

REGION 74

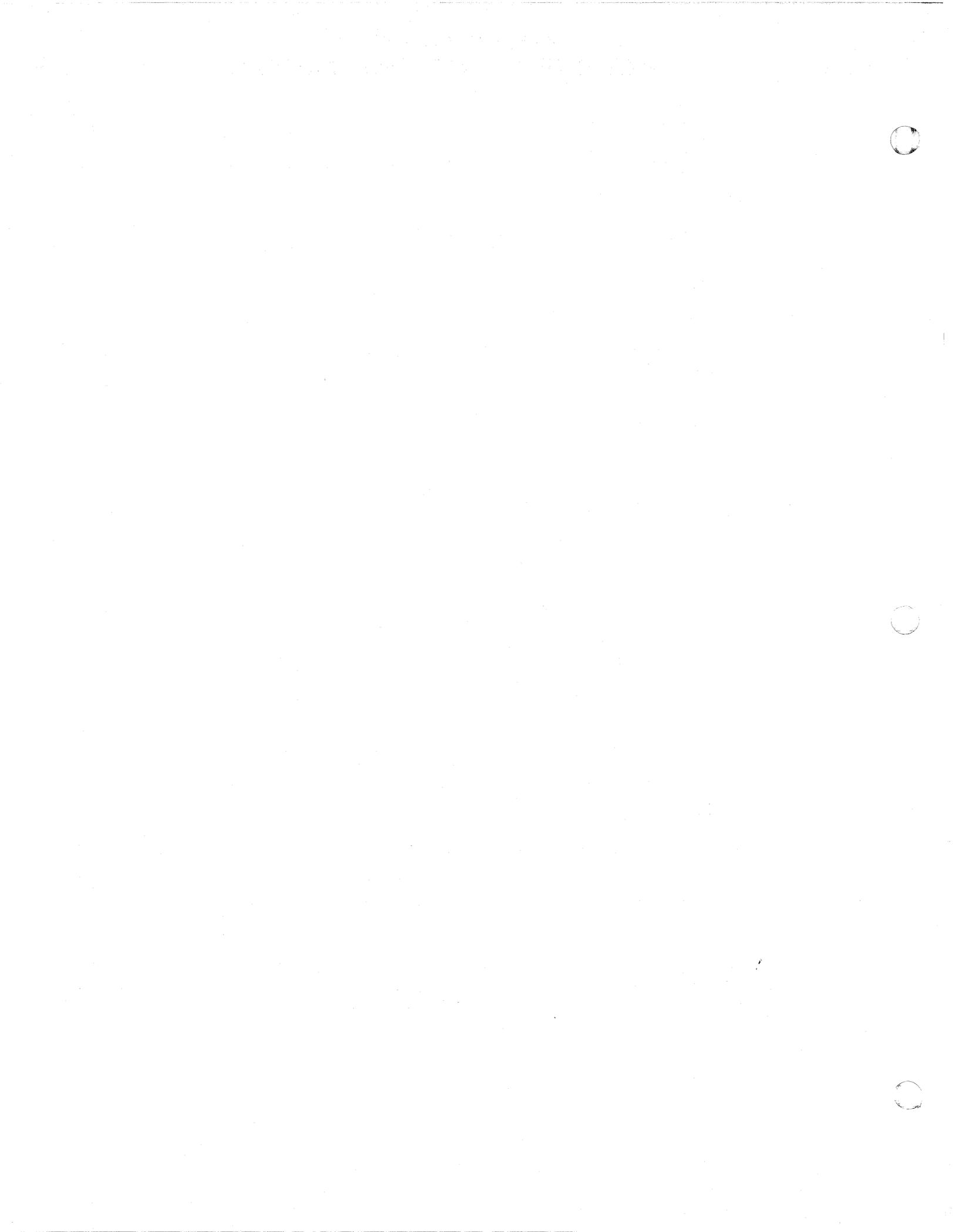
S-C 4020 SUBPROGRAMS

Table of Contents

<u>Section</u>	<u>Page</u>
74.0 Introduction	
Table of Contents	74.00.01
Introduction	74.00.05
Description of Machine	74.00.06
Recording the Image	74.00.07
Resolution	74.00.07
Size of Character Characters	74.00.07
Projected Master Form	74.00.07
Other Machine Features	74.00.07
Output Considerations	74.00.08
74.1 The S-C 4020 First Reader	
Preliminary Discussion	74.10.01
Subprogram Considerations	74.10.01
Selecting the Camera: CAMRAV	74.10.01
The Identification Frame	74.10.03
Advancing the Film: FRAMEV	74.10.03
Frame Count Retrieval: NØFRV	74.10.04
Hand Plotting Versus S-C 4020 Plotting	74.10.04
Generating a Linear Grid: GRID1V	74.10.04
Machine Computation of GRID1V Arguments: DXDYV	74.10.13
Intensity Selection: BRITEV, FAINTV	74.10.13
Plotting, Printing, and Labeling: Using Character Characters	74.10.15
Point Plotting Subprograms	74.10.15
Plotting an Array of Points: APLØTV	74.10.15
Plotting Individual Data Points: PØINTV	74.10.19
The Basic Plot-Print Subprogram: PLØTV	74.10.19
Printing and Labeling Subprograms	74.10.21
Typewriter Printing: PRINTV	74.10.21
Vertical Titles: APRNTV	74.10.21
Display of Non-FORTRAN Characters	74.10.23
Fixed Point Labels: LABLV	74.10.23
Scientific Notation Labels: LABLV	74.10.24
Conversion: NXV, NYV	74.10.25
Line Generation: LINEV, XAXISV, YAXISV	74.10.25
Error Marks	74.10.25
Review of First Reader	74.10.27
74.2 Special Printing and Form Overlay	74.20.01
Line Printer Simulation on the S-C 4020: SCØUTV	74.20.02

<u>Section</u>	<u>Page</u>
Basic Specifications	74.20.02
SCOUTV Variations	74.20.07
Altering the Starting Line: LINE1	74.20.07
Altering the Starting Line and Line Spacing Increments: LINE1, INC	74.20.07
Precise Control of Raster Positions for SCOUTV Printing: LØCSTV	74.20.11
Preservation (or Retrieval) of Raster Positions Used by SCOUTV: LØCSAV	74.20.15
Printing with Program-Generated Characters	74.20.21
The Vector Character Writing Subroutine: RITE2V	74.20.21
Variation in Size of Vector Characters: CHSIZV	74.20.25
Change Vector Character Spacing; Selecting Alternate Character Sets: RITSTV	74.20.25
Retrieval of RITE2V Terminal Coordinates: RITXYV	74.20.25
Display an Individual Vector Character: VCHARV	74.20.31
Available Vector Character Tables: TABL1V, TABL2V, and TABL3V	74.20.35
Special Uses of APRNTV	74.20.41
Use of the Form Slide	74.20.51
 74.3 Grid Variations	
GRID1V Controls	74.30.01
Grid Margin Variation: SETMIV, SETMØV	74.30.01
Combining a Graph and Printed Information on One Frame	74.30.06
Providing for Special Label Characters: SETCIV, SETCØV	74.30.09
Holding Margins from Graph to Graph: HØLDIV, HØLDØV	74.30.09
Log and Semi-Log Plotting	74.30.21
Establishing the Logarithmic Mode: SMXYV, MSXYV	74.30.21
Restrictions on Logarithmic Mode	74.30.21
Labels in Scientific Notation	74.30.24
Additional Modules for Special Grids: LINRV, NØNLNV	74.30.27
 74.4 Scaling and Conversion	
Basic Scaling Subprograms: XSCALV, YSCALV	74.40.01
Basic Equations for Computing Scale Factors	74.40.01
Conversion of Data: NXV, NYV, IXV, IYV	74.40.03
Basic Equations for Data Conversion	74.40.03
The Inverse of Conversion: UXV, UYV	74.40.03
Retrieval of Scaling Information: SCLSAV	74.40.05
Resetting Scaling Information: RESCLV	74.40.05
Nonlinear Scaling and Conversion	74.40.07
Linear-Nonlinear Scale Mode Indicators: SMXYV, MSXYV	74.40.07

<u>Section</u>	<u>Page</u>
Nonlinear Transformation: XMØDV, YMØDV	74.40.07
Off-Scale Error Detection	74.40.09
Set Conversion Error Indicators: SCERRV	74.40.09
Saving and Resetting Error Indicator Cells: SERSAV, SERREV	74.40.10
74.5 Further Aspects of Grid Construction	
Operational Details of GRID1V	74.50.01
Determining Grid Boundary Positions	74.50.01
GRID1V Error Procedure	74.50.02
Examples of GRID1V Characteristics	74.50.05
Building Special Grids: LINRV, NØNLNV	74.50.21
74.6 Further Aspects of Printing	74.60.01
Details of Charactron Characters	74.60.02
Typewriter Mode Considerations	74.60.03
System Subprograms and the Typewriter Mode	74.60.03
"Stop Type" Routines: STØPTV, RESETV	74.60.03
Setting the Current Point	74.60.04
Special Printing and Labeling Aids	74.60.11
Isolating One Character from a BCD Array: HØLLV	74.60.11
Obtaining BCD Equivalent of a Floating Point Number: BNBCDV	74.60.11
Vector Character Construction and Use	74.60.31
Vector Character Formation	74.60.31
Character Size	74.60.31
Character Orientation and Placement	74.60.31
Pattern Specification	74.60.32
Furnishing a Single Pattern to VCHARV	74.60.33
Table Structure	74.60.33
Writing More Than One Vector Character: RITE2V	74.60.34
Character Orientation and Placement: RITE2V	74.60.34
Furnishing a Single Pattern to RITE2V: RITSTV	74.60.34
74.9 S-C 4020 Reference Material	
Transmission of Commands: (PLØT), (CLEAN	74.90.11
S-C 4020 Call Statement Summary	74.90.21
Alphabetical List of S-C 4020 Subprograms	74.99.01



10-15-63

S-C 4020 SUBPROGRAMSINTRODUCTION

Designed and manufactured by General Dynamics Electronics, the S-C 4020 (as used at N.A.A.) is a peripheral system which reads magnetic tape output from a computer program and produces graphical, pictorial, or alphanumeric output. As the tape is read, the desired lines and characters are displayed on a specialized cathode ray tube. Sensitized paper and/or film is exposed to the display and the results are developed and returned to the programmer.

The obvious application of such a device is the rapid production of labeled graphs. Results in graphical form are usually much easier to analyze than are the same results in printed tabular form.

Less obvious, but often highly desirable, is the exploitation of the S-C 4020 as a high-speed printer. For certain applications, this affords distinct advantages over the use of a standard printer. For one thing, printed records can be obtained on 35 mm film, which greatly reduces the amount of space required to store the records. A form suited to the printed information can be generated or superimposed by the S-C 4020.

A variation is the production of a frame that combines tabular data with a plot of the related curve. Still another use of S-C 4020 techniques is the creation of diagrams and line drawings.

A complex of S-C 4020 output routines is available

to write the S-C 4020 control instructions on an output tape. These are either in FORTRAN function or subroutine format. (These two types of routines are described on pages 60.41.01 and 60.42.01, respectively. It is recommended that the programmer be familiar with them before attempting to use the S-C 4020 subprograms.)

