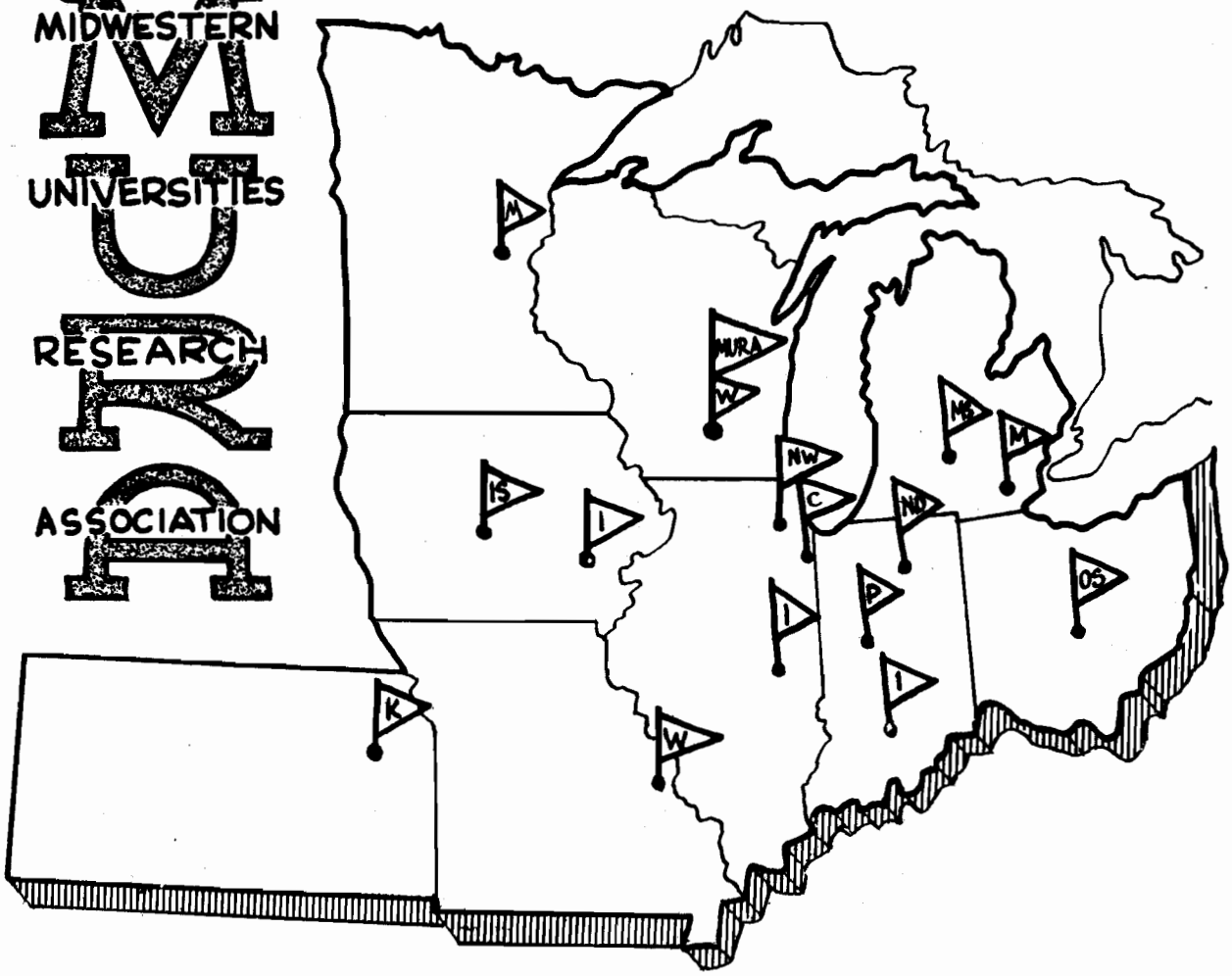
Fermilab Library
0 1160 0038687 4

ZGS LIBRARY

360

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION



A MANUAL FOR THE USE OF MURA IBM 704 PRODUCTION PROGRAMMES

By Jess Anderson

REPORT **NUMBER** 360

MURA-360

INTRODUCTION

This manual is intended to provide any person familiar with the MURA production programmes as described in a series of MURA Internal Reports with the technical instruments of problem preparation and calculator operation. Naturally this manual must presume that the internal reports and their adjuncts have been studied carefully, since an attempt has been made to avoid redundancy.

The minimum equipment used by the majority of these programmes is as follows: an IBM 704 EDPM with 8192 words of core storage, on-line punch and printer, and card input. The reader and punch boards are wired straight 72-72, but the printer is in all cases equipped with the MURA PRINTER BOARD 1, a wiring diagram for which is attached at the end of this introduction. Problems using scope output require in addition 740 Cathode Ray Display and Recorder units. No tapes or drums are used.

The programme master decks are contained in a file of two boxes, and an index is provided both in the boxes and at the very end of this manual.

A standard IBM card has 80 columns numbered 1-80 left to right and 12 rows numbered 9, 8, 7, 6, 5, 3, 4, 2, 1, 0, X, Y bottom to top. The standard IBM 024 or 026 key punch code has been modified so that the symbols below correspond to the punches given in the table. All other key punch symbols are standard.

Symbol	Rows Punched
+	Y
-	X
=	8, 3
(	Y, 4, 8
)	0, 4, 8

Input data cards are of four general forms, viz., fixed-point integers, fixed-point fractions, floating-point numbers, and octal numbers. The first two are referred to more simply as integers and fractions. Special types will be discussed as needed in the comments on individual programs.

The first three types have some properties in common: commas and decimal points are not punched, each card has an address followed by the value to be assigned to a given datum, and only one datum (with its address) is placed on a given card.

The expression "12R" refers to the field of the card from columns 37 thru 72. Indicator words placed in this field are to begin in column 37 and use as many succeeding columns as necessary. The spaces, punctuation, and spelling indicated should be rigidly preserved. Columns in this field other than those so used should be blank. All information in columns 73-78 is ignored by the calculator, allowing the user to place there any identification symbols or other information he may desire.

Fixed-point integers are punched according to the following format:

1. The address may have from one to five digits punched in column 1 and the requisite number of succeeding columns.
2. Following the last digit of the address will be one, and only one, blank column.
3. The value (a signed integer, or an unsigned integer which is then considered to be positive) is punched in the columns following this blank. The integer I must obey the restriction

$$|I| \leq 2^{35} - 1 \equiv 34359738367$$

Fixed-point fractions are punched in a similar way, except that the datum may not exceed 11 digits nor .99999999998 in absolute value.

Floating-point numbers are punched as follows:

1. The address is punched in the same fashion as for fixed-point numbers.
2. The column following the last digit of the address is taken up by the sign, which is plus, minus, or blank, where the last has the same effect as the first.
3. The sign is followed by value, which may not exceed nine digits in length.
4. The column following the last digit of the value is taken by the sign of the exponent, which is plus, minus, or blank, where the last has the same effect as the first.
5. The exponent is punched in the columns which follow, taking up to two columns.
6. The value of the exponent is chosen so that the decimal point (which is of course not punched) lies after the first digit of the absolute value of the number. The range of floating-point numbers is

$$1.46936795 \times 10^{-39} \leq |N| \leq 1.70141181 \times 10^{38}$$

Samples of each of the three major types follow. A space is considered to be a blank column, and all punching starts in column 1.

Fixed-point integers:

```
30 14000
3115 -0
304 62
```

Fixed-point fractions:

```
3585 -0000016
316 752
42 -0
```

Floating-point numbers:

100 5678-10
100-5678 10
100-56789-12

Although no advantage is gained by using octal numbers in the programmes Algytee and Algytee Scope, rules for card format are given here. It is presumed that the user knows how to represent octal floating-point numbers in the calculator's language. All address are five digit octal numbers; all values are 12 digit octal numbers. The five address digits occupy columns 1-5, the value digits, columns 6-17 of a card. For example, a fixed-point decimal integer 1023 at decimal address 512 would be punched, starting in column 1, as below:

01000000000001777

There are several rows of lights on the console which indicate the state of several registers in the calculator. Only three of these are normally of interest to the user, viz., the three long rows, the SR, the AC, and the MQ. These give a binary representation of their contents by causing neons to be lighted for bit positions containing ones and leaving unlighted those containing zeroes. These bits may be grouped in threes to give the less cumbersome octal equivalent of the contents of a register, according to the table below:

000 = 0
001 = 1
010 = 2
011 = 3
100 = 4
101 = 5
110 = 6
111 = 7

Thus the 36 bits of the long registers (disregarding the P and Q bits of the AC) may be written as 12 octal digits. In the list of programme stops associated

with most programmes the terms HTR and HPR appear. The numbers following them are octal. A programme stop lights the upper-left-most indicator light on the console and displays all the registers. The stop instruction is contained in the SR, for example:

$$\text{HTR } 5 = 000000000005_8$$

$$\text{HPR } 1750 = 042000001750_8$$

During the reading of programme and output decks, a check is made to ensure that the information on the card corresponds to that in the calculator. The read-in subroutine forms the logical sum of the information words as they are read, and compares this check sum with that on the card. If a read-in error causes these two sums to differ, a characteristic programme stop, HTR 5, occurs. In the case of output decks from problems which are subsequently used as input for others, this check sum failure could be an error on the card. In the case of programme master decks, however, no such errors are found, and a check sum stop indicates a computer error. In such an event, it is always wise to start the problem over, reading in all the cards again. In the former case, nothing can be done except to obtain a new deck by rerunning the problem which produced the faulty deck.

A propos of errors while running these problems it may happen that some unforeseen limitation is violated. The problem arises of distinguishing such stops from computer errors. If the reason for a programme or other stop is not obvious, and an explanation is desired, the following procedure should be instituted:

1. Copy the condition of the following on a sheet of paper.

APPROVED BY
J N Snyder

IDENTIFICATION

Octal Dump, MU OCD1

L. D. Fosdick - November 5, 1956

Midwestern Universities Research Association, Madison, Wisconsin

PURPOSE

To print as octal numbers via the 716 on-line printer, the contents of a block of core storage starting at location A and ending at A + N.

RESTRICTIONS

The state of the console and index registers is not preserved. The contents of locations 0 - 10_{10} (0 - 12_8) are destroyed. The contents of the memory block 0 - 119_{10} (0 - 167_8) are punched onto five cards, 24 words per card. (The first 12 words have already been destroyed at the time of punching.) After punching of the cards the contents of locations 0 - 99_{10} (0 - 143_8) are overwritten by the remainder of the OCTAL DUMP program and cannot be recovered.

The MURA OCTAL DUMP BOARD is required. See attached sheet for wiring diagram.

USAGE

Self Loading: Ready the cards in the reader and push the LOAD CARDS button. The first card contains a read in interlude program which will punch out the five cards described above under RESTRICTIONS. When this punching is completed a halt on HPR 24_{10} (30_8) is encountered. Pushing the LOAD CARDS button will allow the remainder of the program to be read into the memory. When the read in is complete a halt on HPR 32767_{10} (77777_8) from 19_{10} (23_8) is reached. At this time the starting address, A, of the block to be dumped, and the final address of the block A + N, are entered into the decrement and address portions of the MQ, respectively. When START is pressed dumping will begin. When dumping is completed the halt HPR 32767_{10} (77777_8) from 19_{10} (23_8) is again encountered. Another block may then be dumped by entering the appropriate addresses into the MQ.

The octal numbers are printed in order, A, A + 1, . . . A + N 4 per line, with the octal location of the first number on the line appearing to the left of the line. The interlude punching of the five cards may be omitted by simply removing the first card of the deck.

Any line which would contain five octal numbers all identically equal to 000000000000 is not printed. This omission is indicated as a line space. The appearance of more than one such line of zeros in a block is indicated by only a single line space, rather than a separate line space for each omitted line.

Space Required: $0 - 99_{10}$ ($0 - 143_8$)

The first card of the binary check, which contains the routine for punching out the contents of locations $0 - 119_{10}$, may also be inserted at the beginning of the MU BPU3 MU FRD1 and MU IND1 binary decks whenever it is desired to preserve the contents of locations $11 - 119_{10}$ before overwriting by these dumping routines takes place. The number in 4LA of this card is the end constant for the punching, and its value is just equal to the number of words punched out starting with location zero. Thus the contents of any desired number of locations may be initially dumped on cards by appropriate setting of 4LA. In the deck supplied to SHARE this end constant appears in the address portion of card 11 and has the value 120_{10}.

CODING INFORMATION

Timing: The printer operates at full speed (150 lines per minute).

