle if $\lvert x_c - x_b \rvert > \epsilon$ (Next cycle to 4+.0)	Overwritten by Main LINK
	0.3 0 60)	Otherwise EXIT.	Overwritten by Main LINK

0+

Address	Instruction	Notes	
0.0	+0	Not used	
1	1.7 7 20	F_aF_b to PQ	F_a and F_c have same sign
	1.6 5 11	$F_a + F_c$ to 1.6	F_a and F_c have same sign
2	1.6 6 25	$F'_b = \dfrac{F_a F_b}{F_a + F_c}$	F_a and F_c have same sign
	0.4+ 0 60		
3	0.1 6 62		F_a and F_c have opposite signs
	1.4 6 00	$x'_b = x_a$	F_a and F_c have opposite signs
4	1.5 6 10		F_a and F_c have opposite signs
	1.7 7 10	$F'_b = \begin{cases} F_a \\ \text{or as above} \end{cases}$	F_a and F_c have opposite signs
5	1.3 6 00	x_c to 6	
	1.4 6 10	$x'_a = x_c$	
6	1.6 5 10	$F'_a = F_c$	
	1.5 6 03	$x_c - x_b$	
7	1+ [0] 72		
	0.5 0 60		

LINEAR INVERSE INTERPOLATION

This subroutine estimates the value of the argument x corresponding to a given value $y = f(x)$ of a function $f(x)$, tabulated at equal intervals of x.

The subroutine searches through the table, starting at $y_0 = f(x_0)$, until it finds two adjacent entries such that

$$y_r < y \leqslant y_{r+1}$$

$$\text{or} \quad y_r > y \geqslant y_{r+1}$$

If there are two or more values of r for which one of these conditions is true, the smaller value of r will be selected. The required value of x is given by

$$x = x_{r+1} - \frac{y_{r+1} - y}{y_{r+1} - y_r} \cdot h$$

where h is the tabular interval $(x_{i+1} - x_i)$.

	R 0 0 -0 3 341 - 28 -
01	0+ [0] 72 0.1+ 0 60
02	0+ [0] 72 0.1 0 60
03	0+ [0] 72 0.3+ 0 60

LN
LIN. INVERSE INT.

R 0 0 -0 1 341 - 02 -	Call for P.P.01
R 3 6 -0 2 341 - 02 -	Call for P.P.02
R 1 6 -0 2 341 - 06 -	Title of Optional Parameter List

ENTRY

0+

Entry	Address	Instruction	Notes
	0.0	+0	+P.P.01 = Address $B.P$ of index
02 →	1	0 7 00	Clear 7
01 →		(25) 7 52	Table number, q, to 7_m
	2	0.0 7 01	$(B.P + q)$ = address of q^{th} word in index to 7_m
		0 [1] 72 7	Block containing index to U1
	3	1.0 7 00 7	(A,n) to 7
03 →		0 [7] 73	Store the Accumulators
	4	0 [1] 72 7	1st block of table to U1
		0 4 00	Clear 4
	5	1.4 2 00	$y_0 = f(x_0)$ to 2
		6 2 03	$y_0 - y$ to 2
	6	1.3 3 00	h to 3
		35 7 01	$7_m = A + 1$
	7	1+ [0] 72	
(to 1+.0.1)		0.1 7 66	$7_m = A + 2$. Unconditional jump

INITIAL SETTINGS

1+

Entry	Address	Instruction	Notes
from 2+.7+ →	1.0	0.0 6 00	x to 6
		1.1 1 10	Plant LINK and Exit
	0.1	0.5 7 05	Clear 7_c
		0.2+ 2 63	If $y_0 < y$ leave 0 in 4
	2	(16) 4 46	If $y_0 \geqslant y$ set 2^{-34} in 4
		0.2 0 65	Correct 4 if $y_0 - y$ has overflowed
	3	1.2 2 00	x_0 to 2
		1.1 7 01	m to 7_c
	4	3+ [0] 72	
to 3+.0 ←		0.0 7 66	$7_m = A + 3$. Unconditional jump
	5	7.7 7 70 0.	Collating Mask
		0	
	6	2 − 00 0.	Optional P.P.01; Index address = 2.0
		0	
	7	0.6 0 60	Optional P.P.02; Loop stop if $f(x)$ out of range of table
		0	

