INTRALAB MEMORANDUM

TO: Distribution

MEMO NO: MX/S 005

FROM: J. Paradiso

DATE: 8/1/77

SUBJECT: VERSATEK SOFTWARE

COPIES TO: All Concerned

This document describes the operation and capability of software employed on the SEL 32 to drive a Versatek printer/ plotter.

Joseph A. Paradiso

J. Paradiso

CONTENTS

I. PREFACE

The Versatek software employed here is a two phase operation. Phase one runs differ for each type of data-set plotted; their primary purpose is to pick up pre-formatted data, convert it if necessary, and output a vector file describing the plots to magnetic tape. Phase one is generally a background process.

Phase Two is one specific program which reads the vector file made by Phase one, rasterizes it, and produces actual plots on the printer/plotter itself. Phase two is a low priority foreground process, and background is kept free for other usage.

Phase One "front end" software is available to plot the three major types of data required from the AIRS system. They are:

Program Name	Data Type
VT#PLC&A	C&A Data
VT#PLFOR	Formatted Data Table
VT#PLDED	Dedicated Data (DS#7)

Two other Phase one programs are available which can plot generalized data structures with the aid of a brief pre-

formatting routine. They are:

Program Name	Application
VT#PLGEN	Plots data in conventional 1 graph/plot form similar to the above routines.
VT#PLMLT	Plots up to 10 graphs/plot. Setup is somewhat unique.

II. OPERATIONAL DESCRIPTIONS

A) Execution Setup

The execution commands governing VT#PLC&A, VT#PLFOR,VT#PLDED, VT#PLGEN are similar. The only major difference is the means of indexing used in each dataset. A general description is given here, and indexing will be dealt with separately for each routine.

The format of a typical execution deck is as follows:

① ——→ $JOB PLOT, VERSATEK

[If the LFC is to be re-assigned from the default, include a $ASSIGN(1,2,3)IN=(NAME) here.]

② ——→ $EXECUTE (Program Name)

③ ——→ (Begin Record), (Stop Record), (Sampling Period),

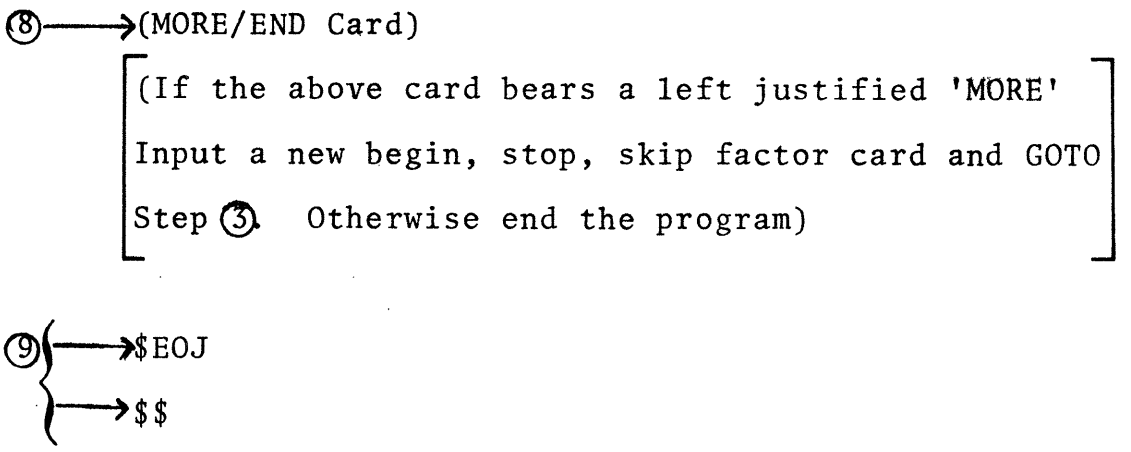
④ ——→ [ⓐ (OPTION CODE)
(As many options and in any order desired)]

⑤ ——→ MAIN TITLE CARD

⑥ ——→ [Indexing cards in any order and in any amount. See independent description of each routine for details.]

⑦ ——→ 999

(Terminates run)



B) Functional Breakdown of Execution Structure

Each execution step will now be discussed in detail:

Step #1: $JOB (title)

This card initiates a job under RTM. The input file phumonic is the character string "IN", and its default assignments are:

Program	"IN" Assignment	
VT#PLC&A	DS4	*Scratch 2 is a 600 sector file
VT#PLFOR	DS7**	on DM08.
VT#PLDED	DS7	**VT#PLFOR also uses the file code
VT#PLGEN	SCRATCH 2*	"CIR" for plotting the circular file.
VT#PLMLT	SCRATCH 2*	It is assigned to DS1 by default.

The input may be re-assigned by placing one of the following cards immediately before the $EXECUTE card:

$ASSIGN1 IN=(file name)	Assign input to another disk file.
$ASSIGN3 IN=MT,TAPE	Assign input to magnetic tape.

When working with magnetic tape, the data should reside in the first file for maximum efficiency. All routines key on dataset ID's, however, so no "skip-file" procedure is necessary.

Step #2: $EXECUTE (program name)

This card causes RTM to load a run module into core and execute the appropriate routine. The program name can be any of the VT#PLxxx files listed above.

Step #3 (Begin record), (Stop record), (Sampling Period),

This card contains three integers which specify restrictions on data acquisition. "Begin record" is the index of the first record retrieved from the input file (starting from record number zero). "Stop record" denotes the final record read. If an EOF is encountered before the stop record, the program will plot all data up to the EOF. "Sampling Period" determines the number of records to skip between plotted points. This number should be rounded such that:

$$[(\text{Stop record}) - (\text{Begin record})] / (\text{Sampling Period}) \leq N$$

Where N is the buffer size. N=2000 normally, and when the "@EXT" option is specified, N=6000. If the above expression is greater than N, the program increases the sampling period such that the restraint is met. Because of SEL Fortran, a comma is required after this entry.

Step #4: @(option code)

This section allows the user to change default parameters if desired, and customize plotting operation. They may be specified in any order and quantity desired. If no options are needed, none need to be declared. The options will hold until the job ends, or until an "@CLR" directive is encountered (A few particular options can not be re-set via "@CLR").

An option specification is denoted via the "@" character followed by a three letter code. Some require a numerical input card after the option specification. Remember to follow all integer entries by a comma!

The following is a list and functional description of all options offered.

@AXX - change size of X axis

Re-settable-Numerical Input card:R1,

R1 specifies the axis size in inches. The default X-axis length is 10 inches. If the X axis is specified longer than 10 inches, the plot if flipped about ninety degrees to allow unlimited axis length.

@AXY - change size of Y axis

Re-settable - Numerical Input card:R1

R1 specifies the axis size in inches. The default Y-axis length is 6.8 inches.

@BLK - Declare "IN" file monitor block

Not re-settable - Numerical input card: Not required

@CIR - Rend circular file, DS1 and 2

Re-settable-Numerical input card: Not required

Used only in VT#PLFOR, to read and decode the circular file rather than the 80-word table in DS7. Uses file code "CIR" assigned to DS1 by default.

@CLR - Return all re-settable options to default values

Numerical input card:Not required

@EXT - Extend buffer size to 6,000 points.

Re-settable - Numerical input card: Not required

Larger buffer enables more points to be plotted. The phase one process will slow down considerably, however.

@END - Supress End Code

Not re-settable - Numerical input card: Not required

Supress the Special End Code (Dual EOF's) placed on the tape at the job's conclusion. This allows one to stack several consecutive sets of plots on one tape

so that phase two will plot them all without halting in between.

@FLT - Filter input data

Re-settable - Numerical input card:TC(FP.Number)

Passes the data through a simple single stage digital filter before plotting. TC is the Time Constant (in units of the sampling period - e.g. minutes for one minute datasets), and it is printed on the plot.

@FRQ - Adjust frequency of marker symbols

Re-settable - Numerical input card:I1

By default, a special marker symbol is plotted on the graph every 200 points. This option allows one to plot the marker symbol every I1 points.

@LGX - Plot X-data on log scale

Re-settable - Numerical input card: Not required

This option is used in the general purpose programs. When specified it plots the X data logarithmically. The X-axis is labeled with engineering units in non-linear fashion.

Warning: When using this option, the X data must not go negative, or an "RT50" will be declared!

@LGY - Plot Y-Data on log Scale

Re-settable - Numerical input card: Not required

Same as "@LGX", but for the Y axis. Do not let the Y-data go negative, or an RT50 will be declared!

@PLT - Automatically activate Phase Two

Not re-settable - Numerical input card: Not required

This option causes the foreground phase two plotter to be activated at the conclusion of a run. The program "VTAUTOPT" is invoked, which grabs the tape drive #0 through no-wait allocation, rewinds it, and dumps commands to the printer/plotter. This option is extremely useful for overnight plotting runs, since it does not require the presence of someone to manually activate "VTPLOTER". Several precautions must be observed when using this option, however:

I) It cannot be used in conjunction with "@END".

II) The plotter must be turned on and ready.

III) The plotter must be initialized.

IV) The plot tape must be mounted on Drive #0.

See the section on Phase two for more detail.

@SCA - Use automatic Scaling

Re-settable - Numerical input card: Not required

Used only in VT#PLFOR, where manual scaling is the default. This option will cause the program to implement automatic max/min scaling.

@SCX - Manually scale X axis

Re-settable - Numerical input card: R1, R2

R1 and R2 are the maximum and minimum values of the X axis (order is non-critical). If the data exceeds these limits, the graph saturates against the axis.

@SCY - Manually scale Y axis

Re-settable - Numerical input card: R1, R2

Same as above, but for Y axis. In PL#FOR it allows one to set new limits for manual scaling, and in all other routines, it overrides the automatic scaling default.

@SIZ - Change size of plot

Re-settable - Numerical input card: R1

Allows one to set the size of the plot. R1 is a multiplicative factor, and its default is 0.9 (a half-size plot would require R1 = 0.45).

@SYM - Change Marker Symbol

Re-settable - Numerical input card: I1

I1 specifies the symbol to be used as a marker on the plot (default is the cross-hatched circle-I1=190). See enclosed symbol table to determine I1.

@WND - Scale about a "window"

Re-settable - Numerical input card: R1

Under this option, the average of the data is calculated, and it is scaled about a window of width R1. The scaling limits will then be: Max = Avg. + R1/2

Min = Avg. - R1/2

If the data exceeds these limits, the plot is saturated against the axis.

The following is a list of each program with corresponding options offered.

Program Name	Options Offered
VT#PLC&A	@AXX, @AXY, @BLK, @CLR, @END, @EXT, @FLT, @PLT, @SCY, @SIZ, @WND
VT#PLFOR	@AXX, @AXY, @BLK, @CIR, @CLR, @END, @EXT, @FLT, @PLT, @SCA, @SCY, @SIZ, @WND
VT#PLDED	@AXX, @AXY, @BLK, @CLR, @END, @EXT, @FLT, @PLT, @SCY, @SIZ, @WND
VT#PLGEN	@AXX, @AXY, @BLK, @CLR, @END, @EXT, @FLT, @FRQ, @LGX, @LGY, @PLT, @SCX, @SCY, @SIZ, @SYM, @WND
VT#PLMLT	@AXX, @AXY, @BLK, @CLR, @END, @LGX, @LGY, @SCX, @SCY, @SIZ

Step #5: Main Title Card

This card bears a twenty character, left-justified title which is printed on every plot. If no options were specified, this card follows step #3.

