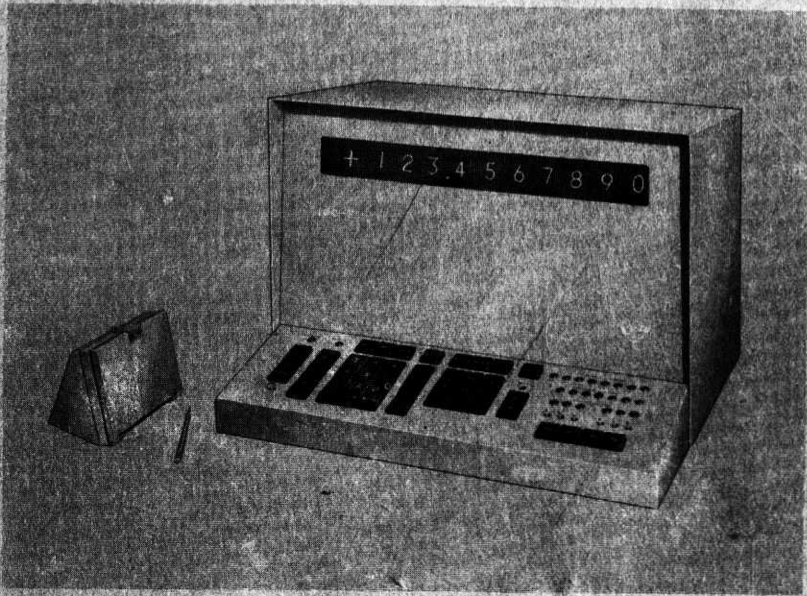
L38-66

LOCI-2 Reference Manual



WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876 TEL. 617 851-7311

Mr FAGE.

LOCI-2

REFERENCE MANUAL

WANG LABORATORIES, INC.
836 NORTH STREET
TEWKSBURY, MASSACHUSETTS

617 851-7311

L38-66

Wang Laboratories, Inc.

LOCI REFERENCE MANUAL

INTRODUCTION

The LOCI is a desk-top computer with many unique and flexible features. The power of the LOCI comes from its ability to automatically generate logarithms and exponentials (anti-logs). With simple keyboard operations, it is possible to do not only +, -, and ÷, but also $\log_e X$, e^x, $\sqrt{X}$, X^2, $1/\sqrt{X}$, and $1/X^2$.

To increase the power of the LOCI, these operations may be linked together by a program. Thus, those problems which require repeated iteration can be done very readily. The power of the program is considerably increased by the addition of a Decrement Counter and the Store and Restore branch commands.

This manual serves to collate together some of the basic operation characteristics of the LOCI. Chapter I deals with the individual machine commands. Chapter II introduces the methods of programming. We note that Chapter II can do no more than simply introduce the idea of programming. The possible combinations in a program are infinite. Chapter III contains some miscellaneous information as an appendix. The final section is a summary of machine functions.

The LOCI-2 is a desk-top computer consisting of the arithmetic processor, the storage registers, the keyboard, and the programming unit. We may visualize its logical organization as in Diagram 1.

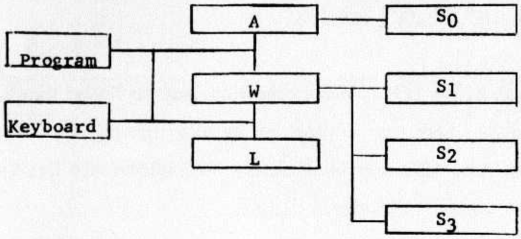


Diagram 1

There are three working registers which are part of the arithmetic unit. The registers are the work register, the accumulator, and the logarithmic register, abbreviated, respectively, as the W-, A-, and L- registers. All of the algebraic functions are carried out by manipulating the contents of these registers.

W-register.

The W-register is the only one displayed visually. There are ten decimal digits of storage in the W-register, plus a sign and a one-digit decimal point indicator. In the display, the decimal point lights up in the proper position between the integer part and the decimal part of the number in W.

We may consider the W-register as the point of input and output for the LOCI-2. Data is entered into W, one digit at a time, either by a key-stroke or by the equivalent program command. (Henceforth, when we use the word "keystroke", it is understood that we include the equivalent program command.) The digits are entered starting at the left-hand, or high-order, end of W. A decimal point may be entered at any time, but only once for a given number. If a sequence of digits is entered without pressing the decimal key, the LOCI-2 tacitly assumes that the decimal point follows the last digit.

Example 1. To enter the number 3.1416, simply key the sequence $\boxed{3}\ \boxed{.}\ \boxed{1}\ \boxed{4}\ \boxed{1}\ \boxed{6}$.

Example 2. To enter 64, we need the sequence $\boxed{6}\ \boxed{4}$.

Example 3. To enter -32.2, key the sequence $\boxed{3}\ \boxed{2}\ \boxed{.}\ \boxed{2}\ \boxed{\pm}$. (Note that the $\boxed{\pm}$ key simply changes the sign of W.)

Special Notes.

1. The range of the display is from $10^{-10}$to 10^{10}-1. (.00000 00001 to 99999 99999).

2. When a number is being entered, the next digit will be blacked out.

3. If the number in W is the result of a computation, i.e., it comes from A or L, the exponent may be beyond the range of the display. When the exponent is between 10^{-11} and 10^{11}, the number is within the capacity of W, although no decimal point will be shown. When the number is outside of this last range, the error light will be turned on.
4. The W-register is used in operations involving both A and L. Details may be found under those headings.

Commands involving W.

Clear W

Prime

All arithmetic commands

All input commands, including $\dot{\leftarrow}$, $\dot{\rightarrow}$, and $\leftarrow$

$W \rightarrow A$

$A \rightarrow W$

$W \rightarrow L$

$L \rightarrow W$

$W \rightarrow S_1, S_2, S_3$

$S_1, S_2, S_3 \rightarrow W$

$W \rightarrow DC$

$DC \rightarrow W$

$W \rightarrow PC$

PC, DC store with $W \rightarrow PC$

Test $W = 0$

Test W for - sign

A-register.

The LOCI accumulator is also called the A-register. It holds the results of additions and subtractions. Thus, when the add key is pressed, the content of W is added to A. The result of the addition is stored in A, while W retains its previous value. We abbreviate the description as follows,

$$A + W \rightarrow A$$

Likewise, pressing the subtract key results in the following,

$$A - W \rightarrow A$$

The capacity of the A-register is identical to that of W, ten decimal digits with the exponent range between 10^{-11} and 10^{11}. There is one important difference: in A, leading zeros are always suppressed. Thus, the number .00001 00000 would be stored as .10000 00000 with an exponent of -4. This clearly increases the precision of intermediate results in A. For example, A may contain $.10000\ 00001 \times 10^{-4}$ after some operation. After an A → W operation, however, .00001 00000 will be displayed in W, although $.10000\ 00001 \times 10^{-4}$ remains in A.

Example. To add 2 and 15.

$\boxed{2}\ \boxed{W \rightarrow A}\ \boxed{1}\ \boxed{5}\ \boxed{+}$

Example. To subtract 21 from 16.3.

$\boxed{1}\ \boxed{6}\ \boxed{.}\ \boxed{3}\ \boxed{W \rightarrow A}\ \boxed{2}\ \boxed{1}\ \boxed{-}$, or alternatively,

$\boxed{PRM}\ \boxed{1}\ \boxed{6}\ \boxed{.}\ \boxed{3}\ \boxed{+}\ \boxed{2}\ \boxed{1}\ \boxed{-}$

Special Notes.

1. If an operation should cause the result to overflow the range of A, the error light will be turned on.

Commands involving A.

+, -

$W \rightarrow A$

$A \rightarrow W$

$A \rightarrow S_0$

$S_0 \rightarrow A$

Test A for 0

Prime

L-register, products, roots, etc.

Much of the power of LOCI is derived from the operation of the L-register, and we shall consider it in some detail. This register is used to "accumulate" products, just as the A-register accumulated sums. More precisely, L accumulates the sum of logarithms. Thus, when the $\boxed{X}$ key is pressed, the natural log of the number in W is generated <u>and added to the contents</u> of L. We abbreviate this statement as follows:

$$X \sim L + \ln(W) \rightarrow L.$$

Other commands in the same class are described below:

$$\div \sim L - \ln(W) \rightarrow L,$$

$$\square \sim L + 2\ln(W) \rightarrow L,$$

$$1/\square \sim L - 2\ln(W) \rightarrow L,$$

$$\sqrt{} \sim L + \tfrac{1}{2}\ln(W) \rightarrow L,$$

$$1/\sqrt{} \sim L - \tfrac{1}{2}\ln(W) \rightarrow L.$$

After performing the above operations, W is automatically cleared to zeros.

Should we be curious to know the value of the number whose log is in L, we may press the LN^{-1} key. This causes the anti-log of L to be generated and stored in W, or, simply

$$\ln^{-1}(L) \rightarrow W.$$

L is cleared to zeros at the end of the operation. We assume that W → L and L → W are self-explanatory. Note that L → W also clears L to zeros.

Example. Assume that L contains zeros, corresponding to the log of 1, and that we wish to compute 2 x 3. After keystrokes $\boxed{2}$ $\boxed{x}$, ln(2) is in L. Then, after the keys $\boxed{3}$ $\boxed{x}$, ln(2) + ln(3) = ln(6) will be in L. To find the result, simply press $\boxed{LN^{-1}}$. This displays 6 in W, clearing L to zeros, simultaneously, for the next operation.

Example. To compute the expression $21 \times 1.1^2 \div \sqrt{9.1}$, the following sequence is needed:

$\boxed{2}$ $\boxed{1}$ $\boxed{x}$
$\boxed{1}$ $\boxed{.}$ $\boxed{1}$ $\boxed{\square}$
$\boxed{9}$ $\boxed{.}$ $\boxed{1}$ $\boxed{1/\sqrt{\ }}$
$\boxed{LN^{-1}}$

It is sometimes useful to consider L as containing the exponent of e. That is, L corresponds to the number e^L. The examples below illustrate this concept.

Example. To find the value of e.

$\boxed{1}$ $\boxed{W \rightarrow L}$	(L contains 1, corresponding to e^1.)
$\boxed{LN^{-1}}$	(W contains 2.71828 18284.)

Example. To find e^{-e}.

$\boxed{1}$ $\boxed{W \rightarrow L}$	
$\boxed{LN^{-1}}$	(W contains 2.71828 18284.)
$\boxed{\pm}$	(W contains -2.71828 18284.)
$\boxed{W \rightarrow L}$	(L contains -e.)
$\boxed{LN^{-1}}$	(W contains e^{-e} = .06598 8 0310.)

Example. To find the tenth root of 1024. ($\sqrt[10]{1024} = 2.$)

$\boxed{1}\ \boxed{0}\ \boxed{2}\ \boxed{4}\ \boxed{x}$	(L contains ln(1024).)
$\boxed{L \rightarrow W}$	(W contains ln(1024).)
$\boxed{x}$	(L contains lnln(1024).)
$\boxed{1}\ \boxed{0}\ \boxed{\div}$	(L contains lnln(1024) - ln(10).)
$\boxed{LN^{-1}}$	(W contains ln(1024)/10.)
$\boxed{W \rightarrow L}$	(L contains ln(1024)/10.)
$\boxed{LN^{-1}}$	(W contains result.)

Example. to find the 11th power of 2. ($2^{11} = 2048$)

$\boxed{2}\ \boxed{x}\ \boxed{L \rightarrow W}$	(W contains ln(2).)
$\boxed{x}$	(L contains lnln(2).)
$\boxed{1}\ \boxed{1}\ \boxed{x}$	(L contains ln(11) + lnln(2).)
$\boxed{LN^{-1}}$	(W contains 11 ln(2).)
$\boxed{W \rightarrow L}$	**(L contains 11 ln(2).)**
$\boxed{LN^{-1}}$	(W contains 2^{11}.)

Special Notes.

1. The L-register consists of 12 decimal digits of storage and two sign bits. Of the 12 digits, two are for the characteristic, and the remaining 10 are for the mantissa. One sign bit indicates whether the log is positive or negative, i.e., whether the number e^L is greater or less than 1 in magnitude (hence, the L Test command). The other sign bit indicates the sign to be attached to the number e^L when the anti-log is taken. Thus, we may consider the L-register as corresponding to the number

$$\pm e^{\pm cc.mmmmmmmmmm},$$

where c stands for a digit of the characteristic and m stands for a digit of the mantissa. With this organization, some interesting anomalies arise in the operation of the machine. These are noted below.

2. As the decimal point position is fixed, leading zeros are not suppressed in L. This is in contrast with the A-register. As a result, leading zeros may appear before the decimal point, when the L → W operation is performed. Thus, if we key the sequence $\boxed{3}\ \boxed{x}\ \boxed{L \rightarrow W}$, the number 01.098 61228 will appear in W, instead of 1.098 61228.
3. Conversely, when the W → L operation is performed, the number in W may be right shifted 1 or 2 digits for decimal alignment. For example, the sequence $\boxed{.}\ \boxed{1}\ \boxed{W \rightarrow L}$, will put 00.100 000000 in L. However, after the operation, 00.100 00000 will appear in W.
4. Note that a loss in precision may result from either the L → W or the W → L operation. In the former case, the 11th digit of L will be lost, as W has a 10-digit capacity. In the latter case, one or two low-order digits may be

lost to W, while one digit may be lost to L. Assume, for example, that W contains .10000 00011. After a W→L, W will retain 00.100 00000, while L will have 00.100 000001. Suppose W initially has 1.1000 00002. Then, after a W→L, 01.100 00000 remains in W. Meanwhile, L sustains no loss and contains 01.100 000002.

5. The operation of the L-register does not permit its value to exceed the range from -41.999 999999 to +41.999 999999. This corresponds approximately to the interval 10^{-15} to 10^{15}. If, in the course of a computation, this range is exceeded, the characteristic will be reset to ±40, ±41, or ±42 in a more or less random fashion. At the same time, the machine-cycles for the offending command will be halted, and a residue may be found in W. No error indication will be given, although any attempt to take an antilog at this time will result in an error signal. Loosely speaking, we may consider the neighborhood around ±40 as the limit of capacity. Multiplication by zero leaves L around -41, while division by zero leaves L around +41. Note that, after appropriate scaling to reduce the L value

to about ±23, corresponding to $10^{\pm 10}$, an anti-log operation can again be performed to display a proper number in W. (Cf. Note #3 on the W-register.) The anti-log of a value in the interval from -41 to -23 generates a zero in W.

6. If the number in W is outside the range of L, and a W→L command is given, the error light will be turned on.
7. In performing x, ÷, □, and 1/□, the LOCI abides by the usual sign conventions. For example, $(-2)x(-3) = +6$, and $(-2)x(-3)^2 = -18$.
8. When a negative number is in W and either the √, or the 1/√ key is pressed, the log of the absolute value is generated.
9. The process of generating logarithms and anti-logarithms introduces some round-off and truncation errors. In general, each operation may introduce an error in the tenth digit, and occasionally the error may propagate to the ninth digit.

Commands involving L.

x, ÷, □, 1/□, √, 1/√
LN^{-1}
W→L
L→W
Test L for - exponent
Prime

Program Counter (PC)

The Program Counter, or PC, is a two digit decimal counter which indicates the next program step to be performed (see page II - 2). It may be compared with a ruler that we may use to indicate our position when looking up entries in a finely printed table. When a program is being executed, the PC generally advances automatically by one count after selecting a program command. For example, if the PC were started at value 00, it will automatically run through the sequence 01, 02, 03, ...

Since there are 80 steps on a program card, the PC can assume the values from 00 - 79. To provide for a second card reader, there is an "X" bit associated with the PC. To indicate the first card reader, the X-bit is zero, while it is a one to indicate the second reader. Thus, to indicate step 10 in the first reader, the PC has a value of 10, with X equal to 0. To indicate step 10 in the second reader, the PC also has the value 10, but with X equal to 1.

Several methods of setting a value into the PC are available. We shall consider them in turn.

1. P_o, P_1, P_2, P_3. The operation, or keystroke, P_o presets the PC to 00 in the first reader and runs the program. Similarly, the operations P_1, P_2, and P_3 sets the PC to 03, 06, and 09 respectively. These keys offer four different ways of starting a program. (See Section II for more details.)

2. W → PC. This command causes the high-order two digits of W to be loaded into the PC. This is done regardless of decimal point position. For example, it W contains either +31000 0000.0 or +3.1000 00000, performing a W → PC sets PC to 31. If the sign of W is positive, execution of a W → PC will set the X bit to 0. Conversely if W is negative, the X bit will be set to 1.

3. W → XPC. This command unconditionally sets the sign of W to negative, and then performs a W → PC. Thus, W → XPC unconditionally selects the second reader.

4. Store & Jump. The execution of this command causes the following events to take place:

   a. PC → PCS,
   b. DC → DCS,
   c. W → PC.

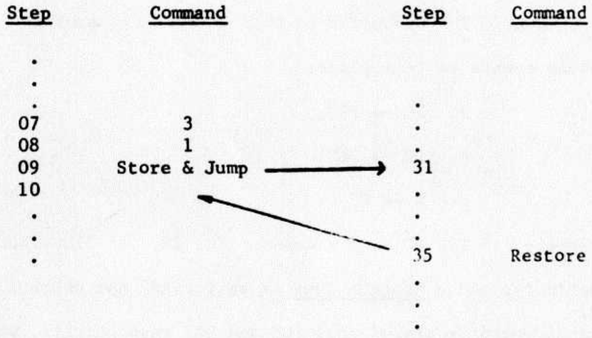
   For example, if PC, DC, and W contain 10, 20, and 31000 00000 respectively, and a Store & Jump is performed, the numbers 10 and 20 would be stored into PCS and DCS respectively, while the value 31 would go into the PC. Thus, the command stores the values in PC & DC, then changes the PC sequence. Both this command and the next one will be discussed more fully in Section II.

5. **Recall PC & DC.** The recall operation exchanges the contents of PC and PCS and also puts the DCS contents into DC. This is abbreviated as follows:

$$PCS \leftrightarrows PC,$$

$$DCS \rightarrow DC.$$

One effect of this is to restore a previously stored state into the PC and DC. If, in the example given above, a Recall PC & DC is given soon after the Store & Jump, the values 10 and 20 would be restored into the PC and DC respectively. The diagram below is illustrative:

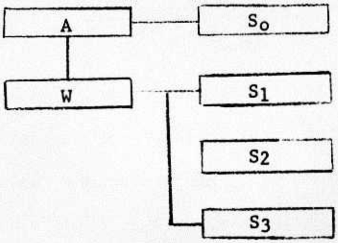


Again, illustrative examples can be found in Section II.