INTERPOLATE

2+

	Order	Notes
0.0	35 7 03	$A + 4 + r$ to 7_m $[= A(x_0) + r]$
	7 1 04	$r + 1$ to 1_m
1	1.0 5 02 7	$y - y_r$ to 5 unless y_r at end of block
	0.2+ 7 66	Jump unless at end of block, $A(x_{r+1})$ to 7.
2	4 5 02	$(y - y_r)$ to 5
	1.0 5 01 7	$(y_{r+1} - y_r)$ to 5
3	1.0 3 20 7	$(y_{r+1} - y).h$ to PQ
	5 6 25	$- \frac{(y_{r+1} - y)}{(y_{r+1} - y_r)} \cdot h$ to 0.0
4	0.0 7 12	
	1+ [1] 72	
5	1.5 1 05	Clear 1_c; $1_m = (r + 1)$
	1 3 20	$(r + 1)h.2^{-13}$ to PQ
6	35 2 22	$x_{r+1}.2^{-13}$ to PQ
	35 6 25	x_{r+1} to 7
7	0.0 7 11	x to 0.0
to 1+.0 ←	0 [7] 72	Restore Accumulators

3+

SEARCH FOR y_r

	Order	Notes
from 1+.4+ → 0.0	7 1 00	$(A + 3, m)$ to 1
	0.3 4 16	Change 0.3+ to 1.0 5 02 7 if $y_0 \geqslant y$
1	0.3 7 66	Unit modify. $7_m = A + 3 + i + 1$
	1.7 4 00	$y_i - y$ to 4 at end of block
2	0 [1] 72 7	Next block of table to U1
	-0	
3	1.0 6 13 7	$y_{i+1} - y$ } If $y_0 \geqslant y$ b-order is 1.0 5 02 7
	1.0 5 00 7	$\pm(y_{i+1} - y)$ }
4	0.5 0 65	Jump if OVR set
	0.7 5 62	Jump when y lies between y_r and y_{r+1}
5	0.1 1 67	Count entries in table (m)
	35 1 01	Form $(A + 0.4, 0)$ in 1 if $f(x)$ outside range of table
6	+0	+P.P.02 } May be loop stop or cue to extrapolation routine
7	2+ [0] 72	y_r found
	0.0 0 60	Enter interpolation sequence

FLOATING-POINT ARITHMETIC

The subroutine is self-preserving and carries out the operations of addition, subtraction, multiplication and division, on the packed floating-point numbers $x = A.2^a$ and $y = B.2^b$, leaving the result in X6.

	R 0 0 -0 2
	610 - 28 -
01	0+ 0 00 0.
	0
02	0
	0+ 0 00 0.

LN
FLOATING-POINT ARITHMETIC

R 0 5 -0 1	Calls for P.P.01
610 - 02 -	
R 0 6 -0 1	
610 - 02 -	
R 0 6 -0 1	
610 - 02 -	
R 1 7 -0 2	Call for P.P.02
610 - 02 -	
R 2 0 -0 3	Call for P.P.03
610 - 02 -	

On entry 1_m = 0 for × and −
= 1 for ÷ and +

0+

Entry	Address	Instruction	Comment
+ −	0	32 1 01	Make 1 negative to indicate + or −
× ÷	3.0	6 4 00	(UNPACK OPERANDS)
	1	3.6 6 05	A to 6 (UNPACK OPERANDS)
		6 4 03	$a + 2^{n-1}$ to 4 (UNPACK OPERANDS)
	2	7 5 00	(UNPACK OPERANDS)
		3.6 7 05	B to 7 (UNPACK OPERANDS)
	3	7 5 03	$b + 2^{n-1}$ to 5 (UNPACK OPERANDS)
		4.2+ 1 63 1	Jump if + or −
	4	3.5 5 01	b to 5 (× or ÷)
		3.7 0 60 1	(× or ÷)
	5	+0	+ P.P.01 = $-2^{n-1}.2^{-38}$
	6	+0	+ (P.P.01) × 2 = $-2^{n}.2^{-38}$
×	7	7 6 20	AB to PQ (×)
		4.2 0 60	(×)

1+

Entry	Address	Instruction	Comment
÷	4.0	7 7 01	$2B$ to 7 (÷)
		7 6 26	$A/2B$ to 6 (÷)
	1	7 6 00	Set OVR if B = 0 (÷)
		(1) 5 44	1 - b to 5 (÷)
	2	5 4 01	$a+b+2^{n-1}$ if ×; $a-b+1+2^{n-1}$ if ÷ (÷; ×)
− (to 2+.1+)		5.1+ 1 62	Jump if × or ÷ (÷; ×; −)
	3	7 7 02	Change sign of B (−)
+		4 5 04	$a-b$ to 5 (+ & −)
	4	4.7 5 62	Jump if $a \geqslant b$; in this case $d = a-b \geqslant 0$ (+ & −)
		5 4 03	$b + 2^{n-1}$ to 4 (If $a < b$; + & −)
	5	7 6 06	Interchange A and B (If $a < b$; + & −)
		6 7 06	Interchange A and B (If $a < b$; + & −)
	6	7 6 06	(If $a < b$; + & −)
		5 5 02	$d = b-a > 0$ to 5 (If $a < b$; + & −)
	7	(0) 5 43 0.	+ P.P.02; $d-(37-n)$ to 5 (+ & −)
to 2+.0+		5.0+ 5 62	Jump if $d \geqslant (37-n)$