Step #6: Indexing Cards

Indexing cards contain at least one entry which specifies the location of the data to be plotted (a 16 character identifier is also included). These cards may be entered in any order and quantity desired. Since each routine employs a different indexing scheme, we'll deal with them separately.

a) C and A

Indexing into the C and A Table is primarily double precision. The three torque words are respectfully 1, 2 and 3, then the DP. index begins with 4,5, and 6 pointing to WPIG 1, 2 and 3,etc. Indices 10, 11, and 12 are not used, but the numbering again proceeds normally with 13 pointing to the 1'st transition matrix entry and 114 pointing to the last entry in the Covariance Matrix diagonal. If necessary, the integers skipped in this data set may be accessed by specifying an index of 115 through 120. This is a single precision index, with 115 pointing to CNTRLWRD, and 120 pointing to CWPNTR. These integers are plotted as is, and no conversions are attempted. The index specified into the

C and A Table is printed on the plot, and a message is printed whenever you access the State Vector estimate or standard deviation (covariance Matrix diagonal). [A state vector index ranging 1→57 is also included]

The index is specified on a card along with a 16 character title describing the item selected, which is printed on the plot. Some Sample C and A index cards are:

1,TORQUE VECTOR#1

22,X RESIDUAL

b) Formatted Data

The formatted data table is accessed using two indices, one ranging from 0 to 79 specifying the particular word in the table, and the other being a sub-index ranging 1 to 3, specifying the item desired in that word. The format of the indexing card is: INDEX, SUB-INDEX, TITLE where the title is a 16 character block printed on the plot. If the sub-index is specified greater than 3, all three items selected by the word index are plotted independently on 3 separate plots, with the specified title appearing on each. Both the word index and sub-index are also printed on every plot. Some sample indexing cards are:

65,1,SFIRJ1 20 VDC RN

74,3,TGG 3 20 VDC RTN

22,4,MTR PWR MON (Prints all 3)

c) Dedicated Data

The dedicated word index ranges 1→25. Value 3 points to Pump Pressure Sum, 4, 5, and 6 to Wheel Power Sums, 7 to OLDMAT, and 8 to fluid temperature. The 80 word Formatted Data Table is then skipped, and Index 9 points to the TOD, 10 to the SF attitude word, etc. When plotting attitude data, a sub index must be specified, pointing to the item (31 and 32 speed) in the selected word. If that index is greater than 2, both speeds are output on two separate plots. If the specified sub-index is negative, the attitude error is calculated and another card is input containing a real number specifying the rate used. When selected, messages are printed on the plot showing Attitude type (phi or Theta), speed, Error flag, and rate (if calculated). An index is also printed on the plot, but this index includes the formatted data Table, and ranges 1→105. The format of the indexing card is: INDEX, TITLE SUBINDEX, [Sub-index begins in column 20, and is followed by a comma].

The sub-index need not be specified if attitude data is not selected. If a sub index greater than two is included with any item other than attitude data, two plots of the selected item are produced. The title, of course, is 16 characters long, and appears on the plot. Same sample indexing cards follows:

```
3,PUMP PRESSURE
11,31 SPEED PHI ATT 1,
12,PHI ATT, WORD#2 3,(Plots both speeds)
14,THETA ATT. #1 ERR-3,(Calculates error for
                        both speeds)
15.5                   (Rate card)
17,DC MATRIX #1 3,     (Makes two plots of
                        the same item)
```

d) General plot program

The generalized plotter, VT#PLGEN, uses only one index which is the pointer into the range buffer used in the "DSKWRITE" call (see next section for details). i.e. index 1 points to YVALS(1), index 2 points to YVALS(2), etc. Sample indexing cards would be:

```
1, FIRST PLOT
50, PLOT #50
```

DATA SET INDEXING

1) C and A: Data set #4

Item	Index
Torque Command Words	1→3
SFIR Head Rate (WPIG)	4→6
Platform Rate (WPIP)	7→9
C and A Control Word	115
CWPNTR	↓ 120
Transition Matrix	13→21
Measurement Residual	22→24
State Vector	25→81
Covarience Matrix Diagonal	82→114

2) Formatted Data: Data set #7

Item	Index	Sub-Index
SFIR-J1 Minor Loop Null Mon SFIR-J2 Minor Loop Null Mon SFIR-J3 Minor Loop Null Mon	0	1 2 3
SFIR-J1 Servo Lag Output Mon SFIR-J2 Servo Lag Output Mon SFIR-J3 Servo Lag Output Mon	1	1 2 3
	↓	
Valve 1B Position Mon Valve 2B Position Mon Valve 3B Position Mon	79	1 2 3

3) Dedicated Data: Data set #7

Item	Index	Sub-Index (if applicable)
MPC at 1 Min.	1	———
ID.=7	2	———
Pump Pressure Sum	3	———
Wheel Power Sums	4→6	———
OLDMAT	7	———
Fluid Temperature	8	———
	80 Word Formatted Data Table is skipped	
Time Of Day	9	———
Super-Fine Attitude Word	10	1=31 speed, 2=32 speed
Ø Attitude angles	11→13	1=31 speed, 2=32 speed
Θ Attitude angles	14→16	1=31 speed, 2=32 speed
Attitude Direction Cosine Matrix	17→25	———

Step #7: 999

This is one card with the number "999" left justified. Its only purpose is to inform the program that the current run segment has ended, and no more indexing cards are to be read.

Step #8: (MORE/END) card

This is one card which specifies whether to terminate the job, or to return to step #3 and begin a new run segment. If this card bears the phrase "MORE", left justified, control is passed back to step #3, and a new "begin-stop-skip" card is read. (All options declared earlier are still pending. They may be removed via "@CLR", or others may be declared). If the card contains the phrase "END", however, the program is terminated, and an "$EOJ" is read.

Step #9: $EOJ
$$

These are conventional RTM directives which terminate a batch job. They should be appended to the end of the deck, after the "END" card.

C) Notes on Plot Format

Each program calculates the actual maximum, the average, and the standard deviation over the presented domain (unaffected by any scaling conditions). This data is printed at the bottom of every plot. A "plot number" is also included on the plots, this refers to the sequential order of plots within a run. All plots also print the Time (Hour, minute, second) taken from the computer clock when the plot was made. The begin and stop records are not printed, but the horizontal axis is labeled in MPC minutes. The record skip factor (ISF) is not printed, but small circular marks are made on the graph after every 200 points are plotted (assuming the @FRQ option has not been specified), so, observing the density of these marks, the amount of points skipped is easily estimated. When plotting attitude data, the validity flag is always checked, and if it is not set, the "pen" is lifted and the "bad" points are not plotted. If, for some reason, there are no valid points found within the specified domain, the message "NO VALID POINTS" is printed and the next item is plotted.

In C and A and Dedicated Data, several different conversions are applied to the data in order to plot in engineering units. The units themselves are printed on the axes (as labels) wherever possible. All items in Formatted Data are converted to and plotted in VOLTS. No scaling factors are used in VT#PLGEN, all variables are plotted just as they appear in the Call to DSKWRITE. VT#PLGEN uses axis titles specified in the call to HEADER.

III. GENERAL PLOTTING

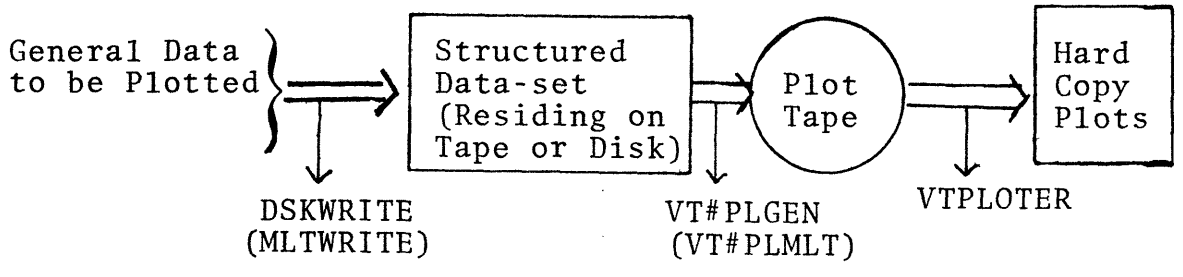
A) Introduction: The general plot software residing on the SEL-32 allows one to plot generalized data on the Versatek by writing a simple preformatting routine. The user is thus protected from the programming horrors inherent with conventional plotter software.

There are two generalized plotter programs on the system:

VT#PLGEN - One graph per frame only. Setup is quite analogous with other plot programs.

VT#PLMLT - Up to ten graphs per fram. Setup is somewhat unique.

These programs read datasets specially created via the DSKWRITE/HEADER/FINISH or MLTWRITE/BEGIN/FINISH subroutines, process this data, and create a phase two plotter tape for "VTPLOTER", as depicted below:



B) Plotting with VT#PLGEN and DSKWRITE: In order to pre-format your data for plotting with VT#PLGEN, the DSKWRITE/HEADER/FINISH subroutine package must be used. (To access these routines, use: $ASSIGN1 LIB=VT%LIBRY DIR=VT%DIRCT before an $EXECUTE GO or CATALOG).

The formatting routines must be initialized, and plotter parameters must be set via a call to HEADER before anything is attempted. If DSKWRITE or FINISH are called before HEADER, your program will be aborted with a "NOINIT" or IO17. The HEADER call line appears as:

```
CALL HEADER (MXT,MYT,IP,IDEV)
```

where:

MXT is an eight character ASCII string which is used as an X-axis label. (MXT usually defines units such as TIME, MPCMIN, etc.)

MYT is an eight by N array of ASCII characters which are used as Y-axis labels. Usually one will declare MYT as: INTEGER*8 MYT(N), where N denotes the number of range variables in the plotter dataset. (There is one Y-axis title paired with each range variable). This array is usually initialized in a DATA statement.

IP denotes the number of range values paired with the domain value (analogous to "N" above). If IP is negative, the program will set the "lift pen" flag, and range values created through "ALIFT" will not be plotted. (This feature is valuable when plotting attitude data, where one wishes to supress "bad" data points.) The maximum number of variables in an unblocked dataset should be kept under 90. (For a blocked dataset the maximum is 20).

IDEV denotes the device on which the structured dataset created by DSKWRITE will be output:

IDEV=1: Data output to magnetic tape "PLTO".

IDEV=2: Data output to disk file "SCRATCH2"

IDEV=3: User must manually assign the file code "PT" to a device.

If IDEV is negative, M:BLOCK is called, and the output file is declared to monitor blocked. In this case the "@BLK" option must be specified when running "VT#PLGEN". (This is strongly recommended when outputing small datasets to a disk file such as SCRATCH2, which has a maximum capacity of 600 blocks).

After Header is called, plotting via DSKWRITE may begin. Whenever DSKWRITE is called, a formatted record is output. The call line to DSKWRITE appears as:

```
CALL DSKWRITE(SVAL,YVALS)
```

where:

XVAL is the current floating point domain value (X-axis)

YVALS is an array of N words (where N=ABS(IP)), specifying the floating point range values (Y axis) paired with XVAL.

If "IP" in the call to HEADER was specified negative, the "pen" may be lifted, allowing us to skip data or re-position the "pen" dynamically without leaving a telltale line. To life the pen for the graph of Y(I), do:

```
Y(I) = ALIFT(0) -- The Zero is used as a dummy argument
  CALL DSKWRITE(X,Y)
    (Y(I) is an array element)
```

The "pen" will be dropped again upon a subsequent normal DSKWRITE call.

To conclude plotting, the FINISH routine is invoked. It is called without arguments:

```
CALL FINISH
```

FINISH writes an EOF on the output device, and dumps a count of the number of records created via DSKWRITE to the "LO" device. (The file code "LO" must be assigned to the SLO before FINISH is called, or the job may abort with an IO 17).

The general structure of a preformatting routine will be:

```
    (Initialization)
    CALL HEADER(---)
        .
        .
        .
┌   (Loop in which data
|   is created or converted)
|       .
|       .
|       .
|   CALL DSKWRITE(---)
└       .
        .
    (Concluding statements)
        .
        .
        .
    CALL FINISH
     END
```

The program VT#PLGEN may be invoked to plot the structured DSKWRITE Datasets. The index specified to VT#PLGEN denotes the entry in "YVALS" which is to be plotted. (i.e. an index of 1 plots YVALS(1), 2 plots YVALS(2), etc.) The operation of VT#PLGEN was discussed earlier in Section II.

C) Plotting with VT#PLMLT and MLTWRITE

This package allows the user to overlay up to ten graphs on one set of axes. The pre-formatting package consists of routines MLTWRITE/BEGIN/FINISH, and resides in VT%LIBRY.