At first glance, the list of S-C 4020 routines on pages 74.00.01 through 74.00.03 seems formidable indeed, but the user should not feel baffled by the great volume of details available. Some routines were developed for special purposes with limited usage. Many are used principally as lower-level modules for more general subroutines, and will seldom be called directly by the programmer.

In order to help the programmer avoid the study of many details extraneous to the solution of the average problem, an S-C 4020 "First Reader" has been prepared, beginning on page 74.10.01. Enough information is included for the preparation of ordinary graphs with plotted points, titles, and labels. Subsequent sections contain more details about the subjects introduced in the First Reader, as well as information on additional ways of using the S-C4020.

In order to become familiar with the vocabulary, the programmer should read the following introduction to the S-C 4020 equipment, before proceeding to the First Reader.

DESCRIPTION OF MACHINE

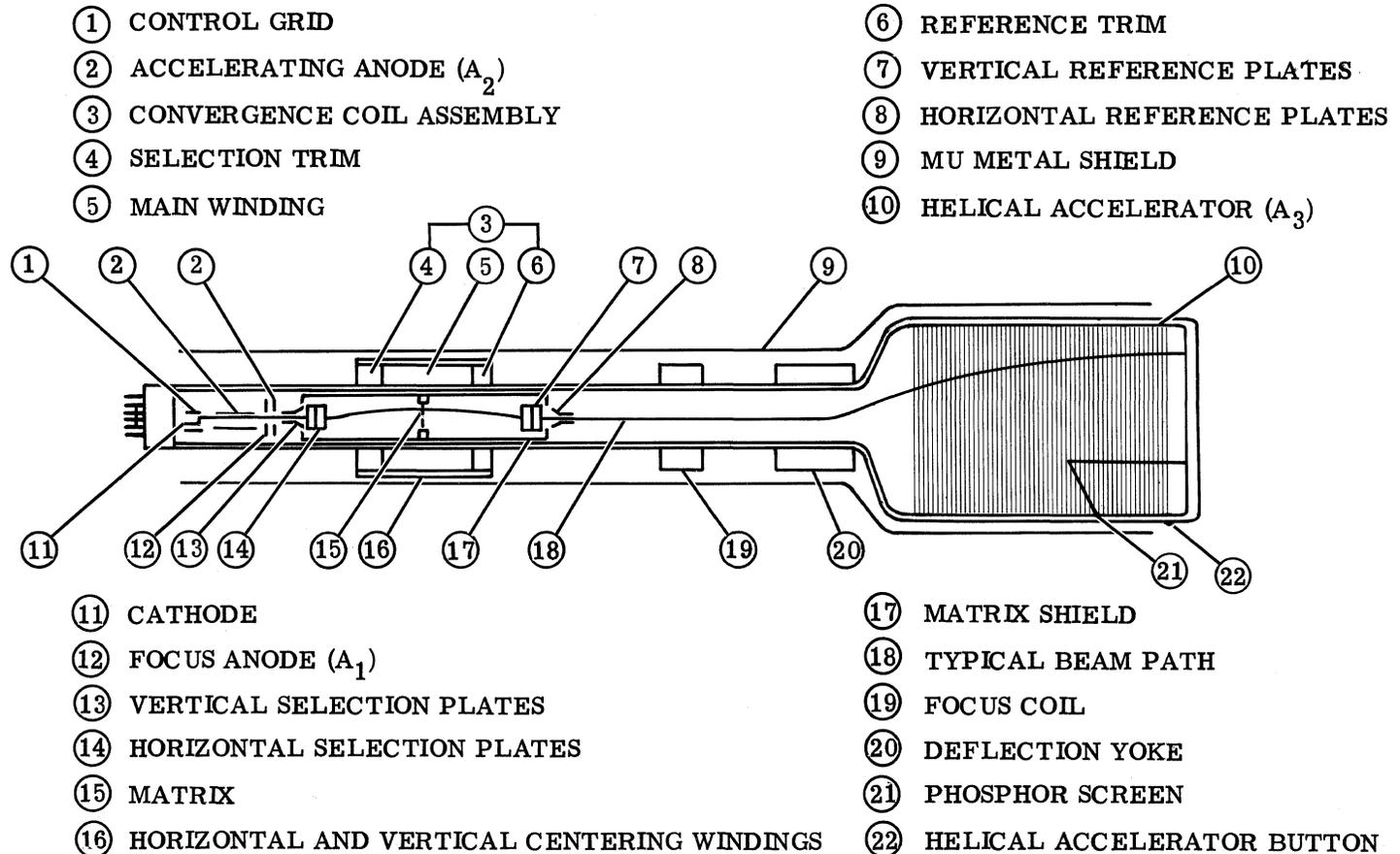
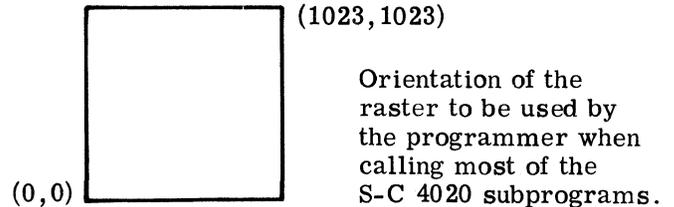
The basic device of the S-C 4020 is the CHARACTRON cathode ray tube illustrated. This shaped beam tube contains a built-in character-forming section consisting of an electron gun, character-selection plates, and a stencil-like matrix employing character-shaped openings. The matrix is a small thin disk of a special alloy; all character apertures are engraved on the matrix in a space less than one-fourth of an inch square. The 64 characters in the matrix are arranged in an 8 x 8 array.

The electron gun directs an electron beam toward the matrix; this beam passes between two sets of electrostatic character-selection plates which deflect the beam toward the matrix aperture desired. When the electron beam strikes the character matrix, a portion of the impinging beam passes through the character-shaped aperture and the emerging "extruded" beam has a cross section matching the character etched in the matrix. After leaving the matrix, the beam is redirected along the tube by reference plates and refined by the focus coil. Finally, the shaped beam passes through the deflection yoke, where it is deflected to

a selected position on the phosphor-coated tube face.

The deflection conversion circuitry allows the beam to be deflected to one of 1024 positions in the horizontal direction and one of 1024 positions in the vertical direction. The raster area on the face of the tube is therefore a lattice of 1024 x 1024 addressable positions. Each position may be addressed by a coordinate system of 0 to 1023 raster counts in the horizontal and in the vertical directions. No fractional coordinate positions may be addressed.

For convenience, the S-C 4020 subprogram system developed at N.A.A. assumes a raster orientation which is different from that built into the machine. In most of the N.A.A. subprograms, the user can think of the (0,0) point as being on the lower left-hand corner and the (1023,1023) point in the upper right-hand corner, like this:



CHARACTRON[®] Shaped Beam Tube

The term "raster count" will be used in referring to coordinate positions on the raster and to dimensions of an image. For example, a horizontal line which spans N addressable positions may be considered to be $N - 1$ raster counts in length. Vertical lines are referred to in a similar way. The length of a diagonal line can be considered to be the resultant of $N-1$ raster counts in the horizontal direction and M raster counts in the vertical direction.

RECORDING THE IMAGE

Two cameras are available which may be used separately or simultaneously to photograph the image on the tube face. The 35 mm camera records a 0.69" square picture on 35 mm film (with sprocket holes). The 9" camera records a 7.5" square picture on sensitized vellum paper. Subprograms are available which allow the programmer to select one or both cameras. "Selection" of a camera opens the shutter of that camera, allowing any image displayed on the phosphor coated CHARACTRON tube face to be recorded on the film (or sensitized paper).

RESOLUTION

Two plotting dots (or two parallel vectors) will appear as resolved dots (or vectors) if positioned three or more raster counts apart.

SIZE OF CHARACTRON CHARACTERS

The largest of the characters occupies an area which is approximately 6 raster counts wide by 9 high. The following is the actual size of the characters on a sample from the 9 inch camera.

THIS IS AN EXAMPLE OF CHARACTRON CHARACTERS.

The CHARACTRON characters cannot be rotated nor changed in size by the programmer. Mechanical variations will cause the size to vary slightly from time to time and may cause the characters to be slightly tilted.

PROJECTED MASTER FORM

A Form Projector allows a Master Form to be superimposed on the S-C 4020 output. Although the "flashing" (exposure) of the form is under program control, the S-C 4020 operator must select and manually insert the form at the beginning of the job. Since preparation of the master is an expensive procedure (it is not a 35 mm slide), such a form is suitable only for certain types of jobs.

For example, in some cases, the form cannot be program generated, or it is repeated so often that the machine time saved by a master justifies its cost.

OTHER MACHINE FEATURES

The Typewriter Mode feature built into the machine simulates typewriter action. Once positioned and started in the typewriter mode, the S-C 4020 will continue displaying word after word of BCD information, typewriter fashion, until a signal is encountered which halts the typing. Each "typed" Charactron character is allowed 8 horizontal and 16 vertical positions on the raster, which gives the necessary spacing to separate adjacent characters both vertically and horizontally. In this mode, a maximum of 128 characters can be typed in one line and 64 lines can be typed on one frame. A "carriage return" indicator is provided which will cause writing on one line to be halted and to be resumed on the next line at the left edge of the frame.

Some printing subprograms provided by the N. A. A. system take advantage of the speed of typewriter mode and the automatic spacing of characters. However, they terminate typing by use of a character count rather than by the "stop type" indicator provided by the manufacturer.

Bright and Faint Intensities of exposure are available for character plotting. In addition, repeated plotting will give heavier exposure, although excessive repetitions will cause haloing (spreading of the light into the surrounding area).

A Vector Generator permits drawing a line from a specified point to the resultant of specified horizontal and vertical components. The length of one vector is limited - neither the horizontal nor the vertical component may be greater than 63 plotting positions. However, a longer line may be formed by a series of vectors plotted head-to-tail.

The Vector Generator facility offers a method of forming characters which may be varied in size and in orientation. Subprograms are available for the generation of vector characters. The following example was displayed in a large size to show details of vector character construction.

VECTORS

An Axis Generator enables a continuous line to be drawn either horizontally or vertically across the film frame. The coordinates of the starting point of each axis must be specified. Because of some current disadvantages in the use of axis lines, the system subprograms, which must produce lines, rely on the use of vector generated lines.

A Current Point Register on the S-C 4020 will hold the coordinates specified by the last operation. This register can be set by the programmer (by plotting a blank, if necessary, at the point desired), but cannot be retrieved by him.

OUTPUT CONSIDERATIONS

A number of variables influence the quality of the pictures produced on the S-C 4020. In addition to variations in the operation of the machine, factors related to the processing of the film or sensitized paper cause some day-to-day changes in output quality.

The S-C 4020 customer engineer and the EDPM Group try to hold these variations to a minimum. Programmers should report instances of substandard quality to the EDPM Group. Form 1028-X-2 is available for this purpose; it can also be used as a request to have pictures "re-shot."

The output tape will be held 48 hours after running on the S-C 4020, in order to allow the programmer to check the output. If the programmer wishes an additional set of pictures, he can request that the tape be reprocessed on the S-C 4020 before the tape is released. If he anticipates that there will be a need to use the tape at a later time, he should, of course, reserve the tape.

Reproductions of S-C 4020 Output

When multiple copies of S-C 4020 pictures are

required, the programmer must consider what method of duplicating is most suitable for his application.