* $k = \min(d, 37-n) \geqslant 0$

2+

Entry	Word	Order	Comment
	0	(25) 5 50	$d-(37-n)$ to 5_m (+ & −)
from 1+.7+	5.0	(0) 7 51 5.	+ P.P.03 Shift argument down k* places (+ & −)
	1	7 6 01	Add arguments (+ & −)
from 1+.2+		5.7 6 60	Jump if argument = 0 (NORMALIZE AND PACK RESULT)
	2	(36) 4 56	Normalize (NORMALIZE AND PACK RESULT)
		3.5 6 03	Round argument (NORMALIZE AND PACK RESULT)
	3	(2) 4 56	Re-normalize (NORMALIZE AND PACK RESULT)
		5.6+ 4 63	Jump if underflow (NORMALIZE AND PACK RESULT)
	4	3.6 6 05	Pack result in 6 (NORMALIZE AND PACK RESULT)
		4 6 01	Pack result in 6 (NORMALIZE AND PACK RESULT)
	5	3.6 4 05	Jump if no overflow
		5.7 4 60	Jump if no overflow
	6	32 0 02	Set OVR if overflow
		0 6 00	Set result = 0
	7	(25) 1 52	EXIT
		0 0 60 1	EXIT

3+

Optional Parameter-list

Word	Contents	Comment
	R 0 0 -0 3	Title
	610 - 06 -	Title
01	-256	$= -2^{n-1}.2^{-38}$ ($n = 9$)
02	28 0 00	$(37-n)$ ($n = 9$)
	0	
03	0	
	28 0 00	$(37-n)$ ($n = 9$)

L

FLOATING-POINT SQUARE ROOT

Method

This subroutine evaluates $F'(6) = \sqrt{F(6)}$ where $F(6) = x = A.2^a$ is the standard floating-point number held in X6.

If we put $A' = A$ and $a' = a$ for a even

and $A' = 2A$ and $a' = a-1$ for a odd,

then $\sqrt{x} = \sqrt{A'}.2^{\frac{1}{2}a'}$.

An iterative Newton-Raphson process is used to find $y = \sqrt{A'}$.

$$y_{n+1} = y_n + d_n \quad \text{where} \quad d_n = \tfrac{1}{2}\left(\frac{A'}{y_n} - y_n\right)$$

$$\text{with} \quad y_0 = \tfrac{1}{2}A' + \tfrac{1}{2}$$

The process terminates when $d_n \geqslant 0$. $y_n.2^{\frac{1}{2}a'}$ is then taken to be the answer and is left in standard floating-point form in X6. There is no need to test for floating-point overflow or underflow since the answer is always nearer to one than the operand.

	R 0 0 -0 1
	611 - 28 -
01	0+ [0] 72
	0.0 0 60

LN
F.P. SQ. ROOT

R 1 5 -0 1	Calls for P.P.01
611 - 02 -	
R 1 6 -0 1	
611 - 02 -	
R 1 6 -0 1	
611 - 02 -	

FERRANTI LTD

PEGASUS LIBRARY PROGRAMME

R 611

0+

Entry	Word	Instruction	Notes
01 →	0.0	6 5 00	x to 5
		6 7 00	x to 7
	1	(1) 7 45	If a even, 0 to 7; if a odd 2^{-38} to 7
		1+[1] 72	
	2	0.2 6 63	LOOP STOP if $x < 0$
		1.6 6 05	A to 6
	3	6 5 03	$a + 2^{(n-1)}$ to 5
		0.5 7 60	Jump if a even
	4	(1) 5 43	$(a - 1) + 2^{(n-1)}$ to 5
		6 6 01	$2A$ to 6
	5	1.5 5 03	$a' + 2.2^{(n-1)}$ to 5 ($a' = a$ or $a - 1$)
		(1) 5 51	$(a'/2) + 2^{(n-1)}$ to 5
	6	1.5 1 10	Plant LINK
		1.5 6 60	EXIT if $A = 0$
	7	6 1 00	A' to 1 ($A' = A$ or $2A$)
		6 7 00	A' to 7

1+

Word	Instruction	Notes
1.0	(1) 7 51	$\frac{1}{2}A'$ to 7
	1.7 7 11	Initially $y_0 = (\frac{1}{2}A' + \frac{1}{2})$ to 1.7; then $y_{n+1} = y_n + d_n$
1	1.7 1 26	$\left(\frac{A'}{y_n} - y_n\right) = 2d_n$ to 7
	1.7 7 03	
2	(1) 7 51	d_n to 7
	1.0+ 7 63	Jump if $d_n < 0$
3	1.7 6 00	$\sqrt{A'}$ to 6
	(76) 5 56	Normalize
4	1.6 6 05	Pack in F.P. form: $\sqrt{x}$ to 6
	5 6 01	
5	+0	$+$ P.P.01 $= -2^{(n-1)}.2^{-38}$ } Overwritten by LINK
6	+0	$+ 2 \times$ P.P.01 $= -2^{n}.2^{-38}$
7	+0.5	Overwritten by y_n; finally $= \sqrt{A'}$