Storage

The LOCI is available with 2, 4, or 16 storage registers. Each storage register is capable of storing a variable in its entirety, consisting of 10 digits, plus one digit of exponent and a sign. Once a variable is transferred into a storage register, the variable may be read out at any time. It is not erased after being read. The only way to change its value is to transfer another variable into the same register. (We could also unplug the power cord, I suppose.)

Consider first the LOCI with 4 storage registers. These registers are christened as S_o, S_1, S_2, and S_3 respectively. Their association with the arithmetic registers is shown below:

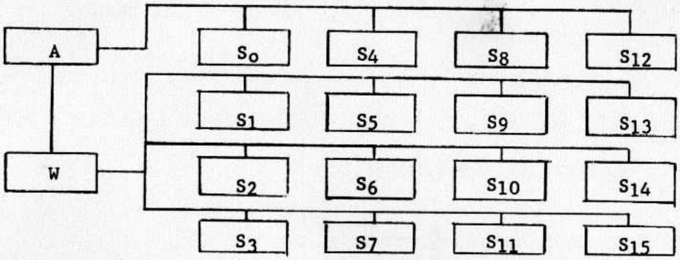


There are eight commands for transferring values in and out of these registers.

1. A-- S_o. The contents of A are transferred into W. A remains unchanged.
2. S_o-- A. The contents of S_o are transferred into A. S_o remains unchanged.
3. W-- S_1. The contents of W are transferred into S_1. W remains unchanged.
4. S_1-- W. The contents of S_1 are transferred into W. S_1 remains unchanged.

5. $W \rightarrow S_2$. Similar to 3.
6. $S_2 \rightarrow W$. Similar to 4.
7. $W \rightarrow S_3$. Similar to 3.
8. $S_3 \rightarrow W$. Similar to 4.

For LOCI-2a, which has 16 storage registers, the arrangement is as follows:



Thus, registers S_o, S_4, S_8, and S_{12} are associated with A, while the other 12 are associated with W. The registers are organized into four groups of four each. Thus S_o - S_3 is one group, and the others are S_4 - S_7, S_8 - S_{11}, and S_{12} - S_{16}. A Memory Selection Counter (MSC) which can assume the values from 0, 4, 8, or 12 designates which group is currently active. The eight keys described above operate upon the 4 registers in the active group. The following table is descriptive of the method.

	MSC value			
	0	4	8	12
Equivalent operations	$A \rightarrow S_o$ $S_o \rightarrow A$	$A \rightarrow S_4$ $S_4 \rightarrow A$	$A \rightarrow S_8$ $S_8 \rightarrow A$	$A \rightarrow S_{12}$ $S_{12} \rightarrow A$
	$W \rightarrow S_1$ $S_1 \rightarrow W$	$W \rightarrow S_5$ $S_5 \rightarrow W$	$W \rightarrow S_9$ $S_9 \rightarrow W$	$W \rightarrow S_{13}$ $S_{13} \rightarrow W$
	$W \rightarrow S_2$ $S_2 \rightarrow W$	$W \rightarrow S_6$ $S_6 \rightarrow W$	$W \rightarrow S_{10}$ $S_{10} \rightarrow W$	$W \rightarrow S_{14}$ $S_{14} \rightarrow W$
	$W \rightarrow S_3$ $S_3 \rightarrow W$	$W \rightarrow S_7$ $S_7 \rightarrow W$	$W \rightarrow S_{11}$ $S_{11} \rightarrow W$	$W \rightarrow S_{15}$ $S_{15} \rightarrow W$

The MSC is reset to 00 upon execution of a ~~CLA~~ PRIME or CLW. It may be stepped to the next higher value by a STEP MSC (Code 10) command. Thus if the MSC has a value of 8, and we push the STEP MSC key, its new value would be 12. The counter will ring around to 00 from the value of 12.

Decision Command

The decision commands are useful for testing programmed conditions. For flexibility of programming, the LOCI has six decision commands. In executing all these commands, except for TEST ERROR, the PC advances to the next step if the condition is not met while PC advances by 4 counts if the condition is met. In the case of the error condition, the PC advances by 4 counts if there is no error, and it advances by 1 count if there is an error registered. Note that the 3 count gap is exactly what we need to branch to some other point in the program (See Section II, Page 3)

1. Test DC = 0. If DC is zero, advance 4 steps. If not, advance 1 step.
2. Test A = 0. If A is zero, advance 4 steps. If not, advance 1 step.
3. Test W = 0. If W is zero, advance 4 steps. If not, advance 1 step.
4. Test W -. If W is negative, advance 4 steps. If it is positive, advance 1 step.
5. Test L -. If the exponent of L is negative, advance 4 steps, otherwise, advance 1 step.
6. Test Error. If the Error Light is <u>not</u> on, advance 4 steps. If the light is on, continue to next step.

Input/Output Commands

The LOCI may be equipped with certain standard input/output devices such as the Teletype Model 33ASR or a column printer. It is also possible to multiplex a number of input devices as well as a number of output devices. We shall consider first the three basic input/output commands. Then the Teletype and multiplex options will be examined in some detail.

1. WRITE (77). This command sends the contents of the W-register to the output channel. The exponent and sign are sent along with the ten-digit mantissa. On the Teletype unit, for example, a WRITE may cause the W-register contents to be printed as below:

   $$\pm\ .\ W_o\ W_1\ .\ .\ .\ .\ W_9\ b\ E\ W_E\ b,$$

   where W_o . . . . W_9 are the 10 digits of W, and W_E is the exponent.

2. CR (75). This command is normally used for returning the carriage and spacing an extra line on a typewriter, or simply for spacing an extra line on a column printer. For other types of devices, the CR command may be used for some other control function.

3. READ (76). The READ command is basically used to input data from external auxiliary equipment. When a READ command is encountered, the following sequence of events occurs:

   A. Switch control to external device, i.e., don't advance PC any longer.

B. Read one command from device.

C. Perform the code read in B as follows:

  i. If the code is RUN, switch control back to main program in card reader. Thus the card program resumes at the last PC setting, one count after the READ command that activated the tape.

  ii. If the code is P_o, P_1, P_2 or P_3, set appropriate value (00, 03, 06, or 09) into PC and switch control to the card reader.

  iii. For any other code (e.g., +, -, x, etc.), execute it and repeat step B.

It is clear that this is a very flexible arrangement. Thus the external device may be used as a method of data entry. It may also be used to control a long program. In that case, the loop portion of the program may be put in the card reader to save time.

Teletype Option C

A standard input/output option on the LOCI is the Teletype Model 33ASR. It has four functional features: output typewriter, input keyboard, paper-tape punch and paper-tape reader. The ASA 8-level code convention is followed (see APPENDIX B).

1. The typewriter is of 80 column width, and a WRITE (77) command from the LOCI will print 16 digits of the W-register as explained in the Input/Output section. Since a WRITE does not automatically begin a new line, two consecutive WRITEs will print the two numbers on the same line. Up to four numbers may be printed on a line. To begin a new line, we may use a CR (75) command.

   By using the WRITE and CR commands judiciously, the LOCI can be used to compute and print a table. A switch is available to allow insertion of table headings. The following example shows a tabulation of SIN (X) and COS (X), where X is given in degrees.

```
      X                 SIN(X)               COS(X)
+.0000000000 E0   +.0000000000 E0   +.1000000000 E1
+.1000000000 E2   +.1736481769 E0   +.9848077526 E0
+.2000000000 E2   +.3422001541 E0   +.9396926233 E0
+.3000000000 E2   +.4999999999 E0   +.8660254038 E0
+.4000000000 E2   +.6427876101 E0   +.7660444429 E0

+.5000000000 E2   +.7660444442 E0   +.6427876101 E0
+.6000000000 E2   +.8660254038 E0   +.4999999999 E0
+.7000000000 E2   +.9396926211 E0   +.3420201441 E0
+.8000000000 E2   +.9848077540 E0   +.1736481769 E0
+.9000000000 E2   +.1000000000 E1   +.0000000000 E0
```

2. Used as an input source, the Teletype keyboard becomes a second source of manual control, similar to the LOCI keyboard. Every Teletype key corresponds to a LOCI operation, (cf. APPENDIX B).

   Each LOCI operation is printed upon the associated Teletype key. Note that a record of the operations performed is printed when the Teletype keyboard is used for control.

3. The paper-tape punch may be used to record either a program or printed data or tape. An independent switch governs the punch. When the switch is on, a WRITE will punch the contents of W on tape as well as print it on the typewriter. In the off position, only the typewriter is operated.

4. The paper tape reader can be operated as an input device, either for a program or for data. We can understand this most readily by considering it as an extension of the keyboard or card reader. Each LOCI operation code is represented by a unique Teletype 8-level code. Appendix B gives the correspondence. For example, pushing the $\underline{\square}$ key on the LOCI is equivalent to reading the tape code

   1010 0110.

   Whenever a READ is encountered in the card program, control is switched to the paper-tape reader. Conversely, when the tape reader encounters a RUN, P_o, P_1, P_2 or P_3, control reverts back to the card program. (Note Input/Output section p. I - 20.) We may note at this point the special case when data is punched on paper tape to be read by a card program. In that case, each data point can be punched simply as a numeral with a decimal point. Following the last digit of each point, there should be a command to terminate the read. This must be one of the following RUN, P_o, P_1, P_2 or P_3. Exactly which one depends on the program, naturally. Consider the following sample program which simply has the loop:

A) read a data point;

B) add it to the accumulated sum;

C) print the sum;

D) repeat A).

Step No.	Command	
00	READ	
01	+	
02	A→ W	Card Program
03	CR	
04	WRITE	
05	P_o	

A typical set of four input points on tape may be as follows:

2	1011 0010	
.	1010 1110	
2	1011 0010	
3	1011 0011	
RUN	1011 1010	
.	1010 1110	
0	1011 0000	
4	1011 0100	
5	1011 0101	
6	1011 0110	Tape input
RUN	1011 1010	
5	1011 0101	
.	1010 1110	
2	1011 0010	
1	1011 0001	

RUN	1011 1010
6	1011 0110
.	1010 1110
1	1011 0001
RUN	1011 1010
STOP	1011 1111

The program may be started by first pushing the PRIME key, then pushing the P_o key. After reading each of the numbers 2.23, .0456, 5.21, and 6.1, the program will print the subtotal and read the next code. After printing the last sum, it will read the STOP code and halt.

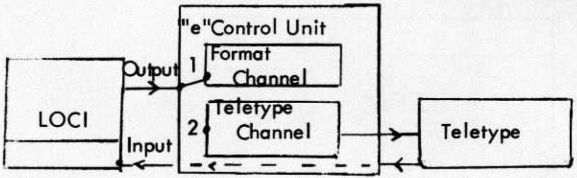
LOCI Option "h" ("e")

Teletype 33ASR

The option "e" Teletype is designed for a greater versatility than option "c", which is superceded by this new model. As in the original model, the Teletype has a typewriter, paper-tape reader and paper-tape punch. Important improvements are incorporated: the ability to format the typewriter output, the ability to read the punched output, and the ability to type alphabetic data on the typewriter.

Output-Typewriter

Physically, the Control Unit for the "e" version is external to the LOCI. Control of the Teletype printer (72 columns) from the LOCI is effected by the LOAD Output Multiplex command. As illustrated below, there are two output channels and an input channel.



To select a channel, we put a one-digit channel designating number, with a "— " sign, in the W-register and send it to the "e" Control Unit by a LOAD Output Multiplex Command (OMX35). This establishes the channel connection and all subsequent WRITE (11) commands will address the channel until a new channel is selected by the OMX command.

Channel Functions

Channel 1 is assigned to the Format Control Register. When Channel 1 is selected, and a WRITE is given, this register stores the first three high-order digits of W ($\overline{W_0 W_1 W_2}$ XX XXXXX) for controlling the format of subsequent typewriter output. The first digit W_0 specifies the number of digits preceding the decimal point, the second digit W_1 controls the number of digits after the decimal point, and the last digit W_2 indicates the number of words of data printed on a line before an automatic carriage return is given. An example given later will be illustrative. The same format holds for subsequent WRITEs until the register is reloaded.

The typewriter is on Channel 2. After an OMX (35) for Channel 2 has been given, a WRITE (11) will send the contents of W to the typewriter. The output format is as follows:

$$S\ \underbrace{X\,.\,.\,.\,X}_{W_o}\,.\,.\,.\,\underbrace{X\,.\,.\,.\,X}_{W_1}\quad \bar{S}\,\tilde{S}\,\Delta\ ;$$

Format Control

S — sign of number, + or – ;

$\bar{S}$ — blank space if number is positive; (S3) if number is negative, This is a non-printing code which will be converted to a ± code.

$\tilde{S}$ → A non-printing code (S6) which will be translated into a RUN code;

Δ — blank space.

Note that the number of digits before and after the decimal point are governed by the Format Control register.

Consider the following example. Assume that we wish to print a table of two columns. Each number is to have three places before the decimal point and four places after. The Format Register may be loaded as follows:

1	21	Select
±	33	Channel 1
OMX	35	
3	23	3 - 4 - 2 into
4	24	Format
2	22	Register
Write	35	

Suppose that storage registers S_1, S_2, S3, and S_0 contain 5, 98.73465128. 7.5 and -104 7500000 respectively. To print these out under control of the above format we may use the following instruction sequence:

2	21	Select
±	33	Channel 2
OMX	35	
S1→W	53	Write S1
WRITE	11	
S2→W	55	Write S2
WRITE	11	
S3→W	57	Write S3
WRITE	11	
S0→A	51	Write S0
A →W	45	
WRITE	11	

The two lines of output would appear as follows on the typewriter:

```
+ △ △ 5.0000△ △ + △ 98.7346 △ △
+ △ △ 7.5000△ △ -  104.7500 △ △
```

Note that the carriage return is automatic. If at this point, we should give a CR (75) command, we would naturally double-space after the last line. An explicit CR(75) command will start a new line, regardless of whether the current line is terminated or not. It also resets the count at the beginning of the line so that the first number after a CR always is printed at the beginning of a line.

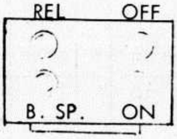
While it takes a few instructions to set-up a format, this may be done on a separate set-up card or a strip of tape. Thus, no program steps will be wasted in the body of the program itself. This set-up may be done manually from the Teletype keyboard by striking those keys corresponding to the LOCI commands when the keyboard is in control of the LOCI. In the example above, we may simply strike the following keys:

LOCI Command	Teletype Key
1	1
±	;
OMX	=
3	3
4	4
2	2
WRITE	)

There is also a Channel 0. When this channel is selected, a WRITE command simply causes 17 spaces to be generated on the typewriter and the paper tape. This offers an additional means of format control.

Output - Paper Tape

The paper tape punch is mechanically linked in parallel to the typewriter. It can be engaged by pushing down the key marked "on", and disengaged by pushing down the key marked "off". The keys are located directly above the punch unit.



Because of the way in which the punch is linked to the typewriter, everything appearing on the typewriter will be punched when the punch is "on". The code used by the Model 33ASR is the 8-level ASA CII code.

Pushing the "REL" key will allow the paper to move freely. Each time the "B. SP" key is pushed, the paper moves back by one line.

Input - Paper Tape

The model "e" paper-tape reader is activated by a READ (76) command in the LOCI. Once a READ has been given, control of the LOCI is transferred to the tape reader and remains there until a RUN, Po, P_1, P_2, or P_3 code is encountered. Then the control reverts back to the card readers just as if the corresponding key is pushed manually.

Once the tape reader is in control, it may operate in one of the following three modes:

Mode 0, for program input. This means that all codes punched on the tape are sent to LOCI. The typewriter is inactive unless a WRITE is given.

Mode 1, for data input. This means that all codes are blocked except the digits 0 - 9, " . ", "; ", and ": ". Note that these codes correspond to the numbers, decimal point, the "+" command, and the RUN command. Again the typewriter is inactive until a WRITE is given.

Mode 2, for typing headings. In this mode, no codes are sent to the LOCI, and the typewriter is activated. Thus, all of the lines punched on the tape are printed, including alphabetic information.

Once the reader is in a particular mode, it remains in that mode until it receives a control code to change into a new mode. In Teletype code, these codes are given by

		Tape Channel								Teletype
Mode	Code	8	7	6	5	4	3	2	1	Keys
0	S0	1	0	0	1	1	0	0	0	(Ctr.) (X)
1	S1	1	0	0	1	1	0	0	1	(Ctr.) (Y)
2	S2	1	0	0	1	1	0	1	0	(Ctr.) (Z)

The (Ctr.) key is similar to the (Shift) key, and assigns a new code to the keys X, Y and Z. The control code can be given either from the tape itself or from the Teletype keyboard.

Control Switches on CU - 3

The Teletype CU-3 control unit has several switches.

1. Power ON-OFF - This switch turns the power either on or off. It controls the power on all of the outlets on the CU-3.

2. Tape - Keyboard - This switch is used to select whether the tape reader or the keyboard is used to control the LOCI. Normally in the running of a program, the tape reader is on. However, sometimes it is desirable to control the system from the keyboard. This occurs when we want to manually change the reading mode of the LOCI. For example, we may be running a tape program in Mode 0. As mentioned earlier, the keyboard is inactive in this mode. It may be desirable, however, to re-punch part of the program because an error has been discovered. The only way to release the keyboard is to turn on the keyboard control switch.

3. Normal - Step - This is a tape reader control switch. When the switch is on "Normal", the tape reader will operate normally. That is, it will automatically sequence through the codes on the tape as described earlier.

   If the switch is on "Step", the tape reader will stop after sending one code to the LOCI. That is, we must press the "READ" key once for each line of code punched on the paper tape. This mode is used mostly to check-out tape programs.

Programming the LOCI.