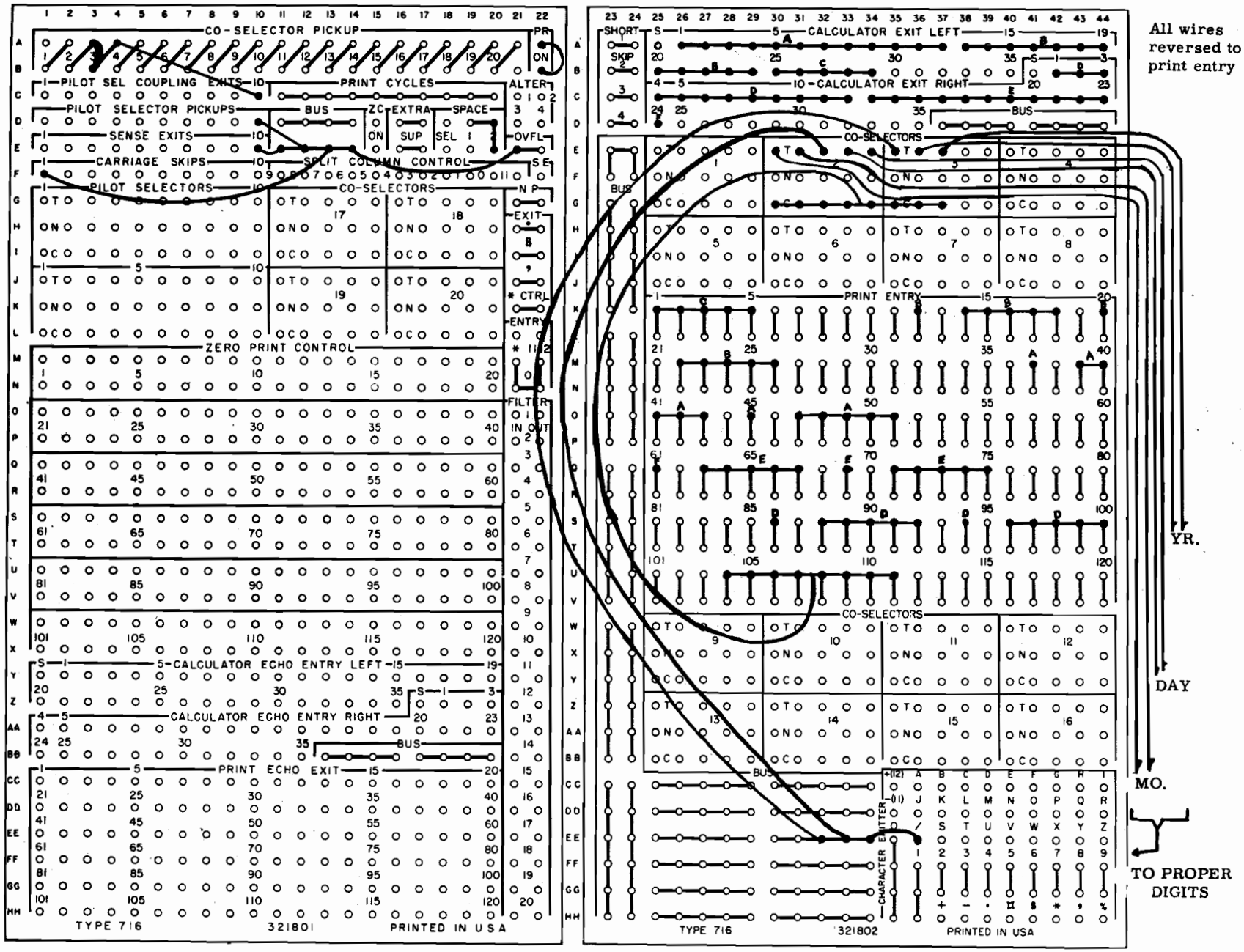
```
      REM OCD1 MURA OCTAL DUMP                                          MU  OCD1
OCD1  CPY 2            BEGIN LOADING PUNCH INTERLUDE                  OCD1  0001
      CPY 0                                                           OCD1  0002
      TRA 0                                                           OCD1  0003
      CPY 3                                                           OCD1  0004
      LXA 6,1          SET IR1=120                                    OCD1  0005
      CPY 4                                                           OCD1  0006
      LXA 10,2         SET IR2=24                                     OCD1  0007
      CPY 5                                                           OCD1  0008
      WPU              SELECT PUNCH                                   OCD1  0009
      CPY 6                                                           OCD1  0010
      CPY 120,1        PUNCH WORDS IN LOCATIONS 0-119                 OCD1  0011
      CPY 7                                                           OCD1  0012
      TNX 10,1,1       TRANSFER TO HPR WHEN PUNCHING COMPLETE         OCD1  0013
      CPY 8                                                           OCD1  0014
      TIX 6,2,1                                                       OCD1  0015
      CPY 9                                                           OCD1  0016
      TRA 4                                                           OCD1  0017
      CPY 10                                                          OCD1  0018
      HPR 24           HERE WHEN DONE PUNCHING                        OCD1  0019
      CPY 1                                                           OCD1  0020
      TRA 3                                                           OCD1  0021
      HTR 0                                                           OCD1  0022
      HTR 0                                                           OCD1  0023
      HTR 0                                                           OCD1  0024
      CPY 2            BEGIN SELF LOADING MAIN PROGRAM                OCD1  0025
      CPY 0                                                           OCD1  0026
      TRA 0                                                           OCD1  0027
      CPY 3                                                           OCD1  0028
      LXA 2,2          SET IR2=0                                      OCD1  0029
      CPY 4                                                           OCD1  0030
      RCD              SELECT CARD READER                             OCD1  0031
      CPY 5                                                           OCD1  0032
      CPY 19,2                                                        OCD1  0033
      CPY 6                                                           OCD1  0034
      CAL 19,2         WORD JUST READ INTO AC                         OCD1  0035
      CPY 7                                                           OCD1  0036
      TXI 9,2,-1       ADD -1 TO IR2                                  OCD1  0037
      CPY 8                                                           OCD1  0038
      TRA 4            END OF RECORD                                  OCD1  0039
      CPY 9                                                           OCD1  0040
      TZE 19           TRANSFER IF WORD JUST READ =0                  OCD1  0041
      CPY 10                                                          OCD1  0042
      TRA 5            LOOP TO CPY                                    OCD1  0043
      CPY 1                                                           OCD1  0044
      TRA 3                                                           OCD1  0045
      HTR 0                                                           OCD1  0046
      HTR 0                                                           OCD1  0047
      HTR 0                                                           OCD1  0048
      HPR 32767        HALT TO INSERT ADDRESSES                       OCD1  0049
      LXA 21,1         SET IR1=2                                      OCD1  0050
      PXD 2,0          CLEAR AC                                       OCD1  0051
      LGL 18           SHIFT ADDRESSES INTO AC                        OCD1  0052
      STO 2,1          PLANT INITIAL, FINAL ADDRESS AT 0,1            OCD1  0053
```

```
TIX 21,1,1      LOOP RE-ENTRY, OUT AFTER 2 PASSES          OCD1  0054
WPR             SELECT PRINTER                             OCD1  0055
SPR 10          JUSTIFY AND DATE                           OCD1  0056
LXA 28,2        SET IR2=12                                 OCD1  0057
PXD 12,0        CLEAR AC                                   OCD1  0058
STO 19,2        CLEAR NUMBER BANK                          OCD1  0059
TIX 29,2,1      LOOP RE-ENTRY, OUT AFTER 12 PASSES         OCD1  0060
CLS 1           - (FINAL ADDRESS)                          OCD1  0061
ADD 0           + (CURRENT ADDRESS)                        OCD1  0062
TPL 19          TO HPR IF DONE                             OCD1  0063
ADD 1           HERE IF NOT DONE                           OCD1  0064
SUB 41          +4                                         OCD1  0065
STA 42          PLANT CALL ADDRESS                         OCD1  0066
STA 57          PLANT CALL ADDRESS                         OCD1  0067
LDQ 0           CURRENT ADDRESS IN MQ                      OCD1  0068
STA 0           PLANT ADDRESS FOR NEXT USE                 OCD1  0069
LXA 41,1        SET IR1=4                                  OCD1  0070
PXD 4,0         CLEAR AC                                   OCD1  0071
ACL 0,1         FORM SUM, ADDRESS PLANTED                  OCD1  0072
TIX 42,1,1      LOOP RE-ENTRY, OUT AFTER 4 PASSES          OCD1  0073
TNZ 48          TRANSFER IF SUM IS NOT ZERO                OCD1  0074
TXH 31,2,1      HERE IF SUM=0                              OCD1  0075
WPR             SELECT PRINTER TO SPACE                    OCD1  0076
TXI 31,2,1      ADD 1 TO IR2                               OCD1  0077
WPR             SELECT PRINTER                             OCD1  0078
LXA 28,1        SET IR1=12                                 OCD1  0079
PXD 0,0         CLEAR AC                                   OCD1  0080
LGL 3           CONSTRUCT CURRENT                          OCD1  0081
ALS 28          ADDRESS AS ID                              OCD1  0082
ORS 19,1        AND STORE                                  OCD1  0083
TIX 50,1,1      LOOP, OUT AFTER 12 PASSES                  OCD1  0084
LXA 41,4        SET IR4=4                                  OCD1  0085
LXA 28,2        SET IR2=12                                 OCD1  0086
LDQ 0,4         NUMBER IN MQ, ADDRESS PLANTED              OCD1  0087
PXD 0,0         CLEAR AC                                   OCD1  0088
LGL 3                                                      OCD1  0089
ALS 25,1                                                   OCD1  0090
ORS 19,2        OCTAL NUMBERS INTO READING BANK            OCD1  0091
TIX 58,2,1      LOOP, OUT AFTER 12 PASSES                  OCD1  0092
TXI 64,1,4      STEP IR1 BY + 4                            OCD1  0093
TIX 56,4,1      LOOP, OUT AFTER 4 PASSES                   OCD1  0094
CAL 97          T=10 INTO AC                               OCD1  0095
ACL 98          REPLACE T BY T-1                           OCD1  0096
SLW 2           STORE T-1 FIXED                            OCD1  0097
SLW 3           STORE T-1 SHIFTING                         OCD1  0098
CAL 98          EXTRACTOR INTO AC                          OCD1  0099
SLW 4           STORE EXTRACTOR                            OCD1  0100
LXA 74,1        SET IR1 EQUAL TO 5                         OCD1  0101
LXA 81,2        SET IR2 EQUAL TO 3                         OCD1  0102
LXA 93,4        SET IR4 EQUAL TO 2                         OCD1  0103
CAL 5,4         SHIFT T AND EXTRACTOR                      OCD1  0104
ARS 4           4 PLACES RIGHT FOR                         OCD1  0105
SLW 5,4         NEXT COLUMN                                OCD1  0106
TIX 74,4,1      LOOP, OUT AFTER 2 PASSES                   OCD1  0107
LXA 28,4        SET IR4 EQUAL TO 12                        OCD1  0108
```

```
CAL 19,4          DIGIT INTO AC                                  OCD1  0109
ANA 4             EXTRACT DIGIT                                  OCD1  0110
CAS 3             COMPARE WITH T                                 OCD1  0111
TRA 84            0 IMAGE BIT IF DIGIT GREATER THAN T            OCD1  0112
CLA 84            1 IMAGE BIT IF DIGIT EQUAL TO T                OCD1  0113
LRS 1             0 IMAGE BIT IF DIGIT LESS THAN T               OCD1  0114
TIX 79,4,1        LOOP, OUT AFTER 12 PASSES                      OCD1  0115
TNX 90,1,1        TRANSFER WHEN ALL COLUMNS PROCESSED            OCD1  0116
TIX 73,2,1        LOOP, OUT AFTER 3 PASSES                       OCD1  0117
STQ 5             STORE LEFT IMAGE                               OCD1  0118
TRA 72            RETURN FOR RIGHT IMAGE                         OCD1  0119
STQ 6             STORE RIGHT IMAGE                              OCD1  0120
CPY 5             COPY LEFT IMAGE                                OCD1  0121
CPY 6             COPY RIGHT IMAGE                               OCD1  0122
CAL 2             CALL IN T                                      OCD1  0123
ANA 98            CLEAN T OFF                                    OCD1  0124
TNZ 66            TRANSFER IF T NOT ZERO                         OCD1  0125
TRA 27            RE-ENTER MAIN LOOP FOR NEXT LINE               OCD1  0126
OCT 500000000000 T EQUAL TO 10                                   OCD1  0127
OCT 740000000000 EXTRACTOR                                       OCD1  0128
HTR 0                                                            OCD1  0129
ENDO                                                             OCD1  0130
```