2+

Word	Notes
R 0 0 -0 1	**Optional Parameter List**
611 - 06 -	
-256	P.P.01 $= -2^{n-1}$ ($n = 9$)

SHORTER FLOATING POINT ARITHMETIC

This is a self-preserving subroutine which performs the operations of addition, subtraction and multiplication on packed floating point numbers and can also be adapted to do division instead of multiplication. The floating point number $x = A.2^{a}$ is held in a single word with the least significant n bits representing the non-negative integer $a + 2^{n-1}$, and the most significant $39-n$ bits representing the fraction A.

	R 0 0 -0 4 612 - 28 -
01	0+ [4] 72 1+ [5] 72
02	[2+ 5] 70 4.4 1 10
03	0+ 0 00 0. 0
04	0 0+ 0 00 0.

LN
SHORTER F.P. ARITHMETIC

R 1 6 -0 1 612 - 02 -	Calls for P.P.01
R 1 7 -0 1 612 - 02 -	
R 1 7 -0 1 612 - 02 -	
R 2 7 -0 1 612 - 06 -	Title of Optional Parameter List

0+

	Address	Order	Comment
+ −	4.0	32 1 01	Make 1 negative to indicate + or −
× ÷		6 4 00	UNPACK OPERANDS (6 4 00 to 7 5 03)
	1	5.7 6 05	A to 6
		6 4 03	$a + 2^{n-1}$ to 4
	2	7 5 00	
		5.7 7 05	B to 7
	3	7 5 03	$b + 2^{n-1}$ to 5
		4.6 1 63	Jump if + or −
÷ to 2+.0	4	(5.6 4 01	a to 4 — Overwritten by orders jumping to 2+.0 when routine adapted for division
		6 7 20)	$A.B$ to 7
÷ from 2+.4	5	5 4 01	$a + b + 2^{n-1}$ if ×; $a - b + 2 + 2^{n-1}$ if ÷
		5.3 0 60	
	6	7 7 12 1	Negate 7 if −
		4 5 04	$a - b$ to 5
	7	(25) 5 50	$a - b$ to 5_m
		5.2 5 62	Jump if $a \geqslant b$

1+

	Address	Order	Comment
	5.0	5 5 02	$b - a$ to 5_m
		(0) 4 41 5	$a + 2^{n-1} + b - a = b + 2^{n-1}$ to 4_c
	1	(0) 6 51 5	Shift A down $b - a$ places
		5.2+ 0 60	
	2	(0) 7 51 5	Shift B down $a - b$ places
		7 6 01	Add Arguments
	3	(40) 4 56	Normalize
		5.6 6 03	Round argument
	4	5.7 6 05 4	Pack result in 6 if exponent $\geqslant -2^{n-1}$
		4 6 01 4	Clear 6 if underflow (exponent $< -2^{n-1}$)
	5	(25) 1 52	Exit provided OVR clear
		0 0 64 1	(otherwise 77 stop in 5.6)
	6	+0	+ P.P.01 $= -2^{n-1}.2^{-38}$ — 77 stop here if OVR is set
	7	+0	+ (P.P.01) × 2 $= -2^{n}.2^{-38}$

Block 2+

		Order	Comment
from 0+.4+ →	4.0	4.0 7 60	Loop stop if zero divisor
		5.6 5 01	b to 5
	1	(2) 5 44	$2 - b$ to 5
		(2) 6 51	¼ A to 6
	2	7 6 26	¼ A/B to 7
		7 6 00	¼ A/B to 6
	3	4.5 7 00 1	Order pair to reset for ÷ or × to 7
		0	
	4	0+[4] 72	Plant reset order pair in 0+.4
← to 0+.5		4.4 7 10	and then return to 0+.5
	5	2+[4] 72	÷ reset order pair
		4.0 0 60	
	6	5.6 4 01	× reset order pair
		6 7 20	
	7	−256	Optional P.P.01 $= -2^{n-1}.2^{-38}$ ($n = 9$)

Notes:

1) The value of the parameter n, specifying the number of bits in the exponent, may not exceed 10 because the order 0 4 41 5 in 1+.0+ assumes that there are not more than 10 bits in $b - a$. The maximum possible value of $b - a$ is $2^n - 1$.

2) The order 5.6 6 03 in 1+.3+ may very occasionally make the argument equal to +½ or −¼, which are not correctly normalized. See section 4 of the Library Specification.

END OF VOLUME