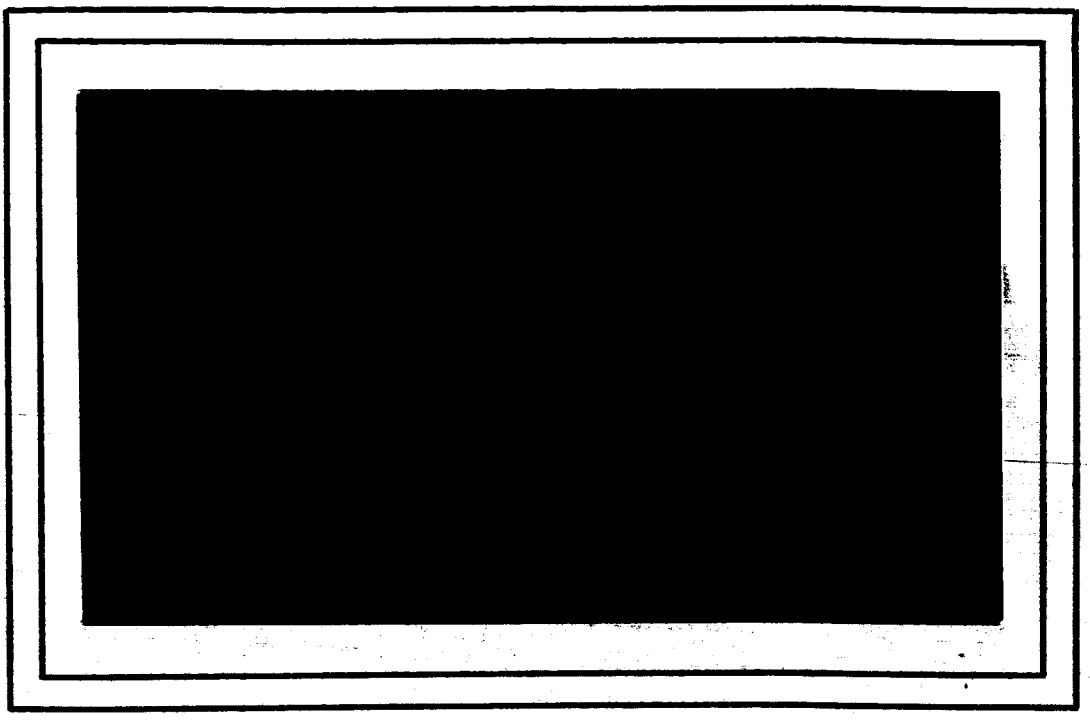


UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

Technical Report TR-67-38 January, 1967
NsG-398

MMPLOT - A Calcomp Plot Routine for MIMIC Programs*

by

Chan M. Park

*This work was supported by National Aeronautics and Space Administration Grant NsG-398 to the Computer Science Center, University of Maryland, College Park, Maryland.

CONTENTS

1. Introduction

This report describes MMPLOT, a Calcomp plot routine for MIMIC and minor modifications made to the original MIMIC package[1] in order to generate output which will be compatible with the MMPLOT routine. Both the modified version of the MIMIC package and the MMPLOT routine are now standard on the University of Maryland (UOM) library tape.

A MIMIC program is very simple to write and extremely useful in solving systems of ordinary differential equations. However, it has been a common desire to have the results plotted out in a form of curves similar to those available with an analog computer in order to depict the resulting functions. The program MMPLOT was developed to fulfill this objective. While there are several ways of obtaining plots, the author chose a Calcomp plot technique because it gives smooth curves and the program package for Calcomp plotting is readily available at the Computer Science Center of the University of Maryland (UOM 0018).

As will be described in the following sections, the principle is that the modified version of the MIMIC package will supply the plotting data to the MMPLOT routine which will plot the data on a Calcomp plot machine. This means that MMPLOT is always dependent on the MIMIC program, and it would be desirable to make MMPLOT a part of MIMIC package. However, MMPLOT was written as a separate program to maximize the available core storages for both MIMIC

and MMPLOT. Thus, the user must call two separate program packages from the UOM library tape; the MIMIC program package, followed by the MMPLOT package.

It should be noted that the Calcomp plot subroutines employed by the MMPLOT package and the control cards presented in this report are those currently used in the University of Maryland and may be different from those of other installations. Therefore, it may be necessary to make adequate changes if MMPLOT is to be used in other installations.

2. Description of Procedure

In the original MIMIC system[1], the plot instruction PLO (A, B, C, D, E, F) is a dummy routine, which each installation is expected to implement in a fashion compatible with its particular plotting devices. Each time the PLO instruction is called the current value of the independent variable A, together with the current values of the variables B, C, D, E, and F (or as many arguments≤6 as are specified) are transmitted for plotting.

In this UOM version these values are simply written on a scratch tape mounted on B-3 (SYSUT2) for actual plotting at a later time. In addition the initial call (when T=0) also transmits the actual symbolic names of the argument variables, which are then used for labeling the plot.

At the completion of the entire MIMIC program, the user must

load and execute the MMPLOT program (see Section 8) which then reads SYSUT2, sorts and interprets the plot data it contains and generates a new tape on B-5 (SYSCK2) which may be plotted off line on a Calcomp plotter.

Each PLO instruction in the MIMIC program will generate a separate labeled plot for each set of parameters. Further each plot will be identified with a figure number to indicate the particular PLO statement and particular parameter set which generated it.

The output plot corresponding to a PLO instruction with $n \leq 6$ arguments will contain n-1 distinct curves. The first variable of the argument string is always assumed to be the independent variable and is plotted along the horizontal axis. Each of the dependent variables is plotted along the vertical axis and is individually scaled to provide an optimum plot. See Appendix I for two typical plots.

3. Data Supplied to MMPLOT by the MIMIC Package

As described in the MIMIC manual [1)], the function PLO supplies the numerical values of the arguments to the user's plot routine for every DT units of T. This is not enough to sort the data at a later time because of the complexities arising from the use of multiple PLO statements in a single program, together with the use of variable numbers of arguments in a PLO instruction. Consequently

the original MIMIC program package (especially SUBROUTINE MIMIO) has been modified to transmit the following additional information to MMPLOT via the scratch tape mounted on B-3;

a) The word BEGIN is transmitted at the beginning (NPLOT=0) of each execution with a new set of parameters.

b) A running number (KOUNT) is transmitted for each PLO statement at each DT increment. KOUNT is initially 0 and is increased by 1 as the time is incremented. (Note: Since each PLO statement will have the same value for KOUNT at a certain time, T, it will be the guide in counting the number of PLO statements later on).

c) The symbolic names of all the variables in the argument list of each PLO statement are transmitted initially at T=0.

(Note: The number of arguments with any PLO statement is at most six. If there are fewer than six arguments in a PLO statement, the blanks will replace the argument names not specified. From the number of blank names in a PLO statement MMPLOT can calculate the number of the actual arguments a programmer specified in his program).

d) The numerical values of the arguments are transmitted for each DT increment.

e) A number 9999999 (LAST) is transmitted at the termination of each run with a particular set of parameters.

f) The word END is transmitted at the end of the whole run.

An example of the way the items are written on the scratch tape is given below. (The flow chart for the supply of data is given in Appendix II together with the listing of modified version of SUBROUTINE MIMIO).

Suppose a programmer has

```
PLO(TIME,CONC1,CONC2)

PLO(CONC1,CONC2)
```

with numerical values (results)

TIME	CONC1	CONC2
0.0	50.0	0.0
10.0	10.1	16.2
20.0	2.0	6.5
30.0	0.4	2.0
40.0	0.1	0.5

Then the scratch tape will have

```
BEGIN ----------------------   indicates the beginning of data
                               with current set of parameters

  0   ----------------------   running number (KOUNT) at T=0

TIME  )
CONC1 |
CONC2 }----------------------   variable names of the arguments
blank |                        in the 1st PLO statement
blank |                        (blank means no argument specified)
blank )
```

```
  0.0
 50.0
  0.0  ----------------------  numerical data of the arguments
 -0.                          in the 1st PLO statement at T=0
 -0.                          (-0. indicates a core constant)
 -0.

    0  ----------------------  KOUNT for 2nd PLO statement at T=0

CONC1
CONC2
blank  ----------------------  variable names of the arguments
blank                         in the 2nd PLO statement
blank
blank

 50.0
  0.0  ----------------------  numerical data of the arguments
 -0.                          in the 2nd PLO statement at T=0
 -0.
 -0.
 -0.

    1  ----------------------  KOUNT for 1st PLO at T=DT

 10.0
 10.1
 16.2  ----------------------  numerical data of 1st PLO at T=DT
 -0.
 -0.
 -0.
```

```
 1      ----------------------    KOUNT for 2nd PLO at T=DT
10.1
16.2
 -0.   ----------------------    numerical data of 2nd PLO at T=DT
 -0.
 -0.
 -0.
 2      ----------------------    KOUNT for 1st PLO at T=DTx2
20.0
 2.0   ----------------------    numerical data of 1st PLO at T=DTx2
 6.5
 -0.
 -0.
 -0.
  2     ----------------------    KOUNT for 2nd PLO at T=DTx2
 2.0
  .
  .
  .
  .
  .
  .
  .
  .
  4     ----------------------    KOUNT for 2nd PLO at T=DTx4
```

```
0.1
0.5
-0.   ---------------------  numerical data of 2nd PLO at T=DTx4
-0.
-0.
-0.

9999999  ---------------------  indicates the last of the data
                                with the current set of parameters

END      ---------------------  indicates the end of whole data
```

4. Data Sorting by MMPLOT Program

When MMPLOT is executed, it rewinds the scratch tape on B-3(SYSUT2) after it writes on-line messages to the operator to mount a tape on B-5 for the Calcomp plot.

The plot data on B-3 is read in the order it was written by the MIMIC program. The following information must be determined;

a) The beginning of plot data associated with a single set of parameters is indicated by the word BEGIN.

b) The number of PLO statements (KK) in the generating MIMIC program is determined by counting the number of KOUNTs with the value zero.

c) The variable names of the arguments of PLO statements are stored in an array ANAME(20,6). (Note the dimension

of this array limits the capacity of MMPLOT to interpret at most 20 PLO instructions)

d) The number of arguments with each PLO statement by counting the number of blanks written out as variable names is determined.

d) The end of plot data associated with a single set of parameters is indicated by KOUNT = 9999999.

f) The end of all plot data is indicated by word END.