The pre-formatting package must be initialized via a call to BEGIN before plotting starts, or the program will abort with a "NOINIT" or "IO17". The BEGIN call line appears in the form:

```
CALL BEGIN (MXT,MYT,IDEV)
```

where:

MXT is an 8-character ASCII string containing the X-axis title.

MYT is an 8-character ASCII string containing the Y-axis title (There is only one!)

IDEV specifies the output device. See the discussion of "HEADER" in Section A for details.

Plotting via "MLTWRITE" may begin after BEGIN has been invoked. Its call line is:

```
CALL MLTWRITE(XVAL,YVAL)
```

where:

XVAL is the floating point X-axis value for the current graph.

YVAL is the floating point Y-axis value for the current graph (not an array).

When a graph is finished, and we wish to plot another over the same set of axes, the routine "CHANGE" is called. CHANGE requires no arguments, and is called:

```
CALL CHANGE
```

Subsequent calls to MLTWRITE will plot a new graph overlaid upon the old one. Up to ten graphs may be specified (at a limit of 6,000 points per graph), thus CHANGE may be called up to nine times.

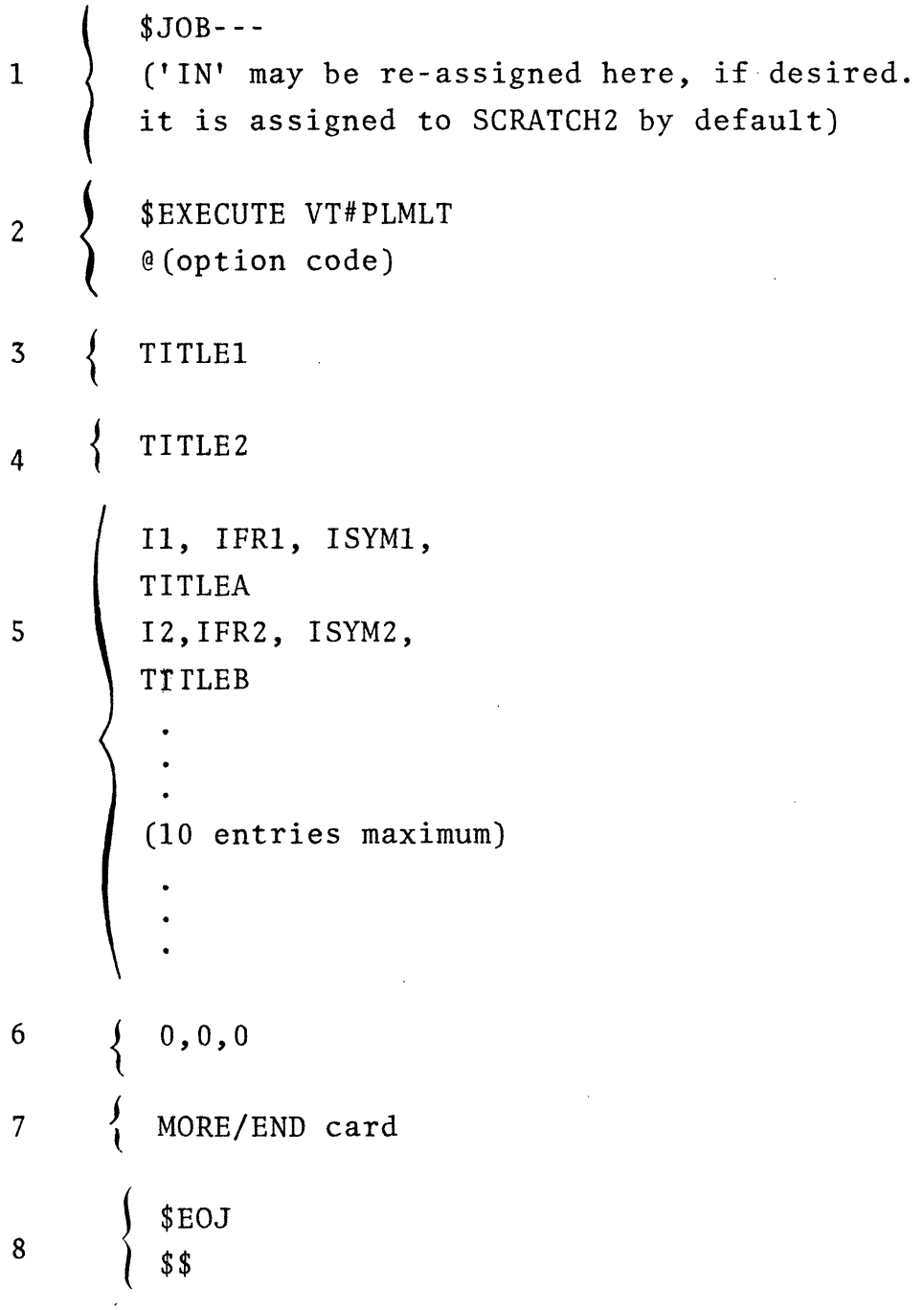
To conclude plotting, FINISH is called to write an EOF on the output device, and dump a final record count to the printer (make sure that the "LO" file code is assigned!). It requires no arguments:

```
CALL FINISH
```

The general structure of a MLTWRITE pre-formatting routine will be:

```
(Initialization)
CALL BEGIN(---)
┌Loop in which data
│for first graph is
│created
└CALL MLTWRITE(--)
CALL CHANGE
┌Loop in which data
│for second graph
│is created
└CALL MLTWRITE(--)
CALL CHANGE
┌ :
└ :  and so on
CALL FINISH
END
```

Datasets created via MLTWRITE are plotted via the program VT#PLMLT. Its execution Deck is setup as shown:



1 - $JOB--

Starts a job under RTM. "IN" may be re-assigned via an $ASSIGN if desired.

2 - @(option code)

Options in VT#PLMLT may be specified if desired. See Section II for more detail.

3 - TITLE1

This is a 20 character title which is printed on the plot.

4 - TITLE2

This is a 16 character title which is printed on the plot.

5 - I1, IFR1, ISYM1

TITLEA

.
.
.

This is a repeating two-card sequence which defines the format of each graph. A pair of cards is included for each graph in the plot.

I1 is an integer which specifies the order of the graph. I1 should start at one and increase sequentially.

IFR1 is an integer which specifies the repetition frequency of marker symbols. (i.e. IFR=50 → symbol drawn every 50 points,etc.) If IFR1 is positive, a visible line is drawn between the marker symbols giving us a "solid" graph. If IFR1 is negative, no line will be drawn between marker symbols, giving us a "point" plot.

ISYM1 is an integer which specifies the symbol to be used as a marker. The symbol table in the appendix lists marker symbols and their associated integer values.

TITLEA is a 16 character title which is used on the plot "legend". This "legend" is a Table which VT#PLMLT prints below the plot which pairs these titles with the marker symbols used to denote the appropriate graph (i.e. "* = TEMPERATURE", etc.).

6 - 0,0,0

This card is analogous to the "999" card used in the other plot programs. It informs the software that no more format specification cards are to be read.

7 - MORE/END

This card serves the same function as its counterpart in the other plot programs. If "MORE" is specified, the software skips back to step#2, reads more options, new titles, and starts again. If "END" is specified,program execution is terminated, and we progress to step#8.

8 - $EOJ

$$

These cards terminate a job under RTM.

IV. PHASE TWO PLOTTING AND PRINTING

A) The Plotting Versatek

All phase-one plot packages described earlier produce "plot tapes" which contain, in binary form, the virtual image of the created plots. This tape must be dumped to the Versatek in order to obtain hard copy. This may be accomplished in either of two ways; by specifying the "@PLT" option, or by using the foreground plotter "VTPLOTER".

The "@PLT" option was described earlier in section II. It enables the background Phase I programs to automatically activate the foreground Phase II plotter "VTAUTOPT". This routine grabs tape drive#0 (make sure that the plot tape has been mounter on drive #0, or else...) via no-wait dynamic allocation, rewinds it, and dumps the plots to the Versatek. This feature can be very useful in making "overnight" plot runs, since it doesn't require anyone present to manually activate "VTPLOTER" (occasionally, when core is filled up, the background phase I program will be rolled out to disk upon activating VTAUTOPT. This stops Phase I before program termination, and locks up background until VTAUTOPT is finished. If this occurs, don't get concerned, it's not a serious problem).

One may also use VTPLOTER to dump plot tapes to the Versatek. Merely rewind the plot tape, key into the teletype "ACTIVATE VTPLOTER", respond to the mount message with an "R", and let

it rip. One may skip plots on the tape, if desired, by skipping files. Merely use the skip file directive in MEDIA to position the tape (do not declare it blocked!). There is an EOF after each plot, so skipping N files skips N plots. VTPLOTER may be activated after the tape has been positioned, and plotting will begin at that point.

The phase two plotters are low-priority foreground jobs, and they keep background free for other usage. They continue dumping plots to the Versatek until a special "end code" is detected on the tape, at which time they automatically issue the "dismount message" and terminate normally.

NOTE#1: Before using these programs, one must be sure that the plotter has been initialized! In order to accomplish this, hit "ATTENTION" on the console, and key into the teletype the sequence:

INITIALIZE @T66,B0000000,00

The plotter is now initialized. If the plotter had not been initialized, it would start spewing out paper filled with commas at the rate of 1.5 feet per second as soon as

the program began execution. If this happens, power down the plotter, abort the program (either VTPLOTER or VTAUTOPT), power the plotter back up, wait for the abort message, initialize the plotter, rewind the plot tape, and activate VTPLOTER.

NOTE#2: These programs allocate temporary files on disk. If they start aborting with a "NOALOC" message, they are unable to allocate these files. This indicates problems with the disk allocation map, and may be remedied by either re-booting or, if unsuccessful, performing a cold-start sysgen.

B) The Printing Versatek

The Versatek may also be used as an auxiliary line printer. In order to accomplish this, all printer output must be loaded onto magnetic tape. This is generally done by an: $ASSIGN3 LO=MT,TAPE before the $EXECUTE card, since "LO" is the listed output file code used in most data products and system programs. An EOF must be written on the tape at the end of the text. This can easily be accomplished via MEDIA. Any tape thus created can be dumped to the Versatek. Merely mount this tape, key in "ACTIVATE VT#PRINT" (make sure that the Versatek has been initialized!), respond to the mount message, and let it run. The Versatek prints faster in this mode than the line printer now used on the SEL.

V. APPENDICES

A) Sample execution decks for VT#PLC&A, VT#PLFOR, VT#PLDED, VT#PLGEN

B) Sample pre-formatting programs using DSKWRITE and their execution setup in VT#PLGEN.

C) A Sample pre-formatting program using MLTWRITE and its execution setup for VT#PLMLT.

D) Listing of the Symbol Table

E) Breakdown of "VT%LIBRY"

F) Listing of files in the "VT" area.

Sheet #1

A) Sample Execution decks for VT#PLC&A, VT#PLFOR, VT#PLDED, VT#PLGEN

```
$JOB PLOT C&A DATA ON VERSATEK -SAMPLE# 1
$EXECUTE VT#PLC&A
0,1500,1,
MAIN TITLE
1,TORQUE WORD#1
4,WPIG#1
22,X RESIDUAL
25,STATE VECTOR#1
999
END
$EOJ
$$
```

This deck illustrates a simple run of the C&A plotter. Records 0⟶1500 are plotted, and every point is sampled. No options are declared, everything is left to default settings. Entries 1,4,22, and 25 into C&A are plotted (4 plots are made). The input data is read directly from DS4 on disk.

Sheet #2

```
$JOB PLOT C&A DATA ON VERSATEK -SAMPLE# 1
$ASSIGN3 IN=MT,DATA
$EXECUTE VT#PLC&A
0,1500,1,
MAIN TITLE
1,TORQUE WORD#1
4,WPIG#1
22,X RESIDUAL
25,STATE VECTOR#1
999
END
$EOJ
$$
```

This deck is identical to the previous sample, except that input data is read from magnetic tape (the $ASSIGN3 card does this). The plotting programs should require no "skip file" directives, they automatically space to the proper point on the data tape.

```
$JOB PLOT C&A DATA ON VERSATEK -SAMPLE# 2
$EXECUTE VT#PLC&A
1000,5000,1,
@EXT
@AXX
15.0
MAIN TITLE
1,TORQUE WORD#1
999
END
$EOJ
$$
```

This deck illustrates a C&A plotter run with options declared. The "@EXT" option extends the buffer capacity to 6,000 points (as opposed to the default of 2,000 points). The "@AXX" option increases the size of the X axis to 15 inches. Only one plot (the first torque word) is made.