Blueline reproductions are of very poor quality, because of the thickness of the paper output. However, reasonably good bluelines can be obtained if a vellum copy is first made on the 914 Xerox and then used as the master copy for blueline reproductions.

If less than four copies of a 9-inch picture are required, 914 Xerox or Xerox Copyflo is satisfactory. Multilith, however, gives much better reproductions, and is preferred if more than ten copies are desired.

Work submitted for Multilith or 914 Xerox reproduction must be cut into 8-1/2 inch sheets. The Xerox Copyflo process can handle a roll of the 9-inch paper output.

Photographic reproductions are recommended for work that must be increased (or decreased) in size, or for work that requires special handling or special quality. If the programmer knows that he will need photographic reproductions, he should produce his output on 35 mm film. (Photographic will have to make an internegative from this film in order for the photo-print to have black lines on a white background.)

Projected Pictures

Output from the 35 mm camera can be projected by a strip film projector, or it can be mounted as 35 mm slides.

Since the space between frames on the 35 mm film is very small, an extra frame advance between each output frame will provide borders useful in either type of projection.

Section 74.1

THE S-C 4020 FIRST READER

The S-C 4020 First Reader has been planned to give the minimum details necessary to use the S-C 4020 as a plotting device. In order to avoid time-consuming descriptions, many subtleties and variations of the routines have been omitted, and other subprograms are left out entirely.

Upon completion of the First Reader, the student will be able to use the S-C 4020 as a plotter. It is hoped that he will be sufficiently interested to study the subsequent sections, which contain descriptions of many additional S-C 4020 facilities.

PRELIMINARY DISCUSSION

To review the Introduction, an area of 1024 x 1024 addressable positions, called the raster, is available for plotting on the face of the Charactron cathode ray tube. For all the routines discussed in the First Reader, the (0,0) point is assumed to be the lower left-hand corner of the raster, and the (1023,1023) point is assumed to be the upper right-hand corner. Other positions may be addressed by specifying the number of raster counts (addressable positions) from the left and from the bottom of the raster. Fractional distances cannot be addressed.

Images displayed on the face of the Charactron cathode ray tube may be recorded on 35mm film and/or 9-inch sensitized vellum, depending upon the camera, or cameras, selected.

Subprogram Considerations

From the programmer's viewpoint, an S-C 4020 display is produced as the result of a series of calls to FORTRAN subroutines (with the occasional use of FORTRAN function-type subprograms). Before attempting to use the S-C 4020, therefore, the programmer must be familiar with the FORTRAN language, and particularly with the operation of FORTRAN subprograms.

To simplify the descriptions, terms have been used which imply that the S-C 4020 display is taking place during execution of the subprograms. Actually, execution of these routines results in writing commands for S-C 4020 display on a magnetic tape. At NAA, the S-C 4020 is a peripheral device; the magnetic tape is physically removed from the 7090/7094 following the computer run, and placed

on the S-C 4020 tape unit, where it controls the S-C 4020 output.

The typical S-C 4020 application involves the display of some physical phenomenon scaled to the dimensions of the available plotting area. For the most part, the programmer can do his planning on the basis of the physical model. The system provides subprograms that will accept information given in terms of the model and perform the necessary scaling and conversion to relate the information to the raster area.

In the subprogram descriptions, floating point variable names have been assigned in accordance with the FORTRAN convention. If integer variables are required, names beginning with I, J, K, L, M, or N are used. The actual floating point number (or integer) may be used in the argument lists in place of a floating point (or integer) variable name, if the argument represents input to the subprogram. Constants should never be substituted for argument names that represent output from the subprogram.

SELECTING THE CAMERA

THE CAMERA (OR CAMERAS) MUST BE SELECTED AT THE OUTSET OF S-C 4020 USAGE IN EVERY JOB OR CHAIN LINK THAT CALLS FOR S-C 4020 OUTPUT.

Camera selection is implemented by the following statement:

CALL CAMRAV (N)

N = 9 The 9-inch camera will be called and the 35mm camera will be dropped out of select.

N = 35 The 35mm camera will be called and the 9-inch camera will be dropped out of select.

N = Any other number. Both cameras will be selected.

Note that selection of one camera causes the S-C 4020 display to be photographed only by the camera selected. That camera will be used until another CALL CAMRAV statement changes the selection. If the 9-inch camera is selected, and a CALL CAMRAV(35) is encountered, usage of the 9-inch

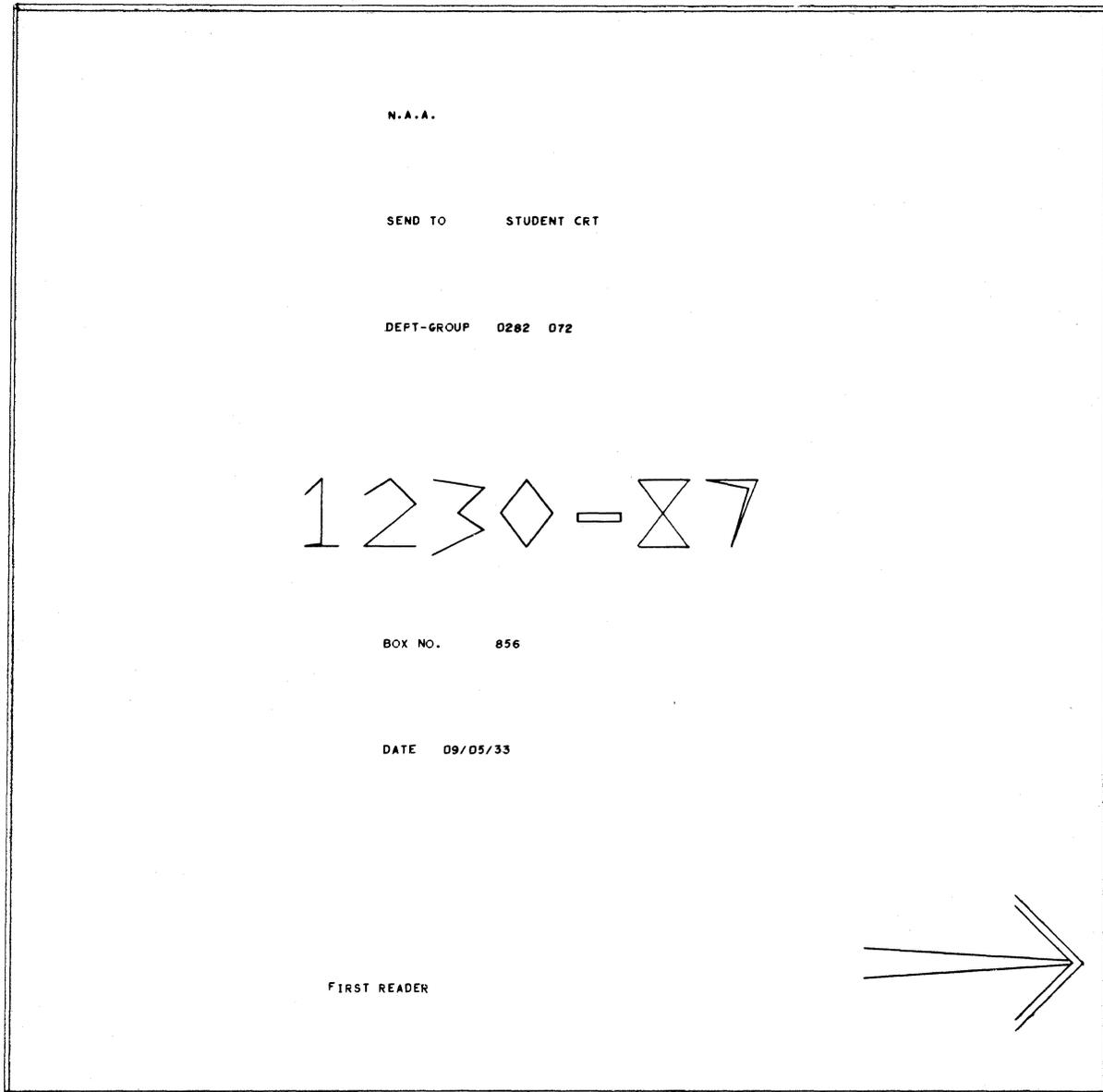


Figure 1-a

AN IDENTIFICATION FRAME

As in other NAA accounting printouts, the machine number and the last digit of the year appear as the last two digits of the date.

10-15-63

camera will be discontinued and only the 35mm camera will be employed.

Therefore, CALL CAMRAV (935) is not equivalent to the two statements

```
CALL CAMRAV (9)
CALL CAMRAV (35)
```

If both cameras are used alternately (rather than simultaneously) in a program, it is recommended that the following sequence be used as the first S-C 4020 statements in the program:

```
CALL CAMRAV (N)      Where N is any
                      number other than 9
                      or 35.

CALL CAMRAV (9)      To select the cam-
or                   era needed for the
CALL CAMRAV (35)     first display. That
                      camera will be used
                      until another CALL
                      CAMRAV is
                      encountered.
```

In this sequence, the first CALL CAMRAV statement provides simultaneous writing of the identification frame on both cameras, since both are going to be used.

(It is possible for the S-C 4020 operator to set a switch which overrides the programmed camera selection. A request for such action should only be made when an unusual need arises.)

THE IDENTIFICATION FRAME

The ID frame (Figure 1-a) is produced by the system when it encounters the first S-C 4020 statement that puts information on the S-C 4020 output tape. This should be during execution of the first CALL CAMRAV statement. The chief purpose of the ID frame, of course, is to enable the job dispatcher to return the developed output to the right programmer.

However, the frame can also serve as a clue to output quality, since it is always generated in the same way. Several machine features are demonstrated. The information printed in small characters is an example of Characteron characters, printed in the typewriter mode, using bright intensity. These characters are displayed only once;

i.e., they are not replotted for greater intensity. Since the typewriter mode is used, the spacing of characters within words is determined by the equipment, not by the programmer.

The large characters in the job number are formed with vectors. (These simplified figures are not examples of the set of standard vector characters for programmer use, however.) Another example of vectors may be seen in the arrow. Since the lines that form the shaft of the arrow are longer than the maximum vector length, each line is made up of three vectors joined end-to-end.

The border is formed by pairs of axis lines. The X axis lines, once started, sweep to the right edge of the frame, while the Y axis lines, once started, sweep to the top of the frame. On the ID frame, the outermost axis lines lie on the outermost raster ordinates. The border, therefore, shows the maximum size of a frame (and, incidentally, shows how close the frame is to being square).

ADVANCING THE FILM

It is the programmer's responsibility to see that the film is advanced - even after display of the identification frame. The only film advance that the system will perform without programmer action is the one that occurs just before the ID frame is written, to ensure a fresh start for the job.

There are two principal ways to advance the film. The first is to use one of the following statements:

```
CALL FRAMEV or CALL FRAMEV(N)
```

Either will advance the frame and update the frame count. If no argument is used, or if the argument (N) is zero, the job number and the frame count for each camera will be displayed in the upper right corner of the frame. If the argument (N) is nonzero, printing of the job number and frame count will be suppressed.

The second way to advance the film is by the selection of the appropriate value for the first argument for the GRIDIV subprogram, which will be described a little later. If GRIDIV is used, the programmer is freed from the necessity of making a separate CALL FRAMEV statement.