Form 24-6127-0

TYPE 716 PRINTER

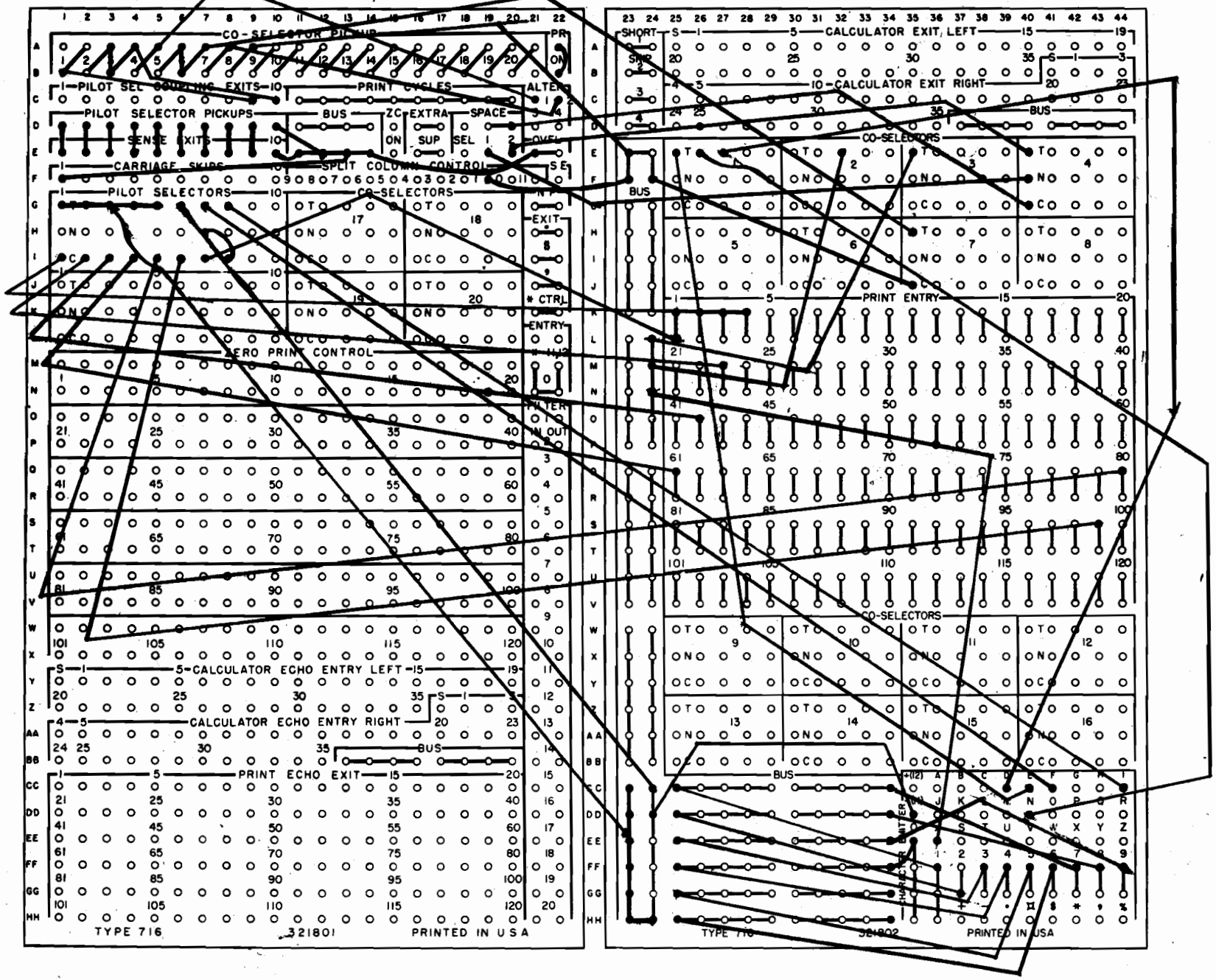


MURA OCTAL DUMP

MURA PRINTER BOARD

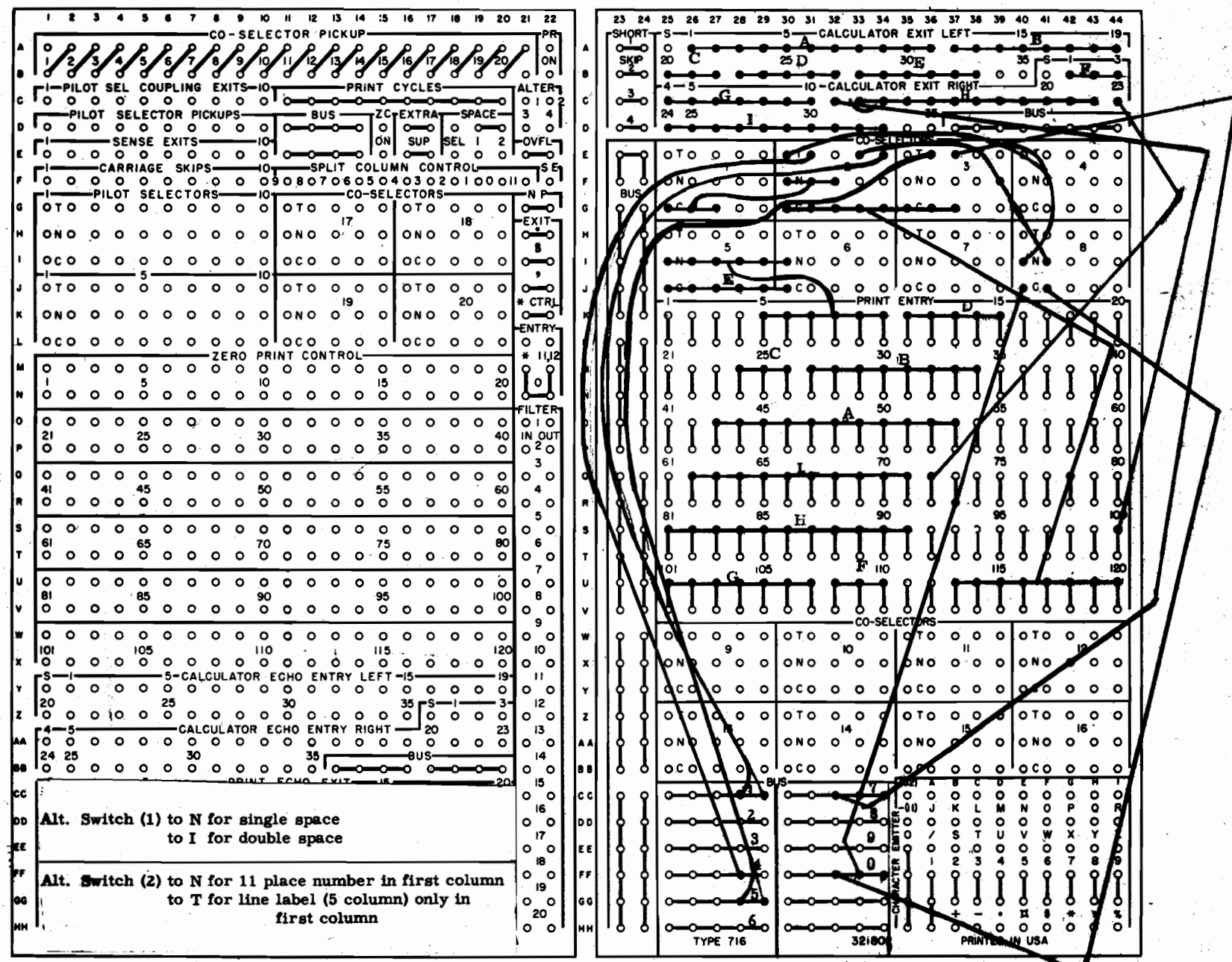
Form 24-6127-0

TYPE 716 PRINTER



Form 24-6127-0

TYPE 716 PRINTER



EXAMPLE DATE 04/01/57

Page 2 of 2

MURA PRINTER BOARD

APPROVED BY

IDENTIFICATION

Transfer Trace Octal Dump, MU TTD1

Jess Anderson - March 7, 1958

Midwestern Universities Research Association, Madison, Wisconsin

PURPOSE

To execute a program in core storage, and for every transfer instruction except TTR of the given program, to print as octal numbers, via the on-line printer, the contents of a specified block of core storage.

RESTRICTIONS

The program to be traced cannot utilize the trapping mode. When this routine is loaded, the contents of index register 4 and the console are destroyed. The MURA OCTAL DUMP BOARD is required. See attached sheet for wiring diagram.

USAGE

The program to be traced is assumed to be in core storage. The trace dump routine is self-loading. It is backed by the requiste control card(s) described below, and readied in the card reader. When the LOAD CARDS button is pushed, the program loads and stops on HPR 77777_8. If START is now pushed, a control card is ingested and control is transferred to the traced program. When a transfer instruction other than TTR is encountered in the traced program, a dump of a specified block of core will be made. Following this dump, control is returned to the traced program, which proceeds up to the next such transfer, when the process is repeated. Notice should be taken in case the instructions HPR or HTR occur as part of the traced program, as it will be necessary to push START to obtain a dump for that instruction.

A new control card may be called for at any time by resetting the console (to turn off the trap indicator if it happens to be on) and transferring manually to location 14_{10} (16_8) which will cause a stop HPR 77777_8, and pressing the START button.

A control card is punched as follows, in binary: 9LD, location of first word of dump; 9LA, location of final word of dump; 9RD, blank; 9RA, starting location of the traced program. If dumping is to be suppressed and only a logical trace is desired, it is only necessary to set the final dump address less than the starting dump address on the control card.

Each dump is preceded by an initial line with a 5 digit label and 4 twelve digit octal numbers. The numbers give the contents of the index registers, the AC, and the MQ at the time the transfer instruction which produced the dump was encountered, and the location of that transfer instruction, in the following format:

LABEL	IR1	IR2	IR4	L O	AC	MQ
00270	0 00000	0 00000	0 00000	0 00000	0 00000 0 00000	0 00000 0 00000

where L = location of the transfer instruction,

O = Overflow bits Q and P as a single octal digit, according to the table below.

Q Bit	P Bit	Digit
0	0	0
0	1	1
1	0	2
1	1	3

This is followed by the dump proper. The octal numbers are printed in order, A, A + 1, . . . , A + N, 4 per line, with the octal location of the first member on the line appearing in the label column to the left of the line. Any line which would contain four octal numbers all identically equal to 000000000000 is not printed. This omission is indicated as a line space although the appearance of more than one consecutive line of zeroes in a block is indicated by only a single line space.

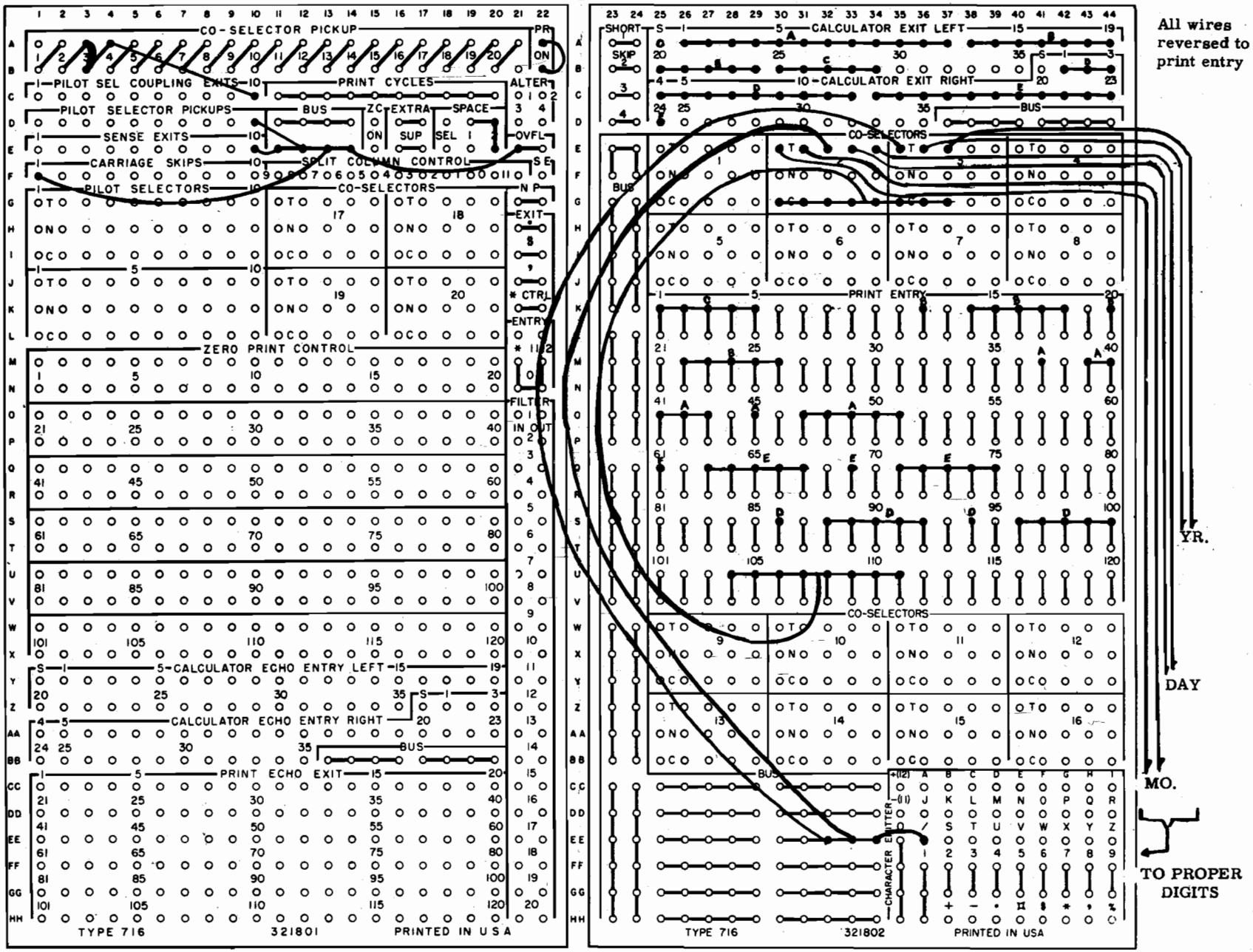
Space Required: $0\text{-}196_{10}$ $(0\text{-}304)_8$.

CODING INFORMATION

Timing: The printer operates at full speed (150 lines per minute).