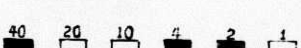
A simple LOCI program may be considered as an extension of some keyboard operations. That is, we merely think of the sequence of keys that must be pushed to perform a calculation, then write it down on a piece of paper. This constitutes a primitive program. For example, the sequence of keystrokes required to calculate $e^{-2.1}$ would be

2
.
1
±
W→L
LN^{-1}
Stop

In the same way, a program to compute e^{-x}, where X is any number in W, is simply

±
W→L
LN^{-1}
Stop

While we may call these sequences of symbols as programs, they are not in a form communicable to LOCI. By searching in the "dictionary", we may find the LOCI-2 code corresponding to each of the keystroke symbols. The code may then be punched upon a program card. To punch the code upon a program card, break the digits of the code into the sum of 40, 20, 10, 4, 2, and 1; then punch a hole at each marked position which is needed to make up the sum. For example, the code for W→L is 46, which is equal to 40 + 4 + 2. It is, therefore, punched as follows:



The LOCI-2 codes for the above programs are:

Step No.	Command	Code	Step No.	Command	Code
00	2	22	00	±	33
01	.	16	01	W→L	46
02	1	21	02	LN^{-1}	14
03	±	33	03	Stop	37
04	W→L	46			
05	LN^{-1}	14			
06	Stop	37			

When a program is punched upon a card, each command in the program occupies one numbered step. At any instant during the execution of the program, the Program Counter indicates the step to be next performed. For example, let us assume that at some point the PC value is 21; then the command punched at step 21 will be performed next. In the two illustrations above, the programs occupy steps 00-06 and 00-03. Thus, if the machine is in the AUTO mode, and if the P_0 key is activated, the PC will advance by one count after selecting each command. When the Stop code is selected, the PC will advance no further until it is activated again, manually, from the keyboard. This may be accomplished by pushing any one of the P buttons, or by merely pushing the Run key, which will start the program from the existing PC position.

Changing the Program Sequence.

While the PC generally advances one count at a time, it is possible to break this somewhat monotonous sequence and branch abruptly to some non-contiguous step. The command W→PC accomplishes this by shifting the first two left-hand digits of W, regardless of decimal point position, into PC. For example, suppose that W contains the number 31000 00004, and that PC just selected step number 22, where we find the code for W→PC. At the completion of this operation, we would find 31 deposited in PC, instead of 23. The next command, of course, would be taken from step 31. A command such as W→PC gives the flexibility of branching from anywhere in the program to anywhere else.

Making Decisions.

The LOCI-2 has six commands which are used to make decisions. These are the test commands which are used to check for the existence of certain conditions. In general, if the condition exists when the command is executed, the PC value will be advanced by three additional counts. Otherwise, it will be advanced by one, as usual. Consider the operation to test DC for 0. Suppose that the command were found at step 72 (i.e., PC was at 72). Then, if DC contains all zeros, PC is increased by three extra counts to 76. Otherwise, if DC were not zero, PC is increased by the normal one count to 73. The diagram below is illustrative.

Step No.	Command	Code	
72	DCT	70	
73	5	25	
74	8	30	
75	W → PC	40	→ DC ≠ 0
76	S_1 → W	53	
.	.	.	DC = 0
.	.	.	
.	.	.	

Another decision command deserving special attention is the test for negative exponent in the L-register. This offers a way of checking the magnitude of numbers, as the exponent will be negative if the number corresponding to the L-register value is less than one in magnitude. As an illustration, assume that L has some unknown value, and we wish to establish if it corresponds to the natural log of a number greater than 10^{-4}. The following sequence of steps may be used:

Step No.	Command	Code	
00	1	21	
01	0	20	
02	0	20	
03	↓□	06	
04	Test L	74	
05	3	23	
06	5	25	
07	W → PC	40	→ $\lvert e^{\overline{L}} \rvert > 10^{-4}$
08	.	.	$\lvert e^{\overline{L}} \rvert < 10^{-4}$
.	.	.	
.	.	.	

NOTE: $\overline{L}$ denotes value of L before multiplication by $(100)^2$

January 65

Programming a Loop.

A program loop is generally understood as a section of program, or a sequence of commands, which may be repeatedly performed. That is, the PC runs through the sequence a successive number of iterations. A simple example of a loop is as follows:

Step No.	Command	Code
10	1	21
11	0	20
12	W→PC	40

Thus, steps 10 and 11 put "10" in W, and step 12 puts this in PC, which brings us back to step 10. An even "tighter" loop is given below:

Step No.	Command	Code
10	1	21
11	2	22
12	W→PC	40

In this example, steps 10 and 11 put "12" in W, and step 12 puts this in PC. Thus, the program is stalled forever at step 12! Non-terminating loops such as these are generally neither intentional nor desirable.

A constructive program loop usually has some provision for making a decision so as to break away from the circle. Consider the following example which computes (11!):

Step No.	Command	Comment
00	Prime	
01	1	Set 11 in DC
02	1	
03	W → DC	
04	DC → W	
05	x	11 x 10 x 9 x ... x 3 x 2 x 1
06	Decrement	Subtract 1 from DC, and
07	DC Test	test if it is zero.
08	0	
09	4	DC not zero, go to step
10	W → PC	04 for another iteration.
11	LN^{-1}	DC is zero, display result
12	Stop	and stop.

The next example checks for the magnitude of a number N, whose log is in L. Assume that we know $|N| > 1$.

Step No.	Command
24	Clear A
25	1
26	0
27	÷
28	1
29	+
30	L test
31	2
32	5
33	W → PC
34	A → W
35	Stop

The idea behind the program is quite simple: divide the contents of L repeatedly, until the result corresponds to a number less than 1 in magnitude. We count the number of divisions by adding 1 to A every time a division is performed. When the characteristic of L becomes negative, then A contains the correct count of magnitude. At this point, the PC will be set to step 34, after performing the "L test" command.

An Example of a Loop
(The SIN (X) Program)

We can calculate the function SIN (X) by using a truncated Taylor series as follows:

$$\text{SIN (X)} = X - \frac{X^3}{3!} + \frac{X^5}{5!} - \ldots + \frac{X^{13}}{13!} \qquad (1)$$

This can be re-written as follows:

$$\text{SIN (X)} = X \left(1 - \frac{X^2}{2 \times 3} \left(1 - \frac{X^2}{4 \times 5} \left(\ldots \frac{X^2}{10 \times 11} \left(1 - \frac{X^2}{12 \times 13}\right)\right)\right)\right) \qquad (2)$$

The value X, in radians, can be put into a storage register. If we refer to the program S100, we find that the loop for the program essentially begins at step #52. The general idea behind the program is quite simple. It computes the quantity

$$X^2 \left(1 - \frac{X^2}{N \times (N+1)}\right) \qquad (3)$$

using the DC to index the value of N. The DC is also used to tell us when to stop.

52	1	21	
53	3	23	
54	W→DC	42	
55	6	26	
56	9	31	
57	W→PC	40	
58	S→W	53	X^2 (....)
59	□	06	
60	DC→W	43	$\left.\begin{matrix} \\ \end{matrix}\right\} \frac{X^2}{2n \times (2n+1)}$ (...)
61	÷	17	
62	Decre.	66	
63	1	21	
64	W→A	44	$\frac{(1 - X^2}{2n \times (2n+1)}$ (..))
65	LN^{-1}	14	
66	—	15	

67	A→W	45	
68	X	12	
69	DC→W	43	$\frac{1}{2n + 1}$ (...)
70	÷	17	
71	Decre.	66	
72	DC test	70	
73	5	25	
74	8	30	
75	W→PC	40	
76	S1→W	53	
77	X	12	
78	LN^{-1}	14	
79	Stop	37	

We begin with the innermost term, where N = 13 which is loaded into W at step # 52. Each time we go through one iteration in the loop (steps # 58 - 75), the value in DC is decreased by 1 twice, once each for N + 1 and N in the expression (3). We begin the first iteration at step # 69 with $\frac{1}{13}$. The program then loop to step # 58 to compute:

$\frac{X^2}{13}$ step # 58 - 59

$\frac{X^2}{12 \times 13}$ step # 60 - 63

$1 - \frac{X^2}{12 \times 13}$ step # 64 - 67

$\frac{1}{11}(1 - \frac{X^2}{12 \times 13})$ step # 68 - 71

At step # 72, the DC is tested for zeroes. This will be the case on the last iteration, at which point the expression

$$(1 - \frac{X^2}{2 \times 3} (1 - \ldots \frac{X^2}{10 \times 11} (1 - \frac{X^2}{12 \times 13}))) \quad (4)$$

will be in L. On the final iteration, the DC test at step # 72 will cause the PC to go to step # 76, where the argument X is multiplied to the expression (4) above for the final result.

The Concept of a Subroutine

We have so far discussed several of the significant concepts of programming:

1. Branching (i.e. W→PC, W→XPC, Store, Restore)
2. Decision making (i.e. DC = 0, L = -, etc.)
3. Looping

Two program examples utilizing the above SIN-COS and ARCTAN have been given to illustrate these ideas.

A fourth important idea is the subroutine. We have seen that a branch command can take us abruptly away from a linear program sequence as illustrated below:

Step	Command	Step	Command
.	.	.	.
.	.	.	.
.	.	.	.
07	2	24	S1→W
08	4	.	.
09	W→PC	.	.

The Store (64) and Restore (65) commands discussed on page I-14 and I-15 offer a somewhat different version of branching. They become useful when we have a part of a program (or a sub-formula) which appears several times.

Suppose that we wish to calculate the expression $(q^3 + r^2)^{1/3} + (s^3 + t^2)^{1/3}$. Assume that the numbers are stored as follows: q in S1, r in S2, s in S5, t in S6. An efficient method of programming this is the sequence below.

Step	Command	Step	Command
00	1	14	S1→W
01	4	15	□
02	Store	16	S1→W
03	W→S3	17	X
04	MS	18	Ln^{-1}
05	1	19	W→A
06	4	20	S2→W
07	Store	21	□
08	W→A	22	Ln^{-1}
09	CLW	23	+
10	S3→W	24	A→W
11	+	25	X
12	A→W	26	L→W
13	Stop	27	X
		28	3
		29	+

Step	Command	Step	Command
		30	Ln^{-1}
		31	W→L
		32	Ln^{-1}
		33	Restore

In this example, the formula $(x^3 + y^3)^{1/3}$ is calculated at step # 14 - 33. The parameters X and Y are assumed to be in the two registers accessible by the S1→W and S2→W keys. As a consequence, by changing the value of the MS Counter, we can address s and t, or r and q.

If we follow the program, we see that at step # 02, we branch to step # 14 with Store (64) command. Thus the number 14 is deposited in PC. At the same time, 03 is stored in PCS (See page I-14, I-15).

Once we are at step # 14, we proceed linearly until step # 33 where the Restore (65) command is encountered. This command simply exchanges the new PC value with the previously stored PCS value. We now have 03 in PC (and 34 in PCS). This returns us to the original program sequence.

Step # 03 simply stores the quantity $(q^3 + r^2)^{1/3}$ in S3. Step # 04 changes the MS Counter to 4 so that the S1→W and S2→ W keys address S5 and S6. Steps # 05-07 once more brings the PC to step # 14. Note that this time, PCS contains 08 so that the Restore at step # 34 will return us to step # 08.

This problem could have been done without the subroutine at steps # 14 - 33. However, the 20 step sequence for $(X^3 + Y^2)^{1/3}$ must be repeated once. This would increase the program length to 47 steps instead of 34. Clearly, if the expression occured more then twice, the economy would be more significant.

Let us consider next a small variation on the above problem to find:

$$\left[(q^3 + r^2)^{1/3} + (s^3 + t^2)^{1/3} \right]^{1/3}$$

A simple change will give us the program:

```
00    1          19    W→A
01    4          20    S2→W
02    Store      21    □
03    W→S3       22    Ln⁻¹
04    MS         23    +
05    1          24    A→W
06    4          25    X
07    Store      26    L→W
08    W→A        27    X
09    CLW        28    3
10    S3→W       29    ÷
11    +          30    Ln⁻¹
12    Restore    31    W→L
13    Stop       32    Ln⁻¹
14    S1→W       33    Restore
15    □          34    2
16    S1→W       35    4
17    X          36    W→PC
18    Ln⁻¹
```

This variation makes use of the exchange feature of the Restore (65) command. The quantity

$$(q^3 + r^2)^{1/3} + (S^3 + t^2)^{1/3}$$

is in A by the time we arrive at step # 12. At this time, the PCS contains 34 from the last time that the subroutine was used. Restore will put the 34 in PC and 13 in PCS. Note that at step # 34, we branch back to the cube root section of the subroutine. After finding the cube root of the quantity in A, the program returns to step # 13 and stops.

A Second Example of a Loop, and also a Subroutine

(The ARCTAN (X) Program)

Theory

An interesting second example of a program is the ARCTAN (X) program. The Taylor series for the function converges very poorly. However, a continued fraction expansion of the function offers excellent convergence for arguments less than 1. The continued fraction is known to be:

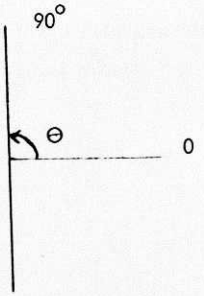
$$\Theta = \text{ARCTAN (X)} = \cfrac{X}{1 + \cfrac{X^2}{3 + \cfrac{(2X)^2}{5 + \cfrac{(3X)^2}{\vdots \; (2N-1) + \cfrac{(NX)^2}{2N+1}}}}} \qquad (1)$$

This is often abbreviated to:

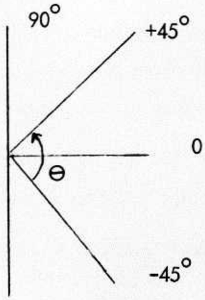
$$\Theta = \text{ARCTAN (X)} = \frac{X}{1+} \frac{X^2}{3+} \frac{(2X)^2}{5+} \cdots \frac{(nX)^2}{(2n+1)+} \cdots \qquad (1a)$$

If 1X1<1, and n = 10, then the function (1) converges to the true value of Θ with an error less than 10^{-6} degrees. This is the same thing as saying that Θ must be in the quandrant from -45° to $+45^{\circ}$ (fig. 1). In the general case where the known tangent Y is between 0 and + 00 (i.e., $0<\Theta<90^{\circ}$), a preliminary transformation can be performed on Y so that the transformed tangent is less than 1 in magnitude. The transformation corresponds to a rotation of 45°. It is given by,

$$X = \frac{Y - 1}{Y + 1} \qquad (2)$$



(Fig. 1)



(Fig. 2)

In short, to find

$$\Theta = \text{ARCTAN }(Y), \qquad 0<Y<\infty$$

is equivalent to finding

$$\Theta = \text{ARCTAN }(X) + 45^\circ$$
$$X = \frac{Y-1}{Y+1}, \qquad 0<Y<\infty$$

Program

The program S101 does the following:

1. Perform the transformation (2),
2. Set-up n=10 in DC,
3. Compute the expression (1),
4. Convert Θ from radians to degrees
5. Add 45° to adjust for the transformation. This also gives the answer.

Referring to the program S121 in Appendix A, we may start at step # 0, where the program puts the argument X in A, puts a high-order 4 in W, and branches to step 09. There, it goes to step 40 with a Store (or Jump 64) code.

From step 41-50, the transformation (X-1)/(X+1) is done. Steps 51-53 load 10 into DC. The succeeding sequence of calculations are:

$2n + Cn\ (10 + 10)$ in A $(Cn = 0$ initially$)$	Steps # 55 - 57,
$(2n)^2\quad(20^2)$ in L	Step # 58,
$2n + 1 + Cn\ (21)$ in A	Steps # 59 - 60,
$n^2 X^2\ (20^2 X^2)$ in L	Steps # 61 - 62
$n^2X^2/(2n+1)\,(20^2X^2/21)$ in L	Steps # 63 - 65
Cn = above	Step # 66

At step # 67, the DC is decreased by 1. Step # 68 test DC for 0. If DC ≠ 0, we loop back to step # 55 to repeat the above sequence. If DC = 0, we go to step # 72. At this point, we have the following expression in L:

$$\cfrac{\bar{X}^2}{3+\cfrac{(2\bar{X})^2}{5+\cfrac{\vdots}{21+\cfrac{(10\bar{X})}{21}}}}$$

Steps # 72-77 completes the expression for equation (1). Step # 78 returns (65) to the source of the last store (64) command. This brings us to step # 10.

At this point, the angle is given in radians. To convert into degrees, we branch to step # 20 from step # 11 to pick up the conversion constant. Dividing the constant $\pi/180$ into L at step # 12, we get the rotated angle. To compensate for the rotation, we add 45° at step # 15 - 18. The final results are now displayed.

Example of An Iterative Process - Newton's Method

(Third Example of Loop and also a Subroutine)

On many occasions, we must resort to a process of calculations which will terminate at some point, even though it is not possible to determine when. An example is the solution of a transcendental equation for which no direct, closed, algebraic solution exists. Illustrative examples are:

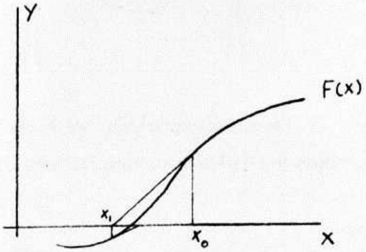
$$\mathrm{SIN}\ KX - X = 0 \cdot \qquad (1)$$

$$X\ \mathrm{Cosh} + K_1 X + K_2 = 0 \qquad (2)$$

A simple classical technique for solving transcendental equations is known as the Newton-Raphson method. We write the equation as

$$F\ (X) = 0 \qquad (3)$$

The function F(X) may be considered as a curve on the X-Y plan. Solution of (3) is equivalent to finding the X - intercept. The motivation for the method is as follows:



A) Pick an X - value, call it X_0

B) Find $Y_0 = F(X_0)$

C) If $Y_0 \neq 0$, follow the slope of the tangent to the curve at (X_0, Y_0) until the tangent intercepts the X - axis.

D) Use the X - intercept, X_1 as the new X - value and repeat B), C), and D) until an X which will satisfy the equation (3) is found.

By using some elementary geometry and the fact that the tangent of F(X) at any point X is given by $F^1(X)$, we easily determine that the new X - value can be found from the old X - value by the following iteration formula:

$$X_{i+1} = X_i - \frac{F(X_i)}{F^1(X_i)} \qquad (4)$$

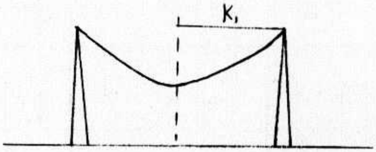
The method is subject to some regularity conditions. However, if our guess of X_0 does not lead to a solution, a solution can generally be found if we change the initial X_0 and start over. Note that we can only find real roots (as opposed to imaginary).

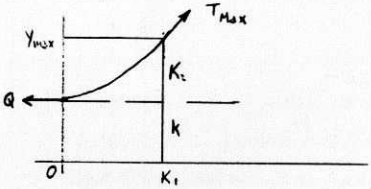
The Catenary Program

We now apply Newton's method to the solution of a Cable Span Problem.

Theory

In the construction of cables supported between two towers, the cable assumes the form of a catenary. One problem that we may encounter is that, given the semi-span (K_1) between towers, the cable sag (K_2), and the unit cable weight (W), we wish to find the proper cable length L and the maximum tension which occurs at the top of the towers.