FRAME COUNT RETRIEVAL: NØFRV

NØFRV enables the programmer to obtain the current frame count. He can use this number as a "cut-off" point to prevent a wasteful loop. (Some programmers have used it, in conjunction with a timing routine, to determine how much computing time was needed to produce a frame, or for other accounting purposes within their programs.) The call statement is

CALL NØFRV (N9, N35)

- N9 Name of location in which current count of 9-inch camera frames will be stored.
- N35 Name of location in which current count of 35mm camera frames will be stored.

Since both arguments represent output from the routine, they must be variables.

HAND PLOTTING VERSUS S-C 4020 PLOTTING

In many ways, plotting on the S-C 4020 is very much like plotting on a sheet of graph paper, but there are also distinct differences. For one thing, the programmer must create the grid - the film frame is completely blank to start with.

Although every line of the grid must be specified on the S-C 4020, there are advantages to this situation. A hand-plotted graph must be adapted to some preprinted form; more frequently than not, this means that some plotting area must be sacrificed in order to use the most convenient scale.

On the S-C 4020, the programmer can select a scale that will be easy to read and that will accommodate the entire range of data. The number of

light grid lines, the number of emphasized grid lines, and the spacing between lines can be chosen to suit the plot. In addition, the programmer is not restricted to the use of a single form for a variety of plots. For each graph, a new grid can be tailored to the data.

In designing a grid, the programmer should consider: (1) how to obtain a meaningful, easy-to-read grid, (2) how to avoid wasted plotting areas, and (3) how to prevent crowding of grid lines and/or line labels.

GENERATING A LINEAR GRID: GRID1V

The easiest way to create a grid for S-C 4020 plots is to call the GRID1V subprogram. (The following description of this subprogram is by no means complete. Additional information on the use of GRID1V will be found in later sections.)

GRID1V will produce a grid which has some lines emphasized and some lines labeled. Margin space (which may be used for titles) will be reserved at the top, left side, and bottom of the grid. Normally, the title margin spaces are 24 raster counts wide.

Upon completion of GRID1V, scale factors will have been established and made available (internally) for the conversion requirements of other subprograms; i.e., the conversion of floating point coordinates into raster coordinates.

At the outset, GRID1V makes certain that the Typewriter Mode is off, and that the Bright Intensity Mode is on.

The call statement for GRID1V appears on the following page, with a description of the arguments.

CALL GRIDIV (L, XL, XR, YB, YT, DX, DY, ±N, ±M, ±I, ±J, ±NX, ±NY)

- L** This integer argument controls the film advance and frame identification display:
- L = 1 The film will be advanced. The job number and frame counts (for both cameras) will be placed in the upper right corner of the frame.
- L = 2 The film will not be advanced and the identification information will not be displayed.
- L = 3 The film will be advanced but the identification information will be suppressed.
- XL, XR** Floating point values of X for the left-most and right-most limits of the grid.
- YB, YT** Floating point values for the bottom limit and the top limit of the grid.
- After margin space for titles and labels has been reserved, the limits of the remaining space are assigned the data values given for XL, XR, YB, and YT. Scale factors are then computed; they will remain in effect until another GRIDIV statement is made (or until other action is taken to compute new scale factors).
- DX, DY** Floating point data increment at which vertical (specified by DX) and horizontal (specified by DY) grid lines will be displayed. If 0.0, no lines will be shown.
- Positions are stepped off in DX increments in the positive and negative directions from $X = 0$, and in DY increments in the positive and negative directions from $Y = 0$. At each position, the data value is converted into raster counts and a grid line is displayed. To prevent crowding of grid lines, DX and DY should be specified to produce less than 120 lines. (This restriction applies to the usual grid. It could be more severe or more lenient in special applications.)
- N, M** Fixed point integers that cause every Nth vertical grid line and every Mth horizontal grid line to be retraced for emphasis. If N (or M) is zero, no vertical (or horizontal) lines will be emphasized.
- To force the grid to be square, a negative sign should be used on N and/or M. (If either N or M is zero, the negative sign should go on both N and M.)
- I, J** Fixed point integers which cause every Ith vertical line and every Jth horizontal line to be labeled. If I (or J) is zero, no vertical (or horizontal) lines will be labeled.
- If I and J are positive, the line labels will lie along the $X = 0$ and $Y = 0$ lines, provided these lines are within the grid limits. If $X = 0$ (or $Y = 0$) does not fall within the grid limits, labels will be placed in a space reserved at the left (or at the bottom) of the grid.
- Negative signs can be used on I and/or J to force labels outside the grid. Label space is reserved at the left if I is negative, or at the bottom if J is negative, and labels will be placed in these reserved spaces. Note that label margin space is in addition to the margin reserved for titles.
- NX, NY** Fixed point integers indicating the number of characters to be displayed in the labels of vertical and horizontal lines.
- +NX, +NY The labels will be in a decimal format similar to the F-type format. In specifying +NX and +NY, a decimal point must be counted as one of the NX or NY characters, but the sign is not counted. The largest number of digits permitted is 6 (or 7 if one character is a decimal point).
- NX, -NY The labels will be in scientific notation. (Example: $1.25 \times 10^{+02}$.) NX indicates the number of significant figures in the labels of vertical grid lines, and NY indicates the same for the labels of horizontal lines. The sign, decimal point, and exponent will be displayed in addition to NX (or NY) characters. NX (or NY) must not be greater than 6.

Examples

Figure 1-b shows a grid produced by GRID1V. The general call statement is repeated here for comparison with the actual call statement for the example.

General

CALL GRID1V (L, XL, XR, YB, YT, DX, DY, ±N, ±M, ±I, ±J, ±NX, ±NY)

Actual

CALL GRID1V (1, -30.0, 30.0, -600.0, 800.0, 1.0, 25.0, 5, 4, 10, 8, 2, 3)

For the sake of convenience, constants were used as arguments in this and many of the subsequent illustrations. It would, of course, be equally satisfactory to use the names of variables which have the desired values.

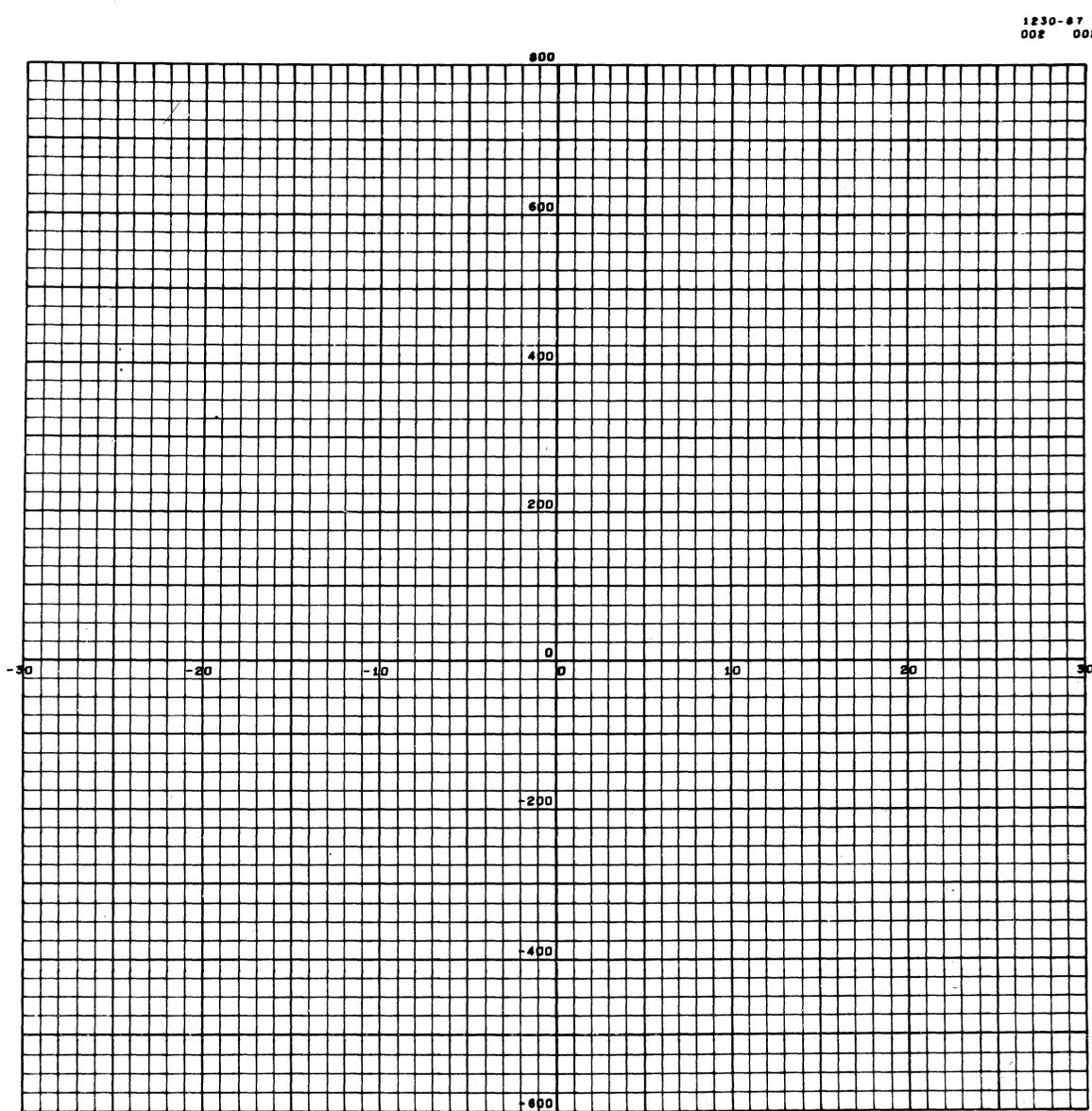


Figure 1-b

The following statement was used to produce this grid:

CALL GRID1V (1, -30.0, 30.0, -600.0, 800.0, 1.0, 25.0, 5, 4, 10, 8, 2, 3)

In studying Figure 1-b, note particularly the following items:

Corner marks radiating out from (0,0), (0,1023), (1023,1023), and (1023,0) are placed on the frame by the system when the film is advanced.

Since the value used for L is 1, the film was advanced, and the job number and frame counts (for both cameras) were placed in the upper right corner.

Note the emphasized lines, the labeled lines, the number of characters in the labels, and the increment of line spacing.

Figures 1-c through 1-g illustrate further aspects of GRID1V.