Sheet #3

```
$JOB PLOT FORMATTED DATA ON THE VERSATEK -SAMPLE#3
$EXECUTE VT#PLFOR
0,3000,2,
@FLT
15.
MAIN TITLE
65,1,SFIRJ1 20VDC RTN
22,3,MTR PWR MON.
2,4,SERVO LAG OUT MON
999
END
$EOJ
$$
```

This deck shows a formatted data run with the filter option declared (a 15 minute time constant is specified). Note that all three triplets are plotted for the last entry (Index=2), since the sub-index =4.

```
$JOB PLOT FORMATTED DATA ON THE VERSATEK -SAMPLE#3
$EXECUTE VT#PLFOR
1000,2500,1,
@SCA
MAIN TITLE#1
22,4,MTR PWR MON.
999
MORE
1000,2500,1,
@CLR
@SCY
3.0,-3.0
MAIN TITLE#2
22,4,MTR PWR MON.
999
END
$EOJ
$$
```

This deck is broken into two portions. In the first part, the automatic scaling option is specified ('@SCA') to override the manual ±5.2 volt default scaling. The "MORE" directive begins a new run in which we clear the automatic scaling option via "@CLR", and impose manual ±3.0 volt Y axis scaling. Both run segments make three plots each (sub-index=4) and plot records 1,000 thru 2,500.

Sheet #4

```
$JOB PLOT DEDICATED DATA ON THE VERSATEK -SAMPLE#4
$EXECUTE VT#PLDED
50,900,1,
MAIN TITLE
3,PUMP PRESSURE
4,WHEEL PWR.#1
11,PHI  ATTITUDE#1  1,
14,THE1 ATTITUDE#1  2,
15,THET ATTITUDE#2  3,
12,PHI RATE ERROR#2 -1,
15.0
8,FLUID TEMP.
999
END
$EOJ
$$
```

This illustrates a sample run of the Dedicated Data plotter without options declared. Pump pressure, fluid temperature, and wheel power Sum #1 are plotted. 31 speed attitude is plotted for index 11, while 32 speed is plotted for index 14. Both 31 and 32 speed attitude is plotted for index 15 (Sub-index=3). The attitude error is calculated for index 12 (31 speed), and a 15 minute rate is input. All in all, 8 plots are made.

```
$JOB PLOT EG-SFIR DATA ON THE VERSATEK - PHASE ONE
$EXECUTE VT#PLGEN
0,1725,1,
@BLK
07-14,15-77 TEST 3109
1,MAGNITUDE OF Z
2,Z1
3,Z2
4,Z3
999
NOMORE
$EOJ
$$
```

This is a simple run of the general plotter VT#PLGEN. The input is assigned to the file SCRATCH2 by default, and the "@BLK" directive declares it to be monitor blocked. The"NOMORE" command is analogous to "END", both can be interchanged.

B) Sample pre-formatting programs using DSKWRITE and their execution setup in VT#PLGEN.

1. Program to format the GP Filter dataset using DSKWRITE.

2. Execution deck to plot the data created in part 1.

3. Sample program to format a SINC function using DSKWRITE.

4. Execution deck to plot the SINC function created in part 3 and exercise plotter options.

5. Re-prints of actual plots created in part 4.

Sheet #5

```
C      PROGRAM TO FORMAT THE GPFILTER DATASET FOR VERSATEK PLOTTING
C
C       INPUTS.........
C            1-25 Y-AXIS TITLES
C
       INTEGER IBUF(192),QBUF(75)
       INTEGER*8 NTIT(75)
       I=0
5      K=3*I+1
       READ ('SI',10,END=20)NTIT(K)
10     FORMAT (2A4)
       DO 1 N=1,2
1      NTIT(N+K)=NTIT(K)
       I=I+1
       GOTO 5
20     J=I-1
       CALL HEADER ('MPC SEC.',NTIT,3*J,1)
25     CALL BUFFER IN ('IN',1,IBUF,192,IST,NWD)
       CALL M:WAIT ('IN')
       CALL STATUS ('IN',IST)
       IF (IBUF(2).NE.17) GOTO 25
       IREC=0
35     IK=0
       DO 30 I=1,J
       DO 30 II=0,2
       IX=I+25*II+2
       IK=IK+1
30     QBUF(IK)=IBUF(IX)
       RBUFF=IBUF(1)/100.
       CALL DSKWRITE (RBUFF,QBUF)
       IREC=IREC+1
       CALL BUFFER IN ('IN',1,IBUF,77,IST,NWD)
       CALL M:WAIT('IN')
       CALL STATUS ('IN',IST)
       IF (IST.NE.3) GOTO 35
       CALL FINISH
       PRINT 50,IREC
50     FORMAT (1X,'TOTAL# OF RECORDS =',I7)
       STOP DONE
       END
```

This program reads GP filter data (file code=IN) re-formats it, packs it into the array QBUF, and outputs it via DSKWRITE. Y-axis titles are read from the "SI" file. The "1" in the call to HEADER specifies that the formatted output is to be routed to magnetic tape in unblocked form.

Sheet #6

```
$JOB PLOT GPFILTER DATA ON VERSATEK
$ASSIGN3 IN=MT,DATA
$EXECUTE VT#PLGEN
200,1560,1,
06-22-77 TEST 3099
1,THETA 1 EVEN(2)
2,THETA 1 EVEN(1)
4,THETA 2 EVEN(2)
7,THETA 3 EVEN(2)
10,THETA 1 ODD(2)
13,THETA 2 ODD(2)
999
NOMO
$EOJ
$$
```

This deck uses VT#PLGEN to plot the data created via DSKWRITE (note the $ASSIGN3 assigning the input to magnetic tape). Indices 1,2,4,7,10, and 13 into the buffer QBUF are plotted.

Sheet #7

```
$JOB TEST OF DSKWRITE/GENPLOT
$OPTION 5
$EXECUTE FORTRAN
C           THIS IS A SAMPLE PROGRAM WHICH DRAWS A SINC FUNCTION
C           BY USING A CALL TO DSKWRITE
C
C           THIS CALL TO HEADER GIVES OUR PLOT THE X-AXIS TITLE 'X-AXIS'
C           AND GIVES THE Y-AXIS TITLE 'YAXIS'
C           WE ALSO DECLARE ONLY ONE VARIABLE TO BE PLOTTED, AND DIRECT
C           THE OUTPUT TO 'SCRATCH2' IN BLOCKED FORMAT.
C           (NO $ASSIGN CARDS ARE NECESSARY)
            CALL HEADER ('X-AXIS','Y-AXIS',1,-2)
C
C           THIS LOOP CREATES THE ACTUAL DATA.  WE CREATE 5001 ACTUAL
C           DATA POINTS, AND OUTPUT THEM VIA 'DSKWRITE'.
C
            DO 5 I=-2500,2500
            R=I*3.1416/625.
            Y=SIN(R)/R+1.0
            CALL DSKWRITE (R,Y)
5           CONTINUE
C
C           THE CALL TO 'FINISH' WRITES AN EOF ON SCRATCH2, AND CONCLUDES
C           PLOTTER OUTPUT
            CALL FINISH
C
            STOP
            END
$OPTION NOMAP
$ALLOCATE 10000
$ASSIGN1 LIB=VT%LIBRY
$ASSIGN1 DIR=VT%DIRCT
$ASSIGN2 LO=SLO,100
$EXECUTE GO
$EOJ
$$
```

This job will create 5,001 DSKWRITE formatted records on SCRATCH2 describing a SINC function.

Sheet #8

```
$JOB PLOT SINC FUNCTION ON THE VERSATEK - TEST OUT OPTIONS
$EXECUTE VT#PLGEN
0,5500,1,
@CLR
NORMAL PLOT
1,SINC FCN
999
MORE
0,5500,1,
@SCX
-6.28,6.28
X AXIS SCALING
1,SINC FCN
999
MORE
0,5500,1,
@CLR
@SCY
1.,1.5
Y AXIS SCALING
1,SINC FCN
999
MORE
0,5500,1,
@CLR
@WND
0.3
WINDOWED PLOT
1,SINC FCN
999
MORE
0,5500,1,
@CLR
@SYM
28,
@FRQ
50,
CHANG OF SYM. AND FRQ.
1,SINC FCN.
999
MORE
0,5500,1,
@CLR
@SIZ
0.5
SMALL PLOT
1,SINC FCN
999
MORE
0,5500,1,
@CLR
@AXY
2.0
SMALL Y-AXIS
```