The numerical values of the specified arguments of all PLO statements are stored as the successive elements of an array DATA(15000) in the order they were generated by the MIMIC program. They must then be sorted for plotting. We illustrate the sorting procedure using the example of the preceding section. After the word BEGIN and KOUNT (=0) are read in, the names of the arguments are read in and stored as follows.

ANAME(1,1) = TIME

ANAME(1,2) = CONC1

ANAME(1,3) = CONC2

ANAME(1,4) = blank

ANAME(1,5) = blank

ANAME(1,6) = blank

From the number of blanks in this list, the number of real arguments in the first PLO statement is determined to be NUM (1) = 3. The numerical values corresponding to the first PLO

statement are read in and only the 3 numbers (others are core constants) which have meanings are stored as DATA such as

DATA(1) = 0.0

DATA(2) = 50.0

DATA(3) = 0.0

Then the next KOUNT is read and checked to see if it is the same as the previous one. In this example, it is the same indicating that there is another PLO statement. So the names of its variables are read in, the number of arguments are determined, and the numerical data is stored consecutively

i.e. ANAME(2,1) = CONC1

ANAME(2,2) = CONC2

ANAME(2,3) = blank

ANAME(2,4) = blank

ANAME(2,5) = blank

ANAME(2,6) = blank

NUM(2) = 2

DATA(4) = 50.0

DATA(5) = 0.0

The next KOUNT is different from the previous one. Consequently MMPLOT will now read and store only numerical values until KOUNT is equal to 9999999 which indicates the last plotting data with the current set of parameters.

DATA(6) = 10.0

DATA(7) = 10.1

DATA(8) = 16.2

DATA(9) = 10.1

DATA(10) = 16.2

.

.

.

In general, the numerical values of the i-th argument of the j-th PLO statement will be stored in the locations displaced by NUM(1)+...+NUM (J-1) + I., e.g., the numerical values of CONC1 in the second PLO statement of the above example (i=1,j=2) are stored in

NUM(1)+I = 3+1 = 4

4+NUM(1)+NUM(2) = 4+3+2 = 9

9 + 5 = 14

14 + 5 = 19

.

.

.

MMPLOT now generates a Calcomp plot on B-5 (SYSCK2) for each PLO statement. Appropriate scaling factors are determined from examination of the associated numerical values in the DATA array. Variable labels are generated from the ANAME array. A detailed flow chart is presented in Appendix III.

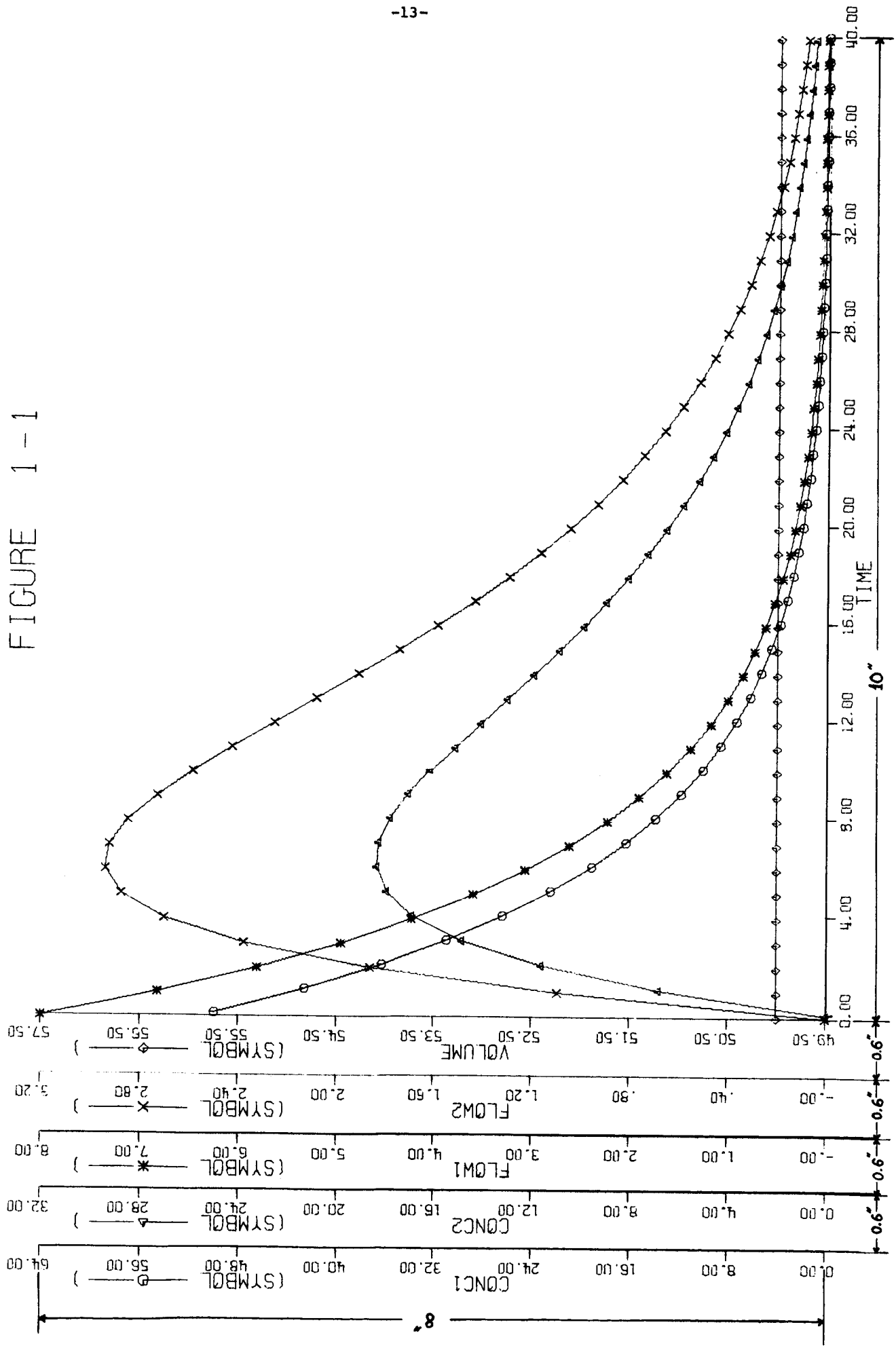
After all the figures are drawn with the first set of parameters, the next set of plotting data will be read from B-3, if any, and the same procedure will be employed to sort and store them.

5. Construction of Figures

As mentioned in Section 2, a PLO statement generates a figure by drawing as many curves as there are dependent variables in that PLO statement, (Note: there is only one independent variable, namely the first argument of a PLO statement). This means that more than one vertical coordinate may be required if the dependent variables have different ranges of magnitude. Therefore, MMPLOT will produce as many vertical coordinates as there are dependent variables. These vertical coordinates are arranged in the order of the appearance of the arguments, i.e., from left to right. Each vertical coordinate is labeled with its variable name, the scale factor if any, and the symbol used in plotting the curve corresponding to that axis. The size of the figure, the spacing between the vertical axes and the symbols used are shown below with a sample statement PLO (TIME, CONC1, CONC2, FLOW1, FLOW2, VOLUME).

The figure numbers are assigned in such a way that the digits to the left of the dash indicate the set number of the parameters and those to the right indicate the corresponding PLO statement number in a user's program. For example FIGURE I-J is the plot

FIGURE 1 - 1


8"
CONC1
0.00
8.00
16.00
24.00
32.00
40.00
48.00
56.00
64.00
(SYMBOL
)
CONC2
0.00
4.00
8.00
12.00
16.00
20.00
24.00
28.00
32.00
(SYMBOL
)
FLOW1
-.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
(SYMBOL
)
FLOW2
-.00
.40
.80
1.20
1.60
2.00
2.40
2.80
3.20
(SYMBOL
)
VOLUME
49.50
50.50
51.50
52.50
53.50
54.50
55.50
56.50
57.50
TIME
0.00
4.00
8.00
12.00
16.00
20.00
24.00
28.00
32.00
36.00
40.00
10"
0.6"
0.6"
0.6"
0.6"

associated with the Ith set of parameters and Jth PLO statement in the generating MIMIC program. Thus if a programmer has two PLO statements and three sets of parameters, then he would expect six figures labeled as FIGURE 1-1, FIGURE 1-2, FIGURE 2-1, FIGURE 2-2, FIGURE 3-1, and FIGURE 3-2.

6. Modifications to the original MIMIC Package

It was the author's intention to preserve as much of the original MIMIC package as possible. However, the following modifications were unavoidable in order to make MMPLOT function correctly. The major change was made to MIM5 - SUBROUTINE MIMIO - to introduce the capability of transmitting the numerical results as well as other necessary items to scratch tape which will be used with MMPLOT as described in section 3. This expansion of MIMIO required more core storage and the dimension of the variable FF in MIMIC package was cut down from 10000 to 8000 (MIM1, MIM2, MIM5, and MIM6). Other modifications include the introduction of an additional subroutine MMSTOP and the change of STOP statements in the original MIMIC package into CALL STOP statements. The purpose of MMSTOP is to provide a safe guard for MMPLOT. There are several situations, arising from user error, in which a MIMIC program will fail to execute. However the MMPLOT is a completely separate program whose execution is normally independent of the preceding MIMIC program. In these cases where the user's MIMIC program does not execute, MMSTOP

first manipulates the location SYSJOB in the UOM monitor to prevent execution of the following MMPLOT program. It then calls STOP in the normal manner. This prevents a needless waste of time which may arise from MMPLOT trying to interpret an erroneous scratch tape on B-3.

7. Limitations

All the limitations described in the MIMIC manual[1] hold with the exception that the machine language program generated by the MIMIC processor should not exceed 8000 instead of 10000 in the original MIMIC package; the other limitation is that a user may not have more than 20 PLO statements in his MIMIC program.