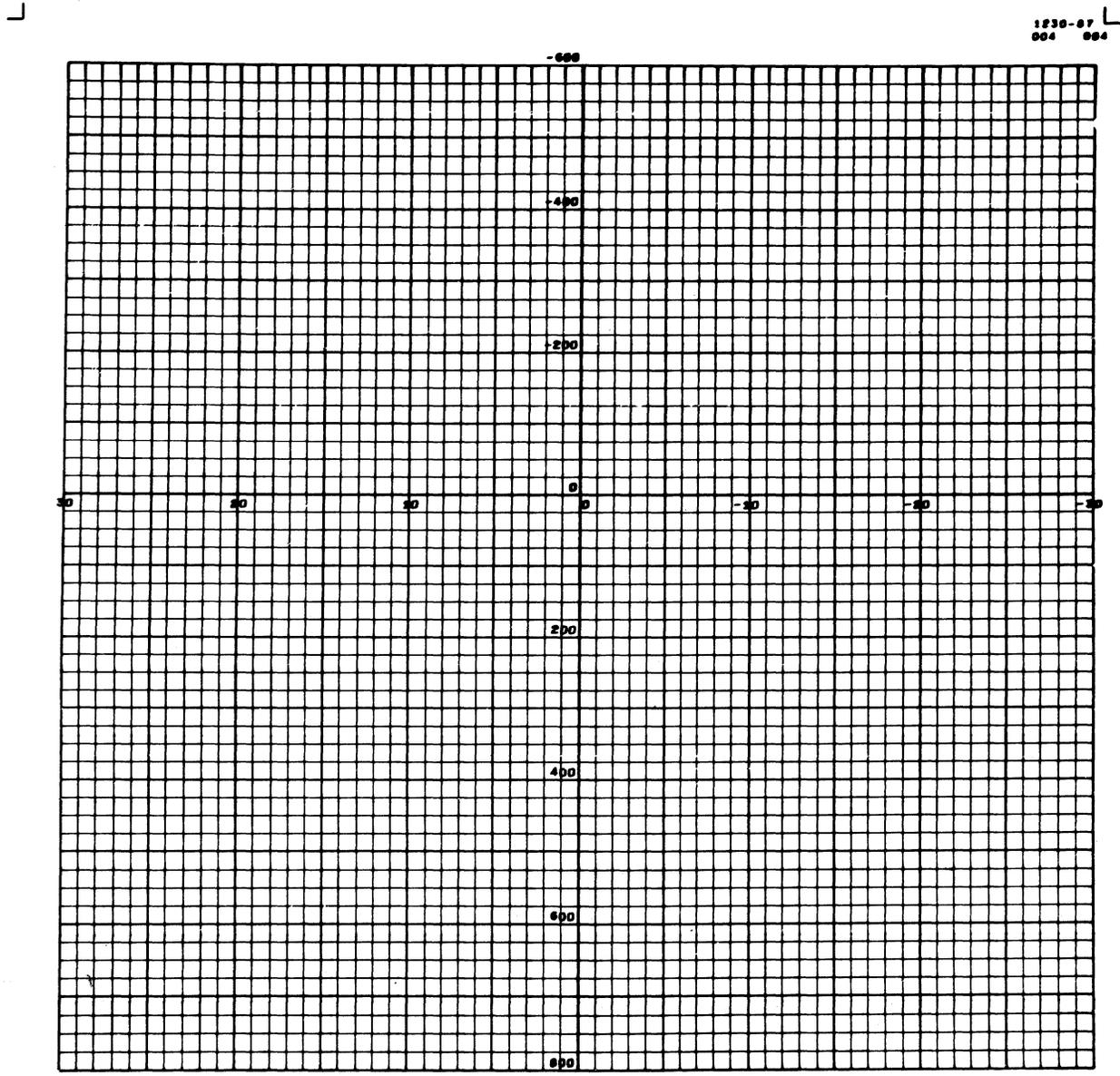


Figure 1-c

CALL GRID1V (1, 30.0, -30.0, 800.0, -600.0, 1.0, 25.0, 5, 4, 10, 8, 2, 3)

This grid shows that no problems arise when X is decreasing from left to right and/or Y is decreasing from bottom to top. It is identical to Figure 1-b except that the values of XL and XR, and of YB and YT, are reversed.

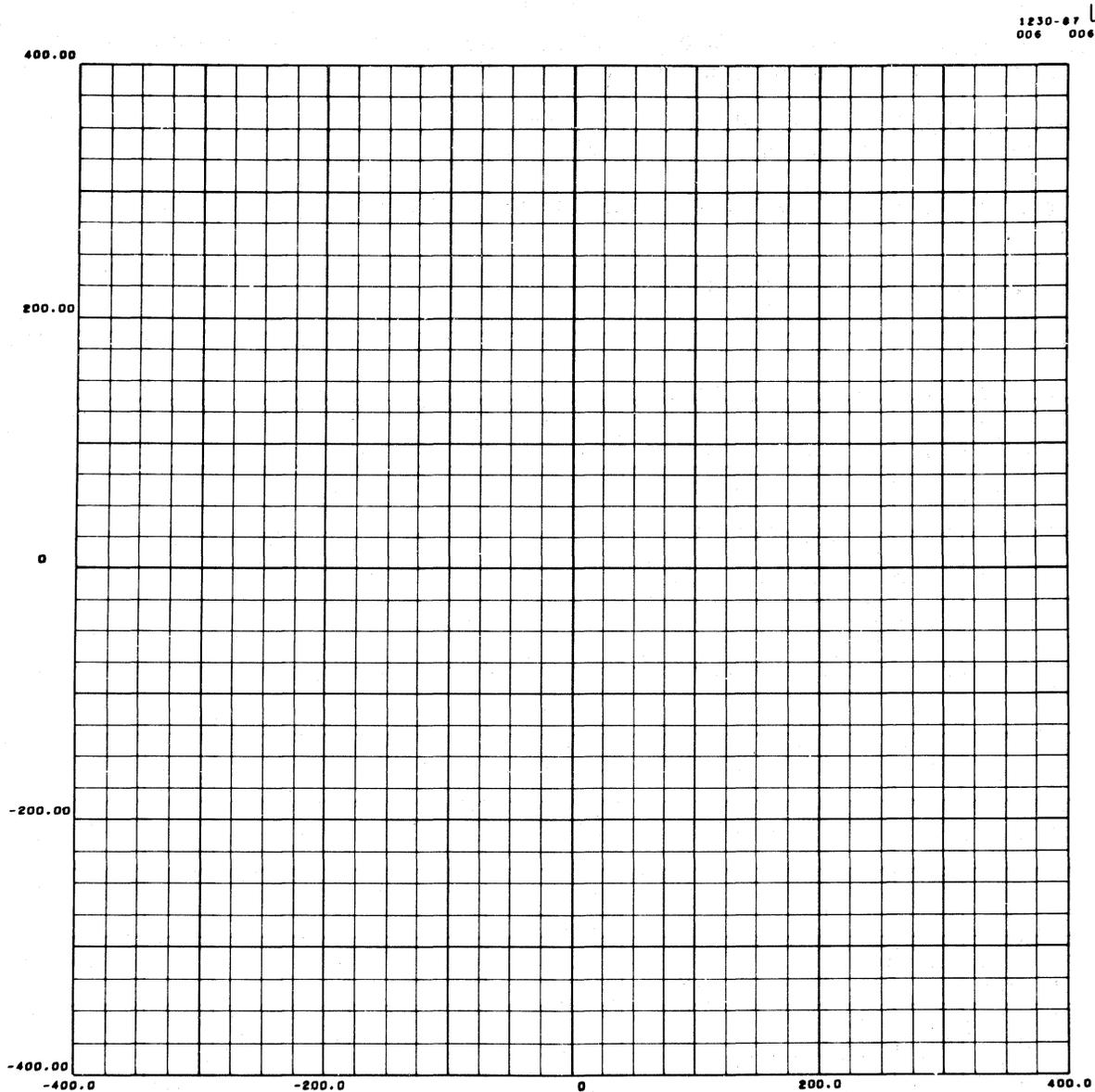


Figure 1-d

CALL GRID1V (1, -400.0, 400.0, -400.0, 400.0, 25.0, 25.0, 4, 4, -8, -8, 5, 6)

Line labels which would otherwise lie along the $X = 0$ and $Y = 0$ lines were forced to lie along the XL and YB lines by assigning negative values to I and J, the 10th and 11th arguments (underlined above). GRID1V assigned additional margin space to accommodate the labels.

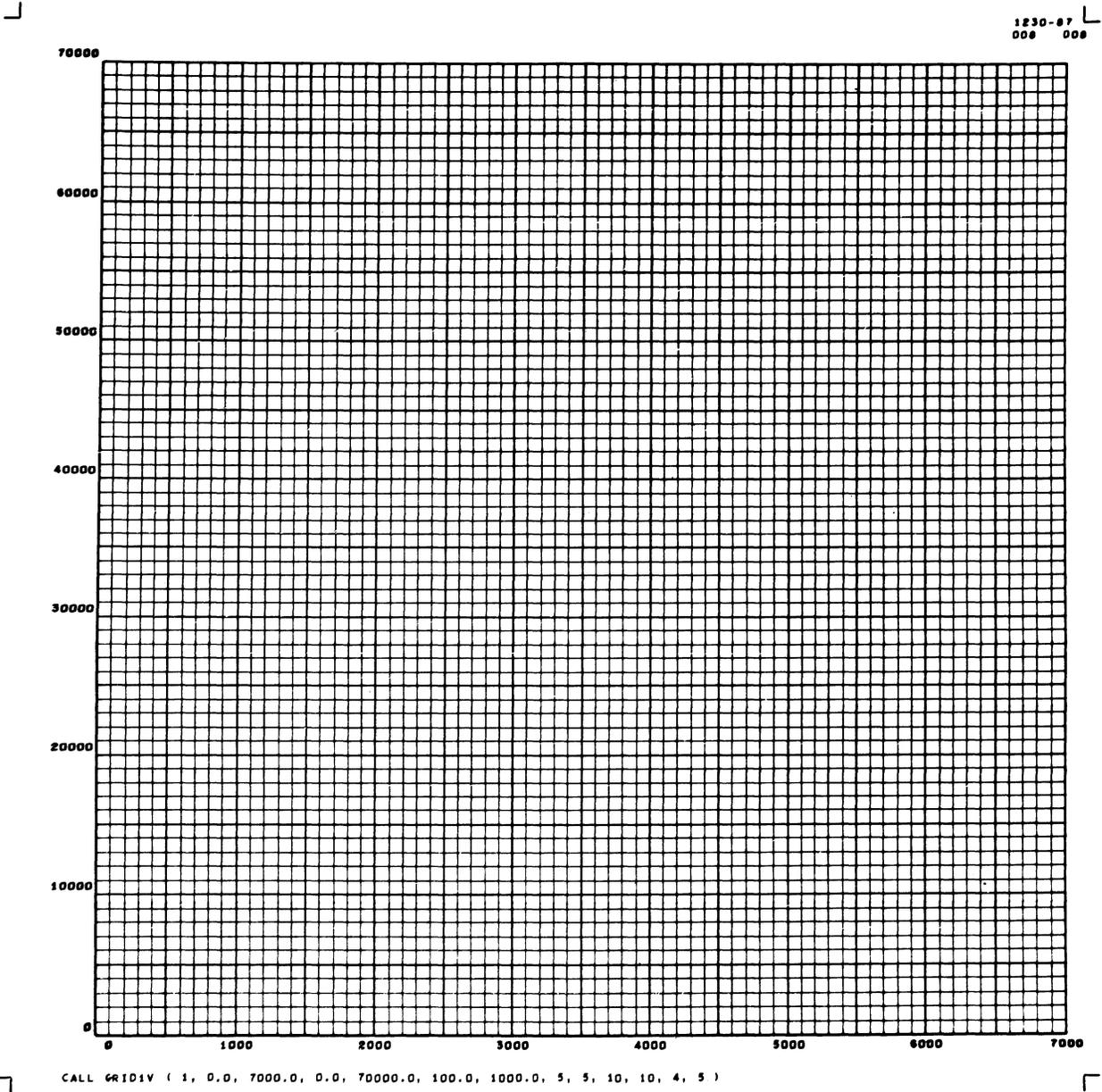


Figure 1-e

CALL GRID1V (1, 0.0, 7000.0, 0.0, 70000.0, 100.0, 1000.0, 5, 5, 10, 10, 4, 5)

This example and Figure 1-f show, for comparison, a grid labeled with fixed point labels and the same grid labeled with scientific labels.