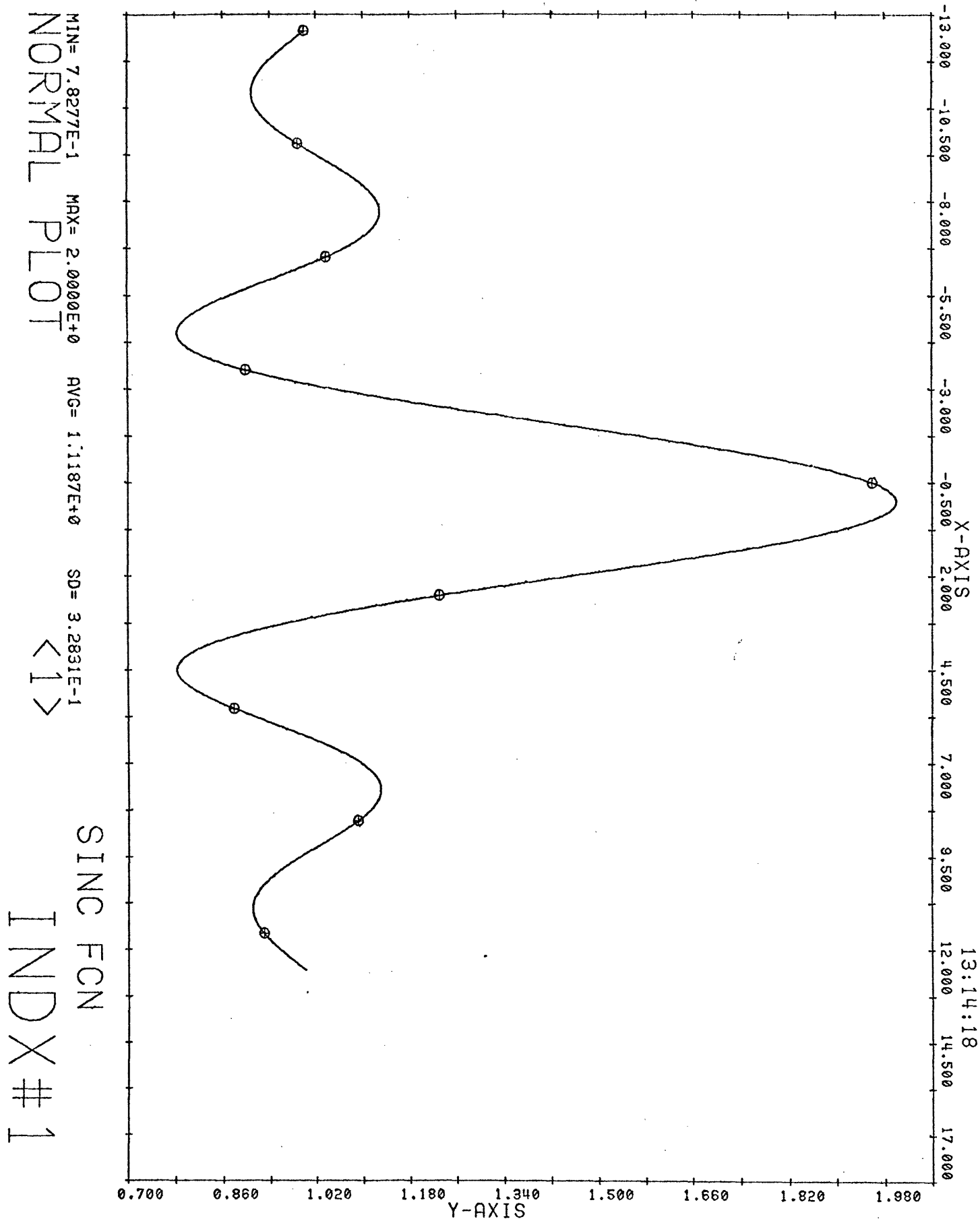
Sheet #9

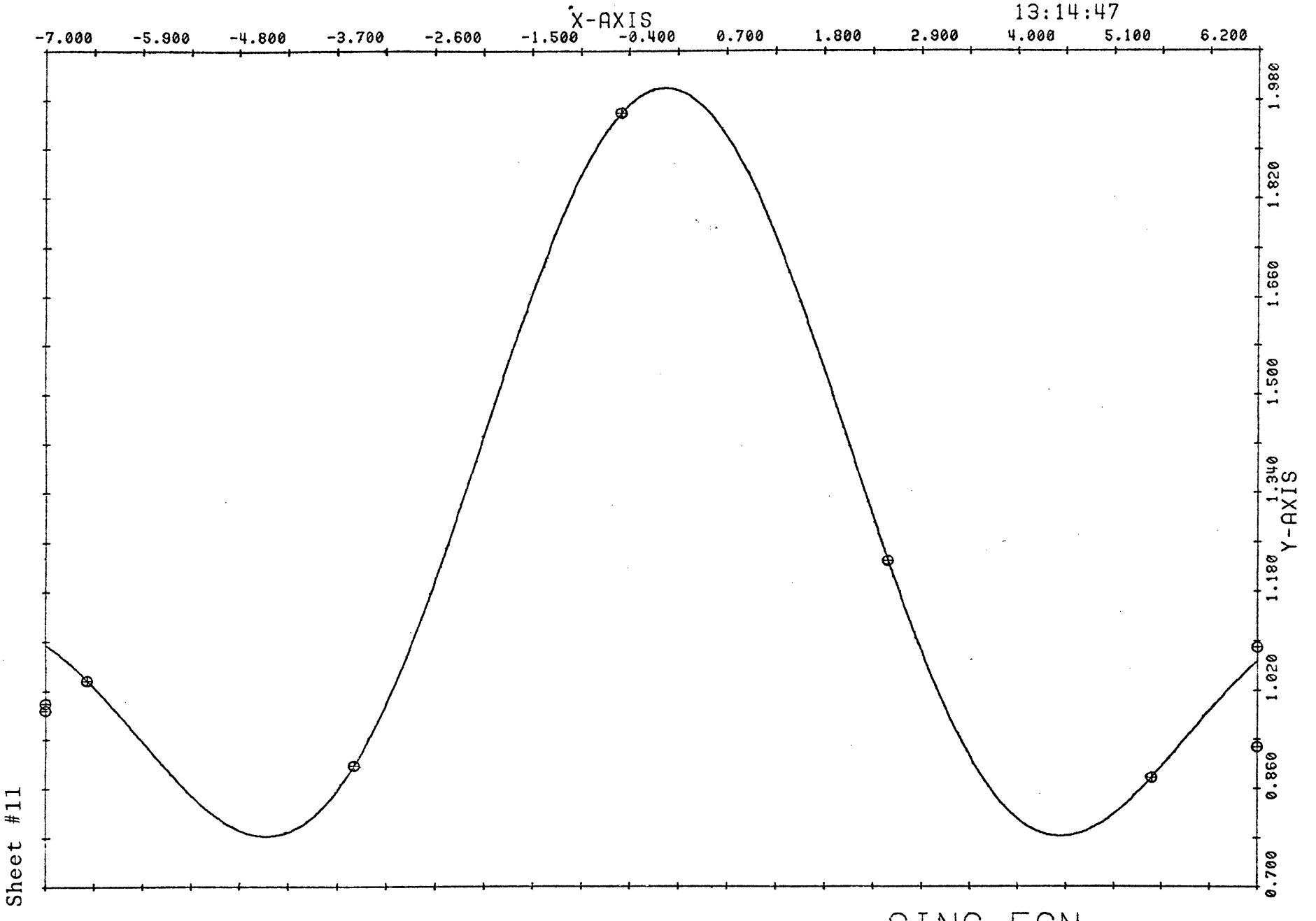
```
1,SINC FCN
999
MORE
0,5500,1,
@CLR
@AXX
5.5
@AXY
5.0
SQUARE PLOT
1,SINC FCN
999
END
$EOJ
$$
```

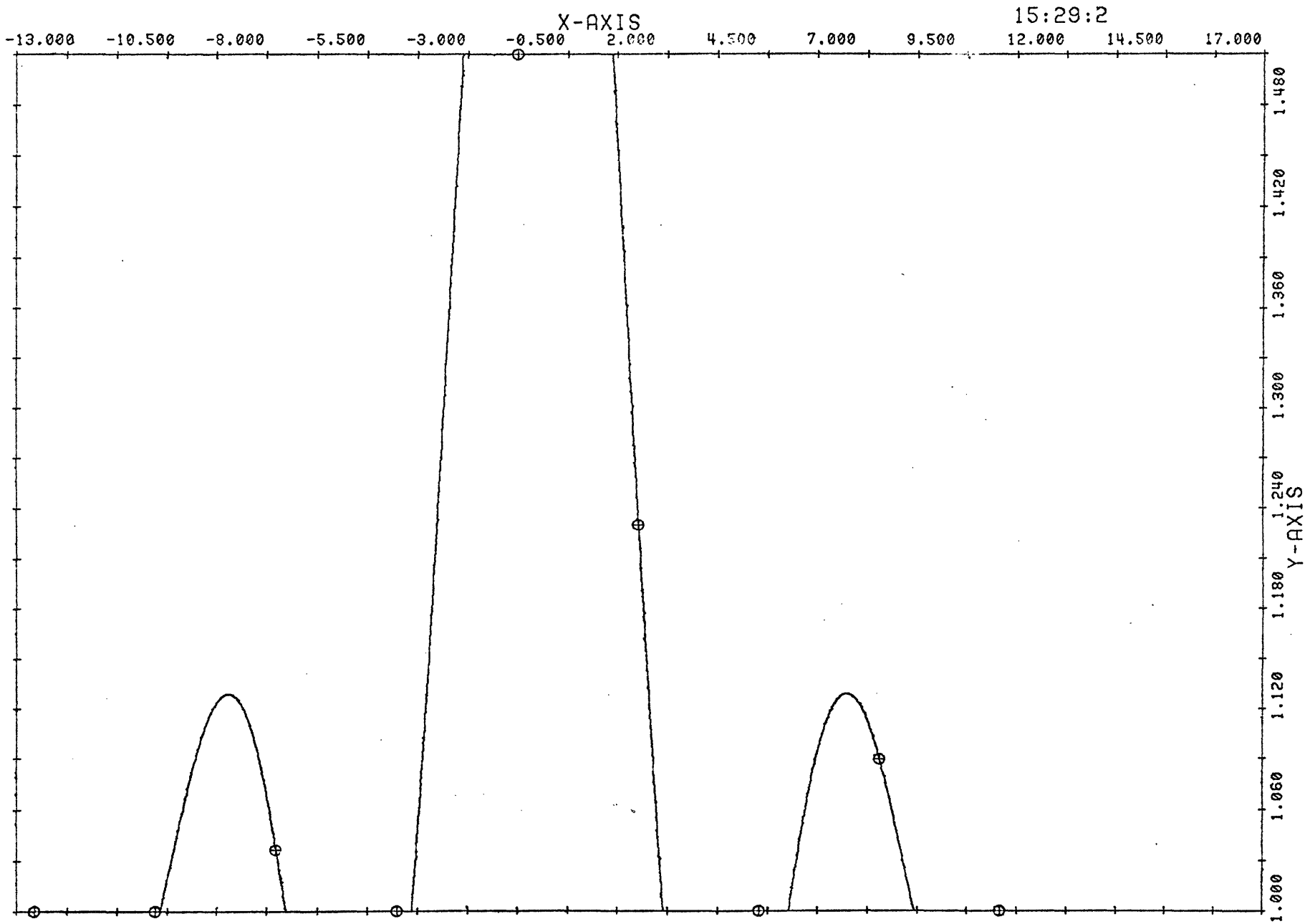
This deck will plot the SINC function which has been created on SCRATCH2 by the previous program. A total of 8 plots are made to illustrate the operation of various options. Copies of the plots made via this job are included.

1 - Normal plot: No options declared
2 - X axis scaling: X axis is scaled to $\pm$ 6.28. Points outside this boundary are saturated on the axes.
3 - Y axis scaling: The Y axis is scaled between 1 and 1.5. All points outside this boundary are saturated on the axes.
4 - Windowed plot: The Y axis is scaled through a "window" of $\pm$ 0.3 units about the average. All points lying outside this "window" are saturated on the axes.
5 - Change of symbol and frequency: The default marker symbol, a circle (code=190) is re-defined to be a cross (code=28). The symbols are drawn every 50 points, as opposed to the default interval of 200 points.
6 - Small plot: A half-size plot is made.
7 - Small Y-Axis: A two-inch Y axis is drawn.
8 - Square plot: Both the X and Y axes are altered to draw an almost "square" plot.

Sheet #10


NORMAL PLOT
MIN= 7.8277E-1 MAX= 2.0000E+0 AVG= 1.1187E+0 SD= 3.2831E-1
<1>
SINC FCN
INDX#1
Y-AXIS
0.700 0.860 1.020 1.180 1.340 1.500 1.660 1.820 1.980
X-AXIS
-13.000 -10.500 -8.000 -5.500 -3.000 -0.500 2.000 4.500 7.000 9.500 12.000 14.500 17.000
13:14:18


X-AXIS
13:14:47
-7.000
-5.900
-4.800
-3.700
-2.600
-1.500
-0.400
0.700
1.800
2.900
4.000
5.100
6.200
Y-AXIS
1.980
1.820
1.660
1.500
1.340
1.180
1.020
0.860
0.700
Sheet #11
MIN= 7.8277E-1 MAX= 2.0000E+0 AVG= 1.1187E+0 SD= 3.2831E-1
X AXIS SCALING
<1>
SINC FCN
INDX#1