Form 24-6127-0

TYPE 716 PRINTER



MURA OCTAL DUMP

```
MUTTD1 REM 0199 10029                                                  TTD1  0001
       REM TRANSFER TRACE OCTAL DUMP                                   TTD1  0002
 TTD1  LXA 0,4          CLEAR IR4                                     TTD1  0003
       CPY 2,4          COPY FIRST CARD                               TTD1  0004
       TXI 1,4,-1       LOOP UNTIL DONE                               TTD1  0005
       HTR 173              CONSTANT                                  TTD1  0006
       LTM              LEAVE TRAPPING MODE                           TTD1  0007
       LXA 3,4          SET IR4 TO 169                                TTD1  0008
       RCD              SELECT CARD READER                            TTD1  0009
       CPY 197,4            COPY ANOTHER                              TTD1  0010
       TIX 7,4,1        CARD                                          TTD1  0011
       TRA 11           PROGRAM LOADED, START                         TTD1  0012
       TRA 6            END OF CARD, GET ANOTHER                      TTD1  0013
       CAL 167          GET TTR 28 WORD                               TTD1  0014
       SLW 1            PLANT IT IN LOCATION 1                        TTD1  0015
       STZ 0            CLEAR LOCATION 0                              TTD1  0016
       HPR, 32767       STOP SHOWING 77777                            TTD1  0017
       RCD              SELECT CARD READER                            TTD1  0018
       CPY 182          COPY 9L OF CONTROL CARD                       TTD1  0019
       CPY 183          COPY 9R OF CONTROL CARD                       TTD1  0020
       CLA 183          GET 9R OF CONTROL CARD                        TTD1  0021
       STA 27           PLANT TRACE PROGRAM STARTING ADDRESS          TTD1  0022
       ARS 18           DECREMENT INTO ADDRESS                        TTD1  0023
       TNZ 23           IF NOT ZERO, STORE AS IS                      TTD1  0024
       COM              IF ZERO, COMPLEMENT AND STORE                 TTD1  0025
       STO 183          HIGHEST TRANSFER TO BE TRACED                 TTD1  0026
       CAL 169          GET PAPER JUSTIFY WORD                        TTD1  0027
       SLW 71           AND STORE IT                                  TTD1  0028
       ETM              ENTER TRAPPING MODE                           TTD1  0029
       TTR **           TRANSFER TO TRACED PROGRAM, ADDR. BY 20       TTD1  0030
       LTM              LEAVE TRAPPING MODE                           TTD1  0031
       SXD 184,1        SAVE IR1                                      TTD1  0032
       SXD 167,2        SAVE IR2                                      TTD1  0033
       SXD 185,4        SAVE IR4                                      TTD1  0034
       STQ 187          SAVE MQ                                       TTD1  0035
       STO 186          SAVE AC                                       TTD1  0036
       ARS 35           Q AND P AS AC ADDRESS                         TTD1  0037
       STA 185          SAVE Q AND P                                  TTD1  0038
       CLA 167          BRING IN OVERFLOW (OV) MARK BOX               TTD1  0039
       SSP              SET ITS SIGN TO PLUS                          TTD1  0040
       TNO 40           IS OV LITE OUT                                TTD1  0041
       SSM              NO, CHANGE ITS SIGN TO MINUS                  TTD1  0042
       STO 167          STORE OV MARK                                 TTD1  0043
       CLA 0            BRING LOCATION OF TRANSFER WORD               TTD1  0044
       STA 50           PLANT IT AS CALL ADDRESSES                    TTD1  0045
       STA 57           FOR RETURN ADDRESS CONSTRUCTION               TTD1  0046
       ALS 3            MULTIPLY BY 8                                 TTD1  0047
       ADD 185          ADD Q AND P                                   TTD1  0048
       STO 185          STORE IN PRINT BOX                            TTD1  0049
       CAL 166          GET RETURN WORD                               TTD1  0050
       ANA 170          CLEAR TAG FIELD                               TTD1  0051
       SLW 166          RESTORE RETURN WORD                           TTD1  0052
       CLA **           GET TRANSFER WORD, ADDR. BY 43                TTD1  0053
       STA 166          PLANT RETURN WORD ADDRESS                     TTD1  0054
       ALS 2            FIRST 2 BITS IN Q,P                           TTD1  0055
       TOV 61           IF TYPE A, RETURN WORD UNALTERED              TTD1  0056
       ANA 171          CLEAN OFF ALL BUT INSTRUCTION BITS            TTD1  0057
       SUB 176          IS IT TSX INSTRUCTION                         TTD1  0058
       TZE 61           YES, TREAT AS TYPE A                          TTD1  0059
       CAL **           NO, GET TRANSFER WORD, ADDR. BY 44            TTD1  0060
```

```
ANA 173          EXTRACT TAG                                TTD1  0061
ACL 166          ADD RETURN WORD                            TTD1  0062
SLW 166          AND RESTORE                                TTD1  0063
CLA 167          GET IR2                                    TTD1  0064
ARS 18           SHIFT TO ADDRESS                           TTD1  0065
STA 184          AND STORE IN PRINT BOX                     TTD1  0066
CLA 179          GET INITIAL PRINT ADDRESSES                TTD1  0067
STA 180          STORE IN FINAL ADDRESS BOX                 TTD1  0068
ARS 18           DECREMENT INTO ADDRESS                     TTD1  0069
STA 181          STORE IN CURRENT ADDRESS BOX               TTD1  0070
CLA 177          GET INITIAL PRINT EXIT ADDRESS             TTD1  0071
STA 78           AND PLANT IT                               TTD1  0072
WPR              SELECT PRINTER                             TTD1  0073
SPR 10           JUSTIFY AND DATE FORM                      TTD1  0074
LXA 73,2         SET IR2 TO 12                              TTD1  0075
PXD 12,0         CLEAR AC                                   TTD1  0076
STO 14,2         CLEAR NUMBER                               TTD1  0077
TIX 74,2,1       CONSTRUCTION BANK                          TTD1  0078
CLS 180          -(FINAL ADDRESS)                           TTD1  0079
ADD 181          +(CURRENT ADDRESS)                         TTD1  0080
TPL **           EXIT IF PLUS, ADDR. BY 70 OR 144           TTD1  0081
ADD 180          ADD FINAL ADDRESS                          TTD1  0082
SUB 86           ADD 4                                      TTD1  0083
STA 87           PLANT CALL ADDRESS                         TTD1  0084
STA 102          TWICE                                      TTD1  0085
LDQ 181          CURRENT ADDRESS IN MQ                      TTD1  0086
STA 181          PLANT FOR FUTURE USE                       TTD1  0087
LXA 86,1         SET IR1 TO 4                               TTD1  0088
PXD 4,0          AC CLEAR                                   TTD1  0089
ACL **,1         FORM SUM, ADDRESS BY 82                    TTD1  0090
TIX 87,1,1       LOOP RE-ENTRY                              TTD1  0091
TNZ 93           PRINT IF NOT ZERO                          TTD1  0092
TXH 76,2,1       GET NEXT FOUR IF SUM IS ZERO               TTD1  0093
WPR              SELECT PRINTER                             TTD1  0094
TXI 76,2,1       INCREASE IR2 BY 1                          TTD1  0095
WPR              SELECT PRINTER                             TTD1  0096
LXA 73,1         SET IR1 TO 12                              TTD1  0097
PXD 5,0          CLEAR AC                                   TTD1  0098
LGL 3            CONSTRUCT                                  TTD1  0099
ALS 28           CURRENT ADDRESS                            TTD1  0100
ORS 14,1         AS LABEL AND STORE                         TTD1  0101
TIX 95,1,1       LOOP 12 TIMES                              TTD1  0102
LXA 86,4         SET IR4 TO 4                               TTD1  0103
LXA 73,2         SET IR2 TO 12                              TTD1  0104
LDQ **,4         NUMBER IN MQ, ADDR. BY 83                  TTD1  0105
PXD 3,0          CLEAR AC                                   TTD1  0106
LGL 3            FORM OCTAL                                 TTD1  0107
ALS 25,1         NUMBERS AND                                TTD1  0108
ORS 14,2         STORE IN NUMBER BANK                       TTD1  0109
TIX 103,2,1      LOOP RE-ENTRY, 12 TIMES                    TTD1  0110
TXI 109,1,4      INCREASE IR1 BY 4                          TTD1  0111
TIX 101,4,1      LOOP 4 TIMES                               TTD1  0112
CAL 175          T=10 INTO AC                               TTD1  0113
ACL 174          REPLACE T BY T-1                           TTD1  0114
SLW 188          STORE T-1 FIXED                            TTD1  0115
SLW 189          STORE T-1 SHIFTING                         TTD1  0116
CAL 174          EXTRACTOR IN AC                            TTD1  0117
SLW 190          STORE EXTRACTOR                            TTD1  0118
LXA 95,1         SET IR1 TO 5                               TTD1  0119
LXA 103,2        SET IR2 TO 3                               TTD1  0120
LXD 127,4        SET IR4 TO 2                               TTD1  0121
```

```
CAL 191,4          SHIFT T AND EXTRACTOR                          TTD1  0122
ARS 4              RIGHT FOUR FOR                                 TTD1  0123
SLW 191,4          NEXT COLUMN                                    TTD1  0124
TIX 119,4,1        TWICE                                          TTD1  0125
LXA 73,4           SET IR4 TO 12                                  TTD1  0126
CAL 14,4           DIGIT INTO AC                                  TTD1  0127
ANA 190            EXTRACT DIGIT                                  TTD1  0128
CAS 189            COMPARE WITH T                                 TTD1  0129
TXI 129,0,2        GREATER, 0 IMAGE BIT                           TTD1  0130
CLA 129            EQUAL, 1 IMAGE BIT                             TTD1  0131
LRS 1              LESS, 0 IMAGE BIT                              TTD1  0132
TIX 124,4,1        LOOP TWELVE TIMES                              TTD1  0133
TNX 135,1,1        IF ALL COLUMNS DONE                            TTD1  0134
TIX 118,2,1        LOOP THREE TIMES                               TTD1  0135
STQ 191            STORE LEFT IMAGE                               TTD1  0136
TRA 117            RETURN FOR RIGHT IMAGE                         TTD1  0137
STQ 192            STORE RIGHT IMAGE                              TTD1  0138
CPY 191            COPY LEFT IMAGE                                TTD1  0139
CPY 192            COPY RIGHT IMAGE                               TTD1  0140
CAL 188            GET T                                          TTD1  0141
ANA 174            CLEAN IT OFF                                   TTD1  0142
TNZ 111            TRANSFER IF NOT ZERO                           TTD1  0143
TRA 72             NEXT LINE RE-ENTRY                             TTD1  0144
CLA 178            GET NEW EXIT ADDRESS                           TTD1  0145
STA 78             AND PLANT IT                                   TTD1  0146
CLA 182            GET DUMP LIMITS                                TTD1  0147
STA 180            PLANT IN FINAL ADDRESS BOX                     TTD1  0148
ARS 18             DECREMENT INTO ADDRESS                         TTD1  0149
STA 181            PLANT IN CURRENT ADDRESS BOX                   TTD1  0150
TRA 72             TRANSFER TO DUMP                               TTD1  0151
CAL 168            SUPPRESS FORM                                  TTD1  0152
SLW 71             JUSTIFY AND DATE                               TTD1  0153
CLA 0              GET TRANSFER WORD ADDRESS WORD                 TTD1  0154
ANA 172            EXTRACT ADDRESS                                TTD1  0155
SUB 183            SUBTRACT HIGHEST TRACED ADDRESS                TTD1  0156
TPL 14             DONE, TO HALT 77777                            TTD1  0157
LXD 184,1          UNDONE, RESTORE IR1                            TTD1  0158
LXD 167,2          RESTORE IR2                                    TTD1  0159
LXD 185,4          RESTORE IR4                                    TTD1  0160
CLA 167                GET OV MARK                                TTD1  0161
TMI 193                MINUS,LEAVE LITE ON                        TTD1  0162
CLA 185                PLUS,RESTORE Q,P BITS                      TTD1  0163
LDQ 186                RESTORE AC                                 TTD1  0164
LLS 35                 AND                                        TTD1  0165
TOV *+1                TURN LITE OUT                              TTD1  0166
LDQ 187            RESTORE MQ                                     TTD1  0167
ETM                ENTER TRAPPING MODE                            TTD1  0168
TTR **             RETURN WORD, ADDR. BY 52                       TTD1  0169
TTR 28             TRANSFER TO START OF TTD1                      TTD1  0170
NOP                NOP                                            TTD1  0171
SPR 10             SPR 10                                         TTD1  0172
OCT 777777077777 MASK FOR TAG CLEAR                               TTD1  0173
OCT 777400000000 SHIFTED INSTRUCTION EXTRACTOR MASK               TTD1  0174
OCT 000000077777 ADDRESS EXTRACTOR MASK                           TTD1  0175
OCT 000000700000 TAG EXTRACTOR MASK                               TTD1  0176
OCT 740000000000 EXTRACTOR FOR DUMP                               TTD1  0177
OCT 500000000000 T=10                                             TTD1  0178
OCT 036000000000 TSX SHIFTED LEFT TWO                             TTD1  0179
HTR 142            INITIAL EXIT ADDRESS                           TTD1  0180
HTR 149            REGULAR EXIT ADDRESS                           TTD1  0181
HTR 187,0,184      INITIAL PRINT ADDRESSES                        TTD1  0182
```

```
HTR **          FINAL ADDRESS BOX                          TTD1  0183
HTR **          CURRENT ADDRESS BOX                        TTD1  0184
HTR **          9L OF CONTROL CARD                         TTD1  0185
HTR **          9R OF CONTROL CARD                         TTD1  0186
HTR 0           PRINT BOX                                  TTD1  0187
HTR 0           X                                          TTD1  0188
HTR 0           X                                          TTD1  0189
HTR 0           X                                          TTD1  0190
HTR 0           TEMPORARY FOR DUMP                         TTD1  0191
HTR 0           X                                          TTD1  0192
HTR 0           X                                          TTD1  0193
HTR 0           X                                          TTD1  0194
HTR 0           X                                          TTD1  0195
CLA 185             RESTORE Q,P BITS                       TTD1  0196
LDQ 186             RESTORE AC                             TTD1  0197
LLS 35              AND                                    TTD1  0198
TRA 164             GO BACK                                TTD1  0199
```

FOROCYL - Programme 13

cf. Internal Report 221

The data deck has both integers and fractions, plus some other types of card. It is presumed that the agenda has been prepared in the fashion indicated in the internal report. Integers are punched in the usual fashion, with the last integer card in the set bearing extra punches in the right half of the card (called 12R) as follows:

a. Col. 37-39: END

b. Col. 40: blank

c. Col. 41-48: INTEGERS

After the integers come fractions, which are also made in the usual fashion, with the last card of the group having END FRACTIONS in 12R.

Fractions are followed by boundary values, if they are called for. Boundary values are entered in the same manner as fractions with N considered as the address. The word BOUNDARY is punched in 12R of every boundary value card.

Following the boundary valuecard are N boundary point cards. They are punched in the same manner as integers with address i and value j. In this case no sign is ever to be entered; i and j are separated by a blank and nothing else. If interpolation is called for an I is punched in column 72 of the relevant point card.

The next boundary value, if any, and its point card(s) follow; this is continued until all boundary values are prepared.

Current values if present come immediately after boundary points and are entered with column 1 blank, followed by the value. The signal CC is punched

in 12R of every current card. Each current card is followed by a current point card, which is exactly identical in form to the boundary point card. It is very important to note that in the case of current values, there must be a current value card for each point which receives that value, unlike boundary values which may be assigned to N points with an appropriate number of boundary point cards.

The very last card in a Forocyl data deck signals the main programme that the data is complete and of the proper form. It is blank except for the word BONJOUR in 12R. In some operating options of the programme there are no boundary values or points and no current values or points. This does not remove the necessity of having a BONJOUR card last in the data deck; it would immediately follow the last fraction datum. The programme is provided with means of examining the data deck as it is being ingested by the calculator. If the rules for format are violated the calculator will quickly halt on a characteristic stop. In particular, if there are fewer or greater than N boundary cards following a given boundary value card, or if END INTEGERS, END FRACTIONS, BOUNDARY, CC, or BONJOUR is misspelled or begun in any column but 37 with more or less than one space between words (where relevant), then the machine can be depended upon to indicate the operator failure. It can certainly come to pass, however, that the machine can be deceived into thinking it has one type of card in hand when in fact it has another. Thus, if the END INTEGERS be omitted on the last integer data card, all the fractions will be converted by the input part of the main programme as integers, and this is naturally an undesirable event. In the same fashion other perhaps far more horrible things can be effected. There is

luckily a dearth of experimental data concerning certain types of human error. It is thus emphasized that all should be in the proper shape before presentation to the calculator.

At this point one can consider how the problem is addressed to the calculator and how the various component machines are affected and called into operation. In short, we have reached the stage of revealing the proper buttons to be pushed and the proper time for so doing.

NORMAL USAGE

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Ready a supply of binary cards in the punch.
3. Set the sense switches on the calculator console as desired.

   a. SS1

      up: continuous iteration

      dn: calculator will stop. One iteration is done for each subsequent push of the START key, so long as the SS1 remains down. If after the above stop, the switch is raised and START pressed, iteration will henceforth be continuous.

   b. SS2

      up: "check point" convergence criterion is employed

      dn: "residual method" convergence criterion is employed

   c. SS3

      up: "cutoff mesh"

      dn: "collapse mesh"

d. SS4

up: field deck is suppressed

dn: field deck is produced

e. SS5

up: field print is suppressed

dn: field print is produced

f. SS6

up: potential punch and print both suppressed

dn: potential punch and print both produced

If neither SS4 nor SS5 is down, fields will not even be computed. It is essential to have one of the three, SS4, SS5, or SS6 down, since there would be no output of any kind otherwise. This would be highly undesirable.

4. Prepare the problem deck as follows:

a. part INPUT of the master

b. the data deck

c. the remainder of the master

5. Ready this deck in the hopper of the reader.

6. CLEAR

7. LOAD CARDS

Pursuant to step 7, the following may be expected to occur:

a. cards read (INPUT), data, and LAPLACE

b. SL1 on

c. Laplace is done, unless intentionally skipped

d. cards read (MAIN)

e. SL2 on

f. Main is done, unless intentionally skipped *

g. cards read (FINAL)

h. SL3 on

i. Final is done, unless intentionally skipped

j. cards read (AEND)

k. potentials are punched

l. potentials are printed

m. cards read (BEND)

n. fields are punched

o. cards read (CEND)

p. Fourier analysis is printed

q. cards read (DEND)

r. fields are printed

s. SL4 on, final stop

TO REMOVE MIDWAY

1. Put SS1 down.

2. If SS6 is not already down, put it down.

3. Calculator stops at end of iteration in progress.

4. Raise all switches other than 1 and 6.

5. LOAD CARDS

6. As soon as output starts, raise SS1 immediately.

The result of this procedure is that first a deck (Resumption Deck) is produced, then a print of it is provided. If very soon after the printing phase of this kind

* But carefully consult cautionary note, page 7.

of output begins, a stop is experienced, SS1 was not raised at the proper time. Raise it and push START to obtain the remaining output.