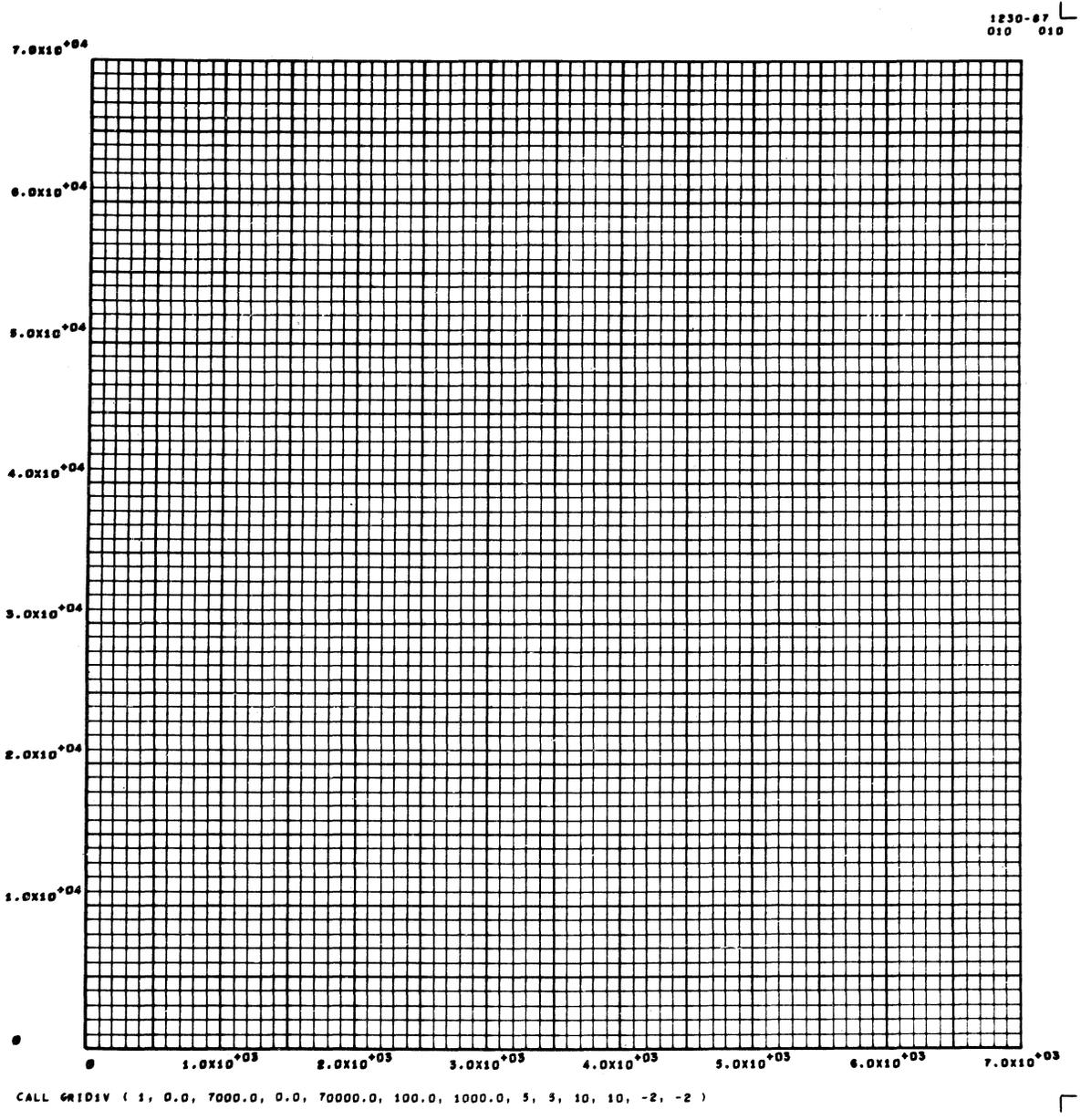


Figure 1-f

CALL GRID1V (1, 0.0, 7000.0, 0.0, 70000.0, 100.0, 1000.0, 5, 5, 10, 10, -2, -2)

The negative signs on the last two arguments result in labels in scientific notation. Note that, for scientific labels, NX and NY specify only the number of significant figures desired.

10-15-63

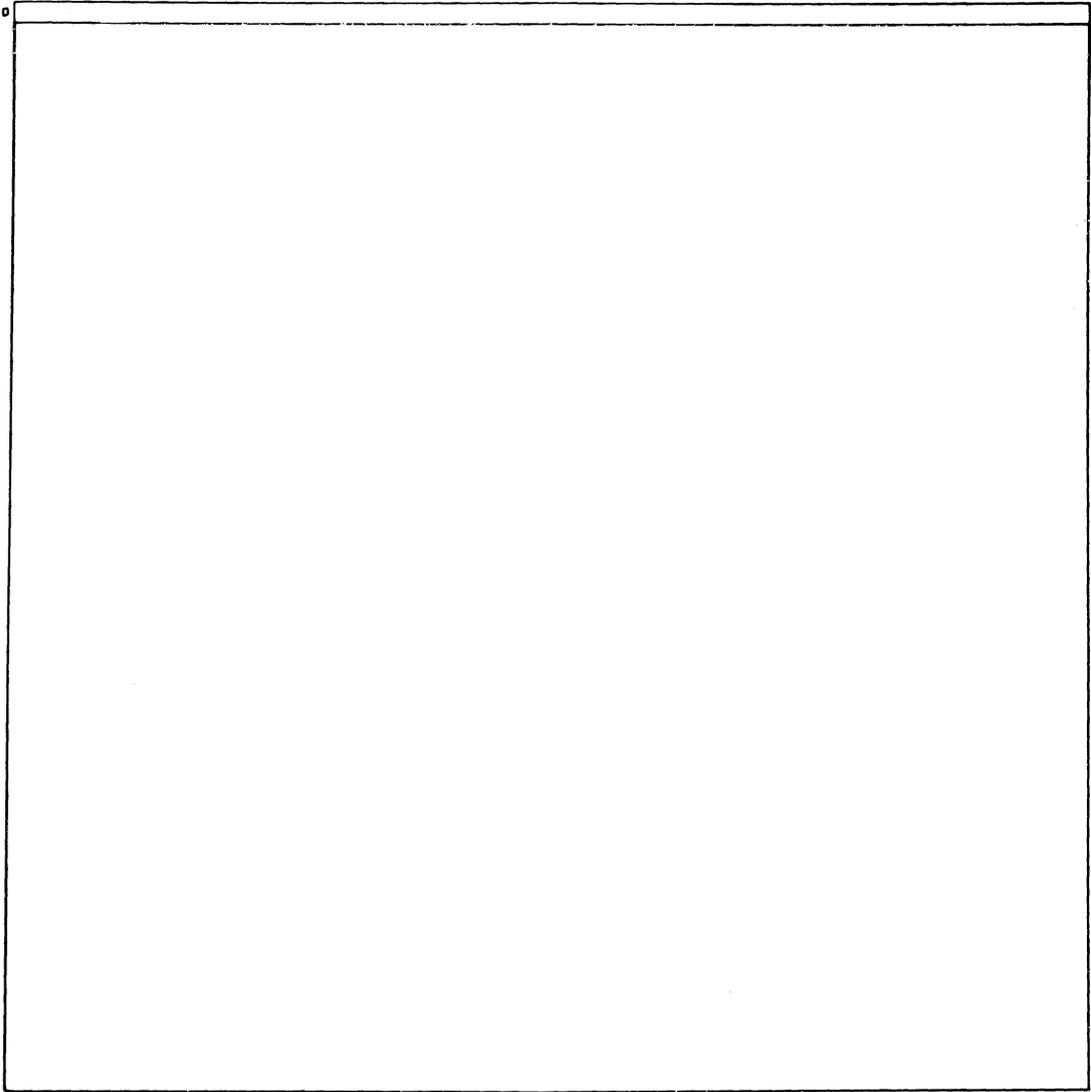
1230-87
012 012100.00
100.00

Figure 1-g

```
CALL GRID1V (1, 50.0, 100.0, 50.0, 100.0, 50.0, 50.0, 1, 1, 2, 2, 3, 6)
```

```
CALL GRID1V (2, 50.0, 100.0, 50.0, 100.0, 50.0, 50.0, -1, -1, 2, 2, 3, 6)
```

This illustration was intentionally double-exposed to show the influence of negative signs on N and/or M (the 8th and 9th arguments). The first CALL GRID1V statement (with positive N and M) produced the grid that is taller than it is wide. This occurred because the total amount of margin space (title margin plus label margin) required on the left and right is greater than the total of the top and bottom margin areas. The second statement, with negative N and M, produced the inner grid. When N and/or M are negative, GRID1V makes an additional margin adjustment so that the remaining grid will be square. (However, machine malfunctions can produce a raster that is not square; in this case a negative N or M will not ensure an absolutely square grid.)

It is particularly important, of course, to square the grid if a circle or square, or some figure approaching these forms, is to be plotted.

This example also shows that the use of 2 as the first argument prevents a frame advance and the display of corner marks and frame identification. (The corner marks and frame identification were displayed only once, by the first statement.)