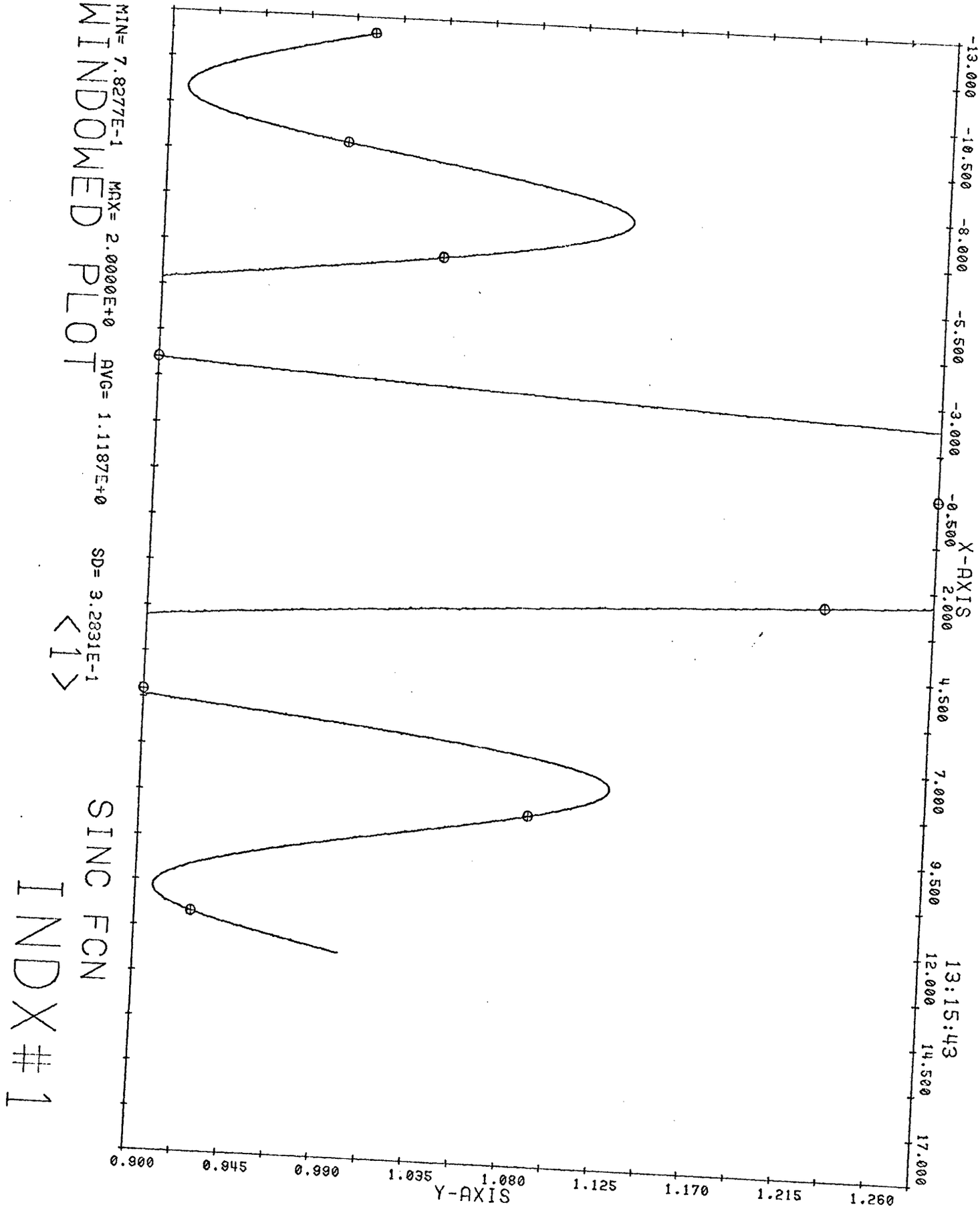
Sheet #12

MIN= 7.8277E-1 MAX= 2.0000E+0 AVG= 1.1187E+0 SD= 3.2831E-1

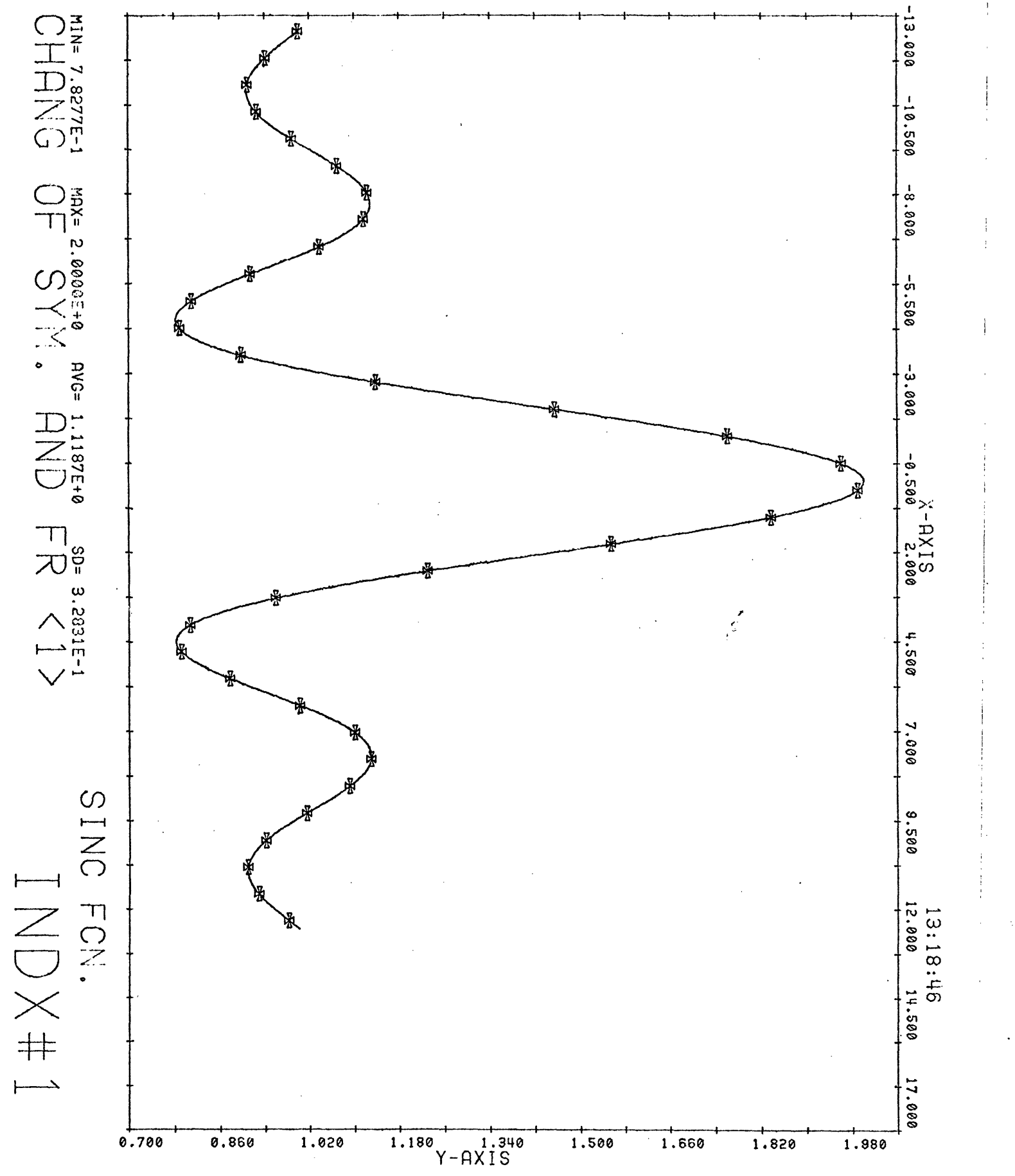
SINC FCN

Y AXIS SCALING <1> INDX#1

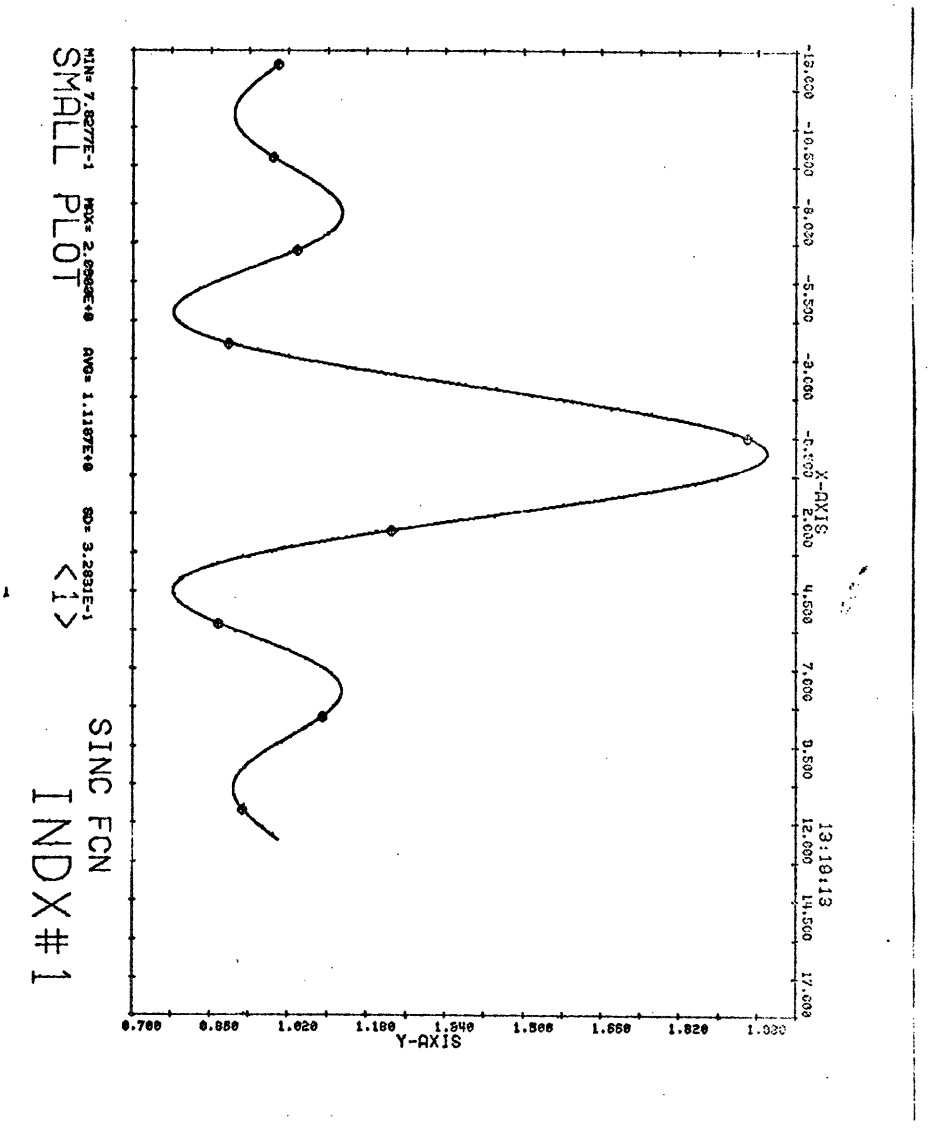
Sheet #13

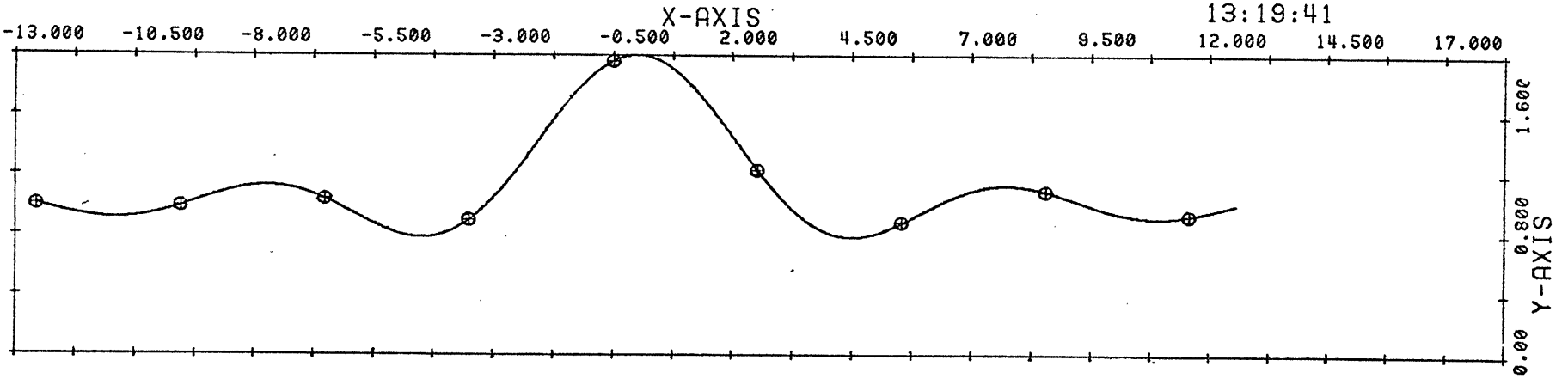

WINDOWED PLOT
MIN= 7.8277E-1
MAX= 2.0000E+0
AVG= 1.1187E+0
SD= 3.2831E-1
<1>
SINC FCN
INDX #1
13:15:43
X-AXIS
-13.000
-10.500
-8.000
-5.500
-3.000
-0.500
2.000
4.500
7.000
9.500
12.000
14.500
17.000
Y-AXIS
0.900
0.945
0.990
1.035
1.080
1.125
1.170
1.215
1.260