8. Control cards

Both the modified version of MIMIC and the MMPLOT program are on the system library tape, and the user does not need to supply the binary decks. The control cards required to use the MIMIC and MMPLOT in the library tape are described below.

a) If there are no PLO statements in the MIMIC program, there is no need to call MMPLOT and the following run set up is sufficient.

```
$EXECUTE          IBJOB
$ID       name * project number* options
$IBJOB            NOSOURCE,FIOCS
$IBLDR MIMIC      LIBE
$DATA

  user's MIMIC program
```

b) If there are PLO statements in the MIMIC program and the user wants to have them plotted at the end of the run, the following run set up is advised.

```
$EXECUTE        IBJOB
$ID      name * project number * options
$IBJOB          NOSOURCE,FIOCS
$IBLDR   MIMIC  LIBE
$DATA

  user's MIMIC program

7}
8} punch 7 and 8 in column 1
$EXECUTE        IBJOB
$ID      name * project number * options
$IBJOB          NOSOURCE,FIOCS
$IBLDR   MMPLOT LIBE
```

It is suggested that the user write the following information in the "COMMENT OR SPECIAL INSTRUCTION" section of a job card when the job is submitted.

"There are 2 jobs (no BEGIN JOB card needed). 2nd job calcomp plot. Plot tape =B5."

c) Sometimes the user may want to analyze the numerical results of MIMIC run before he attempts plotting the values. In such cases, the following procedures may be used. However it is advisable to plot immediately after the MIMIC run as described in b) unless prior analysis is really necessary.

In this latter case the user must submit the job twice, once

to get the numerical results both printed out and transmitted by means of PLO statements to a blank tape mounted on B-3. The tape B-3 is then saved. Later the MMPLOT routine may be submitted using the saved tape with plotting data mounted on B-3.

The control cards for the first run are:

```
$EXECUTE        IBJOB
$ID      name * project number * options including
         tape save
$*       PLEASE MOUNT A BLANK TAPE ON B3(RING IN)
$*       PLEASE SAVE TAPE AFTER RUN
$PAUSE
$IBJOB          NOSOURCE,FIOCS
$IBLDR   MIMIC  LIBE
$DATA

 user's MIMIC program
```

After the first run, the user will be informed of the reel number of the tape saved for him. This reel number should be specified in the control cards of his second run.

The control cards for the second run are:

```
$EXECUTE        IBJOB
$ID      name * project number * options including
         tape save
$*       PLEASE MOUNT TAPE NO. XXX ON B3 (RING OUT)
$IBJOB          NOSOURCE,FIOCS
$IBLDR   MMPLOT LIBE
```

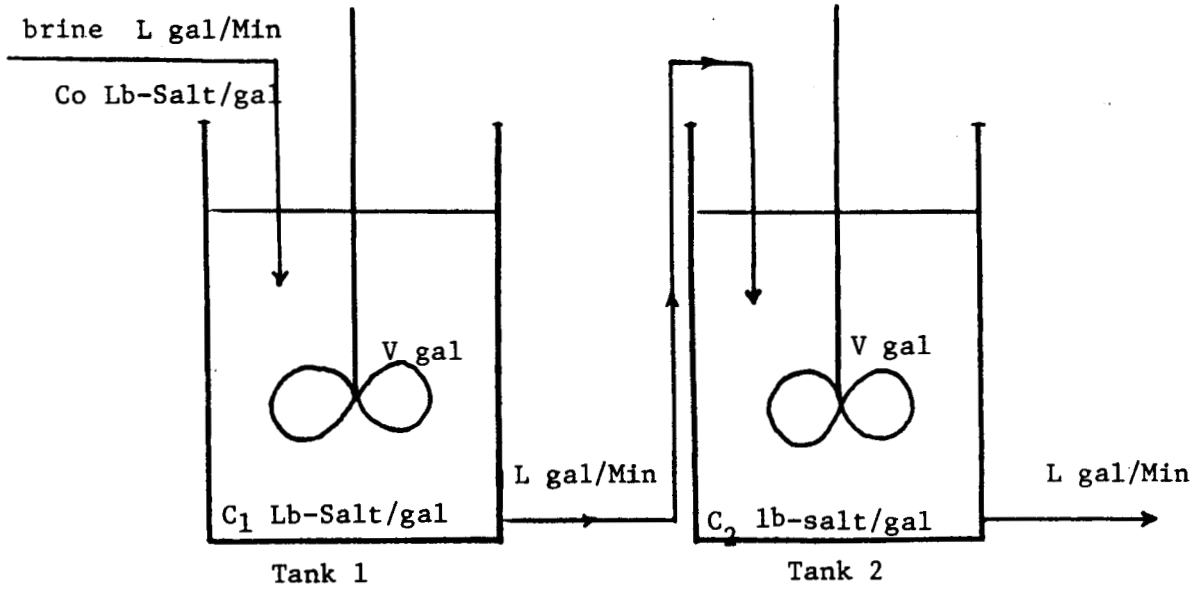
Once again, the tape mounting instructions must also be written in the section of "COMMENT or SPECIAL INSTRUCTION" of the job submittal cards.

APPENDICES

APPENDIX I. Examples

A. Mixing tank problem

The following diagram shows a mixing system with two stirred tanks connected in series. It is desired to find the concentrations in tank 1 and 2 as a function of time, respectively.



At t=0, $C_1 = C_A$ lb/gal

$C_2 = 0$ lb/gal (pure water)

The equations are

$$\frac{dC_1}{dt} = \frac{L}{V} C_0 - \frac{L}{V} C_1 \quad \text{.....(1)}$$

$$\frac{dC_2}{dt} = \frac{L}{V} C_1 - \frac{L}{V} C_2 \quad \text{.....(2)}$$

The MIMIC program with PLO statement and the results are shown in the following pages. The run was made with the following constants;

$L = 8$ gal/min

$V = 50$ gal

$C_0 = 0.0$ lb/gal

$C_A = 50.0$ lb/gal

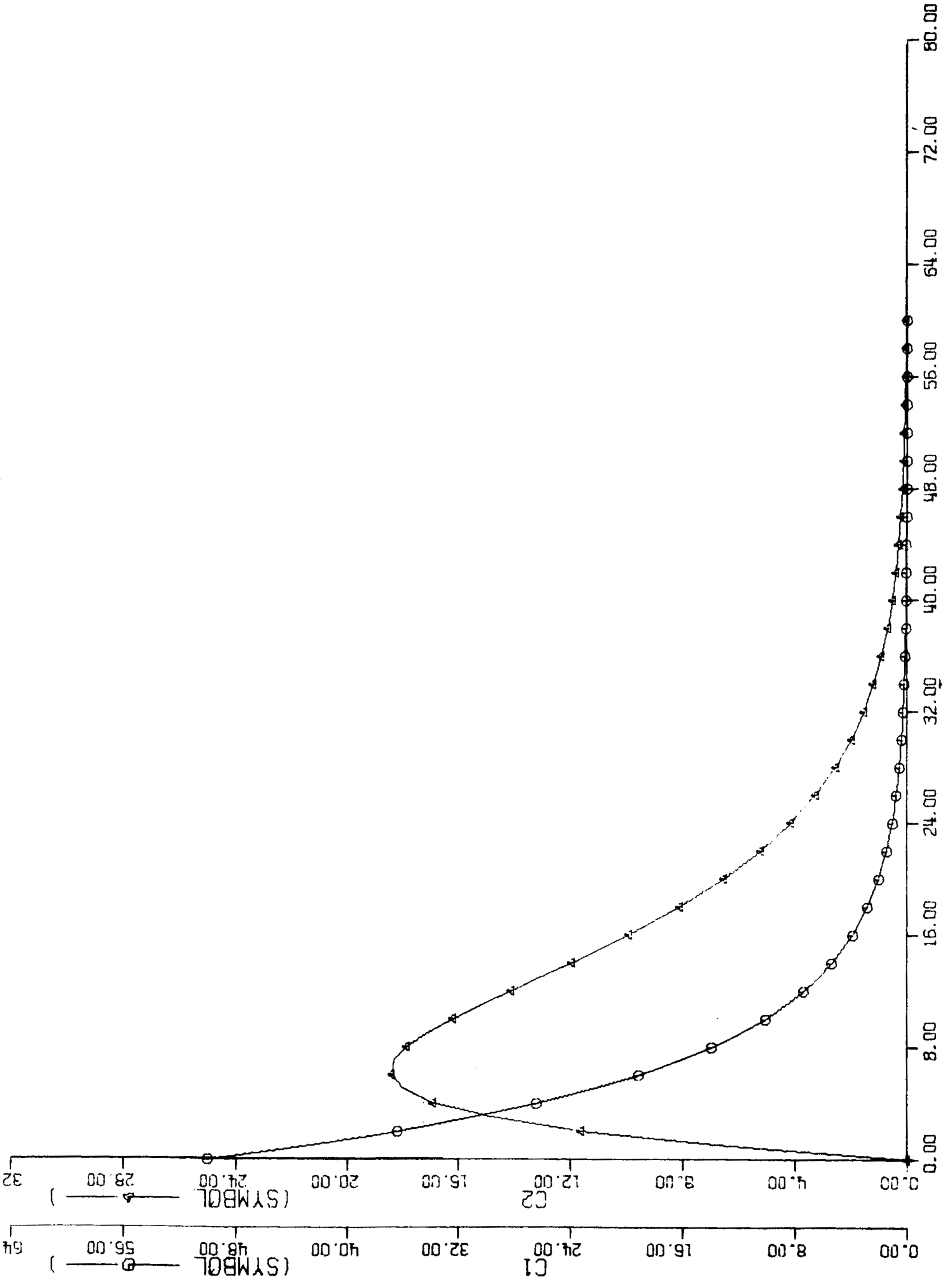
```
$EXECUTE        IBJOB
$ID   PARK, CHAN MO   *001/01/019    *2M*100P
$IBJOB          NOSOURCE,FIOCS
$IBLDR MIMIC    LIBE
$DATA
EXAMPLE A ... MIXING TANK PROBLEM
INITIAL CONDITIONS ... C1=50 AND C2=0 AT T=0
CONSTANTS
                    CON(L,V,MAXT)
PARAMETERS
                    PAR(C0,DTMIN,DT)
MAIN PART
          LOVERV    DIV(L,V)
          DC1DT     LOVERV*(C0-C1)
          C1        INT(DC1DT,50.0)
          DC2DT     LOVERV*(C1-C2)
          C2        INT(DC2DT,0.0)
FINISH STATEMENT
                    FIN(T,MAXT)
PLOT STATEMENT
                    PLO(T,C1,C2)
HEADER STATEMENTS
                    HDR(TIME,C1,C2)
                    HDR
READ OUT STATEMENT
                    OUT(T,C1,C2)
END STATEMENT
                    END
 8.0          50.0          60.0
 0.0          0.0001       1.0
'
$EXECUTE        IBJOB
$ID   PARK, CHAN MO   *001/01/019   *2M*100P
$IBJOB          NOSOURCE,FIOCS
$IBLDR MMPLOT   LIBE
```