From static analysis, we know that if the lowest point of the cable is k above the origin, then the following expressions hold:

$$y = k \cosh \frac{x}{k} \qquad (5)$$

$$s = k \sinh \frac{x}{k} \qquad (6)$$

and

where $k = \frac{Q}{W}$,

Q = tension at lowest point,

W = unit weight of cable (7)

s = cable length over semi-span

If we substitute the end point values K_1, and y_{max} into (5) we have

$$y_{max} = k \cosh \frac{K_1}{k}$$

or

$$k + K_2 = k \cosh \frac{K_1}{k} \qquad (8)$$

This reduces to

$$k\ (\cosh \frac{K_1}{k} - 1) - K_2 = 0 \qquad (9)$$

where K_1 and K_2 are the semi-span and sag respectively, and are known. Once we have solved the transcendental equation (9) for k, we can apply (6) to find the length of the cable for the semi-span.

To use Newton's method, we consider the left side of (9) as a function F(k), then

$$F^1\ (k) = (\cosh \frac{K_1}{k} - 1) - \frac{K_1}{k} \sinh \frac{K_1}{k} \qquad (10)$$

Newton's method says that if we make an initial guess for the value of k, call it k_0, then by applying the following iterative formula, we shall obtain a solution:

$$K_{i+1} = K_i - \frac{F(k_i)}{F^1\ (k_i)},\ i = 0, 1, 2, \ldots . \qquad (11)$$

The method is subject to certain regularity conditions.

A solution such as this is tedious to do by manual means. A LOCI-2 program (A102), applies this method of solution.

Once k has been found, then the cable length for the semi-span is given by

$$s = k \sinh \frac{K_1}{k} \qquad (12)$$

From the definition of k (7), we can find the lowest tension Q as follows:

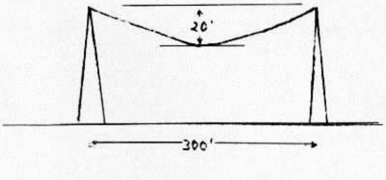
$$Q = Wk \qquad (13)$$

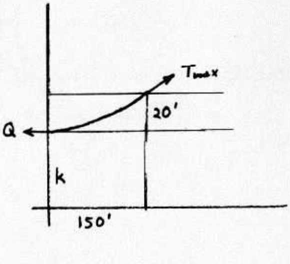
The maximum tension at the support is given by summing the horizontal force Q,

$$T_{max} = \sqrt{W^2k^2 + W^2s^2} = W\sqrt{K^2 + s^2} \qquad (14)$$

EXAMPLE

Suppose we have a cable weighing 1 lb./ft. It spans 300 feet and can have a sag of 20 feet. We wish to find the required cable length s and the maximum tension T_{max}.





We have therefore,

$K_2 = 20$ ft.

$W = 1$ lb./ft.

$K_1 = 150$ ft.

Equation (9) becomes:

$$k \left(\cosh \frac{150}{k} - 1\right) - 20 = 0$$

If we make the initial guess $k_o = 300$, then by using LOCI program A102, we obtain

$$k = 565.80\,024 \text{ ft.}$$

By applying (12), we have cable length

$$s = 565.8 \text{ SINH } \left(\frac{150}{565.8}\right) - 151.76327 \text{ ft.}$$

and the total length is 303.52 ft. The tension at the supports is given by (14),

$$T_{max} = 1 \sqrt{565.8^2 + 151.76^2}$$
$$= 585.80 \text{ lbs.}$$

Program Description

The program uses 4 storage registers assigned as follows:

S0 —— $F^1(x_i)$

S1 —— K_1/X

S2 —— K1

S3 —— K2

The algorithm is as follows:

1) Steps 00-04: Compute K1/Xi and store in S1.

2) Steps 05-07: Go to the subroutine at Steps 56-70 to find COSH $\frac{K1}{Xi}$ - 1.

3) Step 08 : Go to subroutine at Steps 71-79-69-70 to find $\frac{K1}{Xi}\,\frac{e}{2}$ K1/Xi and add to COSH $\frac{K1}{Xi}$ - 1.

4) Steps 09-10: Go to same subroutine and add $-\frac{K1}{Xi}\,\frac{e^{-K1/Xi}}{2}$ to results of 3) and 4). At this point, we have F^1 (Xi) in A and $\frac{Xi}{K1}$ in W.

LOCI Program

Date:

a^x, $\sqrt[x]{a}$, $\log_a x$, $\log_e a$,

Enter a, P_3 $(\log_e a)$ disp.

Enter x, P_o get a^x

P_1 get $\sqrt[x]{a}$

P_2 get $\log_a x$

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	W→ S_1	52		20	S_1→ W	53		40	S_1→ W	53		60			
	2	22			X	12			X	12					
	W→ PC	40			A→ W	45			L→ W	47					
	W→ S_1	52			X	12			X	12					
	3	23			LN^{-1}	14	$x\log_e a$ in W		A→ W	45					
05	W→ PC	40		25	W→ L	46		45	÷	17		65			
	W→ S_1	52			LN^{-1}	14			LN^{-1}	14					
	4	24			Stop	37	a^x in W		Stop	37					
	W→ PC	40													
	X	12													
10	L→ W	47		30	S_1→ W	53		50				70			
	W→ A	44			÷	17									
	Stop	37			A→ W	45									
					X	12									
					LN^{-1}	14									
15				35	W→ L	46		55				75			
					LN^{-1}	14									
					Stop	37	$a^{\frac{1}{x}}$ in W								

S102

LOCI Program Jν (X) **Date:** 21 Mar. '65

S103

$$J\nu(x) = \sum_{o}^{\infty} \frac{(-1)^n}{n!} \frac{1}{(\nu+n)!} \left(\frac{x}{2}\right)^{\nu+2n}$$

1. Enter order ν
2. P_3, when it stops
3. Enter x
4. RUN
   read result

$N = 21 + 2\nu$ iterations

$S_o = \nu$

$S_1 = \frac{x}{2}$

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00				20				40				60	4	24	
					X	12							3	23	
					2	22							W→ PC	40	
					÷	17	$x/2$	→	S_1→ W	53			S_o→ A	51	
					LN^{-1}	14			□	06	$\left(\frac{x}{2}\right)^2$		A→ W	45	
05				25	W→ S_1	52		45	S_o→ A	51		65	W→ DC	42	ν
					4	24			DC→ W	43			7	27	
					3	23			+	13	ν + n		4	24	
					Jump	64			÷	17	$\frac{1}{n}\left(\frac{x}{2}\right)^2$		W→ PC	40	
	W→ A	44							A→ W	45		→	S_1→ W	53	$\left(\frac{x}{2}\right)$
10				30				50	÷	17	$\frac{1}{n(\nu+n)}\left(\frac{x}{2}\right)^2$	70	X	12	
	A→ S_o	50			Stop	37			1	21			DC→ W	43	$\frac{1}{\nu!}$
	+	13	2						W→ A	44			÷	17	
	2	22							LN^{-1}	14			Decrement	66	
	1	21							—	15	$1-\frac{1}{n(\nu+n)}\left(\frac{x}{2}\right)^2(\cdots)$	→	DC=0	70	
15	+	13	21 + 2ν	35				55	A→ W	45		75	6	26	
	A→ W	45							X	12			9	31	
	W→ DC	42							Decrement	66			W→ PC	40	
			n=21+2ν								Blank		LN^{-1}	14	
	Stop	37							DC=0	70			Return	65	

LOCI Program **Hyperbolic Functions** Date: 28 Feb. '65

P_o - $SINH^{-1}$

P_1 - $COSH^{-1}$

P_2 - SINH

P_3 - COSH

$SINH(X) = 1/2\ (e^X - e^{-X})$

$COSH(X) = 1/2\ (e^X + e^{-X})$

$SINH^{-1}(X) = \ln / X + \sqrt{X^2 + 1} /$

$COSH^{-1}(X) = \ln / X + \sqrt{X^2 - 1} /$

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	$W \rightarrow S_1$	52		20	Stop	37		40	$S_1 \rightarrow W$	53		60			
	4	24							□	06					
	W→ PC	40							LN^{-1}	14					
	$W \rightarrow S_1$	52							W→ A	44					
	7	27							1	21					
05	W→ PC	40		25				45	+	13		65			
	$W \rightarrow S_1$	52							A→ W	45					
	3	23							√	04					
	W→ PC	40							$S_1 \rightarrow W$	53					
	W→ L	46							W→ A	44					
10	LN^{-1}	14		30	$S_1 \rightarrow W$	53		50	LN^{-1}	14		70	$S_1 \rightarrow W$	53	
	W→ A	44			W→ L	46			$W \rightarrow S_1$	52			□	06	
	÷	17			LN^{-1}	14			+	13			LN^{-1}	14	
	LN^{-1}	14			W→ A	44			A→ W	45			W→ A	44	
	+	13			÷	17			X	12			1	21	
15	A→ W	45		35	LN^{-1}	14		55	L→ W	47		75	—	15	
	X	12			—	15			Stop	37			4	24	
	2	22			1	21							6	26	
	÷	17			5	25							W→ PC	40	
	LN^{-1}	14			W→ PC	40									

S104

LOCI Program Variance σ^2

Date:

$$\sigma^2 = \frac{\sum(X_i - \bar{X})^2}{n-1} = \frac{1}{n-1}\left[\overbrace{\sum X_i^2}^{s_1} - \frac{1}{n}\overbrace{(\sum x_i)^2}^{s_o}\right]$$

S_o : $\sum X_i$

S_1 : $\sum X_i^2$

DC : 99 - N

1. P_3 to start
2. Enter X_i and RUN, continue for all X_i.
3. P_1 gives -N, then
4. RUN gives variance σ^2

   Standard deviation $\sigma = \sqrt{\sigma^2}$

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00				20	$A \rightarrow W$	45		40				60	Stop	37	
				21	$W \rightarrow S_1$	52						61	÷	17	
				22	$S_o \rightarrow A$	51						62	1	21	
03	5	25		23	Stop	37						63	+	13	
04	$W \rightarrow PC$	40	5	24	$S_o \rightarrow A$	51						64	$A \rightarrow S_o$	50	
05				25	+	13		45				65	LN^{-1}	14	
				26	$A \rightarrow S_o$	50						66	$W \rightarrow A$	44	
				27	□	06						67	$S_1 \rightarrow W$	53	
				28	DC Test	70						68	+	13	
09	Prime	36		29	1	21						69	$A \rightarrow W$	45	
10	$W \rightarrow DC$	42		30	5	25		50	$S_o \rightarrow A$	51		70	X	12	
11	$A \rightarrow S_o$	50		31	$W \rightarrow PC$	40		51	$A \rightarrow W$	45		71	$S_o \rightarrow A$	51	
12	2	22		32	Stop	37		52	□	06		72	$A \rightarrow W$	45	
13	DC	66						53	$DC \rightarrow W$	43		73	÷	17	
14	$W \rightarrow PC$	40						54	$W \rightarrow A$	44		74	LN^{-1}	14	
15	DC	66		35				55	CLW	2		75	±	33	
16	$S_1 \rightarrow W$	53						56	9	31		76	Stop	37	
17	$W \rightarrow A$	44						57	9	31					
18	LN^{-1}	14						58	—	15					
19	+	13						59	$A \rightarrow W$	45					

S105

LOCI Program Y = bX + C Linear Least Square Approx. (1st Card) Date: 8/2/65

S106

$Y = bX + C$ $b = \dfrac{\Sigma XY - 1/n\ (\Sigma X)(\Sigma Y)}{\Sigma X^2 - 1/n\ (\Sigma X)^2}$

$C = 1/n\,\Sigma Y - b/n\,\Sigma X$

S_o: ΣX^2 (note; lost when computing b)
S_1: ΣX
S_2: ΣXY
S_3: ΣY
DC: 99 - n

1. To initiate program, PRIME, P1, W—S1, W—S2, W—S3, A—S0.
2. Key in X_i, RUN
   Key in Y_i, RUN Repeat for all (X_i, Y_i).
3. For "b", push Po.
4. For "c", go to second card.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	4	24		20	+	13		40	A→ W	45		60	X	12	
	5	25			A→ S_o	50	ΣX_i^2		W→ S_2	54			S_3→ W	57	ΣY
	W→ PC	40			ER ?	67	X_i neg ?		P_2	62			X	12	
	0	20			±	33							7	27	
	W→ DC	42			CL ER	01							A→ S_o	50	
05	DEC	66		25				45	S_1→ W	53		65	Store	64	
	Stop	37			√	04			□	06	$(\Sigma X)^2$		—	15	
	W→ A	44	X_i		Stop	37			7	27			A→ W	45	
	W= -?	73			W→ A	44	y_i		Store	64	$1/n\ (\Sigma X)^2$		P_2	62	
	ER ?	67			X	12			—	15					
10				30	S_3→ W	57		50	A→ W	45		70	9	31	
					+	13			÷	17	$\boxed{\Sigma XY}$		9	31	
	CL ER	01			A→ W	45			S_2→ W	55	$\Sigma XY/$		W→ A	44	
	□	06			W→ S_3	56	Σy_i		X	12	$[\Sigma X^2 - 1/n(\Sigma X)^2]$		DC→ W	43	
	S_1→ W	53			S_2→ W	55			LN^{-1}	14			—	15	
15	+	13		35	W→ A	44		55	W→ A	44		75	A→ W	45	
	A→ W	45	ΣX_i		LN^{-1}	14			X	12			÷	17	
	W→ S_1	52			+	13	$\Sigma X_i Y_i$		S_2→ W	55			LN^{-1}	14	
	LN -1	14			DEC	66			÷	17			S_o→ A	51	
	S_o→ A	51			DC = 0?	70			S_1→ W	53			Restore	65	

LOCI Program Y = bX + C (2nd Card) Date: 8/2/65

Y = bX + C C = 1/n Σ Y - b/n Σ X

S_1: Σ X

S_2: Σ XY

S_3: Σ Y

1. Make sure b is in W.
2. Push $\underline{P_o}$.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	X	12	b.	20				40				60			
	S_1 → W	53													
	X	12	Σ X												
	7	27													
	Store	64	b/n Σ X												
05	W → S_1	52		25				45				65			
	S_3 → W	57	Y												
	X	12													
	7	27													
	Store	64	1/n Σ Y												
10	W → A	44		30				50				70	9	31	
	S_1 → W	53											9	31	
	−	15											W → A	44	
	A → W	45											DC → W	43	
	Stop	37											−	15	
15				35				55				75	A → W	45	
													÷	17	
													LN^{-1}	14	
													Restore	65	

S106A

LOCI Program Least Square Fit $Z = B + Ae^{\alpha t}$

Date: 11 Feb. '65

P_1: to start

RUN: to continue

P_o: to find answer

$z(t) = B+Ae^{\alpha t}$

$\ell n A+\alpha t = \ell n(z-\beta) = y$

Given: $\{i\Delta t\}, \{y_i\}$, $i=1, \ldots, N$
LSF for α

$$\alpha = \frac{\Sigma i\ Y_i - \frac{1}{2}(N+1)\Sigma Y_i}{\{\Sigma i^2 - \frac{1}{4} N[N+1]^2\}}$$

$$\tilde{\alpha} = \frac{\Sigma i\ Y_i - \frac{1}{2}(N+1)\Sigma Y_i}{\frac{1}{12} N(N+1)(N-1)} \Rightarrow \alpha = \frac{1}{\Delta t}\tilde{\alpha}$$

S_o: i

S_1: ΣY_i

S_2: $\Sigma\, iY_i$

S_3: background B

1) Key in background and start at P_1

2) Enter each point Z_i and push RUN

3) To find $\tilde{\alpha}$, push P_o

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	4	24		20	A→ W	45	Σy_i	40	S_o→ A	51		60	÷	17	N
	W→ PC	40			W→ S_1	52			1	21			1	21	
					S_o→ A	51			+	13			+	13	
	W→ S_3	56			1	21			A→ W	45			A→ W	45	
	Prm	36			+	13	i = i+1		X	12	N + 1		÷	17	N + 1
05	A→ S_o	50		25	A→ S_o	50		45	2	22		65	2	22	
	W→ S1	52			A→ W	45			÷	17			−	15	
	W→ S2	54			X	12	i Y i		S_i→ W	53			A→ W	45	
	A→ W	45			S_2→ W	55			X	12	$\frac{1}{2}(N+1)\Sigma Y_i$		÷	17	N - 1
	Stop	37			W→ A	44			S2→ W	55	$\Sigma i\ Y_i$		LN^{-1}	14	
10	W→ A	44	z_i	30	LN^{-1}	14		50	W→ A	44		70	Stop	37	
	S_3→ W	57	B		+	13			LN^{-1}	14					
	−	15			A→ W	45			−	15	$\Sigma_i Y_i-\frac{1}{2}(N+1)$ ΣiY_i				
	A→ W	45	z_i - B		W→ S_2	54	$\Sigma_i\ Y_i$		A→ W	45					
	X	12			S_2→ A	51			X	12					
15	S_1→ W	53	$\Sigma 4_i$	35	A→ W	45		55	1	21		75			
	W→ A	44			P_3	63			2	22					
	L→ W	47	$y_i=\ell n(Z_i-B)$						X	12					
	+	13	ΣY_i						S_o→ A	51					
	X	12	Y_i in L						A→ W	45					

S107

Wang Laboratories, Inc. 836 NORTH STREET TEWKSBURY, MASSACHUSETTS

L3-65
©

LOCI Program N! and Stirling's Approximation Date: 17 May '65

$\Gamma(x+1) \sim e^{(\ln x)(x+1/2)-x}(2\pi)^{1/2} = e^{(\ln x)(x+1/2)-x+1/2\ln 2\pi}$

$S_1 = x$ in Stirling's Approximation

Put x in W, then
P_o = $x!$ (x integer)
P_1 = $\ln\Gamma(x+1)$ then RUN gives $\log_{10}\Gamma(x+1)$
P_2 = gives 2π

S108

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	W→ DC	42		20	6	26		40	Stop	37		60	A→ W	45	
	3	23			•	16			S_1→ W	53			Stop	37	$\ln\{\Gamma(+1)\}$
	W→ PC	40			2	22			W→ A	44			1	21	
	W→ S_1	52			8	30			X	12			0	20	
	P_3	63			3	23			•	16			X	12	
05				25	1	21		45	5	25		65	L→ W	47	
	2	22			8	30			+	13	$x + 1/2$		÷	17	
	Store	64			5	25			L→ W	47			A→ W	45	
	Stop	37			2	22			X	12			X	12	
	4	24			Restore	65			A→ W	45			LN $^{-1}$	14	
10	1	21		30	DC→ W	43		50	X	12		70	Stop	37	$\log\{\Gamma(x+1)\}$
	W→ PC	40			W→ A	44			LN $^{-1}$	14	$(x-1/2)\ln x$				
					A→ W	45			W→ A	44					
					X	12			S_1→ W	53					
					Dec.	66			—	15	$(x-1/2)(\ln x)-x$				
15				35	DC= 0	70		55	2	22		75			
					3	23			Store	64					
					W→ PC	40			$\sqrt{}$	04					
									L→ W	47	$1/2\ln(2\pi)$				
					LN $^{-1}$	14			+	13					