Sheet #14


CHANG OF SYM. AND FR <1>
MIN= 7.8277E-1 MAX= 2.0000E+0 AVG= 1.1187E+0 SD= 3.2831E-1
SINC FCN.
INDX#1
Y-AXIS
0.700 0.860 1.020 1.180 1.340 1.500 1.660 1.820 1.980
X-AXIS
-13.000 -10.500 -8.000 -5.500 -3.000 -0.500 2.000 4.500 7.000 9.500 12.000 14.500 17.000
13:18:46

Sheet #15


SMALL PLOT
MIN= 7.8277E-1 MAX= 2.6982E+0 AVG= 1.1197E+0 SD= 3.2831E-1
<1>
SINC FCN
INDX#1
-13.000 -10.500 -8.000 -5.500 -3.000 -0.500 2.000 4.500 7.000 9.500 12.000 14.500 17.000
X-AXIS
0.700 0.850 1.020 1.180 1.340 1.500 1.660 1.820 1.980
Y-AXIS
13:10:13



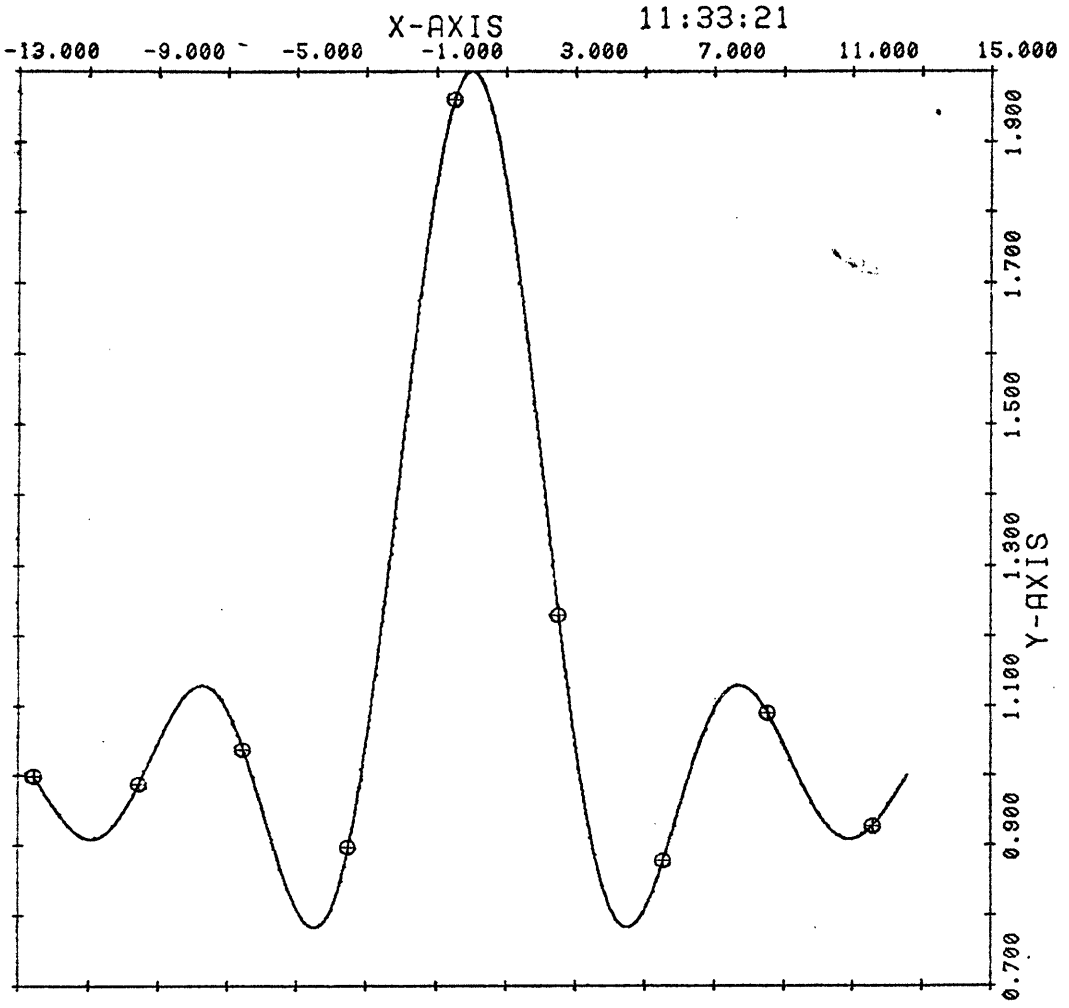
MIN= 7.8277E-1 MAX= 2.0000E+0 AVG= 1.1187E+0 SD= 3.2831E-1

SMALL Y-AXIS <1>

SINC FCN

INDX#1

Sheet #16



MIN= 7.8277E-1 MAX= 2.0000E+0 AVG= 1.1187E+0 SD= 3.2831E-1

SQUARE PLOT

<3>

SINC FCN

INDX#1

Sheet #17

C) A Sample pre-formatting routine using MLTWRITE and its execution setup for VT#PLMLT.

1. Job to create data on SCRATCH2
2. Job to plot this data via VT#PLMLT
3. Copy of plot created above.

```
$JOB CREATE DATA FOR MULTIPLGRAPH PLOTTING
$OPTION 5
$EXECUTE FORTRAN
C           THIS PROGRAM USES THE MLTWRITE/BEGIN/FINISH SUBROUTINE PACKAGE
C           TO PLOT FOUR ACTUAL GRAPHS.
C           THE FIRST IS A SINUSOID WITH RANDOM 'NOISE' ADDED IN.
C           THIS IS USED TO SIMULATE ACTUAL EXPERIMENTAL 'NOISY' DATA.
C           THE NEXT GRAPH IS AN UNDISTORTED SINUSOID, USED TO SIMULATE
C           THE IDEAL CURVE 'FITTED' TO THE DATA.
C           THE NEXT TWO GRAPHS ARE CONSTANT HORIZONTAL LINES USED TO
C           SIMULATE DATA LIMITS.
C
            CALL BEGIN ('**TIME**','**DATA**',-2)
C           THIS CALL TO BEGIN SPECIFIES OUR X-AXIS TO BE LABELED
C           **TIME**, AND OUR Y-AXIS TO BE LABELED **DATA**.
C           THE FORMATTED OUTPUT IS ROUTED TO SCRATCH2, AND IS DECLARED
C           MONITOR BLOCKED.
C
C           THIS LOOP INITIALIZES THE RANDOM NUMBER GENERATOR
            DO 1 I=1,1000
            CALL RANDOM (J,K,RAN)
1           J=K
C
C           THIS LOOP CREATES THE NOISY 'EXPERIMENTAL' DATA USED IN
C           GRAPH# 1.
C
            DO 10 I=1,50
C           (MAKE 50 DATA POINTS)
            X=I/8.
            Y=SIN(X)
            CALL RANDOM (J,K,RAN)
            J=K
C           ADD IN RANDOM 'NOISE' TO SINUSOID
            Y=Y+0.5*RAN-0.25
C           OUTPUT DATA PAIR VIA MLTWRITE
            CALL MLTWRITE (X,Y)
10          CONTINUE
C
C           CHANGE GRAPHS....
            CALL CHANGE
C
C           THE SECOND GRAPH DISPLAYS THE IDEAL, UNDISTORTED SINUSOID.
C           THIS LOOP CREATES 500 DATA POINTS.
C
            DO 20 I=1,500
            X=I/80.
            Y=SIN(X)
C           OUTPUT DATA PAIR VIA MLTWRITE...
            CALL MLTWRITE (X,Y)
20          CONTINUE
C
C           CHANGE GRAPHS....
            CALL CHANGE
C
```

```
C           THE THIRD GRAPH IS A HORIZONTAL LINE WHICH SYMBOLIZES THE
C           UPPER DATA LIMIT.  THIS LOOP CREATES 25 DATA POINTS.
C
            Y=1.0
            DO 30 I=1,25
            X=I/4.
C           OUTPUT DATA PAIR VIA MLTWRITE
            CALL MLTWRITE (X,Y)
30          CONTINUE
C
C           CHANGE GRAPHS....
            CALL CHANGE
C
C           THE FOURTH AND FINAL GRAPH IS A HORIZONTAL LINE WHICH
C           SYMBOLIZES THE LOWER DATA LIMIT. THIS LOOP CREATES 25 DATA
C           POINTS.
C
            Y=-1.0
            DO 40 I=1,25
            X=I/4.
C           OUTPUT DATA PAIR VIA MLTWRITE
            CALL MLTWRITE (X,Y)
40          CONTINUE
C
C           ALL DATA CREATION IS NOW CONCLUDED.  WE CAN CALL FINISH
C           AND BE DONE!!!
C
            CALL FINISH
C
            STOP ALLDONE
            END
$OPTION NOMAP
$ASSIGN1 LIB=VT%LIBRY
$ASSIGN1 DIR=VT%DIRCT
$ASSIGN2 LO=SLO,100
$ALLOCATE 10000
$EXECUTE GO
$EOJ
$$
```

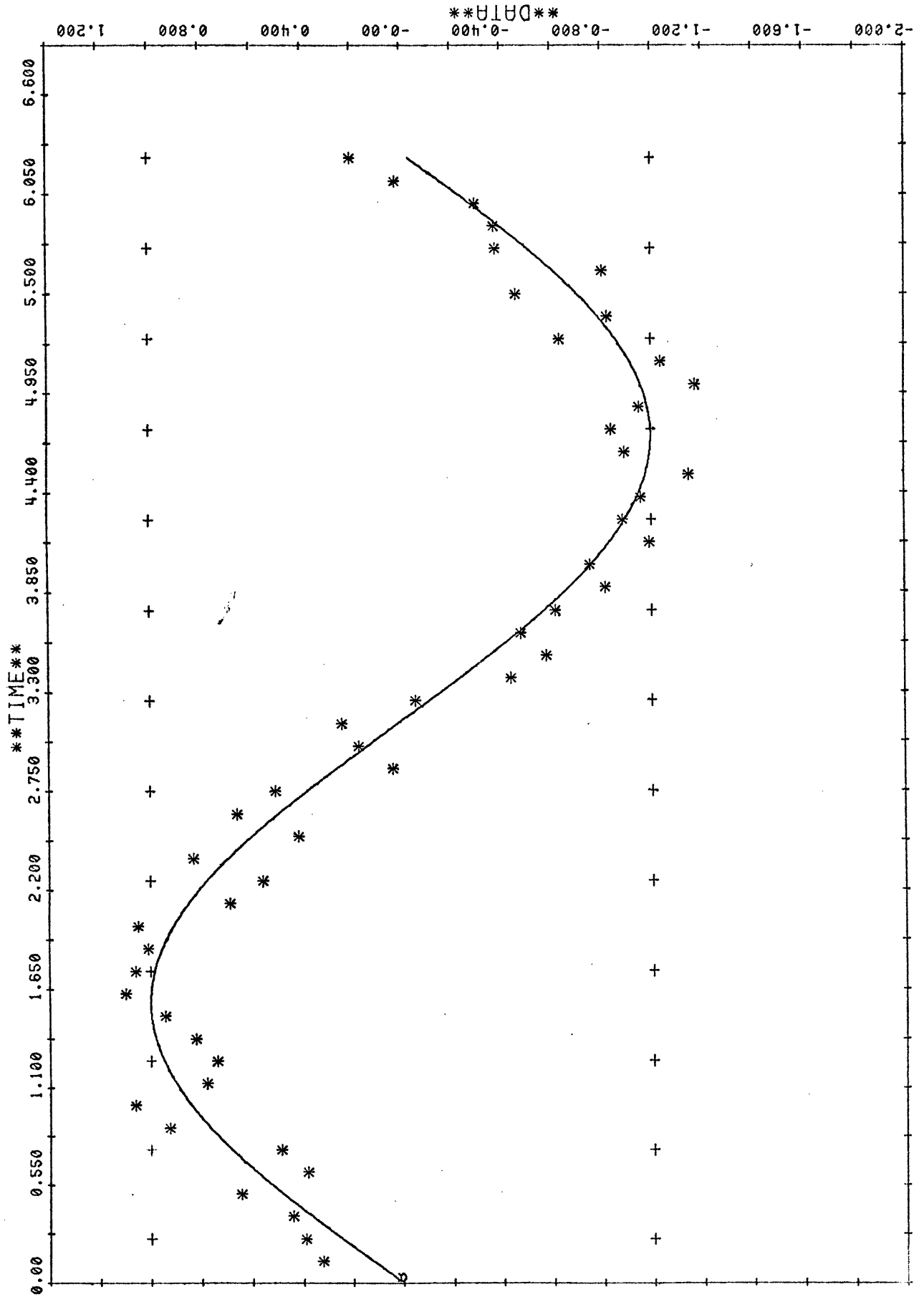
```
$JOB RUN MLTWRITE TEST DATA THROUGH VT#PLMLT
$EXECUTE VT#PLMLT
@BLK
MLTWRITE TEST
SINUSOID...
1,-1,84,
DATA POINTS
2,1000,2,
IDEAL CURVE
3,-2,86,
UPPER LIMIT
4,-2,86,
LOWER LIMIT
0,0,0,
END
$EOJ
$$
```

This job plots the graphs created earlier via VT#PLMLT. The first graph is called "Data Points". The -1 specifies a point plot and the 84 specifies an asterisk to be used as a marker symbol. The second graph is called "Ideal Curve". The 1,000 causes a solid line to be drawn, and the 2 produces no marker symbol. The third and fourth graphs are point plots, plotted every other point (the -2 does this). The 86 specifies a plus sign to be used as the marker symbol.

```
+    = LOWER LIMIT
+    = UPPER LIMIT
.    = IDEAL CURVE
*    = DATA POINTS
```

This legend is printed by VT#PLMLT below every plot.