RESUMPTION PROCEDURE

1. Perform steps 1, 2, and 3 of NORMAL USAGE, except see warning concerning SS3 (below).
2. Prepare the problem deck as follows:
   a. Resumption Deck
   b. master with all of INPUT and the first card of LAPLACE removed
   c. ready the deck in the hopper of the reader
3. CLEAR
4. LOAD CARDS

Depending on the phase at the time of prior removal, the problem now will behave as though it had never been removed. In a very long problem it is advisable to perform the ritual of resumption every several hours, since this provides a point of return if machine error destroys any part of the calculations. One may also decide upon seeing intermediate output that enough accuracy has been obtained in a given part, and wish to progress manually to the next. This may always be done, as follows:

TO PROGRESS TO THE NEXT PHASE

1. Put SS1 down and SS6 up (if it was down).
2. The calculator stops.
3. LOAD CARDS
4. If this brings one to the desired phase, raise SS1. If not, repeat step 3 until the desired phase is reached.

5. START

CAUTION!! CAUTION!! CAUTION!!

During the resumption procedure, care must be exercised with respect to SS3. If the resumption deck was obtained from a problem in the final phase with SS3 down, keep SS3 up when resuming and never again during the course of the same problem put it down. In all other cases SS3 may be set as required.

At no time may MAIN be completedly skipped over if FINAL is to be done, or if Fields are to be computed.

CHANGE K, N, OR 1/W

1. Perform steps 1 and 2 of NORMAL USAGE.
2. Put SS1 down, SS3 up, SS2, 4, and 5 as required, and SS6 up.
3. Prepare problem deck as follows:
   a. obtain the proper potential deck (or resumption deck)
   b. leave its first card in place, but remove its 2d, 3rd, and 4th cards
   c. follow the potential deck with INPUT with its first card removed
   d. insert before the last card of INPUT the overwrite named DIRTY
   e. next the data deck
   f. next the remainder of the master
4. Ready this deck in the hopper of the reader.
5. LOAD CARDS
6. The calculator will stop on HPR 77_8.
7. Raise SS1.
8. Set SS6 now as desired.
9. LOAD CARDS

Output will issue forth.

INITIAL TOTAL MESH LOAD

1. Perform steps 1 thru 3 of NORMAL USAGE.
2. Prepare the problem deck as in step 3 of CHANGE K, N. OR 1/W.
3. Perform steps 5 thru 7 of NORMAL USAGE.

Behaviour is like that of NORMAL USAGE.

INCREASE Ω OVERWRITE

To employ this overwrite it is necessary to insert its two cards in the master deck before the last card of the section MAIN and proceed in the usual fashion.

PROGRAMME STOPS

1. HPR 1	Inconsistent data
2. HPR 7	SS1 down in Laplace phase
3. HPR 77	SS1 down in Main phase
4. HPR 777	SS1 down in Final phase
5. HPR 7777	SS1 down in AEND
6. HPR 77770	SS4 and SS5 both up in AEND, end of problem if no fields desired
7. HPR 7070	SS1 down in BEND
8. HPR 77777	SS5 up in CEND or any switch arrangement in DEND. Either signifies the end of the problem. In the latter case the AC and MQ are entirely lit.

Only stops 6 and 8 are normal, and 1, 5, and 7 are always error stops.

FORMESH - Programme 26

cf. Internal Report 222

The data deck employs both integers and fractions. It is presumed that an agendum has been prepared in accordance with the requirements stated in the internal report. Integers are punched in the usual fashion, with the words FINISH INTEGERS punched in 12R of the last integer card. Fractions are punched in the usual way also, with the words FINISH FRACTIONS punched in 12R of the last fraction card. A series of runs can be made without entering every value in succeeding runs since some (but not all) of them are retained in the computer's memory from one run to the next, unless a subsequent data card with a new value is specifically entered. There must always be at least one integer and one fraction to a given run, however; otherwise some completely inane behaviour can well be expected from the machine. After the last fraction card of the last run in a series two special integer cards are needed to cause the characteristic stop which signifies the end of the series. The first has address 7597 and value 0 with BON in 12R; the second, address 7620, value 0, with NUIT in 12R.

The problem is addressed to the machine as follows:

NORMAL USAGE

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set the sense switches on the console as desired. Each sense switch is tested on each iteration, so in all cases the result of pushing START after some stop will depend on the setting of sense switches other than the one responsible for the stop.

a. SS1

up: stop, do one iteration per START

dn: continuous iteration

b. SS2

up: call in next problem immediately

dn: continue problem in hand until it is done

c. SS3

up: stop, do one iteration per START

dn: continuous iteration

d. SS4

up: stop, do one iteration per START

dn: continuous iteration

e. SS5

up: stop, do one iteration per START

dn: continuous iteration

f. SS6

up: search, i.e., stop at end of first run which is stable for N_e steps

dn: continue computing succeeding parts until BON, NUIT cards are encountered

3. Prepare the problem deck as follows:

a. Forocyl, Formerge, or Tempermesh field deck

b. master deck

c. data deck(s)

4. Ready this deck in the reader hopper.

5. CLEAR

6. LOAD CARDS

Pursuant to step 6, the following may be expected to occur:

a. cards read (FIELD DECK and PRE)

b. print information concerning mesh

c. read cards (MAIN and data)

d. compute and print

e. read cards

If these cards are BON, NUIT, stop; otherwise have a new problem and repeat d. and e. until all the parts are processed.

FORMESH INVARIANTS

This programme is used in an identical fashion to Formesh, except that it computes invariants if SS3 is down and omits them if SS3 is up. It further limits the size of the mesh used, as explained in a memorandum relative to allowable mesh size. The extra fractions are inserted with the fraction group of the first Formesh run.

OPEN-ENDED FORMESH

Because of the nature of the programme it is not possible to make the end constant N_e greater than 7680. It is possible to run the problem any number of steps if the open-ended procedure is employed. To do this, insert the card called the Formesh Infinity Card before the last card of the Formesh master deck. This will disable the counting provision of the programme. To terminate a given run it will be necessary for the operator to intervene, since otherwise

the problem would endure until some computational or mechanical error was made. The problem in progress is terminated by raising SS2 until the card reader starts. It is then lowered immediately, since either the next problem or the terminating cards have just been read.

PROGRAMME STOPS

In PRE:

1. HPR 1	1/w too big
2. HPR 2	k' too big
3. HPR 3	N too big
4. HPR 4	$\frac{1}{2\pi}\sqrt{\left(\frac{1}{\omega}\right)^2 + N^2}$ too big
5. HPR 5	N too small

In MAIN:

6. HPR 7	SS3 up
7. HPR 77	SS4 up
8. HPR 777	SS5 up
9. HPR 77777	SS1 up
10. HPR 77770	SS6 up and search is ended
11. HTR 16722	end of a series of runs

SCOFFLAW - Programme 30

cf. Internal Report 223

The tables of $F_p/100$ and $F_y/100$ are entered as fractions on the agendum sheets, but their addresses do not appear on these sheets. The data deck has the following form:

a. The table of $F_p/100$ is punched in the same fashion as a set of ordinary fractions, with the first value having an address 1000, the second, 1001, and so on. The last entry has the signal FPEE TABLE punched in its 12R field. Even if there is no table of values of $F_p/100$, a zero should be entered in address 1000, with the signal in 12R.

b. The table of $F_y/100$ is prepared exactly the same way, except that the addresses are 2000, 2001, etc., and the last card has the 12R signal FWYE TABLE punched. Again, if no table is desired, a zero should be entered in address 2000, with the 12R signal above.

c. Integers are straightforward, and have the signal COMPLETE INTEGERS in 12R of the last integer card.

d. Fractions are also of the usual form, with the 12R signal COMPLETE FRACTIONS.

e. The last card of a series of runs has address 500, value 0, and the 12R words GOOD NIGHT.

The problem is addressed to the calculator as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.

2. Set all sense switches down.

3. Prepare the problem deck as follows:

    a. master deck

    b. data deck(s)

4. Ready this deck in the reader hopper.

5. CLEAR

6. LOAD CARDS

The final stop is HTR 7164_8. All other stops are errors.

JOYBUCKETS - Programme 32

cf. Internal Report 224

Agenda are prepared and punched. All data are in the standard floating point notation, and the card with address 304 always has END in its 12R field. The data deck has as many of these sets as needed, and is ended by two blank cards.

It is addressed to the calculator as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set the sense switches as desired.
3. Ready the problem deck in the hopper of the reader. It is of the form:
    a. master
    b. data deck(s)
    c. 2 blank cards
4. CLEAR
5. LOAD CARDS

TTT - Programme 39

cf. Internal Report 225

As will be recalled from the internal report, there are two distinct sections to a TTT data deck, viz., the scenario and the runs. We will consider the first one first. The scenario number is punched beginning in column 1 and as many succeeding columns as may be needed, provided that the number is 8 digits or less; otherwise the value used will be zero. The signal SENARIO NUMBER will appear in 12R. It is necessary to be sure that the word scenario is misspelled as it is above and not some other way. Following the scenario number card will come the remainder of the description of a machine, in the order on the agendum sheet, one card per line. Although some of the data in the scenario are not fractions, it will be useful to think of them as such. In column 1 will be punched the χ_1 digit of the first component to be described. Column 2 is blank. Column 3 contains the χ_2 digit, or a blank in the case of a foil. 12R contains the appropriate sign, ANGLE, GAP, or FOIL. Each component card is followed by a set of value cards which should be punched as though they were all fractions with addresses as given in the Type column of the agendum.

The last card of the scenario is blank but for its 12R signal, which is CURTAIN.

Following the scenario will be a run or series of runs, which consists of integers and fractions, in that order, both in the standard form, with the final integer card bearing the 12R mark END INTEGERS, the last fraction card having in its 12R the words END FRACTIONS.

After the last run of a series there will be one blank card.

The problem is then addressed to the calculator.

1. Ready the MURA PRINTER BOARD 1 in the printer.

2. Set the sense switches as desired.

   a. SS1

      up: stop, one iteration per START

      dn: continuous iteration

   b. SS2

      up: call for next run immediately

      dn: wait until this run is done

   c. SS3

      up: stop, one iteration per start

      dn: continuous iteration

   d. SS4

      up: α, ϕ, E, t print suppressed

      dn: α, ϕ, E, t printed

   e. SS5

      up: ϕ, E, t, x, p print suppressed

      dn: ϕ, E, t, x, p printed

   f. SS6

      up: search

      dn: continue series until blank card is read

3. Prepare the problem deck as follows:

   a. master

b. data deck

(1) scenario

(2) run(s)

c. 1 blank card

4. Ready the problem deck in the reader hopper.

5. CLEAR

6. LOAD CARDS

EXTRA PRINT OVERWRITE

To employ this overwrite one must insert it (3 cards) before the last card of the master deck at the outset of a problem that is to use the facility.

PROGRAMME STOPS

1. HPR 1	Run ID number 0, and scenario card of wrong form
2. HPR 2	SENARIO not correctly misspelled
3. HPR 3	12R punch on wrong card of scenario
4. HPR 4	ANGLE, GAP, or FOIL misspelled
5. HPR 15	END INTEGERS misspelled
6. HPR 16	END FRACTIONS misspelled
7. HPR 77	SS1 up
8. HPR 777	SS3 up
9. HPR 7777	SS6 up, end of series if search
10. HPR 77777	final stop, end of series

There are other stops, but none of them is normal or of interest to the user.

WELL TEMPERED FIVE - Programme 46

cf. Internal Report 226

The data deck of this programme has two types of data card, integers and floating point numbers. It is assumed that an agendum has been prepared in accordance with the rules given in the internal report. Integers are prepared in the usual fashion, and are followed by a card with a 1 in column 1 and the word STOPINHERE in 12R. This card is followed by the floating point cards which are punched in the usual fashion, followed by a card with a 1 in column 1 and the word STOPFLOPHERE in 12R. A series of these may be placed one after the other, and the whole series is terminated with a special card with 1 in column 1 and the word OSTOPIT in 12R.

The problem is entered into the calculator in the following way:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set the sense switches as desired. Their functions are explained in the internal report. SS4 and SS5 are not used.
3. Prepare the problem deck as follows:
   a. master
   b. data deck(s)
4. Ready this deck in the reader hopper
5. CLEAR
6. LOAD CARDS

PROGRAMME STOPS

1. HPR 7 $\epsilon = 0$ and SS3 down, sense light 1 on
2. HPR 77 $i_{max} = 0$ and SS3 up, sense light 2 on

3. HPR 777	SS3 down, and $i_{max} > 24$
4. HPR 7777	$\mu_{0,m} \equiv \nu_{0,m} \equiv 0$ for all m
5. HPR 7	SS1 up, end of integration cycle
6. HPR 1	SS6 up, end of search
7. HTR 1751	STOPINHERE or STOPFLOPHERE cards missing
8. HPR 77777	final stop

Other stops exist but are not of interest to the user.

FORMERGE - Programme 50

cf. Internal Report 227

There are three data cards, an integer and two fractions. The integer has the 12R signal PARTITIONS, as indicated on the agendum. With this we are ready to enter the problem into the calculator. This is done as follows:

1. Ready a supply of cards in the punch.
2. Set SS1 and SS2 up.
3. Prepare problem deck as follows:
    a. Forocyl A potential deck
    b. master
    c. data
    d. Forocyl B potential deck with its 1st four cards removed
4. Ready the problem deck in the reader hopper.
5. CLEAR
6. LOAD CARDS

There is no printed output. Printout of the merged potentials may be obtained in the manner prescribed in the internal report.

PROGRAMME STOPS

1. HPR 7	sum check failure in punching of potentials
2. HPR 77	sum check failure in punching of current values
3. HPR 77777	final stop

FORANAL - Programme 52

cf. Internal Report 228

The data deck is of a special form. Column 1 is always blank and the values are entered without addresses immediately following this blank. The 12R signals ALPHA, RHO, and LAMBDA appear on the appropriate cards, and the very last card is blank save for the word in 12R, which is OMEGA.