CO	DTMIN	DT
0.	1.00000E-04	1.00000E 00

TIME	C1	C2
0.	5.00000E 01	0.
1.00000E 00	4.26072E 01	6.81715E 00
2.00000E 00	3.63075E 01	1.16184E 01
3.00000E 00	3.09392E 01	1.48508E 01
4.00000E 00	2.63646E 01	1.68734E 01
5.00000E 00	2.24664E 01	1.79732E 01
6.00000E 00	1.91446E 01	1.83789E 01
7.00000E 00	1.63140E 01	1.82717E 01
8.00000E 00	1.39019E 01	1.77944E 01
9.00000E 00	1.18464E 01	1.70588E 01
1.00000E 01	1.00948E 01	1.61517E 01
1.10000E 01	8.60224E 00	1.51399E 01
1.20000E 01	7.33035E 00	1.40743E 01
1.30000E 01	6.24651E 00	1.29927E 01
1.40000E 01	5.32292E 00	1.19234E 01
1.50000E 01	4.53590E 00	1.08862E 01
1.60000E 01	3.86524E 00	9.89501E 00
1.70000E 01	3.29374E 00	8.95897E 00
1.80000E 01	2.80674E 00	8.08341E 00
1.90000E 01	2.39174E 00	7.27090E 00
2.00000E 01	2.03811E 00	6.52195E 00
2.10000E 01	1.73676E 00	5.83552E 00
2.20000E 01	1.47997E 00	5.20950E 00
2.30000E 01	1.26115E 00	4.64103E 00
2.40000E 01	1.07468E 00	4.12677E 00
2.50000E 01	9.15782E-01	3.66313E 00
2.60000E 01	7.80378E-01	3.24637E 00
2.70000E 01	6.64994E-01	2.87277E 00
2.80000E 01	5.66670E-01	2.53868E 00
2.90000E 01	4.82885E-01	2.24059E 00
3.00000E 01	4.11487E-01	1.97514E 00
3.10000E 01	3.50646E-01	1.73921E 00
3.20000E 01	2.98801E-01	1.52986E 00
3.30000E 01	2.54621E-01	1.34440E 00
3.40000E 01	2.16974E-01	1.18034E 00
3.50000E 01	1.84893E-01	1.03540E 00
3.60000E 01	1.57556E-01	9.07520E-01
3.70000E 01	1.34260E-01	7.94819E-01
3.80000E 01	1.14409E-01	6.95605E-01
3.90000E 01	9.74927E-02	6.08355E-01
4.00000E 01	8.30778E-02	5.31698E-01
4.10000E 01	7.07942E-02	4.64410E-01
4.20000E 01	6.03269E-02	4.05397E-01
4.30000E 01	5.14072E-02	3.53681E-01
4.40000E 01	4.38063E-02	3.08396E-01
4.50000E 01	3.73293E-02	2.68771E-01
4.60000E 01	3.18099E-02	2.34121E-01
4.70000E 01	2.71066E-02	2.03842E-01
4.80000E 01	2.30987E-02	1.77398E-01
4.90000E 01	1.96834E-02	1.54318E-01
5.00000E 01	1.67731E-02	1.34185E-01
5.10000E 01	1.42931E-02	1.16632E-01
5.20000E 01	1.21798E-02	1.01336E-01
5.30000E 01	1.03789E-02	8.80133E-02

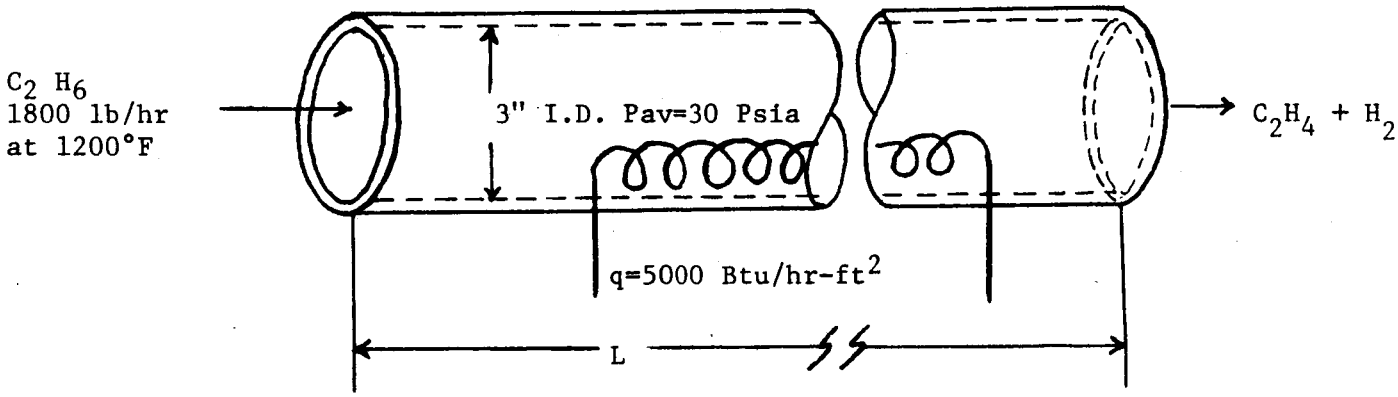
5.40000E 01	8.84434E-03	7.64151E-02
5.50000E 01	7.53665E-03	6.63225E-02
5.60000E 01	6.42231E-03	5.75439E-02
5.70000E 01	5.47273E-03	4.99113E-02
5.80000E 01	4.66355E-03	4.32778E-02
5.90000E 01	3.97402E-03	3.75147E-02
6.00000E 01	3.38643E-03	3.25098E-02

FIGURE 1-1


T
0.00
8.00
16.00
24.00
32.00
40.00
48.00
56.00
64.00
72.00
80.00
C2
(SYMBOL —△—)
0.00
4.00
8.00
12.00
16.00
20.00
24.00
28.00
32.00
C1
(SYMBOL —⊖—)
0.00
8.00
16.00
24.00
32.00
40.00
48.00
56.00
64.00

B. Pyrolysis of ethane problem

The pyrolysis of ethane is to be carried out in a tubular furnace and it is desired to compute the length of tube required to produce 75% decomposition of ethane. The equations and constants are given below, but details may be found elsewhere.[2)]



The equations and constants are:

$$\frac{dZ}{dL} = \frac{AP}{N_0 R}\ \frac{k}{T}\ \frac{1-Z}{1+Z} \qquad \text{.....(1)}$$

$$\frac{dT}{dL} = \frac{Q - \Delta H_R \frac{dZ}{dL}}{(1-Z)C_{PC_2H_6} + (Z\ C_{PC_2H_4} + C_{PH_2})} \qquad \text{.....(2)}$$

$C_{pC_2H_6} = 3.75 + 35.7 \times 10^{-3}\ TK - 10.12 \times 10^{-6} TK^2$ cal/gm-mol-°K

$C_{pC_2H_4} = 5.25 + 24.2 \times 10^{-3}\ TK - 6.88 \times 10^{-6} TK^2$ cal/gm-mol-°K

$C_{pH_2} = 7.00 - 0.385 \times 10^{-3}\ TK + 0.6 \times 10^{-6} TK^2$ cal/gm-mol-°K

$\Delta H_R = 1.8 \times (32{,}732 + 8.50(TK-298) - 5.94 \times 10^{-3}(TK^2 - 298^2)$

$+ 1.28 \times 10^{-6}\ (TK^3 - 298^3))$ BTU/lb-mole

$k = 5.764 \times 10^{16} \times \exp\left(\frac{-41{,}310}{TK}\right)$ sec^{-1}

$R = 10.73$ psia-ft^3/°R-mole

$A = 0.04906$ ft^2

$P = 30.0$ psia

$N_0 = 0.0166667$ lb-mole/hr

$Q = 65.41$ BTU-hr/ft-lb-mole

```
$EXECUTE        IBJOB
$ID   PARK, CHAN MO  *001/01/019  *2M*100P
$IBJOB          NOSOURCE,FIOCS
$IBLDR MIMIC    LIBE
$DATA
EXAMPLE B ... PYROLYSIS OF ETHANE PROBLEM
CONSTANTS
                    CON(C1,C2,C3,C4,C5,C6)
                    CON(C7,C8,C9,C10,C13,C14)
                    CON(R,NO,A,P,Q)
PARAMETERS
                    PAR(DT,DTMIN,FINALC,MAXL)
FIRST EQUATION
          CONST     A*P/(NO*R)
          K         MPY(C13,EXP(DIV(C14,TK)))
          TERM1     DIV(K,TT)
          TERM2     DIV((1.-Z),(1.+Z))
          DZDL      MPY(CONST,TERM1,TERM2)
          Z         INT(DZDL,0.0)
SECOND EQUATION
          TKSQ      TK*TK
          TKCUB     TK*TK*TK
          SQRE      298.*298.
          CUBE      298.*298.*298.
          DELTHR    1.8*(C7+C8*(TK-298.)-C9*(TKSQ-SQRE)+C10*(TKCUB-CUBE))
          CP26      C1+C2*TK-C3*TKSQ
          CP242     C4+C5*TK-C6*TKSQ
          TERM3     Q-DELTHR*DZDL
          TERM4     (1.-Z)*CP26+Z*CP242
          DTDL      TERM3/TERM4
          TT        INT(DTDL,1660.)
          TF        TT-460.
          TK        DIV(TT,1.8)
          L         EQL(T)
FINISH STATEMENTS
                    FIN(Z,FINALC)
                    FIN(T,MAXL)
PLOT STATEMENT
                    PLO(L,TF,Z)
                    PLO(TK,DELTHR,K)
HEADER STATEMENTS
                    HEA(LENGTH,TEMP.,CNVRSN)
                    HDR(FT.,DEGF,FRACTN)
                    HDR
READ OUT STATEMENT
                    OUT(T,TF,Z)
END STATEMENT
                    END
 3.75           35.70E-03    10.12E-06   12.25        23.815E-03     6.28E-06
 32732.          8.5          5.942E-03    1.280E-06   5.7640E+16    -41310.
  10.73        0.01666667     4.906E-02     30.0        65.41
   5.0          0.0005           0.75       1000.0
$
$EXECUTE        IBJOB
$ID   PARK, CHAN MO  *001/01/019   *2M*100P
$IBJOB          NOSOURCE,FIOCS
$IBLDR MMPLOT.  LIBE
```