Sheet #21



SINUSOID...

PLT #4

ML TWRITE TEST

D) Listing of the Symbol Table

A listing of the entire symbol table is presented here, showing each symbol with its accompanying integer representation. One may select a particular symbol as a marker via the @SYM option or the ISYM parameter in the VT#PLMLT routine by specifying an integer from this table.

Sheet #22

 26

 54

 82

 110

 138

 24

 52

 80

 108

 136

 22

 50

 78

 106

 134

 20

 48

 76

 104

 132

 18

 46

 74

 102

 130

 16

 44

 72

 100

 128

 14

 42

 70

 98

 126

 12

 40

 68

 96

 124

 10

 38

 66

 94

 122

 8

 36

64

 92

 120

 6

 34

 62

 90

 118

 4

 32

 60

 88

 116

 2

 30

 58

 86

 114

 0

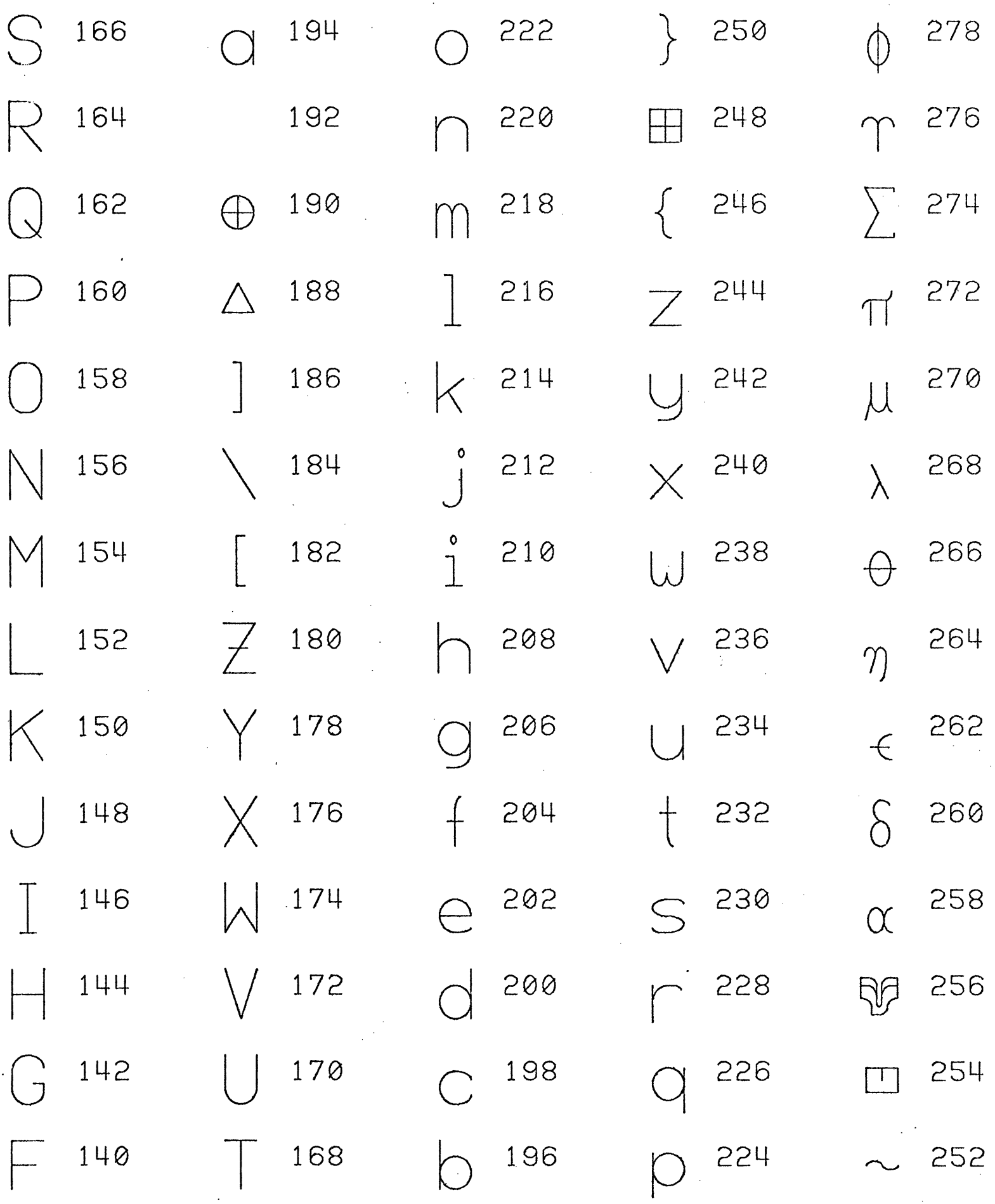
 28

 56

 84

 112

Sheet #23


S 166
a 194
O 222
} 250
ϕ 278
R 164
192
n 220
⊞ 248
ϒ 276
Q 162
⊕ 190
m 218
{ 246
∑ 274
P 160
△ 188
] 216
Z 244
π 272
O 158
] 186
k 214
y 242
μ 270
N 156
\ 184
j 212
x 240
λ 268
M 154
[182
i 210
w 238
θ 266
L 152
Z 180
h 208
v 236
η 264
K 150
Y 178
g 206
u 234
ϵ 262
J 148
X 176
f 204
t 232
δ 260
I 146
W 174
e 202
s 230
α 258
H 144
V 172
d 200
r 228
256
G 142
U 170
c 198
q 226
254
F 140
T 168
b 196
p 224
~ 252

E) Breakdown of VT% LIBRY

All Versatek subroutines are in the plotter library, and may be accessed by: $ASSIGN1 LIB=VT%LIBRY

$ASSIGN1 DIR=VT%DIRCT

These subroutines are broken into several different categories:

1) Versatek system routines - Not to be used:
IRAM, DATAOUT, LIST, FIND, VERS, SPLS, SPLT, MARK, IB, COPYFIL, RENN, CHAR

2) Versatek plotting routines - See "Versatek graphics programming manual" for details:
TONE, FORM,SCAN,AXES, DRAW,MODE, NOTE

3) Pen-plotter emulation routines - Conventional "Calcomp" format.
AXIS, SCALE, LINE, PLOTS, PLOT, FACTOR, WHERE, SYMBOL, NUMBER, NEWPEN, LOGLOG(used in AXIS), OFFSET

4) Sigma-5 Emulation Routines - These routines emulate the Sigma-5 software. They are used by all "VT#" modules.
CLRPLT, PLOT1, SYMBOL1, WHERE1, TIME, MARKER, CHECK, FLIP, FLIPX

5) Utility Routines - "Handy" routines used by most software:
ISL, ISA, IXOR, RANDOM

6) Pre-formatting routines-DSKWRITE/MLTWRITE package:
DSKWRITE, HEADER, ALIFT, MLTWRITE, BEGIN, CHANGE,FINISH

7) Three dimensional Animation Library-Strictly Experimental:
MM, VM, CLR, LOAD, SWAP, TRANS, SCL, XROT, YROT,
ZROT, AROT, WRIT, TDMP, MOVE, CENTR, VIEW, SHOW,
BORDER, INITIAL, CODE

The JCL file "VT$MAKLI" compiles and assembles source from VT$LIB01 thru VT$LIB04 to make up this object library.

F) Listing of Files in the "VT" area

All Versatek files reside in the "VT" area, and are listed below.

PREFIX = VT TOTAL #BLOCKS USED = 1656

FILENAME	LENTH	TYPE
VT#GPFLT	32	00
VT#PLC&A	96	00
VT#PLDED	96	00
VT#PLFOR	88	00
VT#PLGEN	96	00
VT#PLMLT	104	00
VT#PRINT	16	00
VT$BLCK1	8	ED
VT$BLCK2	40	ED
VT$BLK2A	48	ED
VT$CAT01	24	ED
VT$CAT02	16	ED
VT$CAT2A	16	ED
VT$GPFLT	16	ED
VT$LIB01	80	ED
VT$LIB02	24	ED
VT$LIB03	8	ED
VT$LIB04	128	ED
VT$MAKLI	8	ED
VT$PLC&A	32	ED
VT$PLDED	32	ED
VT$PLFOR	32	ED
VT$PLGEN	32	ED
VT$PLMLT	24	ED
VT$PRINT	16	ED
VT$SYMBL	16	ED
VT$TEST	168	ED
VT%DIRCT	8	00
VT%LIBRY	72	00
VT*JOB*	104	00
VT*WRKFL	80	F2
VTAUTOPT	48	00
VTPLOTER	48	00

Filename	Description
VT#GPFLT	Load module to pre-format the GP Filter dataset via DSKWRITE.
VT#PLC&A	Load module to plot the C&A dataset.
VT#PLDED	Load module to plot the Dedicated Data dataset
VT#PLFOR	Load module to plot Formatted Data from DS7
VT#PLGEN	Load module to plot data formatted via DSKWRITE
VT#PLMLT	Load module to plot data formatted via MLTWRITE
VT#PRINT	Load module (foreground) to use the Versatek as a printer.
VT$BLCK1	Phase one block data parameters
VT$BLCK2	Phase two mainline and block data parameters for VTPLOTER
VT$BLK2A	Phase two mainline and block data parameters for VTAUTOPT
VT$CAT01	Catalogs all phase one software.
VT$CAT02	Catalogs VTPLOTER
VT$CAT2A	Catalogs VTAUTOPT
VT$GPFLT	Catalogs VT#GPFLT
VT$LIB01	Source of basic Versatek subroutines
VT$LIB02	Source of plotter emulation subroutines
VT$LIB03	Source of assembly language subroutines
VT$LIB04	Source of 3 dimensional animation subroutines
VT$MAKLI	Creates object library
VT$PLC&A	Source of C&A plotter
VT$PLDED	Source of Dedicated Data plotter
VT$PLFOR	Source of Formatted Data plotter

VT$PLGEN	Source of DSKWRITE plotter
VT$PMLT	Source of MLTWRITE plotter
VT$PRINT	Catalogs VT#PRINT
VT$SYMBL	Plots symbol table
VT$TEST	Versatek test programs - Source
VT%DIRCT	Subroutine library directory
VT%LIBRY	Subroutine library (object)
VTAUTOPT	"Automatic" phase two plotter
VTPLOTER	"Manual" Phase two plotter