The problem is given to the calculator in the following way:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set SS1 as desired.
3. Ready the problem deck in the reader hopper. It has the form
    a. master
    b. data deck
4. CLEAR
5. LOAD CARDS

PROGRAMME STOPS

1. HPR 1	read-in error
2. HPR 2	too many data
3. HPR 77777	final stop

The other stops that exist denote exceeding of scale limits.

ATEMESH - Programme 53

cf. Internal Report 229

This programme is in its operational respects identical with Formesh and Formesh Invariants, q.v.

EQUICYL - Programme 54

cf. Internal Report 230

The data deck is slightly different from usual in that it mixes fractions and integers. One simply punches all numbers as though they were fractions, using the addresses given for the first five cards and those entered in the column marked 100i+j on the agendum, taking the values from the adjacent columns for the other cards. The first five cards have the 12R signal PARAMETERS; the others have no 12R signal, save for the last card, which is blank except for the 12R word END, as indicated on the agendum form. One may now enter the problem into the calculator in the following manner:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set all sense switches up.
3. Prepare the problem deck as follows:
    a. Forocyl or Formerge potential deck
    b. master
    c. data deck(s)
    d. blank card
4. Ready this deck in the reader hopper
5. CLEAR
6. LOAD CARDS

PROGRAMME STOPS

1. HPR 1	1/w too big
2. HPR 2	k' too big

3. HPR 3 N too big

4. HPR 77777 final stop, end of series

There are other stops which denote exceeding of scale limits, and all are errors.

FORFIXPOINT - Programme 55

cf. Internal Report 231

The data deck consists of integers and fractions, which are punched in the usual manner, with the former having the 12R signal END INTEGERS on the last card, the latter, the 12R signal END FRACTIONS on the last card. The data deck is combined with the other necessary cards and fed into the calculator as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set the sense switches as desired. Normally all are down, but either the fixed point search or the rotation number and invariants computation may be omitted by raising the proper switch, as indicated on the agendum sheet.
3. Prepare the problem as follows:
   a. field deck produced by Forocyl or Formerge
   b. part I of the master
   c. data deck
   d. part II of the master
4. Ready the problem deck in the reader hopper
5. CLEAR
6. LOAD CARDS

PROGRAMME STOPS

In part I:

1. HPR 7 SS1 up

In part II:

2. HPR 77777 final stop

SIXTEENMESH - Programme 56

cf. Internal Report 232

This problem is identical in its technical operation to Formesh and Formesh Invariants, q. v.

FORERUNNER - Programme 57

cf. Internal Report 240

The data deck consists of both integers and fractions. The final integer in a given run has the 12R words END INTEGERS punched; the final fraction in a given run has the 12R words END FRACTIONS punched. Having accomplished this much, we are now ready to enter the problem into the calculator. This is done in the following way:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set the sense switches as desired.

   a. SS1

      up: stop

      dn: continuous iteration

   b. SS2

      up: call in next run immediately

      dn: complete run in progress

   c. SS3

      up: stop

      dn: continuous iteration

   d. SS4

      up: stop

      dn: continuous iteration

   e. SS5

      up: stop

      dn: continuous iteration

f. SS6

up: search

dn: do all runs, whether stable or not

3. Prepare the problem deck as follows:

a. master deck

b. data deck(s)

c. extra integer card, address 100, value 0, 12R word END SERIES

4. Ready this deck in the reader hopper

5. CLEAR

6. LOAD CARDS

PROGRAMME STOPS

1. HPR 7	SS1 up
2. HPR 777	SS3 up
3. HPR 7777	SS4 up
4. HPR 77777	SS5 up
5. HPR 777777	SS6 up, end of search
6. HTR 2114	SS6 down, end of series

ALGYTEE - Programme 58

cf. Internal Report 233

The data deck consists of integers and floating point numbers. Integers are punched in the usual manner, with the 12R words END INTEGERS on the last integers card. They are followed by the floating point cards, also punched in the usual manner. After the last floating point datum in any run there is a special card which has a 1 in column 1 and the 12R signal END FLOAT. POINT NOS. If a series of runs is desired, this card is followed by the first integer card of the next problem, and so on. It is also possible to enter the floating point numbers as octal data. The floating point numbers in question are omitted from the above preparations and punched in the form and format given in the introduction. The octal cards are not put with the ordinary integer and floating point number data, and may be entered only at the very beginning of a series of runs, that is, they must remain unchanged throughout a given series.

The problem is given to the calculator as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set sense switches 1, 2, 3, and 5 down. Set SS4 up if bumps desired, down if not; SS6 up for search feature, down for continuous running.
3. Prepare the problem deck as follows:
   a. master deck with octal data (if any) before its last card
   b. data deck(s)
4. Ready this deck in the reader hopper.
5. CLEAR
6. LOAD CARDS

PROGRAMME STOPS

1. HPR 77777 end of a series of runs

2. HPR 6 SS6 up, end of search

Other stops indicate data error or incorrect sense switch settings.

FORMESH FUMBLEBUMPS - Programme 60

cf. Internal Report 234

This problem is prepared in the same way as Formesh. Extra integer data are placed with the integers of the first run of a Formesh series; extra fraction data are placed with the fractions of that run. To discontinue application of the transformations, put SS4 up. Returning it to the down position will cause the transformations to be resumed. Phase will not be lost during the interruption.

FORMESH FUMBLEBUMPS INVARIANTS

This programme is used in a fashion identical to Formesh Fumblebumps except that it computes invariants if SS3 is down, omitting them if it is up. It further limits the allowable mesh size. Extra fraction data are inserted with the fractions of the first Formesh run in a series.

FORMESH MUMBLEBUMPS - Programme 61

cf. Internal Report 235

The Formesh part of the data is prepared in the usual manner. The extra integers for bumps are placed with the integer group of the first Formesh run in a series. The two extra fraction data are not placed in the normal data deck, but are inserted into the master deck, as will be explained below. The second of the two cards bears the 12R words MUMBLEBUMPS DATA. The problem is given to the calculator in the same way as Formesh, with two exceptions: the bumps may be discontinued by raising SS4; the problem deck has a very complex form, and extreme care should be paid to the task of preparing it, which is done as follows:

a. mesh A (normal mesh), with its 2d and 3rd cards removed and placed aside for use later

b. cards A1, A2, and SBL of the master

c. mesh B (bumped mesh), with its 1st, 2d, and 3rd cards removed

d. cards B1, B2, and BL of the master

e. two cards from mesh A which were laid aside above

f. part I of the master

g. two extra fraction data cards

h. remainder of master deck (part II)

i. normal data deck(s)

FORMESH MUMBLEBUMPS INVARIANTS

This programme is identical with Formesh Mumblebumps except for the fact that it computes invariants if SS3 is down and omits them if it is up. It places a further restriction on the mesh size allowable.

FORFIXPOINT FUMBLEBUMPS - Programme 66

cf. Internal Report 231

This programme is used in the same manner as Forfixpoint. Extra integer data are placed with the integer group of the first run in a series; extra fraction data are placed with the fraction group of the same run.

FORFIXPOINT GRUMBLEBUMPS - Programme 67

cf. Internal Report 231

This programme is used in the same fashion as Forfixpoint. Extra integer data are placed with the integer data of the first run in a series; extra fractions are placed with the fraction data of that run.

DUCK ANSWER - Programme 75

cf. Internal Report 237

The data deck consists of integers and fractions, both punched in the normal format. The final integer card has the 12R words END INTEGERS, and is followed by the fraction data, the last card of which bears the 12R words END FRACTIONS. After a series of runs a special terminating card is needed, with address 30, value 0, and 12R words STOP SERIES. The problem is placed in the calculator as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Set the sense switches as desired.

   a. SS1

      up: stop

      dn: continuous iteration

   b. SS2

      up: call for next problem immediately

      dn: continue problem in hand until completion

   c. SS3

      (not used)

   d. SS4

      (not used)

   e. SS5

      up: stop

      dn: continuous iteration

f. SS6

up: search

dn: process all runs available, whether stable or not

3. Prepare the problem deck as follows:

a. master deck

b. data deck(s)

4. Ready this deck in the reader hopper

5. CLEAR

6. LOAD CARDS

PROGRAMME STOPS

1.	HPR 7	SS1 up
2.	HPR 77	SS5 up
3.	HPR 7777	SS6 up, end of search
4.	HPR 1750	end of series

TEMPERMESH - Programme 76

cf. Internal Report 241

The data deck has integers and floating point numbers, both punched in the normal format. After the integers there is a special card with a 1 in column 1 and the 12R word STOPINHERE. After the floating point numbers there is a special card with 1 in column 1 and the 12R word STOPFLOPHERE. This card is followed by another with a 1 in column 1 and the 12R word OSTOPIT. The problem is addressed to the calculator as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Ready a supply of binary cards in the punch.
3. Set the sense switches as desired. Normally all are down except 3, which is set as indicated on the agendum. Printing of the fields can be suppressed by raising SS4, though in this case the initial conditions are printed out anyway.
4. Prepare the problem deck as follows:
   a. master deck
   b. data deck(s)
5. Ready this deck in the reader hopper.
6. CLEAR
7. LOAD CARDS

The final stop is HPR 77777. The field deck output must have a card placed in front of it before it is usable. This card should be the Binary Loader MU LBL3.

INVARIANT DUCK BUMPS - Programme 77

cf. Internal Report 238

This programme is prepared in the same manner as Duck Answer except that the extra integer data (in the case of bumps) are placed with the integer data of the first run of the Duck Answer data deck, and the extra fraction data are in the case of bumps (or invariants) placed with the fractions of that run. The problem is addressed to the calculator in the same way as Duck Answer, except for the use of two sense switches, which is modified in the manner indicated in the Internal Report.

MESSY MESSY - Programme 78

cf. Internal Report 239

The data deck for this programme is of a special sort. The first card is an integer of the normal kind, bearing a 12R signal ID NUMBER. It is followed by a Type card with column 1 blank and a Type digit in column 2. This card is followed by a δ card, which is punched in the same fashion as a floating point number, except that it has no address, and therefore column 1 is blank or plus, or minus, as the case may be. This card is followed by a t card, which is of the same form as the δ card. A series of these three, Type, δ, and t is repeated as needed, and the last card in a given problem bears the 12R word FLOATING POINTS.

The problem is addressed to the calculator as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Prepare the problem deck as follows:
   a. master
   b. data deck(s)
3. Ready this deck in the reader hopper.
4. CLEAR
5. LOAD CARDS

FORMESH FIELD PRINT OVERWRITE - Programme 86

cf. Internal Report 241

This programme is used by inserting it before the last card of the master deck, and employing the otherwise normal usage of the Formesh programme.

FORSUPS - Programme 99

cf. Internal Report 338

There is no data deck for this problem. The problem deck consists of the two sets of potentials and the master deck, which are used as follows:

1. Ready the MURA PRINTER BOARD 1 in the printer.
2. Prepare the problem deck as follows:
   a. potential deck F (squares)
   b. master
   c. potential deck G (products)
3. Ready this deck in the reader hopper
4. CLEAR
5. LOAD CARDS

To perform squares only, simply do not put a G deck after the master.

PROGRAMME STOPS

1. HTR 24	check sum failure in reading of G deck
2. HPR 7	a and b of G deck not identical to those of F deck
3. HPR 70	error, part of G deck missing
4. HPR 77777	final stop

FORMESH (INVARIANTS) SCOPE - Programme 113

FORMESH (INVARIANTS) FUMBLEBUMPS SCOPE - Programme 114

FORMESH (INVARIANTS) GRUMBLEBUMPS SCOPE - Programme 115

cf. Internal Report 296

Each of these three programmes is used in the same way as its non-scope correspondent, with the addition of integer data to the integer group of the first in a series of Formesh runs and fraction data to the fraction group of that run. Finally, one should of course make certain that the 740 Recorder is furnished with film and the slide removed prior to running the problem. Scoping will be discontinued as long as SS5 is raised.

INTEGER SUPPRESS OVERWRITE

To employ this device, place it before the last card of the master and proceed as usual.

SCOPE INVARIANT DUCK BUMPS - Programme 116

cf. Internal Report 298

This programme is used the same way as Invariant Duck Bumps, with additional integer and fraction data placed in their respective groups. If SS5 is raised, scoping will be discontinued until the switch is depressed. Of course the user should be certain that the camera slide has been removed.

INTEGER SUPPRESS OVERWRITE

To employ this device, place it before the last card of the master and proceed as usual.

EQUICYL SCOPE - Programme 118

cf. Internal Report 297

Extra fraction cards, each with the 12R word PARAMETER, are placed after the 5th card of an Equicyl data deck, which is then used in the normal fashion, except that scoping is discontinued if SS5 is raised.

FRAME ADVANCE OVERWRITE

To use this overwrite, insert it before the last card of the master and proceed as usual.

MAGNIFY SCALE AND SUPPRESS OVERFLOW OVERWRITE

To use this overwrite, insert it before the last card of the master and proceed as usual.