DT	DTMIN	FINALC	MAXL
5.00000E 00	5.00000E-04	7.50000E-01	1.00000E 03

LENGTH FT.	TEMP. DEGF	CNVRSN FRACTN
0.	1.20000E 03	0.
5.00000E 00	1.21150E 03	5.86790E-05
1.00000E 01	1.22291E 03	1.37713E-04
1.50000E 01	1.23422E 03	2.43466E-04
2.00000E 01	1.24542E 03	3.84022E-04
2.50000E 01	1.25647E 03	5.69525E-04
3.00000E 01	1.26736E 03	8.12535E-04
3.50000E 01	1.27806E 03	1.12837E-03
4.00000E 01	1.28853E 03	1.53536E-03
4.50000E 01	1.29872E 03	2.05497E-03
5.00000E 01	1.30858E 03	2.71165E-03
5.50000E 01	1.31807E 03	3.53237E-03
6.00000E 01	1.32712E 03	4.54574E-03
6.50000E 01	1.33567E 03	5.78060E-03
7.00000E 01	1.34368E 03	7.26422E-03
7.50000E 01	1.35109E 03	9.02030E-03
8.00000E 01	1.35788E 03	1.10668E-02
8.50000E 01	1.36401E 03	1.34145E-02
9.00000E 01	1.36950E 03	1.60656E-02
9.50000E 01	1.37436E 03	1.90141E-02
1.00000E 02	1.37861E 03	2.22461E-02
1.05000E 02	1.38231E 03	2.57417E-02
1.10000E 02	1.38550E 03	2.94764E-02
1.15000E 02	1.38824E 03	3.34238E-02
1.20000E 02	1.39060E 03	3.75567E-02
1.25000E 02	1.39262E 03	4.18486E-02
1.30000E 02	1.39436E 03	4.62752E-02
1.35000E 02	1.39586E 03	5.08143E-02
1.40000E 02	1.39717E 03	5.54467E-02
1.45000E 02	1.39832E 03	6.01556E-02
1.50000E 02	1.39934E 03	6.49269E-02
1.55000E 02	1.40025E 03	6.97488E-02
1.60000E 02	1.40108E 03	7.46117E-02
1.65000E 02	1.40183E 03	7.95074E-02
1.70000E 02	1.40254E 03	8.44295E-02
1.75000E 02	1.40319E 03	8.93727E-02
1.80000E 02	1.40382E 03	9.43327E-02
1.85000E 02	1.40441E 03	9.93060E-02
1.90000E 02	1.40499E 03	1.04290E-01
1.95000E 02	1.40554E 03	1.09282E-01
2.00000E 02	1.40609E 03	1.14281E-01
2.05000E 02	1.40662E 03	1.19285E-01
2.10000E 02	1.40714E 03	1.24293E-01
2.15000E 02	1.40766E 03	1.29304E-01
2.20000E 02	1.40817E 03	1.34318E-01
2.25000E 02	1.40867E 03	1.39333E-01
2.30000E 02	1.40918E 03	1.44350E-01
2.35000E 02	1.40968E 03	1.49367E-01
2.40000E 02	1.41018E 03	1.54386E-01
2.45000E 02	1.41068E 03	1.59404E-01
2.50000E 02	1.41118E 03	1.64423E-01
2.55000E 02	1.41168E 03	1.69442E-01
2.60000E 02	1.41218E 03	1.74461E-01

2.65000E 02	1.41268E 03	1.79480E-01
2.70000E 02	1.41318E 03	1.84499E-01
2.75000E 02	1.41368E 03	1.89517E-01
2.80000E 02	1.41418E 03	1.94535E-01
2.85000E 02	1.41468E 03	1.99552E-01
2.90000E 02	1.41518E 03	2.04569E-01
2.95000E 02	1.41569E 03	2.09586E-01
3.00000E 02	1.41619E 03	2.14601E-01
3.05000E 02	1.41670E 03	2.19617E-01
3.10000E 02	1.41720E 03	2.24631E-01
3.15000E 02	1.41771E 03	2.29645E-01
3.20000E 02	1.41822E 03	2.34659E-01
3.25000E 02	1.41873E 03	2.39671E-01
3.30000E 02	1.41924E 03	2.44683E-01
3.35000E 02	1.41975E 03	2.49694E-01
3.40000E 02	1.42027E 03	2.54705E-01
3.45000E 02	1.42079E 03	2.59715E-01
3.50000E 02	1.42130E 03	2.64724E-01
3.55000E 02	1.42182E 03	2.69732E-01
3.60000E 02	1.42234E 03	2.74739E-01
3.65000E 02	1.42286E 03	2.79746E-01
3.70000E 02	1.42339E 03	2.84751E-01
3.75000E 02	1.42391E 03	2.89756E-01
3.80000E 02	1.42444E 03	2.94760E-01
3.85000E 02	1.42497E 03	2.99763E-01
3.90000E 02	1.42550E 03	3.04765E-01
3.95000E 02	1.42603E 03	3.09767E-01
4.00000E 02	1.42657E 03	3.14767E-01
4.05000E 02	1.42710E 03	3.19766E-01
4.10000E 02	1.42764E 03	3.24764E-01
4.15000E 02	1.42818E 03	3.29761E-01
4.20000E 02	1.42873E 03	3.34758E-01
4.25000E 02	1.42927E 03	3.39753E-01
4.30000E 02	1.42982E 03	3.44747E-01
4.35000E 02	1.43037E 03	3.49740E-01
4.40000E 02	1.43092E 03	3.54731E-01
4.45000E 02	1.43147E 03	3.59722E-01
4.50000E 02	1.43203E 03	3.64711E-01
4.55000E 02	1.43258E 03	3.69700E-01
4.60000E 02	1.43314E 03	3.74687E-01
4.65000E 02	1.43371E 03	3.79672E-01
4.70000E 02	1.43427E 03	3.84657E-01
4.75000E 02	1.43484E 03	3.89640E-01
4.80000E 02	1.43541E 03	3.94622E-01
4.85000E 02	1.43598E 03	3.99603E-01
4.90000E 02	1.43656E 03	4.04582E-01
4.95000E 02	1.43714E 03	4.09560E-01
5.00000E 02	1.43772E 03	4.14536E-01
5.05000E 02	1.43830E 03	4.19511E-01
5.10000E 02	1.43889E 03	4.24484E-01
5.15000E 02	1.43948E 03	4.29456E-01
5.20000E 02	1.44007E 03	4.34427E-01
5.25000E 02	1.44067E 03	4.39395E-01
5.30000E 02	1.44127E 03	4.44362E-01
5.35000E 02	1.44187E 03	4.49328E-01
5.40000E 02	1.44248E 03	4.54292E-01
5.45000E 02	1.44309E 03	4.59254E-01
5.50000E 02	1.44370E 03	4.64214E-01
5.55000E 02	1.44431E 03	4.69173E-01
5.60000E 02	1.44493E 03	4.74130E-01

5.65000E 02	1.44556E 03	4.79085E-01
5.70000E 02	1.44618E 03	4.84038E-01
5.75000E 02	1.44682E 03	4.88989E-01
5.80000E 02	1.44745E 03	4.93938E-01
5.85000E 02	1.44809E 03	4.98885E-01
5.90000E 02	1.44873E 03	5.03830E-01
5.95000E 02	1.44938E 03	5.08773E-01
6.00000E 02	1.45003E 03	5.13714E-01
6.05000E 02	1.45069E 03	5.18653E-01
6.10000E 02	1.45135E 03	5.23589E-01
6.15000E 02	1.45201E 03	5.28523E-01
6.20000E 02	1.45268E 03	5.33455E-01
6.25000E 02	1.45336E 03	5.38385E-01
6.30000E 02	1.45403E 03	5.43312E-01
6.35000E 02	1.45472E 03	5.48236E-01
6.40000E 02	1.45541E 03	5.53158E-01
6.45000E 02	1.45610E 03	5.58078E-01
6.50000E 02	1.45680E 03	5.62995E-01
6.55000E 02	1.45751E 03	5.67909E-01
6.60000E 02	1.45822E 03	5.72820E-01
6.65000E 02	1.45893E 03	5.77728E-01
6.70000E 02	1.45965E 03	5.82634E-01
6.75000E 02	1.46038E 03	5.87536E-01
6.80000E 02	1.46111E 03	5.92436E-01
6.85000E 02	1.46185E 03	5.97332E-01
6.90000E 02	1.46260E 03	6.02226E-01
6.95000E 02	1.46335E 03	6.07116E-01
7.00000E 02	1.46411E 03	6.12002E-01
7.05000E 02	1.46488E 03	6.16885E-01
7.10000E 02	1.46565E 03	6.21765E-01
7.15000E 02	1.46643E 03	6.26641E-01
7.20000E 02	1.46722E 03	6.31514E-01
7.25000E 02	1.46801E 03	6.36382E-01
7.30000E 02	1.46882E 03	6.41247E-01
7.35000E 02	1.46963E 03	6.46108E-01
7.40000E 02	1.47045E 03	6.50965E-01
7.45000E 02	1.47127E 03	6.55818E-01
7.50000E 02	1.47211E 03	6.60666E-01
7.55000E 02	1.47295E 03	6.65510E-01
7.60000E 02	1.47381E 03	6.70349E-01
7.65000E 02	1.47467E 03	6.75184E-01
7.70000E 02	1.47554E 03	6.80014E-01
7.75000E 02	1.47642E 03	6.84840E-01
7.80000E 02	1.47731E 03	6.89660E-01
7.85000E 02	1.47822E 03	6.94475E-01
7.90000E 02	1.47913E 03	6.99285E-01
7.95000E 02	1.48005E 03	7.04089E-01
8.00000E 02	1.48099E 03	7.08888E-01
8.05000E 02	1.48193E 03	7.13681E-01
8.10000E 02	1.48289E 03	7.18467E-01
8.15000E 02	1.48386E 03	7.23248E-01
8.20000E 02	1.48484E 03	7.28023E-01
8.25000E 02	1.48584E 03	7.32791E-01
8.30000E 02	1.48685E 03	7.37552E-01
8.35000E 02	1.48787E 03	7.42306E-01
8.40000E 02	1.48891E 03	7.47054E-01
8.45000E 02	1.48996E 03	7.51793E-01