Wang Laboratories, Inc. 836 NORTH STREET TEWKSBURY, MASSACHUSETTS

LOCI Program Erf(x), N(x) - Rational Approximation

Date: 19 May '65

$$\mathrm{Erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = 1 - (a_1 t + a_2 t^2 + a_3 t^3)\, e^{-x^2}$$

$$= 1 - t(a_1 + t(a_2 + a_3 t))\, e^{-x^2}$$

Error $\le 2.5 \times 10^{-5}$

1. Put x in W
2. Push P_o for N(x)
   Push P_2 for Erf(x)

Normal probability integral

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{\frac{-t^2}{2}} dt = \frac{1}{\sqrt{2}} \mathrm{Erf}(x)$$

$t = \dfrac{1}{1 + px}$

$p = .47047$

$a_3 = .74785\ 56$

$a_2 = -.09587\ 98$

$a_1 = .34802\ 42$

S_i: t

S_2: x

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	X	12		20	$\div$	17		40	9	31		60	1	21	
	2	22			LN^{-1}	14			8	30	a2		$W \to A$	44	
	$1/\sqrt{\ }$	05			$W \to S_1$	52	$t = \frac{1}{1 + px}$		Restore	65	$t(a_2 + a_3 t)$		LN^{-1}	14	
	LN^{-1}	14			·	16			·	16			—	15	
					7	27			3	23			$A \to W$	45	
05				25	4	24		45	4	24		65	Stop	37	
	$W \text{--} S_2$	54	x		7	27			8	30					
	X	12			8	30			0	20					
	·	16			5	25			2	22					
	4	24			5	25			4	24					
10	7	27		30	6	26		50	2	22		70	Restore	65	
	0	20			$W \to A$	44			Restore	65	$t(a_1 + \ldots.)$		$W \to A$	44	
	4	24			Store	64			$L \to W$	47			LN^{-1}	14	
	7	27			·	16			$W \to A$	44			+	13	
	X	12			0	20			$S_2 \to W$	55			$A \to W$	45	
15	1	21		35	9	31		55	□	06		75	X	12	
	$W \to A$	44			5	25			LN^{-1}	14			$S_1 \to W$	53	
	LN^{-1}	14			8	30			—	15	$\ln\{t(\cdots)\} - x^2$		X	12	
	+	13	$1 + px$		$\pm$	33			$A \to W$	45			7	27	
	$A \to W$	45			7	27			$W \to L$	46			$W \to PC$	40	

S109

LOCI Program

Date:

Resonant Frequency in C/S

$$f = \frac{10^3}{2\pi\sqrt{LC}} = 159.\text{xxx}\ \frac{1}{\sqrt{L}} \cdot \frac{1}{\sqrt{C}}$$

$$C = (159.\text{xxxx})^2 \cdot \frac{1}{L} \cdot \frac{1}{f^2}$$

1) Enter L in henries, $\underline{P_o}$

2) Enter C in μf, RUN
   Read frequency in CPS

or

1) Enter frequency in CPS, $\underline{P_2}$

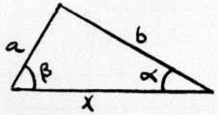
2) Enter L in henries or C in μf, <u>RUN</u>.

Read result for C in uf or L in henries.

S110

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	$W \to S_1$	52		20	X	12		40	2	22		60			
	Stop	37			$S_1 \to W$	53			X	12					
	$1/\sqrt{}$	05			1/□	07			$S_1 \to W$	53					
	3	23			LN^{-1}	14			$1/\sqrt{}$	05					
	$W \to PC$	40			Stop	37			LN^{-1}	14					
05	Stop	37		25				45	Stop	37		65			
	$W \to S_1$	52													
	Stop	37													
	÷	17													
	2	22													
10	5	25		30	1	21		50				70			
	3	23			5	25									
	3	23			9	31									
	0	20			.	16									
	.	16			1	21									
15	2	22		35	5	25		55				75			
	9	31			4	24									
	5	25			9	31									
	9	31			4	24									
	2	22			3	23									

LOCI Program Triangulation Date: Mar. 27, '65



$$a = \frac{\sin\alpha}{\sin(\alpha+\beta)}$$

$$b = \frac{\chi \sin\beta}{\sin(\alpha+\beta)}$$

S_0: β

S_1: work

S_2: α

S_3: χ, then sin(α+β)/χ

1. Prime
2 Key in χ , $\underline{P_0}$
3. Key in α in radians, W→ S_2
4. Key in β in radians, P_3
5. 1st stop, read "a", then RUN
6. 2nd stop, read "b"

$\underline{P_3}$ recomputes a & b, but must put χ in S_3.

$\underline{P_2}$ gives $\frac{\pi}{180}$

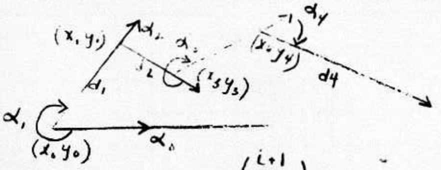
No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	W→ S_3	56	χ	20	S_3→ W	57		40	1	21		60	Dec	66	
	Stop	37			÷	17	sin(α+β)/χ		7	27			1	21	
	P_3	63			LN $^{-1}$	14	↓		4	24			W→ A	44	
	1	21			W→ S_3	56	S_8		5	25			LN $^{-1}$	14	
	3	23			S_2→ W	55	α		3	23			−	15	
05	W→ PC	40		25	W→ S_1	52		45	2	22		65	A-- W	45	
	3	23			5	25			9	31			X	12	
	8	30			Jump	64			2	22			DC→ W	43	
	W→ PC	40			S_8→ W	57			3	23			÷	17	
			α		÷	17	χ sinβ χ sinα/sin(α+β)		Stop	37			Dec	66	
10				30	LN $^{-1}$	14	↓	50	1	21		70	DC = 0	70	
	W→ A	44	β		Stop	37	a or b		3	23			5	25	
	A→ S_0	50	β		S_1→ A	51			W→ DC	42			6	26	
	S_0→ A	51			A→ W	45			6	26			W→ PC	40	
	S_2→ W	55			W→ S_1	52	β		9	27			S_1→ W	53	
15	+	13	α + β	35	2	22		55	W→ PC	40		75	X	12	
	A→ W	45			6	26			S_1→ W	53			Return	65	
	W→ S_1	52			W→ PC	40			□	06					
	5	25			.	16			DC→ W	43					
	Jump	64			0	20			÷	17					

S111

LOCI Program Traverse Problem

Date:

$S_o : \Sigma_i x_i$
$S_1 : \Sigma \alpha_i$
$S_2 : \Sigma_i y_i$
$S_3 : d_i$



$$y_{i+1} = y_i + d_{i+1} \text{ SIN} \left(\sum^{i+1} \alpha_j\right), \qquad X_{i+1} = X_i + d_{i+1} \text{ COS} \left(\sum^{i+1} \alpha_j\right)$$

NOTE: SIN-COS series uses Taylor series.
Last term of SIN is $\frac{X^{29}}{29!}$ COS is $\frac{X^{28}}{28!}$

Convergence to 8 places if $\Sigma\alpha \leq 2\pi$

1. Clear S_o, S_1, S_2, S_3 to zeroes.
2. X_o in S_o, y_o in S_2, α_o (radians) in S_1.
3. For each point, put d_i in S_3, put α_i (radius) in A, push P_1 to get y_i, then <u>RUN</u> to get X_i } REPEAT

NOTE: P_o gives 2π for conversion aid.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	1.	21		20	1	21		40	+	13	Σ y	60	$S_1 \rightarrow$ W	53	
	Jump	64			Jump	64			A→ W	45			◻	06	
	Stop	37			—	15			W→ S_2	54			Test Error	67	
	$S_1 \rightarrow$ W	53	Assume α_i is		÷	17			Stop	37			$S_1 \rightarrow$ W	53	
	+	13	in A		L= -?	74			Dec	66			X	12	SIN return
05	A→ W	45		25	A→ W	45	← $\alpha > 2\pi$	45	5	25		65	Return	65	
	W→ S_1	52	$\Sigma\alpha$		W→ S_1	52			6	26	COS		DC→ W	43	
	X	12			W→ S_1	52			Jump	64			÷	17	
	2	22			LN^{-1}	14	← $\alpha < 2\pi$		$S_3 \rightarrow$ W	57			Dec	66	
	W→ PC	40			2	22			X	12			1	21	
10	6	26		30	9	31		50	$S_o \rightarrow$ A	51		70	W→ A	44	
	•	16			W→ DC	42			LN^{-1}	14			LN^{-1}	14	
	2	22			5	25			+	13			—	15	
	8	30			6	26			A→ S_o	50	Σx		A→ W	45	
	3	23			Jump	64	SIN		A→ W	45			X	12	
15	1	21		35	$S_3 \rightarrow$ W	57		55	Stop	37		75	DC=0?	70	
	8	30			X	12	d SIN $\Sigma\alpha$		DC→ W	43			5	25	
	5	25			$S_2 \rightarrow$ W	55			÷	17			6	26	
	2	22			W→ A	44			Dec	66			W→ PC	40	
	Return	65			LN^{-1}	14			DC= 0?	70			Return	65	COS Return

S112

LOCI Program $ax^2 + bx + c = 0$ Quadratic **Date:**

S113

S0: $\frac{-b}{2a}$

S1: a

S2: b

S3: b

$$ax^2 + bx + c = 0$$

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

1) Enter a, P0
2) Enter b, RUN
3) Enter c, RUN
4) P_3 for 1st root
5) RUN for 2nd root

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	W → S1	52	Enter a	20	□	06	b^2	40	3	23		60	A → S$_0$	50	
1	Prime	36		21	LN^{-1}	14		41	0	20		61	+	13	
2	Stop	37		22	W → A	44		42	0	20		62	+	13	$(-b + \sqrt{\Delta})/2a$
3	W → S2	54	Enter b	23	S1 → W	53		43	0	20		63	Stop	37	
4	Stop	37		24	X	12	a	44				64	S$_0$ → A	51	
05				25	S3 → W	57		45	1/□	07	$\Delta < 9.10^{-6}$?	65	A → W	45	
6	W → S3	56	Enter c	26	X	12	c	46	L=-?	74	$\Delta < 9.10^{-6}$?.	66	Stop	37	
7	Stop	37		27	4	24		47	Prime	36	Yes, set	67			
8				28	X	12	4ac	48	□	06	$\Delta = 0$	68			
9	S2 → W	55		29	LN^{-1}	14		49	L=-?	74		69			
10	X	12	b	30	—	15	$\Delta = b^2 - 4ac$	50	LN^{-1}	14	$\Delta > 0$	70			
11	S$_1$ → W	53		31	A → W	45		51	A → W	45		71			
12	÷	17		32	W=-?	73		52	√	04		72			
13	2	22		33	÷	17	$\Delta > 0$	53	S$_1$ → W	53		73			
14	÷	17	b/2a	34	4	24		54	÷	17	$\sqrt{\Delta}/a$	74			
15	LN^{-1}	14		35	W → PC	40		55	2	22		75			
16	÷	33		36	Stop	37		56	÷	17		76			
17	W → A	44		37				57	LN^{-1}	14	$\sqrt{\Delta}/2a$	77			
18	A → S$_0$	50	$\frac{-b}{2a}$ in S$_0$	38				58	S$_0$ → A	51	$-b/2a$	78			
19	S$_2$ → W	55		39				59	—	15	$(-b - \sqrt{\ })/2a$	79			

Cubic Eqn.

LOCI Program $X^3 + a_2 X^2 + a_1 x + a_o = 0$ Real Root

Date: 29 May

$X = -\frac{a_2}{3} + S_1 + S_2$

$S_1 = \left\{r + (q^3+r^2)^{1/2}\right\}^{1/3}$

$S_2 = \left\{r - (q^3+r^2)^{1/2}\right\}^{1/3}$

$r = 1/6(a_1a_2 - 3a_0) - 1/27a_2^3$

$q = \frac{a_1}{3} - \frac{a_2^2}{9}$

S_0: $-r^r$

$S_1:a_1$ $-q$

$S_2:a_2$

$S_3:a_0$

NOTE: The following conditions must be met:
$q^3 + r^2 > 0$

1) Put a_1 in S_1
a_2 in S_2
a_0 in S_3

2) Push $\overline{P_0}$. When it stops, put 2nd card $\overline{\text{in}}$ reader

3) $\underline{P_0}$ again. This should give the answer.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	3	23		20	W→ S_1	52	a_1	40	S_3→ W	57	a_o	60	S_1→ W	53	
	4	24			3	23			−	15			X	12	
	W→ PC	40			X	12			−	15			3	23	
	W→ A	44	δ		A→ W	45			−	15	$a_1a_2 - 3a_o$		÷	17	
	X	12			÷	17			A→ W	45			LN^{-1}	14	
05	1	21		25	LN^{-1}	14	$\frac{3}{1+\delta}$	45	X	12		65	W→ A	44	
	+	13			$\underline{+}$	33			6	26			S_2→ W	55	
	A→ S_o	50	$1 + \delta$		W→ S_3	56	a_o		÷	17			□	06	
	4	24							LN^{-1}	14			3	23	
	X	12			4	24			W→ A	44	$\frac{1}{6}(a_1a_2-3a_o)$		1/□	07	
10	6	26		30	$\underline{+}$	33		50	S_2→ W	55		70	LN^{-1}	14	
	W→ A	44			W→ S_2	54	a_2		□	06	a_2^2		−	15	$\frac{1}{3}a_1-\frac{1}{9}a_2^2$
	LN^{-1}	14							S_2→ W	55			A→ W	45	
	+	13							X	12	a_2^3		W→ S_1	52	q
	A→ W	45	$6 + 4\delta$		S_1→ W	53			2	22			Stop	37	
15	X	12		35	X	12		55	7	27		75			
	S_o→ A	51			S_2→ W	55			÷	17					
	A→ W	45			X	12			LN^{-1}	14					
	÷	17	$\frac{6+4\delta}{1+\delta}$		LN^{-1}	14			−	15	$r=\frac{1}{6}(a_1a_2-3a_o)$				
	LN^{-1}	14			W→ A	44	a_1a_2		A→ S_o	50	$-1/27\ a_2^3$				

LOCI Program Cubic Equation 2nd card Date: 7/16/65

$S_1 = \left\{ r + (q^3 + r^2)^{\frac{1}{2}} \right\}^{1/3}$

$S_2 = \left\{ r - (q^3 - r^2)^{\frac{1}{2}} \right\} 1/3$

$\lambda = - a_2/3 + S_1 + S_2$

$S_3 : S_1$

Push P_o, when it stops, result is displayed in W.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	5	25		20	1	21		40				60	W→ S_3	56	S_1
	5	25			0	20			X	12			0	20	
	Store	64			0	20			L→ W	47			3	23	
	S_o→ A	51			0	20			X	12			Store	64	$\sqrt{r^2 + q^3}$
	A→ W	45			1/□	07			3	23					
05	□	06	r^2	25	L= -?	74		45	÷	17		65	—	15	$r - \sqrt{}$
	LN^{-1}	14			CLA	03	$\sqrt{} = 0$		LN^{-1}	14			Restore	65	$\left\{r - \sqrt{}\right\}^{1/3}$
	W→ A	44							W→ L	46			W→ A	44	
	S_1→ W	53							LN^{-1}	14			S_3→ W	57	
	□	06			LN^{-1}	14			Test ER	67			+	13	$S_1 + S_2$
10	S_1→ W	53		30	A→ W	45		50	±	33		70	S_2→ W	55	a_2
	X	12			$\sqrt{}$	04			CL ER	01			X	12	
	LN^{-1}	14			LN^{-1}	14	$\sqrt{q^3 + r^2}$						3	23	
	+	13	$r^2 + q^3$		S_o→ A	51			Restore	65			÷	17	
	A→ W	45			Restore	65							LN^{-1}	14	$a_2/3$
15	W= -?	73		35	A→ W	45	$r \pm \sqrt{q^3 + r^2}$	55				75	—	15	
	÷	17			±	33							A→ W	45	
	2	22			W= -?	73			Restore	65			Stop	37	
	W→ PC	40			CL ER	01			+	13	$r + \sqrt{} \}^{1/3}$				
	Stop	37							Restore	65					

S114A

Wang Laboratories, Inc. 836 NORTH STREET TEWKSBURY, MASSACHUSETTS

LOCI Program $Xe^x + K = 0$ (Newton's Method) Date: 2/4/65

$f(X) = Xe^x + K = 0$

$f^1(X) = e^x(1 + X)$

$$X_{i+1} = X_i - \frac{fi}{fi^1} = X_i - \frac{Xe^{x_i} + K}{e^x(1 + X_i)}$$

S_o: K

S_1: x_i

P_0: Initialize with K
Key in first guess, then
P_1 for automatic computation

P_3 to see result one iteration at a time. In this case, push RUN for additional iteration. Answer in S_1.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	W→ A	44		20	A→ W	45	$Xe^x + K$	40	S_1→ W	53		60	S_1→ W	53	
	A→ S_o	50			X	12	$e^{-x}(Xe^x+K)$		W→ L	46			±	33	
	Stop	37			1	21			X	12	xe^x		W→ L	46	e^{-x}
	W→ S_1	52	x_{i0}		W→ A	44			LN^{-1}	14			A→ W	45	f
	4	24			S_1→ W	53			S_o→ A	51			X	12	$e^{-x}f$
05	W→ PC	40		25	+	13	1 + X	45	+	13	$xe^x + K$	65	1	21	
					A→ W	45			A→ W	45			W→ A	44	
					÷	17			X	12			S_1→ W	53	
					S_1→ W	53	fi/fi^1		1	21			+	13	$1 + x$
	W→ S_1	52			W→ A	44			0	20			A→ W	45	
10	W→ L	46		30	LN^{-1}	14		50	0	20		70	÷	17	fi/fi^1
	X	12			—	15	$X - fi/fi^1$		0	20			S_1→ W	53	
	LN^{-1}	14	Xe^x		A→ W	45			□	06			W→ A	44	
	S_o→ A	51			P_3	63			L=-?	74			LN^{-1}	14	
	+	13							LN^{-1}	14			—	15	$x_i - fi/fi^1$
15	A→ W	45		35				55	6	26		75	A→ W	45	
	Stop	37	$Xe^x + K$						W→ PC	40			P_1	61	
	S_1→ W	53							S_1→ W	53					
	±	33							Stop	37					
	W→ L	46	e^{-x}												

LOCI Program Newton's method soln. for $U^2 - e^{KU} = 0$ Date: June 14, '65

$g(U) = U^2 - e^{KU}$

$g^1(U) = 2U - Ke^{KU}$

$U_{i+1} = U_i + \left(\frac{-g_i}{g_i^1}\right)$

S_o: K

S_1: $g(U_i)$

S_2:

S_3: U_i

1. Enter K or compute it and put it in S_o.
2. Enter a first guess for U_o. (Often a good guess is U_o=0)
3. Push P_o.
4. When machine stops, read resulting U.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	W→ A	44		20				40	A→ W	45		60	0	20	
	4	24							W→ S_3	56	U_i		◻	06	
	W→ PC	40							X	12			L = -?	74	$\lvert f_i \rvert < 10^{-6}$?
									S_o→ W	51	K		LN^{-1}	14	No
					Dec	66			A→ W	45			7	27	
05				25	A→ W	45	K	45	X	12	KU_i	65	W→ PC	40	
					X	12	Ke^{KU_i}		LN^{-1}	14			LN^{-1}	14	Yes
					S_3→ W	57			W→ L	46			S_3→ W	57	
					W→ A	44			LN^{-1}	14	e^{KU_i}		Stop	37	
					+	13	$2U_i$		W→ A	44					
10				30	LN^{-1}	14		50	S_3→ W	57		70	S_o→ A	51	K
					−	15	$2U_i - Ke^{KU_i}$		◻	06			A→ W	45	
					A→ W	45			LN^{-1}	14			X	12	
					÷	17			−	15	$e^{KU_i} - U^2$		S_3→ W	57	
					S_1→ W	53	$- g_i$		A→ W	45			X	12	KU_i
15				35	X	12	$-g_i/g_i$	55	W→ S_1	52	$-f_i$	75	LN^{-1}	14	
					S_3→ W	57	U_i		X	12			W→ L	46	
					W→ A	44			1	21			2	22	
					LN^{-1}	14			0	20			4	24	
					+	13	$U_i - g_i/g_i^1$		0	20			W→ PC	40	

Wang Laboratories, Inc.