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION
COMPUTER DIVISION

PROGRAM INDEX

Program Sequence Number	Name	SHARE Mnemonic	MURASS Identification	Production Number	Status	Author
1	MURA Assembler	MURASS	LL101			Fosdick
2	Transfer Test (Visual)	MU TTV1	LN102			Snyder
3	Effective Address Search Routine	MU EAS1	LN302		Obsolete	Snyder
4	Effective Address Search Routine	MU EAS2	LN303			Snyder
5	Lower Binary Loader (One Card)	MU LBL1	LI105		Obsolete	Snyder
6	Upper Binary Loader (One Card)	MU UBL1	LI106			Snyder
7	Relocatable to Absolute Translator	MU RAT1	LL102			Snyder
8	Read Decimal Integer Routine	MU RDI1	LI302			Snyder
9	Read Decimal Fraction Routine	MU RDF1	LI303		Obsolete	McNall
10	Read Octal Number Routine	MU RON1	LI202			Storm
11	Fixed Point Runge-Kutta	MU RKY1	LD202			Fosdick
12	Variable Column Fraction Print	MU PRF1	LJ308			Fosdick
13	FOROCYL			P013		Snyder
14	Six Column Fraction Print	MU PRF2	LJ309			Fosdick
15	Variable Column Integer Print	MU PRI1	LJ310			Fosdick
16	Six Column Integer Print	MU PRI2	LJ311			Fosdick
17	Binary Punch Routine	MU BPU1	LJ103			Zographos
18	Read Decimal Integer Routine	MU RDI2	LI304			McNall
19	Read Decimal Fraction Routine	MU RDF2	LI305		Obsolete	McNall
20	Fixed Point Square Root	MU SQR1	LB403		Obsolete	Fosdick
21	Binary Punch Routine	MU BPU2	LJ104			Zographos
22	Self Loading Binary Punch Routine	MU BPU3	LJ105			Zographos
23	Fraction Dump	MU FRD1	LN202			Fosdick
4	Octal Dump	MU OCD1	LN204			Fosdick
25	Fixed Point Sine	MU SIN1	LB106		Obsolete	Fosdick
26	FORMESH			P026		Snyder

27	Fixed Point Exponential, Base e	MU EXP1	LB304			Fosdick
28	Fixed Point Logarithm, Base e	MU LOG1	LB305		Obsolete	Fosdick
29	Integer Dump	MU IND1	LN203			Zographos
30	SCOFFLAW			P030		Snyder
31	LIOUVILLE TEST 1			P031		Fosdick
32	JOYBUCKETS			P032		Joyce
33	DUCK CALL			P033	Obsolete	Ballance and Brons
34	SET OF EXPONENTIALS			P034		McNall
35	Read Floating Decimal Routine	MU RFD1	LI301		Obsolete	McNall
36	Variable Column Floating Decimal Print	MU PFD1	LJ312			McNall
37	Fixed Point Logarithm, Base 2	MU LOG2	LB306			Zographos
38	Fixed Point Exponential, Base 2	MU EXP2	LB307			Fosdick
39	TTT			P039		Snyder
40	Double Precision Floating Point Addition	MU DPA1	LA113			Zographos
41	Division Round	MU DVR1	LA114		Obsolete	Zographos
42	Multiplication Round	MU MPR1	LA115		Obsolete	Zographos
43	Binary Check Sum Checker and Corrector (One Card)	MU CSC1	LQ102		Obsolete	Ballance
44	Lower Binary Loader	MU LBL2	LI107		Obsolete	Ballance
45	Floating Point Runge-Kutta	MU RKY2	LD203		Obsolete	Zographos
46	WELL TEMPERED FIVE			P046		Fosdick
47	JOYBUCKETS CHECK			P047		Joyce
48	POINT 34 INTEGRATION			P048		Parker
49	Floating Point Cube Root	MU CRT1	LB404			Storm
50	FORMERGE			P050		Snyder
51	Fixed Point Arctangent	MU ATN1	LB108			King
52	FORANAL			P052		Snyder
53	ATEMESH			P053		Snyder
54	EQUICYL			P054		Snyder
55	FORFIX POINT			P055		Snyder

56	SIXTEENMESH			P056		Snyder
57	FORERUNNER			P057		King
58	ALGYTEE			P058		Storm
59	Shifting Binary Loader (One Card)	MU SBL1	LI108		Obsolete	Snyder
60	FORMESH FUMBLEBUMPS			P060		Snyder
61	FORMESH MUMBLEBUMPS			P061		Snyder
62	FORMESH GRUMBLEBUMPS			P062		Snyder
63	LOG AND EXP TEST			P063		Storm
64	LIOUVILLE TEST 2			P064		Zographos
65	FLOATING SINE TABLE			P065		Joyce
66	FORFIXPOINT WITH FUMBLEBUMPS			P066		Snyder
67	FORFIXPOINT GRUMBLEBUMPS			P067		Snyder
68	TESTS OF MU RKYZ			P068		Zographos
69	Floating Decimal Dump	MU FDD1	LN205			McNall
70	Lower Binary Loader (One Card)	MU LBL3	LI109			Snyder & Storm
71	Gauss Quadrature (Three Point)	MU GQF1	LD101			Zographos
72	Read Decimal Fraction Routine	MU RDF3	LI310			McNall
73	Read Decimal Fraction Routine	MU RDF4	LI311			McNall
74	Double Precision Addition (Fixed Point)	MU DPA2	LA116			Snyder
75	DUCK ANSWER			P075		Snyder
76	TEMPERMESH			P076		Fosdick
77	INVARIANT DUCKBUMPS			P077		Snyder
78	MESSY MESSY			P078		Zographos
79	VAN BLASUM			P079		McNall
80	PARKBUCKS			P080		Parker
81	Variable Column Integer or Fraction Print	MU PIF1	LJ315			Zographos
82	Self Loading Binary Punch Routine	MU BPU4	LJ106			Storm
83	24 Word Per Card Binary Loader (One Card)	MU LBL4	LI110			Storm
84	Variable Column, Variable Digit Rounded Fraction Print	MU PRF3	LJ316			Zographos
85	A BRUCK CONJECTURE			P085		Lordan

86	FORMESH FIELD PRINT OVERWRITE			P086		Snyder
87	The Reflective 704	MU 704R	LO101			Snyder
88	Fixed Point Square Root	MU SQR2	LB407			Fosdick
89	Shifting Binary Loader (One Card)	MU SBL2	LI111			Snyder
90	Binary Check Sum Checker and Corrector (One Card)	MU CSC2	LQ103			Ballance
91	Integer Cathode Ray Tube Display	MU SCP1	LJ607			Storm
92	Fixed Point Sine	MU SIN2	LB112			Zographos
93	Fixed Point Sine	MU SIN3	LB113			Zographos
94	Variable Column Floating Decimal Print	MU PFD2	LJ318			McNall
95	Read Floating Decimal Routine	MU RFD2	LI312			McNall
96	6 Column Fraction Cathode Ray Tube Display	MU SCP2	LJ608			King
97	General Alphanumeric Cathode Ray Tube Display	MU SCP3	LJ606			Fosdick
98	Random Number (Modular Method)	MU RAN1	LG502			Snyder
99	FORSUPS			P099		Anderson
100	Cathode Ray Tube Point Plotter	MU SCP4	LJ609			Zographos
101	Fixed Point Logarithm, Base e	MU LOG3	LB314			Fosdick
102	Six Column Fraction Print	MU PRF4	LJ319			Storm
103	Floating Point Runge-Kutta	MU RKY3	LD207			Zographos
104	Floating Decimal Dump	MU FDD2	LN208			McNall
105	Variable Column Fraction Cathode Ray Tube Display	MU SCP5	LJ610			King
106	Six Column Integer Cathode Ray Tube Display	MU SCP6	LJ611			King
107	Variable Column Integer Cathode Ray Tube Display	MU SCP7	LJ612			King
108	Fixed Point Square Root	MU SQR3	LB417			Fosdick and Snyder
109	Fixed Point Cube Root	MU CRT2	LB418			Snyder
110	SAW TOOTHED POLLY			P110		Clark
111	Floating Multiply and Round or Floating Add and Round	MU MAR1	LA126			Storm
112	MONTE CRYSTO I			P112		Fosdick
113	SCOPE FORMESH (INVARIANT)			P113		Snyder
114	SCOPE FORMESH (INVARIANT) FUMBLEBUMPS			P114		Snyder
115	SCOPE FORMESH (INVARIANT) GRUMBLEBUMPS			P115		Snyder
116	SCOPE INVARIANT DUCK BUMPS			P116		Snyder

117	LINEAR ALGYTEE		P117			Hutchinson
118	EQUICYL SCOPE		P118			Snyder
119	TTT SCOPE		P119			Snyder
120	CHANNEL 5		P120			Fosdick
121	ALGYTEE SCOPE		P121			Storm
122	CROCOR		P122			McNall

MURA 704 PERFORMANCE STATISTICS

CURRENT MONTH (Dec. 1 - Dec. 31, 1956)
(One machine - one shift - 181.33 hours)

A. AVAILABLE		86.13%
1 Compute	69.11%	
2 Checkout	8.08%	
3 Assembly	6.28%	
4 Data Error	.13%	
5 Computer Error	2.53%	
6 Idle	--	
B. NOT AVAILABLE		13.87%
1 Maintenance, scheduled	9.36%	
2 Maintenance, unscheduled	4.51%	
3 Air conditioning and power failure	--	

ACCUMULATIVE (Nov. 7 - Dec. 31, 1956)
(One machine - one shift - 352.78 hours)

A. AVAILABLE		83.76%
1 Compute	69.07%	
2 Checkout	5.90%	
3 Assembly	6.48%	
4 Data Error	.86%	
5 Computer Error	1.45%	
6 Idle	--	
B. NOT AVAILABLE		16.24%
1 Maintenance, scheduled	9.05%	
2 Maintenance, unscheduled	6.72%	
3 Air conditioning and power failure	.47%	

MURA 704 PERFORMANCE STATISTICS

CURRENT MONTH (Jan. 1 - Jan. 31, 1957)
(One machine - one shift - 228.12 hours)

A. AVAILABLE		88.35%
1 Compute	71.44%	
2 Checkout	9.24%	
3 Assembly	6.60%	
4 Data Error	.66%	
5 Computer Error	.41%	
6 Idle	--	
B. NOT AVAILABLE		11.65%
1 Maintenance, scheduled	8.97%	
2 Maintenance, unscheduled	2.68%	
3 Air conditioning and power failure	--	

ACCUMULATIVE (Nov. 7, 1956 - Jan. 31, 1957)
(One machine - one shift - 580.90 hours)

A. AVAILABLE		85.57%
1 Compute	70.02%	
2 Checkout	7.21%	
3 Assembly	6.52%	
4 Data Error	.78%	
5 Computer Error	1.04%	
6 Idle	--	
B. NOT AVAILABLE		14.43%
1 Maintenance, scheduled	9.02%	
2 Maintenance, unscheduled	5.13%	
3 Air conditioning and power failure	.28%	

MURA 704 PERFORMANCE STATISTICS

CURRENT MONTH (Feb. 1 - Feb. 28, 1957)
(One machine - one shift - 206.72 hours)

A. AVAILABLE		87.43%
1 Compute	79.55%	
2 Checkout	4.60%	
3 Assembly	2.95%	
4 Data Error	0.03%	
5 Computer Error	0.30%	
6 Idle	--	
B. NOT AVAILABLE		12.57%
1 Maintenance, scheduled	9.31%	
2 Maintenance, unscheduled	2.25%	
3 Air conditioning and power failure	1.01%	

ACCUMULATIVE (Nov. 7, 1956 - Feb. 28, 1957)
(One machine - one shift - 787.62 hours)

A. AVAILABLE		86.05%
1 Compute	72.51%	
2 Checkout	6.52%	
3 Assembly	5.59%	
4 Data Error	0.58%	
5 Computer Error	.85%	
6 Idle	--	
B. NOT AVAILABLE		13.95%
1 Maintenance, scheduled	9.10%	
2 Maintenance, unscheduled	4.38%	
3 Air conditioning and power failure	.47%	

MURA 704 PERFORMANCE STATISTICS

CURRENT MONTH (March 1 - March 31, 1957)
(One machine - one shift - 213.70 hours)

A. AVAILABLE		91.08%
1 Compute	83.79%	
2 Checkout	2.85%	
3 Assembly	3.23%	
4 Data Error	.17%	
5 Computer Error	1.04%	
6 Idle	--	
B. NOT AVAILABLE		8.92%
1 Maintenance, scheduled	6.46%	
2 Maintenance, unscheduled	2.10%	
3 Air conditioning and power failure	.36%	

ACCUMULATIVE (Nov. 7, 1956 - March 31, 1957)
(One machine - one shift - 1001.32 hours)

A. AVAILABLE		87.13%
1 Compute	74.91%	
2 Checkout	5.74%	
3 Assembly	5.09%	
4 Data Error	.50%	
5 Computer Error	.89%	
6 Idle	--	
B. NOT AVAILABLE		12.87%
1 Maintenance, scheduled	8.53%	
2 Maintenance, unscheduled	3.89%	
3 Air conditioning and power failure	.45%	

MURA 704 PERFORMANCE STATISTICS

CURRENT MONTH (April 1 - April 30, 1957)
(One machine - one shift - 224.75 hours)

A. AVAILABLE		92.26%
1 Compute	84.62%	
2 Checkout	4.08%	
3 Assembly	3.14%	
4 Data Error	--	
5 Computer Error	.42%	
6 Idle	--	
B. NOT AVAILABLE		7.74%
1 Maintainance, scheduled	7.63%	
2 Maintainance, unscheduled	.11%	
3 Air conditioning and power failure	--	

ACCUMULATIVE (Nov. 7, 1956 - April 30, 1957)
(One machine - one shift - 1226.07 hours)

A. AVAILABLE		88.07%
1 Compute	76.70%	
2 Checkout	5.43%	
3 Assembly	4.73%	
4 Data Error	.41%	
5 Computer Error	.80%	
6 Idle	--	
B. NOT AVAILABLE		11.93%
1 Maintainance, scheduled	8.37%	
2 Maintainance, unscheduled	3.20%	
3 Air conditioning and power failure	.36%	

MURA 704 PERFORMANCE STATISTICS

CURRENT MONTH (May 1 - May 31, 1957)
(One machine - one shift - 248.57 hours)

A. AVAILABLE		85.23%
1 Compute	74.43%	
2 Checkout	4.22%	
3 Assembly	3.22%	
4 Data Error	.23%	
5 Computer Error	3.13%	
6 Idle	--	
B. NOT AVAILABLE		14.77%
1 Maintainance, scheduled	8.47%	
2 Maintainance, unscheduled	5.91%	
3 Air conditioning and power failure	.39%	

ACCUMULATIVE (Nov. 7, 1956 - May 31, 1957)
(One machine - one shift - 1474.30 hours)

A. AVAILABLE		87.59%
1 Compute	76.31%	
2 Checkout	5.23%	
3 Assembly	4.48%	
4 Data Error	.38%	
5 Computer Error	1.19%	
6 Idle	--	
B. NOT AVAILABLE		12.41%
1 Maintainance, scheduled	8.39%	
2 Maintainance, unscheduled	3.65%	
3 Air conditioning and power failure	.37%	

MURA 704 PERFORMANCE STATISTICS

CURRENT MONTH (June 1 - June 30, 1957)
(One machine - one plus shift - 281.48 hours)

A. AVAILABLE		83.65%
1 Compute	74.05%	
2 Checkout	5.50%	
3 Assembly	3.40%	
4 Data Error	--	
5 Computer Error	.70%	
6 Idle	--	
B. NOT AVAILABLE		16.35%
1 Maintenance, scheduled	6.73%	
2 Maintenance, unscheduled	7.11%	
3 Air conditioning and power failure	2.51%	

ACCUMULATIVE (Nov. 7, 1956 - June 30, 1957)
(One machine - one plus shift - 1755.78 hours)

A. AVAILABLE		86.96%
1 Compute	75.97%	
2 Checkout	5.25%	
3 Assembly	4.30%	
4 Data Error	.32%	
5 Computer Error	1.12%	
6 Idle	--	
B. NOT AVAILABLE		13.04%
1 Maintenance, scheduled	8.12%	
2 Maintenance, unscheduled	4.21%	
3 Air condi tioning and power failure	.71%	

MEMORANDUM

TO: EVERYBODY

No. 2 of January 9, 1957

FROM: JIM SNYDER

For reasons to be explained below, the methods suggested in a recent memorandum for correlating runs with the submitters of those runs have been found to be unsatisfactory. The memorandum in hand is to replace the recent memorandum on this subject which should be carefully ferreted out and destroyed.