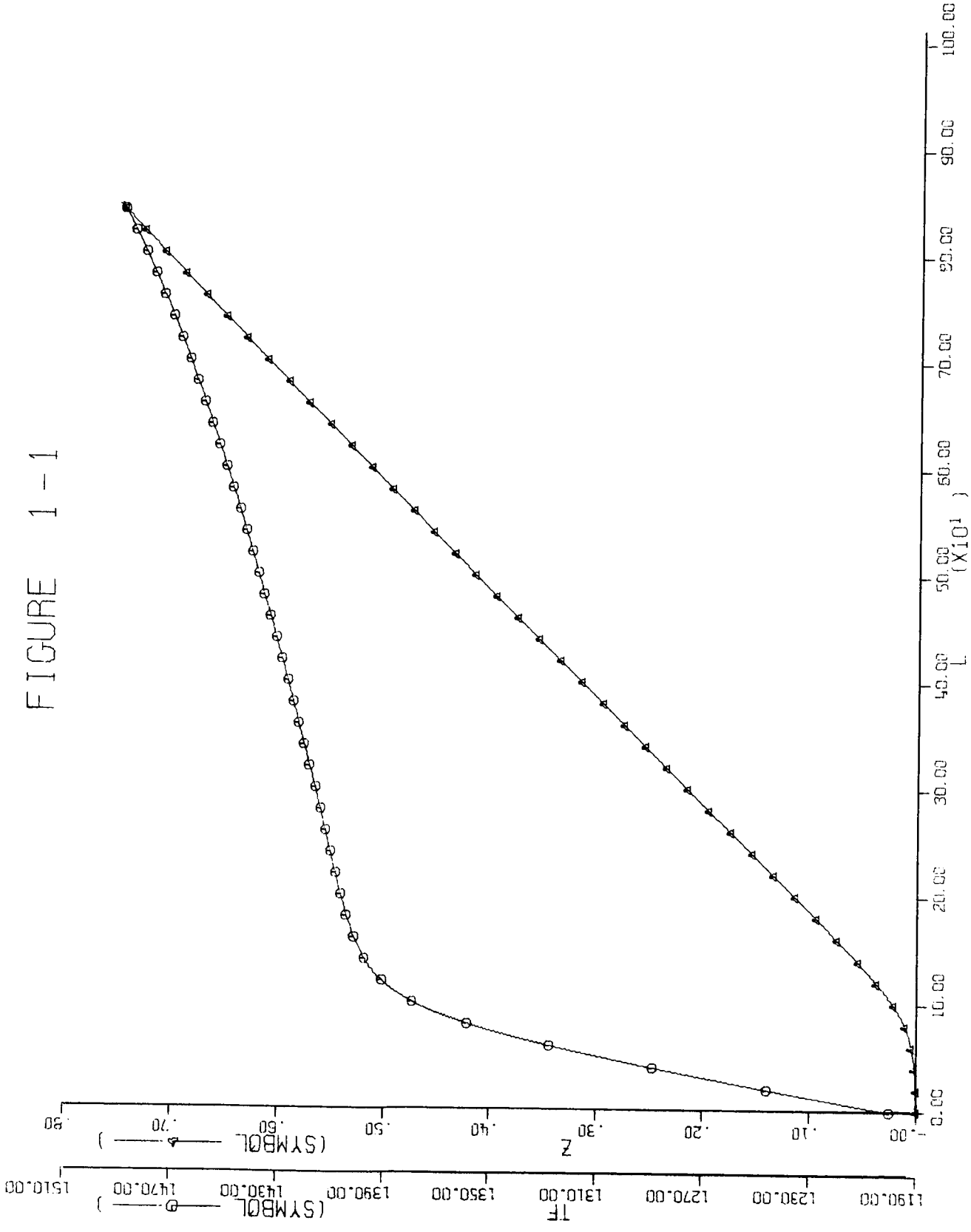

0.00
10.00
20.00
30.00
40.00
50.00
60.00
70.00
80.00
90.00
100.00
L
(X10⁴)
1190.00
1230.00
1270.00
1310.00
1350.00
1390.00
1430.00
1470.00
1510.00
TF
(SYMBOL)
.00
.10
.20
.30
.40
.50
.60
.70
.80
Z
(SYMBOL)

FIGURE 1-1

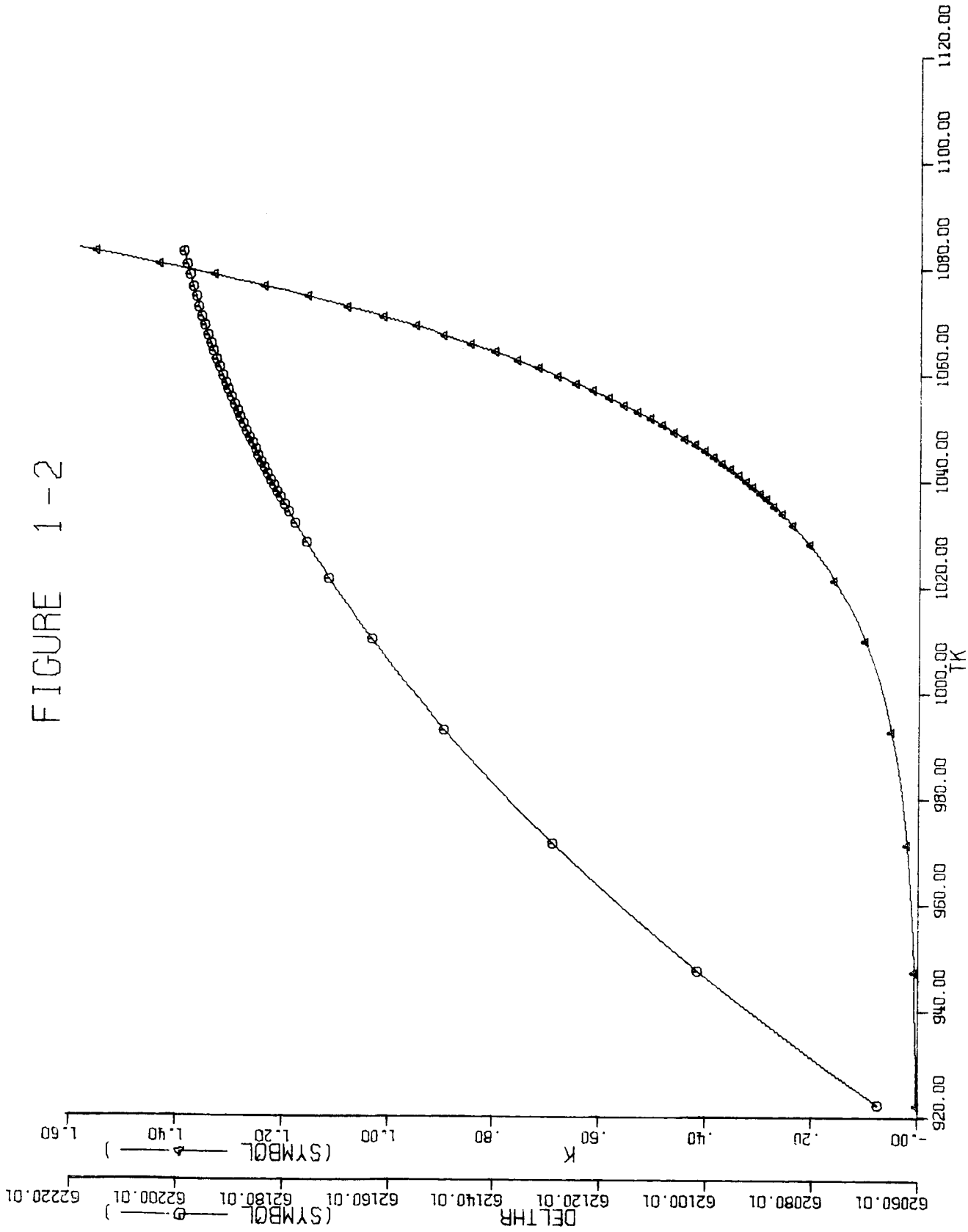
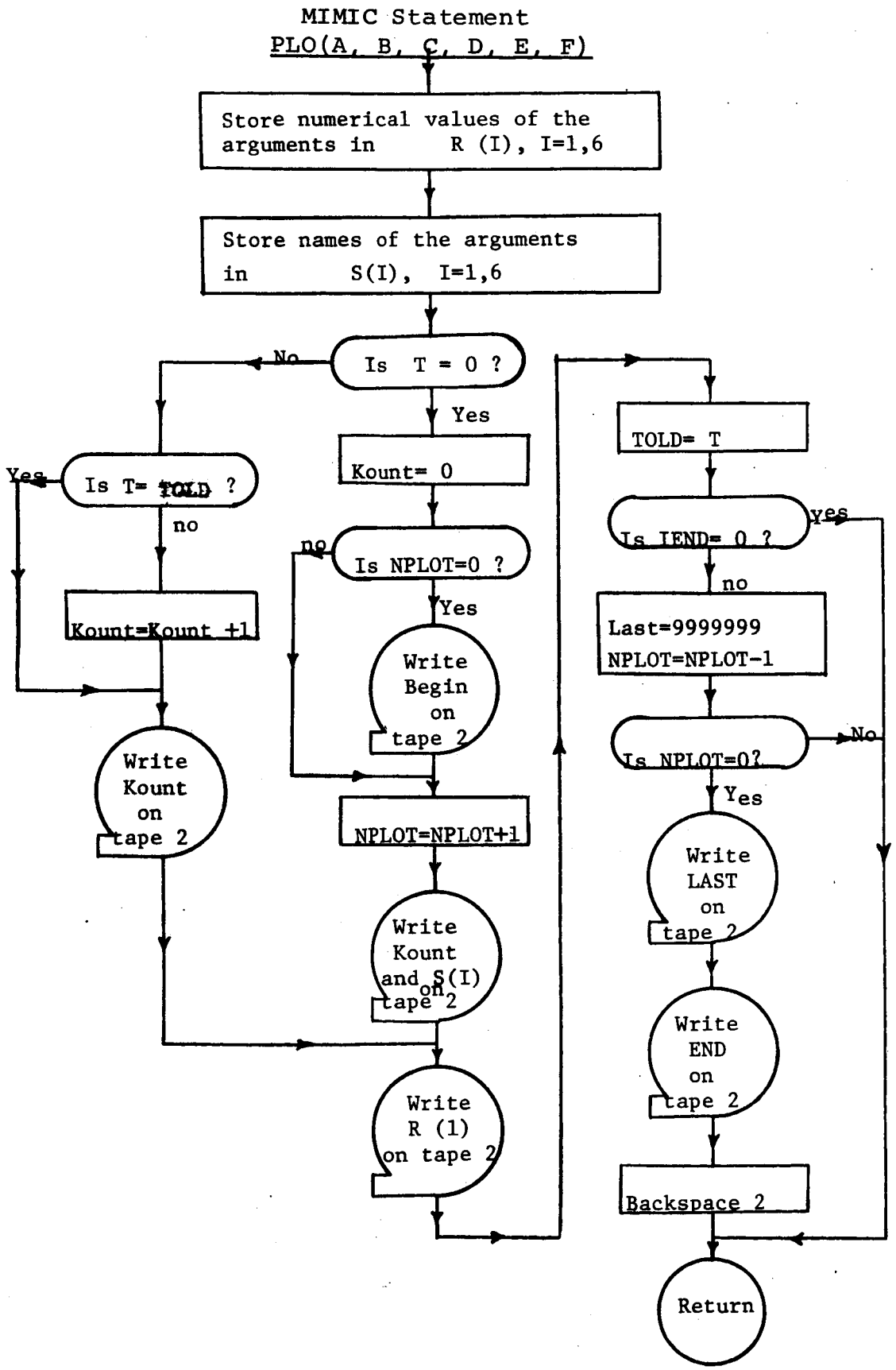