7/8/65

S117

Direct Solution for Quartic

Eqn. $X^4 + a_3X^3 + a_2X^2 + a_1X + a_o = 0.$

I. Card #1 computes the following:

$$S = a_1a_3 + a_2^{\ 2} - 4a_o;$$

$$R = a_1a_2a_3 - a_oa_3^{\ 2} - a_1^{\ 2};$$

$$a = S - \frac{4}{3}\,a_2^{\ 2};$$

$$b = \frac{1}{27}\,(18a_2S - 16a_2^{\ 3}) - R.$$

II. Card #2 calculates

$$\Delta = b^2/4 + a^3/27,$$

and determines the following conditions:

1. When the program stops with a $\perp$ in W, $\Delta > 0$. In this case, use Card #3 next to compute

$$X_n = \left(\frac{-b}{2} + \sqrt{\frac{b^2}{4} + \frac{a^3}{27}}\right)^{\frac{1}{3}} + \left(\frac{-b}{2} - \sqrt{\frac{b^2}{4} + \frac{a^3}{27}}\right)^{\frac{1}{3}}$$

2. When the program stops with a 2 in W, Δ=0. Compute

   $X_1 = 2\sqrt[3]{\frac{-b}{2}}$ and $X_2 = X_3 = -\sqrt[3]{\frac{-b}{2}}$ on the keyboard.

   Note that the quantity $\frac{-b}{2}$ is in A when the program stopped.

3. When the program stops with a 3 in W, $\Delta < 0$. For this case, use Card #4 to evaluate ϕ (given in degrees) where ϕ is defined by

   $$\cos\ \phi = \frac{-b}{2} \div \frac{a^3}{27}.$$

   After getting ϕ, use the SIN-COS program to find

$$X_1 = 2\left[\sqrt{-a/3}\ \cos\left(\frac{\emptyset}{3}\right)\right]$$

$$X_2 = 2\left[\sqrt{-a/3}\ \cos\left(\frac{\emptyset}{3} + 120^\circ\right)\right]$$

$$X_3 = 2\left[\sqrt{-a/3}\ \cos\left(\frac{\emptyset}{3} + 240^\circ\right)\right]$$

III. Having found X_n appropriately, choose the algebraically smallest X_n such that

$$X_n + \frac{2}{3}a_2 \leq \frac{a_3{}^2}{4}\ .$$

Then $pq \equiv X_n + \frac{2}{3}a_2$, store this value (in S_1) as indicated on Card #5.

IV. After choosing X_n, use Card #5 to find

$$S = \frac{a_2 - pq}{2} + \sqrt{\left(\frac{a_2 - pq}{2}\right)^2 - a_o},$$

$$r = \frac{a_o}{S},$$

$$q = \frac{a1 - a_3S}{r - s},$$

$$p = a_3 - q.$$

V. Finally, use Card #6 to find the 4 roots.

$$\lambda_{1,2} = \frac{-p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - r}\ ,$$

$$\lambda_{3,4} = \frac{-q}{2} \pm \sqrt{\left(\frac{q}{2}\right)^2 - s}\ .$$

LOCI Program Quartic (#1) - Six-Card Program **Date:** 7/9/65

$X^4 + a_3X^3 + a_2X^2 + a_1X + a_o = 0$

$S = a_3a_1 + a_2^2 - 4a_o$

$R = a_3a_2a_1 - a_oa_3^2 - a_1^2$

$a = S - \frac{4}{3} a_2^2$

$b = \frac{1}{27} (18a_2S - 16a_2^3) - R$

1) Enter a_3, <u>P_o</u>;
2) Enter a_2, <u>RUN</u>;
3) Enter a_1, <u>RUN</u>;
4) Enter a_o, <u>RUN</u>;
5) Put in second card

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	$W \to S_3$	56	a3	20	$S_3 \to W$	57		40	$W \to S_1$	52	S in S_1	60	$W \to A$	44	
	X	12			☐	06	a_3^2		$S_2 \to W$	55			$S_2 \to W$	55	
	Stop	37			Stop	37			☐	06			☐	06	
	$W \to S_2$	54	a_2		$W \to S_3$	56	a_o in S_3		4	24			$S_2 \to W$	55	
	X	12	a_2a_3		X	12	$a_oa_3^2$		X	12			X	12	
05	Stop	37		25	LN^{-1}	14		45	3	23		65	4	24	
	$W \to S_1$	52	a_1		—	15	R		÷	17	$\frac{4}{3}$ a_2^2		☐	06	$16a_2^3$
	X	12	$a_1a_2a_3$		$A \to S_o$	50			LN^{-1}	14			LN^{-1}	14	
	LN^{-1}	14			$S_1 \to W$	53	a_1a_3		—	15			—	15	
	$W \to A$	44			$W \to A$	44			$A \to W$	45	a		$A \to W$	45	
10	$S_1 \to W$	53	a_1	30	$S_3 \to W$	57	a_o	50				70	X	12	
	☐	06			—	15			$W \to S_3$	56	a in S_3		2	22	
	LN^{-1}	14			—	15			$S_1 \to W$	53	S		7	27	
	—	15	$a_1a_2a_3 - a_1^2$		—	15			X	12			÷	17	
	$S_3 \to W$	57			—	15	$a_1a_3 - 4a_o$		$S_2 \to W$	55	a_2		$S_o \to A$	51	
15	X	12		35	$S_2 \to W$	55	a_2	55	X	12		75	LN^{-1}	14	
	$S_1 \to W$	53			☐	06	a_2^2		1	21			—	15	
	X	12	a_1a_3		LN^{-1}	14			8	30			$A \to W$	45	-b
	LN^{-1}	14			+	13	$a_1a_3+a_2^2-4a_o$		X	12			±	33	
	$W \to S_1$	52	a_1a_3 in S_1		$A \to W$	45			LN^{-1}	14	$10a_2S$		Stop	37	

S117a

LOCI Program Quartic (#2) Date: 7/9/65

$$\Delta = \frac{b^2}{4} + \frac{a^3}{27}$$

Case 1, $\Delta > 0$; use Card #3 to get X_n

Case 2, $\Delta = 0$; $\frac{-b}{2}$ is in A.

Case 3, $\Delta < 0$; use Card #4 to find Ø.

Routine to determine the 3 cases.

1) Push $\underline{P_o}$

2) When program stops, the case number is indicated by the number in W.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	A→ W	45		20	Test ER	67		40	÷	17		60			
	±	33	b						LN^{-1}	14	$\frac{-b}{2}$				
	W→ S_1	52	b in S_1		3	23	$\Delta<0$, Case 3		W→ A	44					
	□	06			Stop	37			2	22	Case 2				
	4	24			X	12			Stop	37					
05	÷	17		25	1	21		45	LN^{-1}	14	$\Delta>0$, Case 1	65			
	LN^{-1}	14			0	20			ER ?	67					
	W→ A	44	$\frac{b}{4}^2$		0	20			CL ER	01					
	S_3→ W	57			0	20									
	□	06			□	06									
10	S_3→ W	57		30	L= -?	74		50	1	21		70			
	X	12	a^3		4	24			Stop	37					
	2	22			5	25									
	7	27			W→ PC	40									
	÷	17	$\frac{a^3}{27}$		LN^{-1}	14	$\Delta = 0$, Case 2								
15	LN^{-1}	14		35	Prime	36		55				75			
	+	13			S_1→ W	53									
	A→ W	45	$\Delta = \frac{b^2}{4} + \frac{a^3}{27}$		±	33									
	W= -?	73			X	12									
					2	22									

S117b

LOCI Program Quartic (#3) Date: July 8, '65

Case 1

$b\ 2/4 + a^3/27 > 0.$

$Xn = \sqrt{-b/2 + \sqrt{b2/4 + a^3/27}} \quad +\sqrt[3]{-b/2 - \sqrt{b2/4 + a^3/27}}$

Push P_o

X_n appears when it stops

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	A→ W	45	Δ	20	S_o→ A	51		40	A→ W	45		60			
	$\sqrt{\ }$	04	$\sqrt{\Delta}$		+	13	Xn, Case a		CL ER	01	Turn on				
	LN^{-1}	14			A→ W	45			W= -?	73					
	W→ A	44			Stop	37			CL ER	01	Turn off if				
	W-- S_3	56									W positive				
05	7	27		25				45				65			
	3	23							X	12					
	Store	64							L→ W	47	ln A				
	4	24							X	12					
	Store	64	$\sqrt[3]{(\ )+(\ )}$						3	23					
10	W→ A	44		30				50	÷	17	$\frac{1}{3}$ ln A	70			
	A→ S_o	50							LN^{-1}	14					
	S_3→ W	57							W→ L	46					
	±	33							LN^{-1}	14	$\sqrt[3]{A}$		S_1→ W	53	
	W→ A	44							ER ?	67			X	12	
15	7	27		35				55	CL ER	01	ER is on,	75	2	22	
	3	23							±	33	root neg.		÷	17	
	Store	64											LN^{-1}	14	
	4	24							Restore	65			—	15	-b/2
	Store	64	$\sqrt[3]{(\ )-(\ )}$										Restore	65	

S117c

LOCI Program Quartic (#4) **Date:** 7/9/65

Case C $b\ ^2/4 + a\ ^3/27 \quad 0,$

$\cos \emptyset = b/2 \sqrt{-a\ ^3/27}$, and

$X_1 = 2\sqrt{-a\ ^3/27}\ \cos(\emptyset/3)$

$X_2 = 2\sqrt{-a\ ^3/27}\ \cos(\emptyset/3 + 120^\circ)$

$X3 = 2\sqrt{-a\ ^3/27}\ \cos(\emptyset/3 + 240^\circ)$

$\mathrm{ARC\ COS}\ y = \mathrm{ARCTAN}\left[\sqrt{(1-y^2)}/y\right]$

Push $\underline{P_o}$.

When it stops, $\frac{1}{3}(\emptyset - 45^\circ)$ appears in W.

Then use SIN-COS program to find X_n.

S117d

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	$S_1 \rightarrow W$	53	b	20	$S_2 \rightarrow W$	56		40				60	+	13	2n+1+Cn+1
	X	12			÷	17			1	21	Transformatio		S_1-- W	53	
	2	22			A→ W	45			—	15			□	06	$n^2x^2 = a_n$
	÷	17	$\frac{b}{2}$		$\sqrt{\ }$	04	$\sqrt{1-y^2}/y$		A→ W	45	X + 1		A→ W	45	
	S3→ W	57	a		LN^{-1}	14			X	12			÷	17	$\frac{a_n}{2n+1+Cn+1}$
05	÷	17		25	W→ A	44		45	2	22		65	LN^{-1}	14	
	3	23			4	24			+	13	X + 1		W→ A	44	C_n
	X	12	$\frac{3b}{2a}$		Store	64			A→ W	45			DEC	66	n - 1
	$S_3 \rightarrow W$	57			.	16			÷	17			DC=0	70	
	1/	05			0	20			LN^{-1}	14			5	25	
10	3	23		30	5	25		50	$W \rightarrow S_1$	52	$\overline{X}=(X-1)/(X+1$	70	5	25	
	$\sqrt{\ }$	04	$\frac{3\sqrt{3}\ b}{2\ a\sqrt{a}}$		2	22			1	21			W→ PC	40	
	LN^{-1}	14	y = COS Ø		3	23			0	20			$S_1 \rightarrow W$	53	
	±	33			5	25			W→ DC	42	N = 10		X	12	
	$W \rightarrow S_2$	54			9	31			Prime	36	$C_N = 0$		1	21	
15	□	06	y^2	35	8	30		55	DC→ W	43	n	75	+	13	$1 + C_2$
	1	21			7	27			+	13			A→ W	45	
	W→ A	44			÷	17			+	13	+ 2n		÷	17	
	LN^{-1}	14			LN^{-1}	14			□	06			RETURN	65	
	—	15	$1 - y^2$		Stop	37	$\frac{1}{3}(\emptyset - 45^\circ)$		1	21					

LOCI Program Quartic Eqn. (#5) **Date:** July 9,'65

Select the smallest Xn such that

$$pq = X_n + \frac{2}{3} a_2 \leq \frac{a_3^2}{4} \qquad (1)$$

Then,

$$S = \left(\frac{a_2 - pq}{2}\right) + \sqrt{\left(\frac{a_2 - pq}{2}\right)^2 - a_o},$$

$r = a_o/S$,

$$q = \frac{a_1 - a_3 S}{r - S},$$

$p = a_3 - q$

At end of program,

p in S_3, r in S_o

q in S_1, S in S_2.

Routine to find p & q

1) Compute pq by (1) and store it in S_1.
2) Enter a_3, push <u>P_o</u>.
3) Enter a_2, push <u>RUN</u>.
4) Enter a_1, push <u>RUN</u>.
5) Enter a_o, push <u>RUN</u>.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00				20	Stop	37		40	X	12		60	A→ W	45	
					W→ S_2	54	a_o in S_2		A→ W	45			÷	17	
					—	15	$( )^2 - a_o$		W→ S_2	54	S in S_2		LN^{-1}	14	8
	W→ S_3	56.	a_3 in S_3		A-- W	45			÷	17			W→ S_1	52	8 in S_1
	Stop	37			$\sqrt{\ }$	04			LN^{-1}	14	$r = a_o/S$		±	33	
05	W→ A	44	a_2	25	S_o→ A	51		45	W→ A	44		65	W→ A	44	
	S_1→ W	53	pq		LN^{-1}	14			A→ S_o	50	r in S_o		S_3→ W	57	
	—	15	a_2 - pq		—	15	$( ) - \sqrt{\ }$		S_1→ W	53			+	13	a_3 - q
	A→ W	45			÷	17			W→ A	44	a_1		A→ W	45	
	X	12			1	21			S_2→ W	55			W→ S_3	56	P in S_3
10	2	22		30	0	20		50	X	12		70	Stop	37	
	÷	17			0	20			S_3→ W	57					
	LN^{-1}	14	$(a_2-pq)/2$		0	20			X	12					
	W→ A	44	in S_o		1/□	07			LN^{-1}	14	a_3S				
	A→ S_o	50			L= -?	74			—	15	a_1-a_3S				
15	□	06	$(\quad)^2$	35	S_o→ A	51	$\sqrt{\quad}$ =0.	55	A→ W	45		75			
	LN^{-1}	14							X	12					
	W↔ A	44							S_o→ A	51					
	Stop	37			LN^{-1}	14			S_2→ W	55					
	W→ S_1	52	a_1 in S_1		S_2→ W	55	a_o		—	15	r - s				

S117f

LOCI Program Quartic (#6) Date: 7/9/65

$$\lambda_{1,2} = \frac{-p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - r}$$

$$\lambda_{3,4} = \frac{-q}{2} \pm \sqrt{\left(\frac{q}{2}\right)^2 - s}$$

$S_o = r$
$S_1 = q$
$S_2 = s$
$S_3 = p$

1) Push P_o for λ_1 and λ_2;

2) a. When program stops, if W is positive, push RUN to get λ_1, then RUN again to set λ_2.

b. If W is negative, W is the imaginary part. To get the real part, push A→ W.

3) Push P_1 for λ_3 and λ_4;

4) Same as 2).

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	2	22	$\lambda_{1,2}$	20	S_o→ A	51	r	40	S_1→ W	53	q	60			
	W→ PC	40			S_3→ W	57	p		X	12					
					X	12			2	22					
	4	24	$\lambda_{3,4}$		2	22			÷	17					
	W→ PC	40			÷	17			LN^{-1}	14					
05				25	LN^{-1}	14		45	W→ S_1	52		65			
					W→ S_3	56	$\left(\frac{p}{2}\right)$ in S_3		□	06	$\left(\frac{q}{2}\right)^2$				
					□	06			LN^{-1}	14					
					LN^{-1}	14			W→ A	44					
	+	13			−	15	$r - \left(\frac{p}{2}\right)^2$		S_2→ W	55	s				
10	A→ W	45		30	A→ W	45		50	−	15		70			
	Stop	37	$( ) + \sqrt{}$		±	33			A→ W	45	$\left(\frac{q}{2}\right)^2 - s$				
	S_3→ W	57			$\sqrt{}$	04			$\sqrt{}$	04					
	−	15			S_3→ W	57			S_1→ W	53					
	−	15			±	33			±	33					
15	A→ W	45	$( ) - \sqrt{}$	35	W→ A	44	$\frac{-p}{2}$	55	W→ A	44	$\frac{-q}{2}$	75			
	Stop	37			LN^{-1}	14			LN^{-1}	14					
					Stop	37			Stop	37					
					W→ S_3	56	$\sqrt{}$		W→ S_3	56					
					P_3	63			P_3	63					

A102

LOCI Program   Catenary Program   Date: 9/10/65

$$f(x) = X(\text{COSH}\ \frac{K_1}{X} - 1) - K_2 = 0$$

$$f'(x) = (\text{COSH}\ \frac{K_1}{X} - 1) - \frac{K_1}{X}\ \text{SINH}\ \frac{K_1}{X}$$

$$X_{i+1} = X_i - \frac{f(x_i)}{f'(x_i)}$$

(Newton's Method)

S_o : f'

S_1 : K_1/X

S_2 : K_1

S_3 : K_2

1) Enter K_1, W→ S_2.