It is the case that FORMESH and all members of the FORMESH family will accept only a 15 bit identification number. This allows only 32767 independently numbered runs to be carried out. If human identification is attempted, two digits of this number must be reserved for that purpose, thereby allowing each such person only 1000 runs. Since programs of this family are subjected to heavy usage, this situation is completely unsatisfactory. Therefore a method has been worked out to alleviate this difficulty. To be complete (and we hope definitive and final), the following suggestion is put forth:

Each person will be assigned a number A to be referred to as the HUMAN ID. Since our programs will be divided into three categories and since each category demands a slightly different treatment of A, several people may have their A different for different categories. A table of suggested HUMAN ID's is:

	HUMAN ID = A	
User	Categories I & II	Category III
J. N. Snyder	00	00
D. W. Kerst	11	11
R. F. King	12	12
J. L. Powell	13	13
L. D. Fosdick	14	14
L. J. Laslett	15	15
A. M. Sessler	16	16
F. Cole	17	17
M. Storm	18	18
E. Zographos	19	19
G. Parzen	21	21
H. Meier	22	22
J. McNall	23	23
C. Joyce	24	24
K. Symon	25	25
J. Anderson	26	26
J. Enoch	27	27
M. Freiser	28	28
R. Haxby	29	29
L. Jones	31	31

F. Mills	32	01
T. Ohkawa	33	02
F. Peterson	34	03
C. Pruett	35	04
K. Terwilliger	36	05
J. Van Bladel	37	06

Programs fall into the following categories:

Category I

FOROCYL (13)
FORMERGE (50)
WELL-TEMPERED FIVE (46)
FORERUNNER (57)
DUCK CALL (33)
TTT (39)
ALGETEE (58)

and, hopefully, all future programs will accept a 35 bit identification number so that the HUMAN ID and the run ID can be computed in a single ID by using the formula:

$$ID = 10^8 A + i$$

where i is the run number. It should be noted that EQUICYL (54) has no additional run ID but is identified by the FOROCYL or FORMERGE problem to which EQUICYL is being applied.

Category II

These programs constitute members of the FORMESH family with the exception of FORMESH MUMBLEBUMPS. To be specific, the programs of Category II are:

FORMESH (26)
FORFIX POINT (55)
FORMESH INVARIANTS
ATEMESH (53)
ATEMESH INVARIANTS
SIXTEEN MESH (56)
SIXTEEN MESH INVARIANTS
FORMESH FUMBLEBUMPS (60)
FORMESH FUMBLEBUMPS INVARIANTS
FORMESH GRUMBLEBUMPS (62)
FORMESH GRUMBLEBUMPS INVARIANTS
FORFIX POINT WITH FUMBLEBUMPS (66)
FORFIX POINT WITH GRUMBLEBUMPS (67)

These programs which already fill the memory have no room to provide an integer print routine which is necessary for the printing of a 35 bit ID number. However, they do allow the printing of fractions so that the HUMAN ID number, A, can be printed as a fraction. An additional piece of data should be given on the agenda in the fraction group. This will be the address 7601 followed by the two digit HUMAN ID. This number will appear on printed output as an eleven digit decimal following the already present run identification number. Thus each person will be allowed up to 32767 runs on any one of these programs. However, the submitter of agenda should be prepared to find his HUMAN ID printed with up to an inaccuracy of 3×10^{-11}, which is standard with our fractional print routines. Thus the HUMAN ID number 12000000000 may appear as 11999999997.

It should be noted that EQUICYL, although not provided with a run ID, is provided with an additional HUMAN ID which will also conform to this rule concerning fractions.

Category III

Members of this category are:

FORANAL (52)
SCOFFLAW (30)
FORMESH MUMBLEBUMPS (61)
FORMESH MUMBLEBUMPS INVARIANTS

and possible future programs if exigencies of fate demand it.

These programs still require the 15 bit ID which is limited by 32767. Since it is improbable that a given person will make more than 1000 runs on each of these programs, the original proposal of assigning an ID:

$$ID = 10^3 A + i$$

will be followed. The A for this category is given in the above table. It should be very carefully noted that FORMESH MUMBLEBUMPS and FORMESH MUMBLEBUMPS INVARIANTS are members of Category III. An attempt to treat them by the fraction method described in Category II would completely foul up the program.

MEMORANDUM

TO: ONE AND ALL

No. 2 of February 11, 1957

FROM: JIM SNYDER

RESTRICTIONS ON MESH SIZE

With a = number of points in width of mesh
b = number of points in height of mesh (including median plane)

the following programs demand the following restrictions:

FORMESH

$3\,ab \leq 7417$

FORMESH WITH INVARIANTS

$3\,ab \leq 7324$

ATEMESH

$3\,ab \leq 7296$

ATEMESH WITH INVARIANTS

$3\,ab \leq 7225$

SIXTEEN MESH

$3\,ab \leq 7265$

SIXTEEN MESH WITH INVARIANTS

$3\,ab \leq 7194$

FORMESH WITH FUMBLEBUMPS

$3\,ab \leq 7187$

FORMESH WITH INVARIANTS AND FUMBLEBUMPS

$3\,ab \leq 7094$

FORMESH WITH MUMBLEBUMPS

$3\,ab \leq 3695$ (each mesh)

FORMESH WITH INVARIANTS AND MUMBLEBUMPS

$3\,ab \leq 3649$ (each mesh)

FORMESH WITH GRUMBLEBUMPS

$3\,ab \leq 7083$

FORMESH WITH INVARIANTS AND GRUMBLEBUMPS

$3\,ab \leq 6990$

FORMESH WITH FORFIXPOINT

$3\,ab \leq 7324$

FORMESH WITH FORFIXPOINT AND FUMBLEBUMPS

$3\,ab \leq 7094$

FORMESH WITH FORFIXPOINT AND GRUMBLEBUMPS

$3\,ab \leq 6990$

If a larger mesh be in hand it can be used with a given program since the program will overwrite the necessary number of extra positions. However, if this is to be done with impunity, the b_{PHONY} procedure should be executed so that b_{PHONY} satisfies the above restrictions.

FOROCYL (POTENTIAL PART)

$a(b - 1) + 2\,N_c \leq 7600$

(N_c = number of points at which current exists)

FORMERGE POTENTIALS

As above in FOROCYL

FORMERGE FIELDS

$3\,ab \leq 7601$

FOROCYL (FIELD PART - FOR PROPER FORMESH INPUT DECK)

$3\,ab \leq 7313$

FOROCYL (FOR PROPER PRINTING AND FOURIER ANALYSIS)

$3\,ab \leq 7201$ For the region processed

EQUICYL

$a(b - 1) + 2N_c \leq 7600$

(N_c = number of points at which current correction exists)

MEMORANDUM

TO: One and All March 20, 1957

FROM: J. Anderson

PROGRAMME RUNNING TIME

FOROCYL

Though estimates vary greatly with mesh size, convergence criteria, etc., about 20 points per minute may be considered an average processing rate. Printing and punching time runs from 2 to 20 minutes depending on mesh size and choice of potential and/or field output.

FORMERGE

Read time: $\frac{N_{\Omega}}{2750}$ minutes

Punch time: $\frac{N_{\Omega}}{2200}$ minutes, where N_{Ω} is the number of potential points.

In addition, one usually obtains fields and Fourier analysis, which means 2-20 minutes more.

EQUICYL

$\left(\frac{a}{150}\right)$ minutes

FORMESH
FORMESH WITH MUMBLEBUMPS
FORMESH WITH FUMBLEBUMPS

4.85 seconds per sector (32 RK steps)
Invariants add a negligible amount.

FORMESH WITH GRUMBLEBUMPS

6.7 seconds per sector if bump is computed once per Runge-Kutta step;
12.5 seconds per sector if 4 times per RK step.

FORFIXPOINT

4.85 seconds per sector.

ATEMESH

9.7 seconds per sector.

SIXTEENMESH

19.4 seconds per sector.

WELL-TEMPERED FIVE

$\{N_e [.0167\, i_{max} (m_{max} + 1) + 0.197] + .430\, (\text{no. lines of print})\}$ seconds,

where m_{max} is the largest m for which μ_{om} and ν_{om} are not zero.

TEMPERMESH

ab seconds, approximately.

DUCK ANSWER

Varies widely with choice of switches, etc., averaging about 12 seconds per 32 RK steps.

FORERUNNER

4.2 seconds per sector (32 RK steps)

ALGYTEE

About 5500 steps per minute. The 2 dimensional problem takes very nearly the same time as the 4 dimensional one.

FORANAL
SCOFFLAW
MESSY MESSY

These take a very short time, less than a minute for any one problem.

TTT

On account of the way this program has been used to date, we can only say that it operates the printer at 150 lines (steps) per minute. With printing suppressed it would naturally proceed faster.

MEMORANDUM

TO: EVERYBODY (and I hope we get through to everybody)

No. 2 of June 12, 1957

FROM: ME

Approximately one week ago International Business Machines Cathode Ray Display Units Models 740 and 780 were attached to our machine. This equipment rents for $2850 per month for a basic 8-hour shift, with a 40% charge for actual use on all time over and above the basic shift. In the hope of utilizing this equipment effectively, all important production routines have been modified and augmented so as to produce generalized (and, we hope, convenient and utilizable) cathode ray output. This necessitated the preparation of new programs which are:

113 SCOPE FORMESH (INVARIANTS)
114 SCOPE FORMESH (INVARIANTS) FUMBLEBUMPS
115 SCOPE FORMESH (INVARIANTS) GRUMBLEBUMPS
116 SCOPE INVARIANTS DUCK BUMPS
118 EQUICYL SCOPE
119 TTT SCOPE
120 CHANNEL FIVE
121 ALGYTEE SCOPE

Descriptionsof these programs and agendum sheets pertinent thereto are being prepared and are being distributed.

It will be noticed that this new equipment can be used either visually or photographically. It will be possible for consumers to request that their runs be made in their presence so that the visual scope may be scrutinized. Mr. Anderson should be consulted on these matters and appointments made. However, it is preferred that scope output be taken photographically. Tentatively we hope to mechanize this operation as follows. The camera will be emptied at the end of each day. This film will be developed as the first order of business on the subsequent morning. The output should then be ready for distribution by the same methods used to distribute printed output, before noon of the day following that on which the material was run.

At great expense a microreader suitable for viewing this film and suitable for transferring it to graph paper has been procured. At present this reader resides in the second room to the right on entry to the Computer Division. It should be noted that the film can also be enlarged and printed.

MEMORANDUM July 22, 1957

TO: Regular Viewers of Channel 704 TV

FROM: The Sponsors

RE: Commercials (what else?)

Preceding each set of scope runs one may expect to find a short commercial giving the date and other useful information, as follows:

I	00001	MONTH	DAY
I	00002	HUMAN ID	
I	00003	MESH NO.	
I	00004	BUMP NO.	

The numbers will be common integers, and if not entered, will be printed as zeroes. To employ this wonderful device, fill in the commercial sheet and make it the top sheet of all in a given set of agenda. Of course any information in the form of an integer may be entered, according to the whim of the user. The use of this programme is strongly advised, if only to get the date on the film.

COMMERCIAL AGENDUM

PROGRAMME NO. 131

To be stapled as the first page in a series of scope agenda.

PARAMETER	ADDRESS	VALUE
Month	1000	
Date	1001	
HID	1002	
Mesh number	1003	
Bump number	1004	
12R = COMMERCIAL	1005	0

MEMORANDUM

TO: One and All — March 20, 1957

FROM: J. Anderson — Revised: August 27, 1957

PROGRAMME RUNNING TIME

FOROCYL

Though estimates vary greatly with mesh size, convergence criteria, etc., about 20 points per minute may be considered an average processing rate. Printing and punching time runs from 2 to 20 minutes depending on mesh size and choice of potential and/or field output.

FORMERGE

Read time: $\frac{N_{\Omega}}{2750}$ minutes

Punch time: $\frac{N_{\Omega}}{2200}$ minutes, where N_{Ω} is the number of potential points.

In addition, one usually obtains fields and Fourier analysis, which means 2-20 minutes more.

EQUICYL

$\left(\frac{a}{150}\right)$ minutes

FORMESH
FORMESH WITH MUMBLEBUMPS
FORMESH WITH FUMBLEBUMPS
FORMESH (INVARIANTS) SCOPE
FORMESH (INVARIANTS) FUMBLEBUMPS SCOPE

4.85 seconds per sector (32 RK steps)
Invariants add a negligible amount.

FORMESH WITH GRUMBLEBUMPS
FORMESH (INVARIANTS) GRUMBLEBUMPS SCOPE
FORMESH WITH STUMBLEBUMPS

6.7 seconds per sector if bump is computed once per Runge-Kutta step;
12.5 seconds per sector if 4 times per RK step.

FORFIXPOINT

4.85 seconds per sector.

ATEMESH

9.7 seconds per sector.

Noted DAF
Noted DAD
Noted
Recd. AUG 29 1957
Action
FILE

SIXTEENMESH

19.4 seconds per sector.

WELL-TEMPERED FIVE
CHANNEL FIVE

$\{N_e[.0167\, i_{max}(m_{max}+1)+0.197]+.430\,(\text{no. lines of print})\}$ seconds,

where m_{max} is the largest m for which μ_{om} and ν_{om} are not zero.

TEMPERMESH

ab seconds, approximately.

DUCK ANSWER
INVARIANT DUCK BUMPS
INVARIANT DUCK BUMPS SCOPE

Varies widely with choice of switches, etc., averaging about 12 seconds per 32 RK steps.

FORERUNNER

4.2 seconds per sector (32 RD steps)

ALGYTEE
ALGYTEE SCOPE

About 5500 steps per minutes, the 2 dimensional problem takes very nearly the same time as the 4 dimensional one.

FORANAL
SCOFFLAW
MESSY MESSY
FORSUPS
COMMERCIAL

These take a very short time, less than a minute for any one problem.

TTT
TTT SCOPE

Depending on the machine described in the scenario, the program computes between 2000 and 10000 steps per minute.