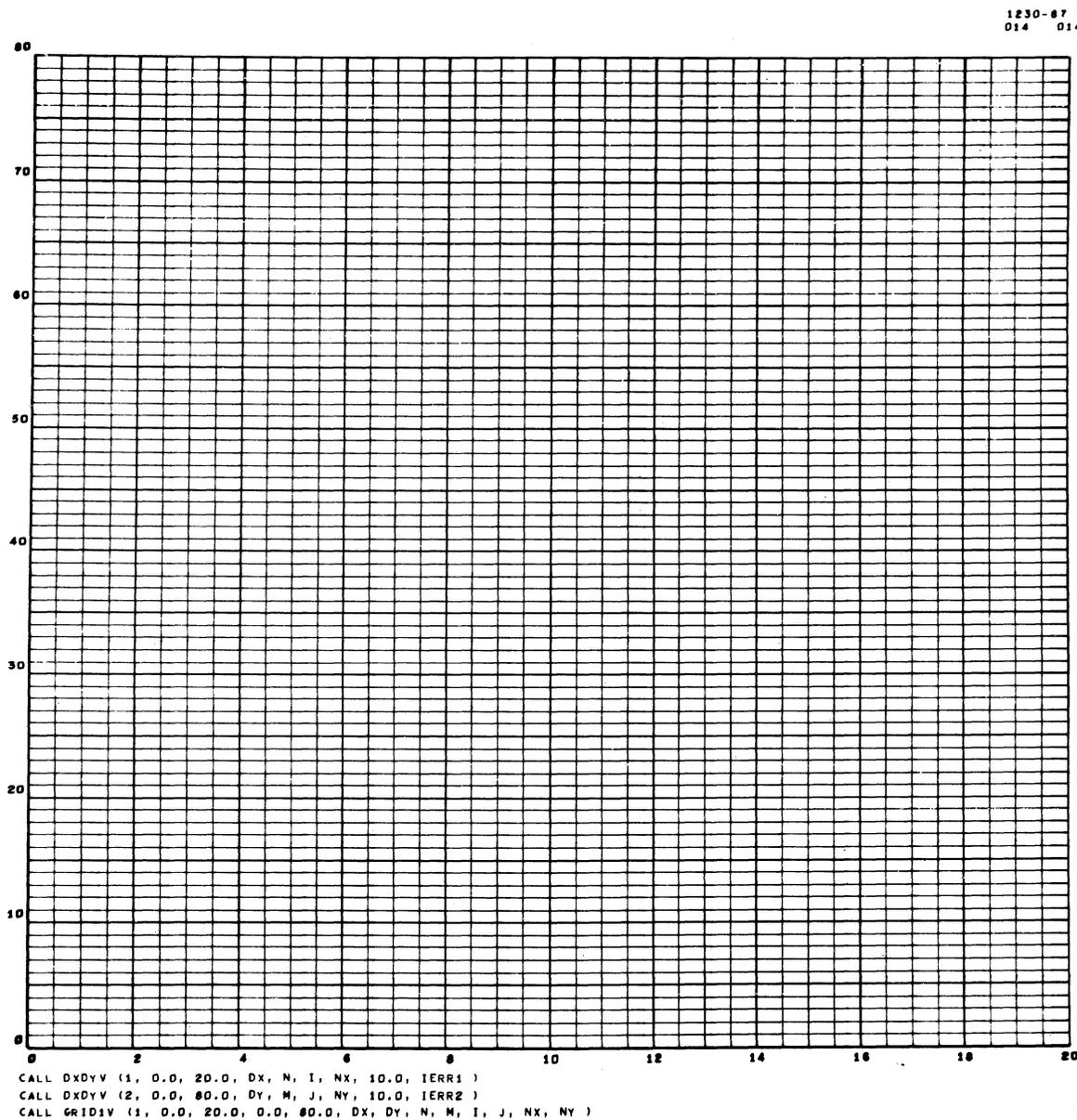


Figure 1-h

CALL DXDYV (1, 0.0, 20.0, DX, N, I, NX, 10.0, IERR1)

CALL DXDYV (2, 0.0, 80.0, DY, M, J, NY, 10.0, IERR2)

CALL GRID1V (1, 0.0, 20.0, 0.0, 80.0, DX, DY, N, M, I, J, NX, NY)

(IERR1 and IERR2 should be tested for nonzero values, and appropriate action taken, before the CALL GRID1V statement.)

10-15-63

MACHINE COMPUTATION OF GRID1VARGUMENTS: DXDYV

It frequently happens that the programmer does not have sufficient advance information about the range of data his program will encounter to be able to assign practical values to all arguments of GRID1V. In this case, a series of FORTRAN statements can be used to determine the upper and lower X and Y bounds. For example, the values of XL and XR for a block of data, X, can be computed as follows:

$$XL = X(1)$$

$$XR = X(1)$$

$$D\emptyset 10 J = 2, NPTS \quad \text{where NPTS is the number of points in the X block of data}$$

$$XL = \text{MIN1F}(XL, X(J))$$

$$10 \quad XR = \text{MAX1F}(XR, X(J))$$

A similar group of statements can be used to compute YB and YT for the Y block of data.

Once XL and XR (or YB and YT) are known, the routine DXDYV is available to compute arguments for line spacing, line emphasizing, and line labeling. Two call statements are available, one for the X direction and one for the Y direction. They are:

$$\text{CALL DXDYV}(1, XL, XR, DX, N, I, NX, DC, \text{IERR})$$

$$\text{CALL DXDYV}(2, YB, YT, DY, M, J, NY, DC, \text{IERR})$$

On each entry to DXDYV, four arguments are furnished by the programmer:

The first argument is a 1 or a 2, to indicate whether DXDYV is being applied in the X direction or in the Y direction.

XL and XR (or YB and YT) are defined as in the summary of GRID1V arguments, on page 74. 10. 05 .

DC represents a floating point quantity which limits the density of the grid. The grid lines drawn by GRID1V, using arguments furnished by DXDYV, will be no closer than DC raster positions. DC should never have a value less than 3.0; values of 8.0 to 20.0 are recommended.

The remainder of the arguments are variables to which DXDYV will assign values. Any value previously assigned these variables will be destroyed during execution of the subroutine. NEVER USE CONSTANTS FOR THESE ARGUMENTS.

DX, N, I, NX (or DY, M, J, NY) are defined as in the summary of the GRID1V arguments, on page 74. 10. 05 .

IERR is an error indicator. It will be set to zero if no error is detected, or to non-zero if an error is encountered. After execution of DXDYV, IERR should always be tested. If it is non-zero, GRID1V will probably not produce a satisfactory picture with the computed argument.

Figure 1-h is an illustration of the use of DXDYV. Note that the argument names used in DXDYV are also used for the corresponding terms in GRID1V.

Note: DXDYV has not been modified since the scientific label capability was added to the system. Until this modification has been made, the programmer who uses DXDYV and also wants scientific labels must include some extra provisions in his program.

For one thing, the values of NX and NY derived from DXDYV must be set to -NX and -NY by the programmer. Also, NX and NY must not exceed 6.

Because the scientific labels are longer (by at least seven characters), there is considerable danger that the labels of adjacent vertical grid lines will run into one another. Usually, this can be prevented by inserting the following statement between the DXDYV statement and the call to GRID1V:

$$I = 2*I$$

If the grid density indicator, DC, is large enough (say, $DC \geq 15.0$), it may be sufficient to space the labels without taking other action.

INTENSITY SELECTION: BRITV, FAINTV

GRID1V ensures that the bright intensity mode is on. Normally, this intensity mode should be left on, since experience has shown that it produces the best results. If the programmer wants to change this setting to the faint mode, he can use the following statement:

$$\text{CALL FAINTV}$$

Then, to restore the bright intensity mode, he can use the statement:

$$\text{CALL BRITV}$$

(Additional forms of these statements will be shown in Section 74.6, in the discussion of the "current point.")

S-C 4020 STANDARD SCIENTIFIC CHARACTERS

<u>NUMERIC</u>		<u>ALPHABETIC</u>		<u>OTHER FORTRAN</u>		<u>NON-FORTRAN</u>	
<u>CHAR</u>	<u>DEC EQUIV</u>	<u>CHAR</u>	<u>DEC EQUIV</u>	<u>CHAR</u>	<u>DEC EQUIV</u>	<u>CHAR</u>	<u>DEC EQUIV</u>
0	0	A	17	=	11	→ ∂	10
1	1	B	18	+	16	"	12
2	2	C	19	-	32	' (PRIME)	13
3	3	D	20	*	44	δ	14
4	4	E	21	/	49	α	15
5	5	F	22	. (PERIOD)	27	?	31
6	6	G	23	,	59	β	29
7	7	H	24	(60	±	30
8	8	I	25)	28	π	26
9	9	J	33	\$	43	→ · (DOT)	42
		K	34	(BLANK)	48	γ	45
		L	35			→ ~	46
		M	36			α	47
		N	37			° (DEGREE)	58
		O	38			∫	61
		P	39			∫	62
		Q	40			□	63
		R	41	→	·, ∂, ~ NOT AVAILABLE IN TYPEWRITER MODE		
		S	50				
		T	51				
		U	52				
		V	53				
		W	54				
		X	55				
		Y	56				
		Z	57				

<u>SUGGESTED PLOTTING CHARACTERS AND COMBINATIONS</u>			
<u>CHAR</u>	<u>DEC EQUIV</u>	<u>CHAR</u>	<u>DEC EQUIV</u>
· (DOT)	42	H	24
O	38	U	52
□	63	∅	∅
+	16	∅	38, 49
X	55	∅	63, 49
*	44	⊗	38, 55
C	19	⊗	63, 55

Figure 1-i

See page 74.60.02 for additional information on the selection and placement of individual Charactron characters.

PLOTTING, PRINTING, AND LABELING: USING CHARACTERON CHARACTERS

The 64 Characteron characters are shown in Figure 1-i, along with the selection code (decimal equivalent) for each. For the standard FORTRAN characters, the octal forms of the selection codes are shown on the table on page 60.91.01, in the columns headed "Storage." Note that, for these characters, the selection codes are the same as the BCD codes recognized by IBM equipment.

Reduced to fundamentals, printing or plotting of one Characteron character involves the selection and display of that character at a specified position on the raster. (A basic subprogram, PLØTV, can be used to select and display one character at a time.) The higher-level subprograms, however, contain features that make each one suitable for a specialized purpose: plotting, printing, or labeling. These specialized features of the higher-level routines make them appear to be distinctly different from each other.

Since some of these routines permit arguments in Hollerith form, it may be helpful to review the discussion of Hollerith arguments on page 60.42.03 before going further.

POINT PLOTTING SUBPROGRAMS

Since point plotting usually involves the scaled representation of a physical phenomenon, most point plotting subprograms accept physical data coordinates for position information. During execution of these subprograms, the data coordinates are converted into raster coordinates. Scale factors must have been established for the plotting routines to use in making these conversions; this requirement can be satisfied by a prior entry to GRID1V.

Scale factors are not normally saved from link to link in a chain job. Consequently, plotting should be done within the link in which scale factors are computed.

(The student is reminded that the First Reader does not attempt to describe all the facilities in the S-C 4020 system. Scale factors can be established by other methods than by entry to GRID1V, and these factors can be carried over to a new chain link. Discussion of these and other features not needed for the average use of the S-C 4020 as a plotter have been purposely deferred to later sections.)

Plotting an Array of Points: APLØTV

APLØTV was designed for situations in which a large number of X values are stored in one array and the corresponding Y values are stored in another array. It is possible to plot the entire set of data with one entry to APLØTV. If desired, only a portion of the data can be plotted.

It is also possible to use different Characteron characters as symbols to identify curves. The programmer furnishes a table (of one or more Characteron characters) that will be selected cyclically.

APLØTV also keeps a tally of the number of off-scale points encountered.