2) Enter K_2, W→ S_3.

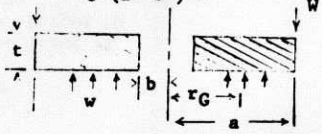
3) Enter first guess X_o, P_o.

4) Read off result.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	÷	17		20	X	12		40	X	12		60	LN^{-1}	14	
1	S_2→ W	55		21	A→ W	45		41	S_o→ A	51	$f'\ (X_i)$	61	W→ A	44	
2	X	12		22	X	12		42	A→ W	45		62	÷	17	
3	LN^{-1}	14		23	LN^{-1}	14	X( )	43	÷	17	f/f'	63	4	24	
4	W→ S_1	52	K_1/X_i	24	W→ A	44		44	S_1→ W	53	Xi	64	÷	17	
05	5	25		25	S_3→ W	57	K_2	45	W→ A	44		65	LN^{-1}	14	
6	6	26		26	—	15		46	LN^{-1}	14		66	+	13	COSH $\frac{K_1}{X1}$
7	Store	64		27	1	21		47	—	15		67	1	21	
8	Restore	65		28	0	20		48	A→ W	45	X_{i+1}	68	—	15	COSH $\frac{K_1}{X_1}$ - 1
9	+	33		29	0	20		49	Dec	66		69	S_1→ W	53	
10	Restore	65		30	0	20		50	P_o	60		70	Restore	65	
11	A→ S_o	50		31	1/☐	07		51				71	W→ L	46	
12	5	25		32	A→ W	45	f (X_i)	52				72	X	12	
13	6	26		33	÷	17		53				73	2	22	
14	Store	64		34	L= -?	74		54				74	÷	17	
15	÷	17		35	S_1→ W	53		55				75	LN^{-1}	14	
16	S_2→ W	55	K_1	36	Stop	37		56	S_1→ W	53	K_1/X_i	76	—	15	
17	X	12		37				57	W→ L	46		77	6	26	
18	LN^{-1}	14		38	LN^{-1}	14		58	2	22		78	9	31	
19	W→ S_1	52	X_i	39	A→ W	45	f (X_i)	59	÷	17		79	W→ PC	40	

$$S = \frac{3W}{\pi m t^2(a^2-b^2)}\left[ma^2 + (m+1)\ a^2 \ln\frac{a}{r_G} - \left(\frac{m+1}{2}\right)\frac{a^2b^2}{r_G^2} - \left(\frac{m-1}{4}\right)\left(\frac{r_G^4+b^4}{r_G^2}\right)\right] ;$$

$$\therefore\ S = \frac{3r_G^2w}{t^2(a^2-b^2)}\left[a^2 + .65a^2\ \left(\ln\frac{a^2}{r_G^2} - \frac{b^2}{r_G^2}\right) - .175\left(r_G^2 + \frac{b^4}{r_G^2}\right)\right],\ \text{for } m = \frac{1}{.3}$$



Typical values:
$a = 39.5''$
$b = 6.625''$
$r_G = 27.75''$
$t = 32''$

$w = 8000$psi.
$W = \pi r_G^2 w$

Answer to above 24797 psi.

0) Put LOCI on AUTO DISP
Enter a, W→ S_1,
" b, W→ S_2,
" c, W→ S_3.
Do this once

1) Push P_o.
2) Enter W, push RUN,
3) Enter t, push RUN,
4) Read result.
One for each calculation

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00				20	5	25		40	+	13	$r_G^2 + b^4/r_G^2$	60	X	12	w
1				21	X	12		41	X	12		61	Stop	37	
2				22	LN^{-1}	14		42	.	16		62	1/□	07	w/t^2
3	S_1→ W	53		23	W→ A	44		43	1	21		63	A→ W	45	
4	□	06		24	S1→ W	53		44	7	27		64	÷	17	
05	S_3→ W	57		25	□	06		45	5	25		65	S_3→ W	57	
6	1/□	07	a^2/ra^2	26	LN^{-1}	14		46	X	12		66	□	06	$r_G^2\ w/t^2$
7	L→ W	47		27	+	13	$a^2+.65a^2(\cdots)$	47	LN^{-1}	14	$.175(\cdots)$	67	3	23	
8	W→ A	44	$\ln\ a^2/r_G^2$	28	A→ S_o	50		48	S_o→ A	51		68	X	12	
9	S_3→ W	57		29	S_3→ W	57		49	—	15		69	S_o→ A	51	
10	1/□	07		30	□	06		50	A→ S_o	50		70	A→ W	45	
11	S_2→ W	55		31	LN^{-1}	14	r_G^2	51	S_1→ W	53		71	X	12	
12	□	06	b^2/r_G^2	32	W→ A	44		52	□	06		72	LN^{-1}	14	
13	LN^{-1}	14		33	S2→ W	55		53	LN^{-1}	14		73	Stop	37	
14	—	15	$\ln a^2/r_G^2-b^2/$	34	□	06		54	W→ A	44		74			
15	X	12	r_G^2	35	S_3→ W	57		55	S_2→ W	55		75			
16	S_1→ W	53		36	÷	17	$b^2/$	56	□	06		76			
17	□	06	$a^2($ $)$	37	LN^{-1}	14		57	LN^{-1}	14		77			
18	.	16		38	□	06		58	—	15	$a^2 - b^2$	78			
19	6	26		39	LN^{-1}	14	b^4/r_G^2	59	Stop	37		79			

APPENDIX B

Standard Code Table

Operation	LOCI Code	Teletype Key	8-Level Code	LOCI Operation	LOCI Code	Teletype Key	8-Level Code
(Reserved)	00	SP	1010 0000	W→ PC	40		1100 0000
Clear Error	01	!	1010 0001	W→ XPC	41	A	1100 0001
Clear W	02	"	1010 0010	W→ DC	42	B	1100 0010
Clear A	03	#	1010 0011	DC→ W	43	C	1100 0011
$\sqrt{\ }$	04	$	1010 0100	W→ A	44	D	1100 0100
$1/\sqrt{\ }$	05	%	1010 0101	A→ W	45	E	1100 0101
□	06	&	1010 0110	W→ L	46	F	1100 0110
1/□	07	'	1010 0111	L→ W	47	G	1100 0111
Step M	10	(	1010 1000	A→ S_o	50	H	1100 1000
Write	11	)	1010 1001	S_o→ A	51	I	1100 1001
X	12	*	1010 1010	W→ S_1	52	J	1100 1010
+	13	+	1010 1011	S_1→ W	53	K	1100 1011
LN^{-1}	14	,	1010 1100	W→ S_2	54	L	1100 1100
—	15	—	1010 1101	S_2→ W	55	M	1100 1101
.	16	.	1010 1110	W→ S_3	56	N	1100 1110
÷	17	/	1010 1111	S_3→ W	57	O	1100 1111
0	20	0	1011 0000	P_o	60	P	1101 0000
1	21	1	1011 0001	P_1	61	Q	1101 0001
2	22	2	1011 0010	P_2	62	R	1101 0010
3	23	3	1011 0011	P_3	63	S	1101 0011
4	24	4	1011 0100	Store & Jump	64	T	1101 0100
5	25	5	1011 0101	Restore	65	U	1101 0101
6	26	6	1011 0110	Decrement	66	V	1101 0110
7	27	7	1011 0111	Test Error	67	W	1101 0111
8	30	8	1011 1000	Test DC=0	70	X	1101 1000
9	31	9	1011 1001	Test A=0	71	Y	1101 1001
RUN	32	:	1011 1010	Test W=0	72	Z	1101 1010
±	33	;	1011 1011	Test W -	73	[	1101 1011
Load Input MX	34	<	1011 1100	Test L -	74	\	1101 1100
Load Output MX	35	=	1011 1101	Car'ge Return	75	]	1101 1101
Prime	36	>	1011 1110	Read	76	↑	1101 1110
Stop	37	?	1011 1111		77	←	1101 1111
			8765 4321				8765 4321
			Channel No.				Channel No.

APPENDIX C Diagnostic Operating Instructions

LOCI Diagnostic Test

#1 & #2	Input - W Register
#3	Input & Accumulator
#4	Accumulator Test
#5	L - Register
#6	Storage Registers
#7 & #8	Programmer

Test Procedures

SIN-COS Program

Arctan

Y - bX + C

LOCI Diagnostic Test #1 & #2

Input - W Register

Machine No. 3010

Date 1 - 6 - 67

Checked by

INSTRUCTIONS & CHECKLIST, TEST #1

1. **Push P_O,**	**result:**	.12345 67890	OK
2. **Push RUN,**	"	2.3456 78901	
3. **Push RUN,**	"	34.567 89012	
4. **Push RUN,**	"	456.78 90123	
5. **Push RUN,**	"	5678.9 01234	

INSTRUCTIONS & CHECKLIST, TEST #2

1. **Push P_O,**	**result:**	67890. 12345	
2. **Push RUN,**	"	78901 2.3456	
3. **Push RUN,**	"	89012 34.567	
4. **Push RUN,**	"	90123 456.78	
5. **Push RUN,**	"	01234 5678.9	

LOCI Diagnostic Test #3

Input & Accumulator

Machine No. 3010

Date 1-6-67

Checked by

INSTRUCTIONS & CHECKLIST

1.	Push P_o,	result:	+1X000 00000	OK
2.	Push RUN,	"	+1.0000 00000	
3.	Push RUN,	"	+1.0000 00000	
4.	Push RUN,	"	+21X00 00000	
5.	Push RUN,	"	+21.000 00000	
6.	Push RUN,	"	+21.000 00000	
7.	Push RUN,	"	+404X0 00000	
8.	Push RUN,	"	-404.00 00000	
9.	Push RUN,	"	-404.00 00000	
10.	Push RUN,	"	-0808X 00000	
11.	Push RUN,	"	-0808.0 00000	
12.	Push RUN,	"	+808.00 00000	
13.	Push RUN,	"	+80808 X0000	
14.	Push RUN,	"	+80808. 00000	
15.	Push RUN,	"	-80000. 00000	
16.	Push RUN,	"	+04040 4X000	
17.	Push RUN,	"	+04040 4.0000	
18.	Push RUN,	"	+40404. 00000	
19.	Push RUN,	"	+80808 08X00	
20.	Push RUN,	"	+80808 08.000	
21.	Push RUN,	"	+81212 12.000	

LOCI Diagnostic Test #4

Accumulator Test

Machine No. 30.10

Date 1-6-67

Checked by

INSTRUCTIONS & CHECKLIST

1. Put AUTO DISP switch off (up).
2. Push P_o, result: +101.01 00000 OK
3. Push RUN, " -101.01 00000
4. Push RUN, " -00404 0404.0
5. Push RUN, " -40404 04.000
6. Push RUN, " +00000 00.040
7. Push RUN, " -40400 00.037
8. Push RUN, " +10101 0101.0
9. Push RUN, " +20202 0202.0
10. Push RUN, " +40404 0404.0
11. Push RUN, " +80808 0808.0
12. Push RUN, " +16161 61616
13. Put AUTO DISP switch on (down)
14. Push P_o, result: +101.01 00000
15. Push RUN, " +.00000 00000
16. Push RUN, " -40404 04.000
17. Push RUN, " -40404 04.000
18. Push RUN, " -40404 04.040
19. Push RUN, " -40400 00.037
20. Push RUN, " +10101 0101.0
21. Push RUN, " +20202 0202.0
22. Push RUN, " +40404 0404.0
23. Push RUN, " +80808 0808.0
24. Push RUN, " +16161 61616

LOCI Diagnostic Test #5

L - Register

Machine No. 3010

Date 1-6-67

Checked by

INSTRUCTIONS & CHECKLIST

1.	Push P_O,	result:	+1.0000 00004	OK
2.	Push RUN,	"	+12.000 00000	
3.	Push RUN,	"	+.02000 00000	
4.	Push RUN,	"	-110.00 00006	
5.	Push RUN,	"	+12100. 00016	
6.	Push RUN,	"	+.00008 99999	
7.	Push RUN,	"	+12.000 00000	
8.	Push RUN,	"	+2.7182 81827	
9.	Push RUN,	"	+.06598 80364	
10.	Push RUN,	"	+00.693 14718	
11.	Push RUN,	"	+1.9999 99999	
12.	Push RUN,	"	+59378 0452.4	

LOCI Diagnostic Test #6

Storage Registers

Machine No. 3010

Date 1·6·67

Checked by

INSTRUCTIONS & CHECKLIST, TEST #6

1. Prime

2.	Push P$_o$	(S$_o$)	Result:	78787 87878	OK
3.	Push RUN	(S$_1$)	"	" "	

If there are only 2 storage registers, stop here.

4.	Push RUN	(S$_2$)	Result:	78787 87878	
5.	Push RUN	(S$_3$)	"	" "	

If there are only 4 storage registers, stop here.

6.	Push RUN	(S$_4$)	Result:	78787 87878	
7.	Push RUN	(S$_5$)	"	" "	
8.	Push RUN	(S$_6$)	"	" "	
9.	Push RUN	(S$_7$)	"	" "	
10.	Push RUN	(S$_8$)	"	" "	
11.	Push RUN	(S$_9$)	"	" "	
12.	Push RUN	(S$_{10}$)	"	" "	
13.	Push RUN	(S$_{11}$)	"	" "	
14.	Push RUN	(S$_{12}$)	"	" "	
15.	Push RUN	(S$_{13}$)	"	" "	
16.	Push RUN	(S$_{14}$)	"	" "	
17.	Push RUN	(S$_{15}$)	"	" "	

LOCI Diagnostic Tests 7 & 8

Programmer

Machine No. 3010

Date 1-6-67

Checked by

INSTRUCTIONS & CHECKLIST #7

1. Make sure Error Light is off.
2. Push P_3.

Result: 00000 00000 OK

PC value: 67

If 99999 99990 appears in W, one of the following errors could occur:

a. PC at 14 or 22, code 72 does not function properly.
b. " " 26 or 34, code 71 " " " "
c. " " 41 or 46, code 73 " " " "
d. " " 53 or 58, code 74 " " " "
e. " " 62 or 70, code 67 " " " "
f. Any other condition, something else is wrong; use other diagnostic cards.

INSTRUCTIONS & CHECKLIST #8

1. Make sure Error Light is off.
2. Push P_o.

Result: 00000 00000 OK

PC value: 11

If machine stops with 99999 99999, one of the following errors occurred:

a. PC at 34, either store of DC (code 64) or restore of DC (code 65) if wrong, or Test DC = 0 (code 70) is wrong.
b. PC at 42, Test DC = 0 (code 70) is wrong.
c. Any other condition, use other diagnostic cards.

LOCI - 2a Testing Procedures

(1) SIN-COS Program

a. Enter 30, push $\underline{P_o}$.

Result is .49999 99999

b. Enter 30, push $\underline{P_2}$.

Result is .86602 54038

c. Enter 180, push $\underline{P_2}$.

Result is -1.0000 00000

d. Enter 90, push $\underline{P_o}$.

Result is 1.0000 00000

e. Enter 45, push $\underline{P_o}$.

Result is .70710 67817

(2) Enter 21.5515, $\underline{W \rightarrow S_2}$. $\underline{P_o}$, when it stops,

$\underline{\Box}$ $\underline{LN^{-1}}$ $\underline{W \rightarrow S_3}$ (this gives SIN^2 21.5515),

$\underline{S_2 \rightarrow W}$ $\underline{P_2}$, when it stops,

$\underline{\Box}$ $\underline{LN^{-1}}$ $\underline{W \rightarrow A}$ $\underline{S_3 \rightarrow W}$ $\underline{\pm}$ $\underline{A \rightarrow W}$

Result displayed should be 1.0000 00006

LOCI - 2a Testing Procedures

Arctan # 1 Program Test

1. PRIME
2. Enter 1, $\underline{P_o}$.
3. Result displayed should be 45.000 00000
4. Enter 99999 99999, $\underline{P_o}$.
5. Result displayed should be 90.000 00079
6. Key the sequence .5 x
   .75 $1/\sqrt{\ }$ LN^{-1}, $\underline{P_o}$.
7. Result displayed should be 29.999 99982
8. Enter 267.87 65432, $\underline{P_o}$.
9. Result displayed should be 89.786 11309

LOCI-2 Test Procedure

Y = bX + C Program

1. Make sure AUTO DISP is up.

2. PRM, W→S1, W→ S2, W→ S3, A→ S0, P_1.

3. Key in:

1 RUN	2 RUN	2 ± RUN	4 ± RUN
5 ± RUN	1 0 ± RUN	1 0 ± RUN	2 0 ± RUN

P_o.

Result should be: +1.9999 99993.

4. PRM, W→S1, W→ S2, W→ S3, A→ SO, P_1

5. Key in:

1 RUN	3 . 8 RUN	1 0 RUN	3 0 . 8 RUN
5 RUN	1 5 . 8 RUN	1 ± RUN	2 . 2 ± RUN
4 ± RUN	1 1 . 2 ± RUN	2 0 ± RUN	5 9 . 2 ± RUN

P_o

Result is: +3.0000 00015.

6. PRM, W→S1, W→ S2, W→ S3, A→ S0, P_1

7. Key in:

0 RUN	. 7 ± RUN	1 RUN	5 . 7 ± RUN
1 0 RUN	5 0 . 7 ± RUN	1 5 RUN	7 5 . 7 ± RUN
3 ± RUN	1 4 . 3 RUN	2 ± RUN	9 . 3 RUN
1 0 ± RUN	4 9 . 3 RUN		

P_o

Result is: -4.9999 99989.