920.00
940.00
960.00
980.00
1000.00
1020.00
1040.00
1060.00
1080.00
1100.00
1120.00
TK
-.00
.20
.40
.60
.80
1.00
1.20
1.40
1.60
K
(SYMBOL
)
(SYMBOL
)
DELTHR
6206O.01
62080.01
62100.01
62120.01
62140.01
62160.01
62180.01
62200.01
62220.01

FIGURE 1-2

APPENDIX II. Data Transmission to MMPLOT (SUBROUTINE MIMIO - modified section)

A. Flow Chart



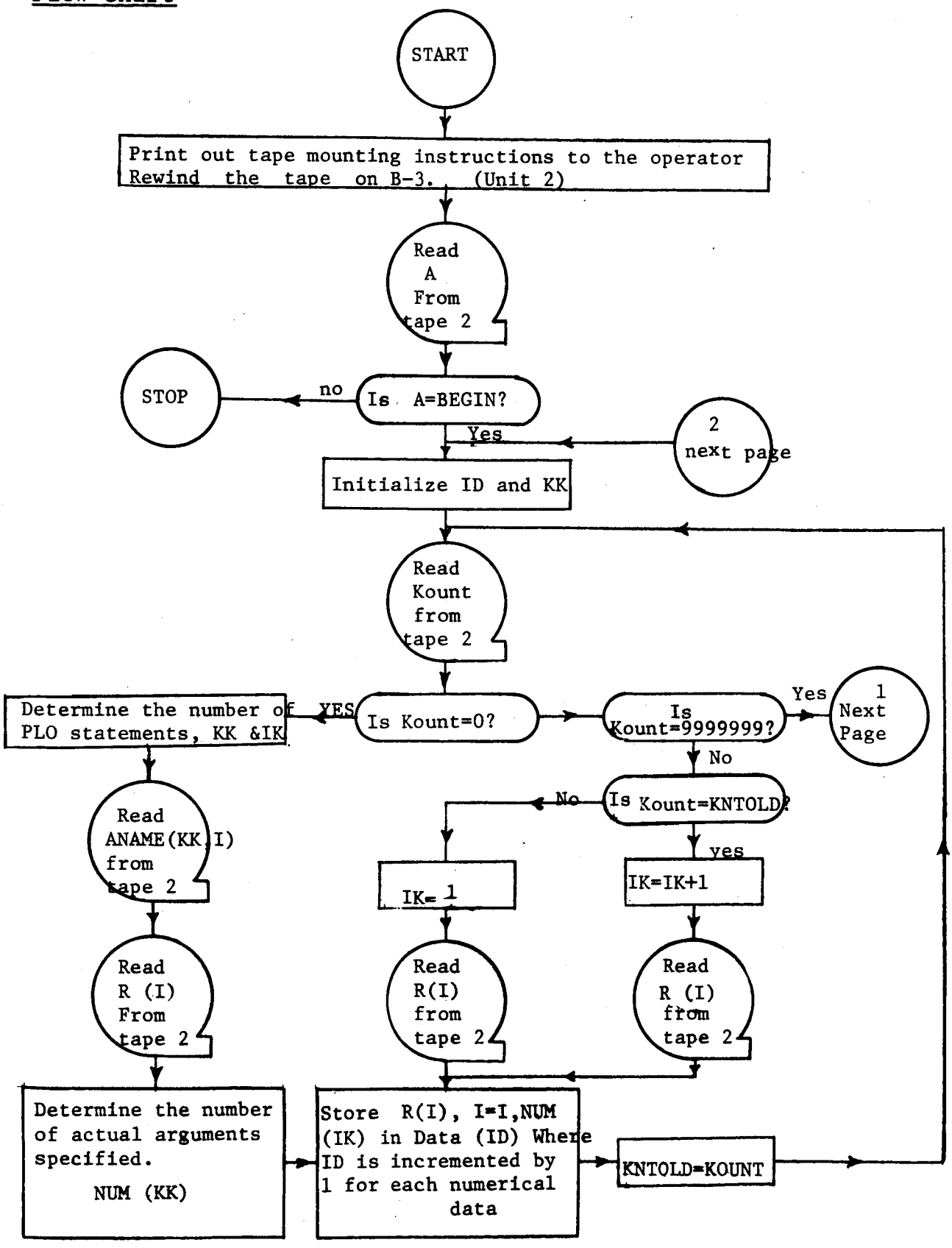
B. Listing

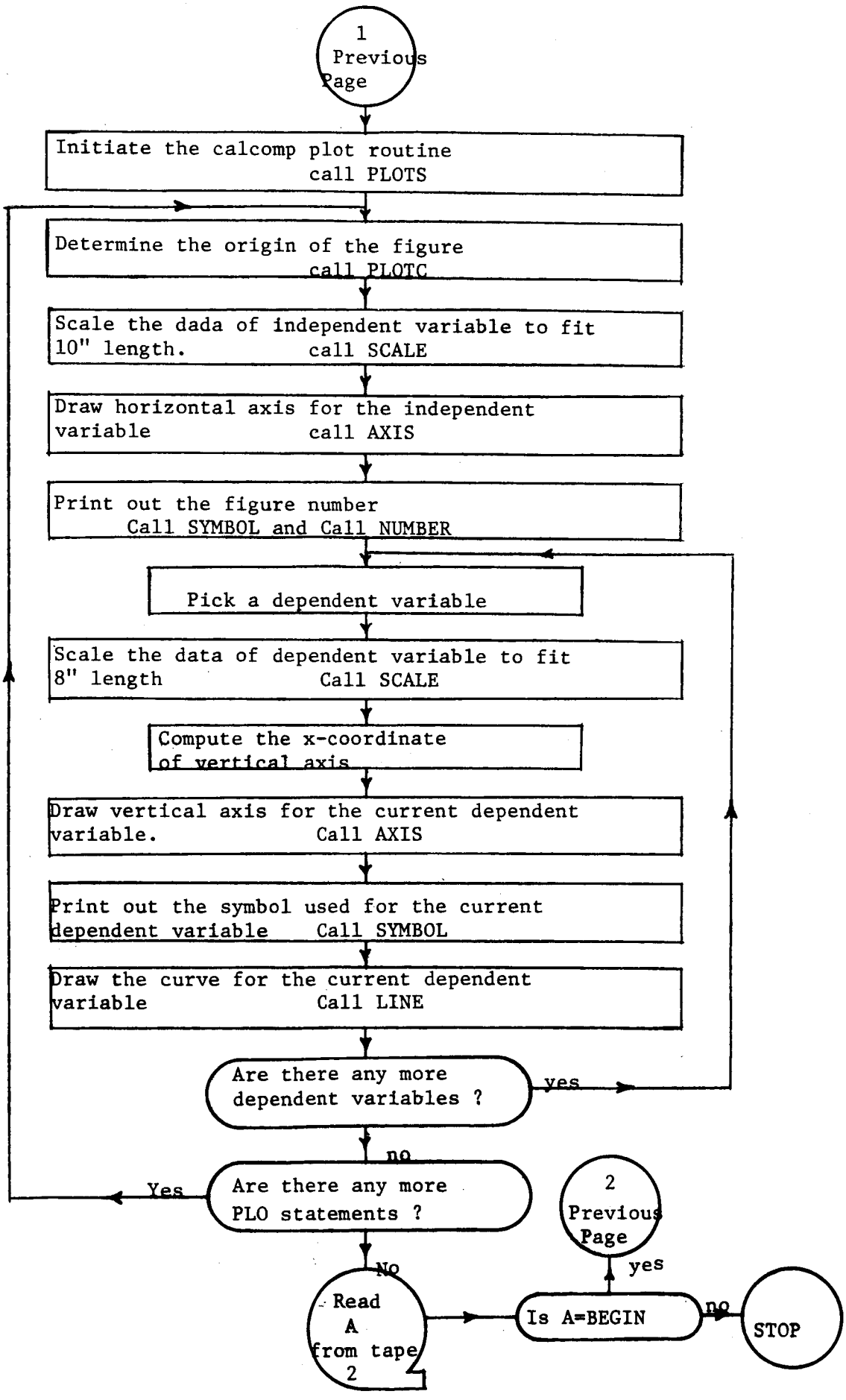
```
$IBFTC MIM5     XR7,M94
       SUBROUTINE MIMIO(A,B,C,D,E,F)
       DIMENSION A(1),B(1),C(1),D(1),E(1),F(1)
       DIMENSION P(95),RX(2000),SX(2000),FF( 8000),R(6),S(6),FMT(8)
       COMMON P,RX,SX,FF,IOUT,IPAR,INOUT,IHDR,IFIN,IEND,NPAR
       COMMON /BLK1/NPLOT
       DATA O1P/0746060600147/,0E205/0732502003305/
       DATA OBL/060606060606C/,014XA6/0730104672106/
       DATA OCP/034606060606C/
       DATA BEGIN/5HBEGIN/
       DATA END/3HEND/
       GO TO (1000,2000,3000,4000,4000),INOUT
C**** COME HERE IF CON OR PAR
 1000  READ(5,1) A(1),B(1),C(1),D(1),E(1),F(1)
       GO TO 3000
C**** COME HERE IF CFN OR PFN
 2000  WRITE(6,3) A,B(2001)
       N=A
       IL=3*N
       READ(5,4)(B(I),I=1,IL)
       WRITE(6,5)(B(I),I=1,IL)
       WRITE(6,2)
       DO 2005 I=3,IL,3
       IF(B(I).NE.0.) RETURN
 2005  CONTINUE
       B(3)=0.
       DO 2010 I=6,IL,3
 2010  B(I)=(B(I-1)-B(I-4))/(B(I-2)-B(I-5))
       B(IL+1)=1.E37
       B(IL+2)=B(IL-1)
       B(IL+3)=0.
       RETURN
C**** COME HERE IF HDR CON OR PAR
 3000  WRITE(6,6)A(2001),B(2001),C(2001),D(2001),E(2001),F(2001)
       IF(INOUT.EQ.1)   GO TO 4000
       RETURN
C**** COME HERE IF OUT CON PAR OR PLO
 4000  R(1)=A
       R(2)=B
       R(3)=C
       R(4)=D
       R(5)=E
       R(6)=F
       S(1)=A(2001)
       S(2)=B(2001)
       S(3)=C(2001)
       S(4)=D(2001)
       S(5)=E(2001)
       S(6)=F(2001)
       IF(INOUT.EQ.5) GO TO 5000
       FMT(1)=O1P
       DO 4020 I=1,6
       FMT(I+1)=0E205
       IF(ERA(S(I),OBL).NE.0.) GO TO 4020
 4010  R(I)=S(I)
       FMT(I+1)=014XA6
 4020  CONTINUE
```

```
      FMT(8)=OCP
      WRITE(6,FMT)R
      IF(INOUT.EQ.1)   WRITE(6,2)
      RETURN
C**** COME HERE IF PLO
 5000 IF(RX(1).NE.0.) GO TO 5020
      KOUNT=0
      IF(NPLOT.EQ.0) WRITE(2) BEGIN
      NPLOT=NPLOT+1
      WRITE(2) KOUNT
      WRITE(2) S
 5010 WRITE(2) R
      TOLD=RX(1)
      IF(IEND.EQ.0) RETURN
      LAST=9999999
      NPLOT=NPLOT-1
      IF(NPLOT.NE.0) RETURN
      WRITE(2) LAST
      WRITE(2) END
      BACKSPACE 2
      RETURN
 5020 IF(RX(1).EQ.TOLD) GO TO 5030
      KOUNT=KOUNT+1
 5030 WRITE(2) KOUNT
      GO TO 5010
 1    FORMAT(6E12.4)
 2    FORMAT(/)
 3    FORMAT(F20.0,11X,A6,3X)
 4    FORMAT(3E12.4)
 5    FORMAT(1P3E20.5)
 6    FORMAT(6(11X,A6,3X))
      END
```

APPENDIX III. Data Sorting and Construction of Figures (MMPLOT)

A. Flow Chart




1
Previous
Page
Initiate the calcomp plot routine
call PLOTS
Determine the origin of the figure
call PLOTC
Scale the dada of independent variable to fit
10" length. call SCALE
Draw horizontal axis for the independent
variable call AXIS
Print out the figure number
Call SYMBOL and Call NUMBER
Pick a dependent variable
Scale the data of dependent variable to fit
8" length Call SCALE
Compute the x-coordinate
of vertical axis
Draw vertical axis for the current dependent