8. If machine is a LOCI-2a, repeat 2) - 7), except at end of 2), 4), 6), push MSC before going to next step.

LOCI Program Input (Test #1)

Date: 23 April

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	·	16		20	6	26		40	2	22		60	5	25	
	1	21			7	27			W→ A	44			6	26	
	2	22			8	30			CLW	02			7	27	
	3	23			9	31			A→ W	45			8	30	
	4	24			0	20			Stop	37			·	16	
05	5	25		25	1	21		45	4	24		65	9	31	
	6	26			W→ A	44			5	25			0	20	
	7	27			CLW	02			6	26			1	21	
	8	30			A→ W	45			·	16			2	22	
	9	31			Stop	37			7	27			3	23	
10	0	20		30	3	23		50	8	30		70	4	24	
	W→ A	44			4	24			9	31			W→ A	44	
	CLW	02			·	16			0	20			CLW	02	
	A→ W	45			5	25			1	21			A→ W	45	
	Stop	37			6	26			2	22			Stop	37	
15	2	22		35	7	27		55	3	23		75			
	·	16			8	30			W→ A	44					
	3	23			9	31			CLW	02					
	4	24			0	20			A→ W	45					
	5	25			1	21			Stop	37					

Wang Laboratories, Inc. 836 NORTH STREET TEWKSBURY, MASSACHUSETTS

LOCI Program Input (Test #2)

Date: April 29,'65

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	6	26		20	2	22		40	7	27		60	0	20	
	7	27			•	16			W→ A	44			1	21	
	8	30			3	23			CLW	02			2	22	
	9	31			4	24			A→ W	45			3	23	
	0	20			5	25			Stop	37			4	24	
05	•	16		25	6	26		45	9	31		65	5	25	
	1	21			W→ A	44			0	20			6	26	
	2	22			CLW	02			1	21			7	27	
	3	23			A→ W	45			2	22			8	30	
	4	24			Stop	37			3	23			•	16	
10	5	25		30	8	30		50	4	24		70	9	31	
	W→ A	44			9	31			5	25			W→ A	44	
	CLW	02			0	20			6	26			CLW	02	
	A→ W	45			1	21			•	16			A→ W	45	
	Stop	37			2	22			7	27			Stop	37	
15	7	27		35	3	23		55	8	30		75			
	8	30			4	24			W→ A	44					
	9	31			•	16			CLW	02					
	0	20			5	25			A→ W	45					
	1	21			6	26			Stop	37					

LOCI Program Test #3 (Input & Accumulator) Date: April 29 '66

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	Prime	36		20	±	33		40	8	30		60	Stop	37	+40404.
	1	21	16		W→ A	44			0	20			8	30	
	Stop	37			Stop	37	-404.		8	30			0	20	
	+	13			CLW	02			Stop	37	+808086		8	30	
	Stop	37	1.		A→ W	45			-	15			0	20	
05	A→ W	45		25	Stop	37	-404.	45	Stop	37	+80808.	65	8	30	
	Stop	37	1.		0	20			A→ W	45	-80000.		0	20	
	2	22			8	30			Stop	37			8	30	
	1	21			0	20			Prime	36			Stop	37	+8080808b
	Stop	37	216		8	30			0	20			+	13	
10	W→ A	44		30	±	33		50	4	24		70	Stop	37	+8080808.
	Stop	37	21.		Stop	37	-0808b		0	20			A→ W	45	
	CLW	02			CLA	03			4	24			Stop	37	+8121212.
	A→ W	45	21.		-	15			0	20					
	Stop	37			Stop	37	-0808.		4	24					
15	Prime	36		35	CLW	02		55	Stop	37	+040404b	75			
	4	24			A→ W	45			+	13					
	0	20			Stop	37	+1808.		Stop	37	+040404.				
	4	24			8	30			CLW	02					
	Stop	37	404b		0	20			A→ W	45					

LOCI Program Diagnostic Test #4 (Accumulator) Date: 30 April

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	0	20		20	—	15	neg. zero	40	0	20		60	0	20	
	1	21			A→ W	45	suppress in A		4	24			1	21	
	0	20			Stop	37	-101.01		—	15			0	20	
	1	21			0	20			Stop	37	+00000 00.040		1	21	
	·	16			0	20			4	24			+	13	
05	0	20		25	4	24		45	0	20		65	A→ W	45	
	1	21			0	20			4	24			Stop	37	+10101 0101.
	W→ A	44	positive		4	24			·	16			+	13	
	A→ W	45	zero suppress in A		0	20			0	20			A→ W	45	
	Stop	37	+101.01		4	24			0	20			Stop	37	+20202 0202.
10	±	33		30	0	20		50	3	23		70	+	13	
	W→ A	44			4	24			+	13			A→ W	45	
	1	21			±	33			A→ W	45			Stop	37	+40404 0404.
	0	20			W→ A	44			Stop	37	-40400 00.37		+	13	
	0	20			Stop	37	-00404 0404.		Prime	36			A→ W	45	
15	0	20		35	A→ W	45		55	1	21		75	Stop	37	+80808 0808.
	0	20			Stop	37			0	20			+	13	
	0	20			·	16			1	21			A→ W	45	
	0	20			0	20			0	20			Stop	37	+16161 61616
	+	13			4	24			1	21					

LOCI Program Diagnostic Test #5 (L-Register) Date: 30 April'65

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	Prime	36		20	2	22		40	•	16		60	Stop	37	+00.693 14718
	LN^{-1}	14			X	12			5	25			2	22	
	Stop	37	+1.0000 00004		LN^{-1}	14			±	33			X	12	
	3	23			Stop	37	-110.00---6		1/□	07			LN^{-1}	14	
	X	12			□	06			9	31			Stop	37	+1.999
05	4	24		25	LN^{-1}	14		45	√	04		65	2	22	
	X	12			Stop	37	+12100.00016		LN^{-1}	14			0	20	
	LN^{-1}	14			1/√	05			Stop	37	+12.000		•	16	
	Stop	37	+12.000		•	16			1	21			2	22	
	X	12			9	31			W→ L	46			0	20	
10	6	26		30	±	33		50	LN^{-1}	14		70	2	22	
	0	20			9	31			Stop	37	+2.7182 81827		0	20	
	0	20			X	12			±	33			2	22	
	÷	17			•	16			W→ L	46			0	20	
	LN^{-1}	14			0	20			LN^{-1}	14			2	22	
15	Stop	37	+.02	35	1	21		55	Stop	37		75	W→ L	46	
	±	33			±	33			2	22			LN^{-1}	14	
	÷	17			X	12	$\frac{1\ (-.01)}{110(-.99)}$		±	33			Stop	37	59378 0452.4
	2	22			LN^{-1}	14			X	12					
	•	16			Stop	37	.00008 9---9		L→ W	47					

LOCI Program LOCI Diagnostic #6

Date: 9 June '65

Test Storage Registers

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	7	27		20				40				60	A→ S_o	50	
	8	30											Prime	36	
	7	27											S_o→ A	51	
	8	30											A→ W	45	
	7	27											Stop	37	
05	8	30		25				45				65	W→ S_1	52	
	7	27											Prime	36	
	8	30											S_1→ W	53	
	7	27											Stop	37	
	8	30											W→ S_2	54	
10	W→ A	44		30				50				70	Prime	36	
	6	26											S_2→ W	55	
	Store	64											Stop	37	
	Step M	10											W→ S_3	56	
	P_o	60											Prime	36	
15				35				55				75	S_3→ W	57	
													Stop	37	
													Restore	65	

LOCI Program Diagnostic #7 **Date:** 9 June '65

Test W = 0

W = -

A = 0

L = -

Test ER

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00				20	Store	64	Wrong	40	Stop	37		60	Store	64	
					Stop	37			±	33			Stop	37	
					A = 0?	71			W = -?	73			CL ER	01	
					7	27	Wrong		7	27			Test ER	67	
					Store	64			Store	64			CL ER	01	
05				25	Stop	37		45	Stop	37		65	Prime	36	
					+	13			L = -?	74			Stop	37	Everything OK
					A = 0?	71			5	25			7	27	
					3	23			3	23			Store	64	
	Prime	36			4	24			W→ PC	40			Stop	37	
10	W= 0?	72		30	W→ PC	40		50	7	27		70	9	31	
	7	27			7	27			Store	64			9	31	
	Store	64			Store	64			Stop	37			9	31	
	Stop	37	Wrong		Stop	37			÷	17			9	31	
	2	22			W = -?	73			L = -?	74			9	31	
15	W = 0?	72		35	4	24		55	7	27		75	9	31	
	2	22			1	21			Store	64			9	31	
	2	22			W→ PC	40			Stop	37			9	31	
	W→ PC	40			7	27			Test ER	67			9	31	
	7	27			Store				7	27			Restore	65	

LOCI Program LOCI Diagnostic #8 **Date:** 9 June '65

If machine stops with 9 ------ 9 at stop 34,
either store of DC is incorrect, or
DC = 0 test is incorrect.

At stop 42, DC = 0 test is wrong.

Anywhere else, if it stops with 9 ------ 9,
something is wrong.

No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment	No.	Cmd	Code	Comment
00	2	22		20	Prime	36		40				60	9	31	
	W→ PC	40			W→ DC	42			Stop	37			9	31	
					9	31			7	27			9	31	
	P_2	62			9	31			Store	64			P_1	61	
					9	31			9	31					
05				25	9	31		45	9	31		65			
	P_3	63			9	31			W→ A	44					
					9	31			DC→ W	43					
					9	31			—	15					
	Prime	36			9	31			A = 0?	71					
10	Stop	37		30	9	31		50	2	22		70	Dec	66	
					9	31			2	22	Store of DC		Dec	66	
					DC = 0?	70			W→ PC	40			Restore	65	
					Stop	37			9	31					
									9	31					
15				35				55	9	31		75			
					Dec	66			9	31					
					DC = 0?	70			9	31					
					Test ER	67			9	31					
									9	31					

IV. SUMMARY OF CONTROLS

A. Switches.

on - off: Circuit breaker at rear of cabinet.

Auto-Display: When DOWN, displays contents of "A" register after a + or - operation.

Operation Code: For program checkout in manual mode. Normally these switches are all down.

Mode Switch: Manual (step-by-step operation, no card needed).
Auto (card - controlled programs)
Step (card - controlled program, one step at a time)

B. Keys.

CL ER	Clears "error" condition (registers overloaded)
CL A	Clears contents of "A" register only
CL W	Clears contents of "W" register only. Sets MSC = 0
PRM	"Prime", clears A, W, & L registers only. Sets MSC = 0
0,1,2,3...9	Enters digits into W register.
.	Enters decimal point in sequence of digits.
± Change Sign	Changes sign of contents of W register only.
$\dot{\rightarrow}$	Moves decimal in W register 1 position to right.
$\dot{\leftarrow}$	Moves decimal in W register 1 position to left.
←	Erases last digit entered in W register. Will be replaced by next digit pressed.
+	Adds content of W register to contents of A register. Does not clear W register.
-	Subtracts content of W register from contents of A register. Does not clear W register.
x	Used in multiplying contents of W register by content of L register (adds ln W to contents of L register). Clears W register.
÷	Used in division. Subtracts ln W from contents of L register. Clears W register.
√	Adds 1/2 ln W to L register. Clears W register.
□	Adds 2 ln W to L register. Clears W register.

$1/\sqrt{\ }$	Subtracts 1/2 ln W from L register. Clears W register.
1/□	Subtracts 2 ln W from L register. Clears W register.
$\ln^{-1}$ or Antilog	Takes antilog of L register (i.e./ raises a to contents of L register. Displayed on W. Clears L. Clears previous W.
$W \rightarrow A$	Transfer contents of W register to A register. Clears previous value of A, maintains W. (does not add since previous A is cleared.)
$A \rightarrow W$	Transfers contents of A register to W. Clears previous W. Maintains A.
$W \rightarrow L$	Transfers contents of W register to L register. Clears previous L. Maintains W.
$L \rightarrow W$	Transfers contents of L register to W register. Clears previous W. Clears L.
$A \rightarrow S_0$	Transfers contents of A register to S_0, S_4, S_8, or S_{12} (depending on value of Memory Selector). Clears previous value of particular storage, Maintains A.
$S_0 \rightarrow A$	Transfers contents of S_0, S_4, S_8 or S_{12} (depending on value of Memory Selector) to A. Clears previous A. Maintains S.
$W \rightarrow S_1$	Transfers value of W register to S_1, S_5, S_9 or S_{13} (depending on value of Memory Selector). Clears previous value of particular storage. Maintains W. Similarly for $W \rightarrow S_2$ and $W \rightarrow S_3$.
$S_1 \rightarrow W$	Transfers contents of S_1, S_5, S_{13} (depending on Memory Selector) to W. Clears previous W. Maintains S. Similarly for $S_2 \rightarrow W$ and $S_3 \rightarrow W$.
MS	Steps value of Memory Selector, i.e., from 0 to 4, 4 to 8, 8 to 12, or 12 to 0. Does not affect any register.
Run	Advances program by 1 step in step mode, continues program which may have been stopped intentionally in Auto mode, executes command manually indicated by toggle switches in Manual mode.
P_0, P_1, P_2, P_3	Starts program at step 00, 03, 06, or 09 in Auto mode.

C. Indicator Lamps.

Error	Indicates overload condition in one of the registers.
Response	Indicates an operation is being performed when lit.
MS	Indicates Memory Selector state (0, 4, 8, or 12).
Decrement Counter	Indicates value of decrement counter in all modes.

D. Automatic Function (no button on keyboard to control).

W→PC	Advances program to step corresponding to first two digits in W register. (I-14)
W→DC	"Loads" decrement counter (used in loops and summations). Clears previous DC value.
DC→W	Presents current value of DC on W register. Maintains DC. Clears previous W.
PCS, DCS, W→PC	Stores Value of decrement counter and program step number. Advances program to step corresponding to first two digits in W register. (I-14)
Recall PC, DC	Resets decrement counter to value previously stored. Advances program to the next step after the PCS, DCS, W→PC step. (I-15)
Decrement DC	Decreases value of decrement counter by one.
TEST ERROR	If no overload of any register, advances PC by 4 counts. If an overload exists, advances PC by 1 count. No effect on contents of any register.
TEST DC=0	If DC is zero, advances PC by 4 counts. If DC is not zero, advances PC by 1 count. No effect on contents of any register.
TEST A=0	If A is zero, advances PC by 4 counts. If A is not zero, advances PC by 1 count. No effect on contents of any register.
TEST W=0	If W is zero, advances PC by 4 counts. If a is not zero, advances PC by 1 count. No effect on contents of any register.
TEST W = -	If W negative, advance PC by 4 counts. If W is not negative, advance PC by 1 count. No effect on contents of any register.
TEST L = -	If contents of L register is negative, advance PC by 4 counts. If L register is not negative, advances PC by 1 count. No effect on contents of any register.

APPENDIX B

Standard Code Table

Operation	LOCI Code	Teletype Key	8-Level Code	LOCI Operation	LOCI Code	Teletype Key	8-Level Code
(Reserved)	00	SP	1010 0000	W→ PC	40	@	1100 0000
Clear Error	01	!	1010 0001	W→ XPC	41	A	1100 0001
Clear W	02	"	1010 0010	W→ DC	42	B	1100 0010
Clear A	03	#	1010 0011	DC→ W	43	C	1100 0011
√	04	$	1010 0100	W→ A	44	D	1100 0100
1/√	05	%	1010 0101	A→ W	45	E	1100 0101
□	06	&	1010 0110	W→ L	46	F	1100 0110
1/□	07	'	1010 0111	L→ W	47	G	1100 0111
Step M	10	(	1010 1000	A→ S_o	50	H	1100 1000
Write	11	)	1010 1001	S_o→ A	51	I	1100 1001
X	12	*	1010 1010	W→ S_1	52	J	1100 1010
+	13	+	1010 1011	S_1→ W	53	K	1100 1011
LN^{-1}	14	,	1010 1100	W→ S_2	54	L	1100 1100
—	15	—	1010 1101	S_2→ W	55	M	1100 1101
.	16	.	1010 1110	W→ S_3	56	N	1100 1110
÷	17	/	1010 1111	S_3→ W	57	O	1100 1111
0	20	0	1011 0000	P_o	60	P	1101 0000
1	21	1	1011 0001	P_1	61	Q	1101 0001
2	22	2	1011 0010	P_2	62	R	1101 0010
3	23	3	1011 0011	P_3	63	S	1101 0011
4	24	4	1011 0100	Store & Jump	64	T	1101 0100
5	25	5	1011 0101	Restore	65	U	1101 0101
6	26	6	1011 0110	Decrement	66	V	1101 0110
7	27	7	1011 0111	Test Error	67	W	1101 0111
8	30	8	1011 1000	Test DC=0	70	X	1101 1000
9	31	9	1011 1001	Test A=0	71	Y	1101 1001
RUN	32	:	1011 1010	Test W=0	72	Z	1101 1010
±	33	;	1011 1011	Test W -	73	[	1101 1011
Load Input MX	34	<	1011 1100	Test L -	74	\	1101 1100
Load Output MX	35	=	1011 1101	Car'ge Return	75	]	1101 1101
Prime	36	>	1011 1110	Read	76	↑	1101 1110
Stop	37	?	1011 1111		77	←	1101 1111
			8765 4321				8765 4321
			[illegible]				
			Channel No.				Channel No.

Wang Laboratories, Inc.

REPRESENTATIVES

PADDOCK - JOSLOW COMPANY
721 Ellsworth Drive
P. O. Boc 192
Silver Spring, Maryland 20907
301 JU 8-7866

WANG LABORATORIES (CANADA) LTD.
Olympia Square
797 Don Mills Road
Don Mills, Ontario, Canada
416 429-0261
Mr. Winston Chen, Mgr., Mr. Robert Carlson
Mr Robert Cunningham, Sls & Svs., Mr. Richard Adam

OVERSEAS

AEROMARITIME ELECTRONIQUE
57, Avenue D'Lena
Paris 16, France
Mr. Jacques Boivin

RIKEI TRADING CO., LTD.
Kozato Kaikan Building 12
2-Chome, Shiba Tamura-Cho
Minato-Ku, Japan

M. STENHARDT, AB
Bjornsensgatran 197
Stockholm-Brumma, Sweden

MR. MAURICE BARBIER
9 Rue de La Fontaine
1211 - Geneve 3, Switzerland

S. T. G. INTERNATIONAL LTD.
52 Nachlat Benyamin Street
Post Office Box 1276
Tel-Aviv, Isreal
Mr. T. Gruber

INDUSTRIAL ELECTRONICS
P. O. Box 22423
Taipei, Taiwan, China
Mr. H. T. Liu, Mgr.

Wang Laboratories was founded in 1951, among the pioneering concerns making New England the heart of the electronics industry.

Wang's modern plant and engineering facilities are situated just off Interstate Highway 495, known as Boston's new Outerbelt. We are 30 minutes by car from Boston.

Our digital systems and electronic products have served a wide range of users in business, industry, the universities and government. We have a standing policy of meeting the needs of our customers first with high grade product design, manufacture, and customer service.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876 TEL. 617 851-7311