variable. Call AXIS
Print out the symbol used for the current
dependent variable Call SYMBOL
Draw the curve for the current dependent
variable Call LINE
Are there any more
dependent variables ?
yes
no
Are there any more
PLO statements ?
Yes
No
Read
A
from tape
2
Is A=BEGIN
no
STOP
2
Previous
Page
yes

B. Listing

```
$IBFTC MMPLOT  FULIST,M94
C**** CALCOMP PLOT ROUTINE FOR MIMIC PROGRAM
      DIMENSION DATA(15000),DIM(1000),ANAME(20,6),R(6)
      DIMENSION NUM(20)
      DATA BLANK/0606060606060/
      DATA BEGIN/5HBEGIN/
      NPAR=0
      PRINT 2000
      REWIND 2
      READ(2) A
      IF(A.NE.BEGIN) STOP
    1 ID=0
      KK=0
      NPAR=NPAR+1
    5 READ(2) KOUNT
      IF(KOUNT.NE.0) GO TO 25
      KK=KK+1
      IK=KK
      READ(2) (ANAME(KK,I),I=1,6)
      READ(2) R
      NUM(KK)=0
      DO 10 I=1,6
      IF(ANAME(KK,I).EQ.BLANK) GO TO 14
      NUM(KK)=NUM(KK)+1
   10 CONTINUE
   14 IF(I.EQ.1) GO TO 200
   15 IH=NUM(IK)
      DO 20 I=1,IH
      ID=ID+1
      DATA(ID)=R(I)
   20 CONTINUE
      KNTOLD=KOUNT
      GO TO 5
   25 IF(KOUNT.EQ.9999999) GO TO 45
      IF(KOUNT.EQ.KNTOLD) GO TO 30
      IK=1
      READ(2) R
      GO TO 15
   30 IK=IK+1
      READ(2) R
      GO TO 15
   45 N=KNTOLD+1
      IP=(N+50)/50
      K=0
      DO 46 I=1,KK
   46 K=K+NUM(I)
      CALL PLOTS(DIM,1000)
      ID=0
      IN=0
      IH=KK
      DO 100 I=1,IH
      JH=NUM(I)-1
      IN=IN+1
      ID=ID+1
      IX=ID
      CALL PLOTC(0.0,-11.0,-3)
      CALL PLOTC(4.0,1.0,-3)
```

```
      CALL SCALE(DATA(ID),10.0,N,K)
      ISUB1=ID+N*K
      ISUB2=ISUB1+K
      CALL AXIS(0.0,0.0,ANAME(IN,1),-6,10.0,0.0,DATA(ISUB1),DATA(ISUB2))
      CALL SYMBOL(3.7,8.1,0.25,6HFIGURE,0.0,6)
      IF(NPAR.GT.9) GO TO 47
      CALL NUMBER(5.465,8.1,0.25,FLOAT(NPAR),0.0,-1)
      GO TO 48
   47 CALL NUMBER(5.25,8.1,0.25,FLOAT(NPAR),0.0,-1)
   48 CALL SYMBOL(5.75,8.1,0.25,1H-,0.0,1)
      CALL NUMBER(6.0,8.1,0.25,FLOAT(I),0.0,-1)
      DO 50 J=1,JH
      ID=ID+1
      JSYMBL=J
      CALL SCALE(DATA(ID),8.0,N,K)
      ISUB3=ID+N*K
      ISUB4=ISUB3+K
      X=-0.6*FLOAT(JH-J)
      CALL AXIS(X,0.0,ANAME(IN,J+1),+6,8.0,90.0,DATA(ISUB3),DATA(ISUB4))
      CALL SYMBOL(X-0.275,5.5,0.15,7H(SYMBOL,90.0,7)
      CALL SYMBOL(X-0.35,6.5,0.45,15,90.0,-1)
      IF(JSYMBL.EQ.3) JSYMBL=11
      CALL SYMBOL(X-0.35,6.95,0.1,JSYMBL,90.0,-1)
      CALL SYMBOL(X-0.35,7.05,0.45,15,90.0,-1)
      CALL SYMBOL(X-0.275,7.6,0.15,1H),90.0,1)
      CALL LINE(DATA(IX),DATA(ID),N,K,IP,JSYMBL)
   50 CONTINUE
      CALL PLOTC(0.0,0.0,-3)
      CALL PLOTC(10.0,0.0,-3)
  100 CONTINUE
      READ(2) A
      IF(A.EQ.BEGIN) GO TO 1
      CALL PLOTC(0,0,999)
      STOP
  200 WRITE(6,1000) KK
      GO TO 5
 1000 FORMAT(1H1/1H010X,26H NO DEPENDENT VARIABLES IN,I3,12HTH PLOT CALL
     1)
 2000 FORMAT(1H010X,44H ***** CALCOMP PLOT WITH MIMIC PROGRAM *****/1H0,
     112X,39H PLOT TAPE = B5.  PLEASE PLOT AFTER RUN/1H020X,22H ***** TH
     2ANK YOU *****)
      END
```

References:

1) Petersen, H. E. and F. J. Sansom; "MIMIC - A DIGITAL SIMULATOR PROGRAM," SESCA Internal Memo 65-12, Wright-Patterson Air Force Base, Ohio, May, 1965

2) Briggs, D. E. et. al. "The Use of Computers in Chemical Engineering Education," University of Michigan, January, 1963