The calling statement is:

```
CALL APLØTV (±N, XARRAY, YARRAY, JX,
             JY, ±NC, MRKPT, IERR)
```

N Controls the number of points to be plotted. N is actually the number of points if all the data points in the arrays are to be plotted in succession. The value of N may be computed by letting K in the following formulas equal the number of points to be plotted:

$$N = K * JX \text{ or } N = K * JY, \\ \text{whichever is larger.}$$

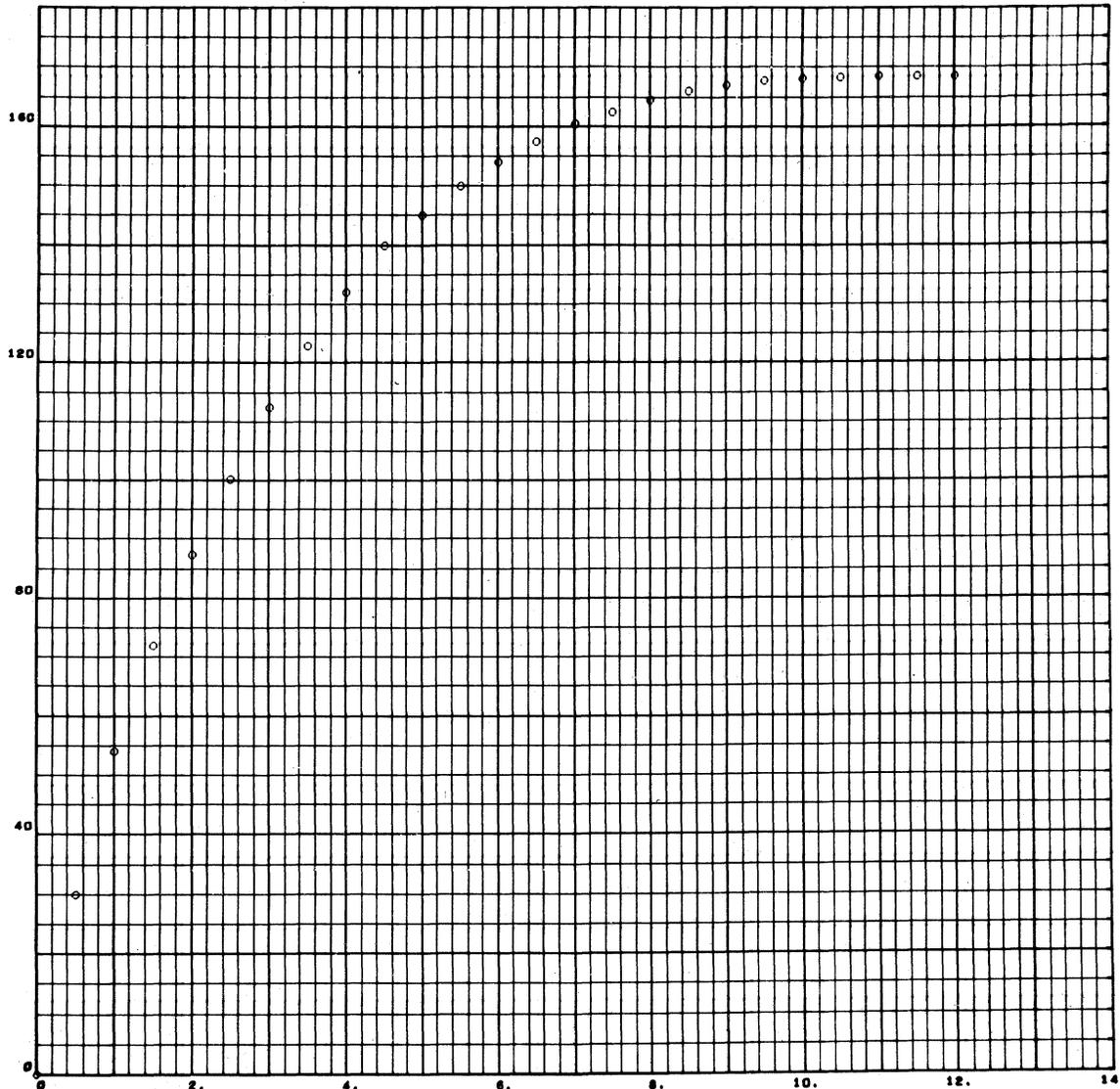
Usually, N should be positive, to indicate that the data arrays are in normal FORTRAN order of storage. However, if the arrays are in increasing order of storage, N should be negative.

XARRAY
YARRAY Normally, the names of the arrays of floating point data to be plotted. Since these arguments must name the locations of the data coordinates of the first point to be plotted, subscripts may be necessary. Both arrays must be in the same order of storage.

JX, JY Fixed point positive integers giving the increments to be added to the subscripts of XARRAY and YARRAY as each point is plotted.

DATA IS STORED X1,X2,...X25 AND Y1,Y2,...Y25

1230-87
018 018



CALL GRID1V (1, 0.0, 14.0, 0.0, 180.0, 0.2, 5.0, 5, 4, 10, 8, 3, 3)
 CALL APL0TV (25, X, Y, 1, 1, 1, 30, IERR) OR CALL APL0TV (25, X, Y, 1, 1, 1, 1HO, IERR)

Figure 1-j

It is assumed that the data are stored in two arrays: $X(1), X(2), \dots, X(25)$, and $Y(1), Y(2), \dots, Y(25)$. A prior entry was made to GRID1V, using the statement:

CALL GRID1V (1, 0.0, 14.0, 0.0, 180.0, 0.2, 5.0, 5, 4, 10, 8, 3, 3)

Note that the X and Y limits cover the range of the data to be plotted. Also, since GRID1V was entered first, the necessary scale factors have been established for APL0TV. The calling statement of APL0TV for this example is on the following page.

10-15-63

CALL APLØTV (25, X, Y, 1, 1, 1, 38, IERR)
 or CALL APLØTV (25, X, Y, 1, 1, 1, 1HØ, IERR)

N = 25 The array size, 25 in this case, is equivalent to the number of points to be plotted. This can only be true when all points in each array are to be plotted.

X, Y Names of the X and Y arrays (properly dimensioned).

JX = 1
JY = 1 Since every X versus Y is to be plotted, the fourth and fifth arguments are set to 1.

NC = 1 Only one plotting symbol is to be used; hence the sixth argument is set to 1.

MARKPT In the two calling sequences shown, one specifies the plotting symbol by an integer; the other uses a Hollerith argument.

NC The number of characters in the array **MRKPT** to be used as plotting symbols (usually the size of the array **MRKPT**). Positive **NC** indicates that **MRKPT** is a normal FORTRAN array. Negative **NC** shows that **MRKPT** is stored in increasing locations in core.

MRKPT The array that contains the selection codes of the plotting character(s) to be used. These will be used cyclically, with the first one being used again after the **NC**th one has been used.

To set up this information when only one character is to be used, the argument can be one of the following:

- a. An integer selection code, as "42" for the plotting dot, "38" for the alphabetic letter Ø, etc.
- b. A Hollerith argument for one FORTRAN character, as "1HØ."
- c. The name of a location containing an integer, as "MRKPT."

If more than one character is desired, the seventh argument

may be one of the following:

- a. An array of integers read in by the I format, or generated by a series of arithmetic statements: **MRKPT(1) = 38**; **MRKPT(2) = 55**, etc. Such an array must have a fixed point variable name.
- b. An array read in by the A format, one character to a location. Such an array is not restricted to integer-type names; it could be called **PTMRK**, for example, if the contents are not used in a computation.

IERR

The name of an error location supplied by the programmer. Any point that falls outside the grid drawn by **GRIDIV** will not be plotted. Instead, a count of such points will be stored in **IERR**.

APLØTV plots only one character per point. If a center dot is desired, **APLØTV** may be repeated, using the plotting dot as the symbol.

Figures 1-j and 1-k illustrate the use of **APLØTV** for one curve and for a family of curves, respectively. The tests for **IERR** are not shown.

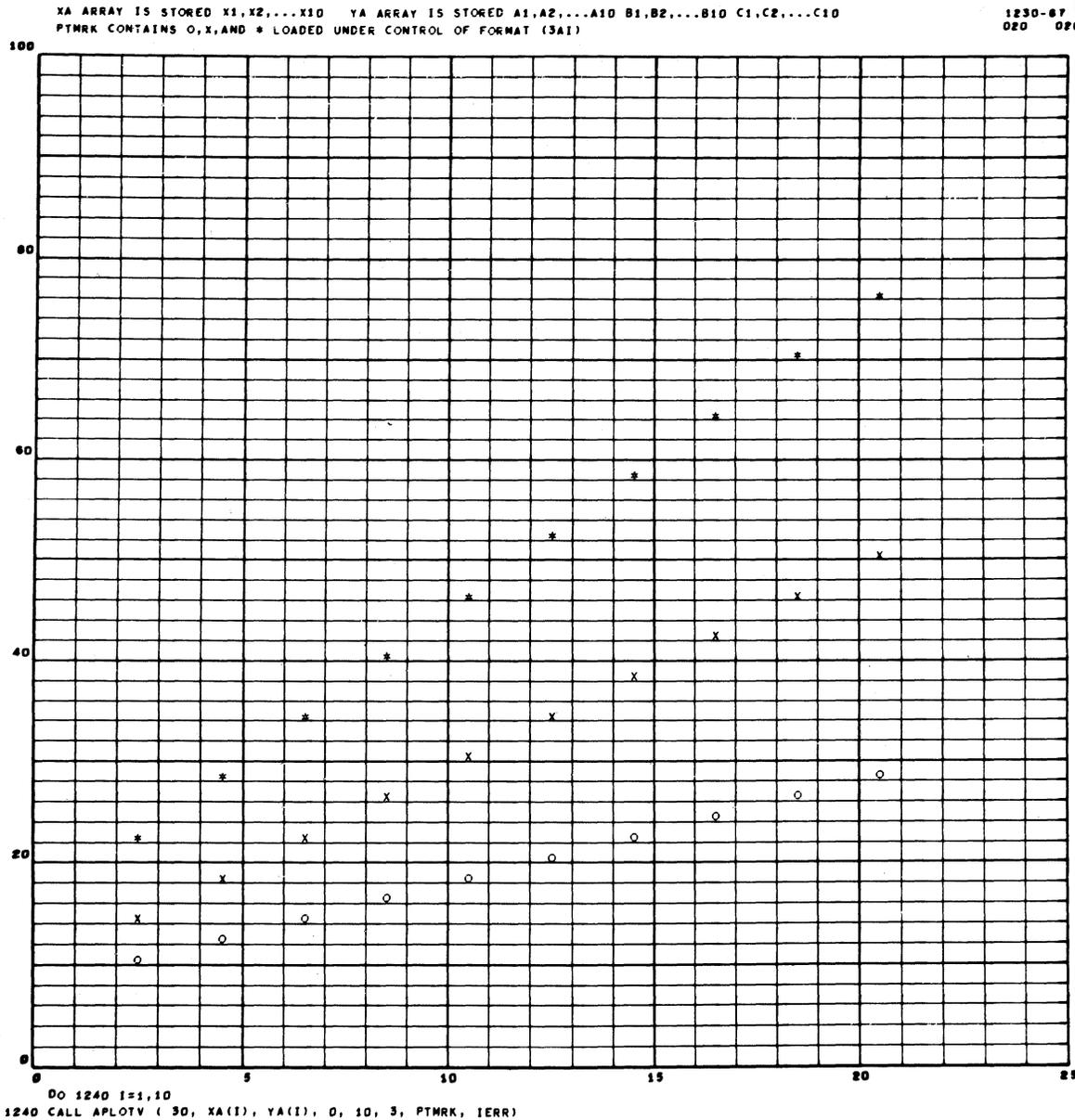


Figure 1-k

This exercise shows one way that APLØTV might be used to plot a family of curves. The example assumes that the XA and YA arrays are properly dimensioned. The data is stored XA(1), XA(2), . . . , XA(10); corresponding Y's for the three curves are stored in the array YA in the order A(1), A(2), . . . , A(10), B(1), B(2), . . . , B(10), C(1), C(2), . . . , C(10). PTMRK is the name of a three-word array which was loaded as the BCD equivalents (read in by the A format) of the characters Ø, X, and *.

In the example, A(I), B(I), and C(I) were plotted versus X(I) each time APLØTV was entered. A DØ loop was used to proceed to the next value of X, so that a total of 10 entries were made to APLØTV. The coding was:

```
DØ 1240 I = 1,10
```

```
1240 CALL APLØTV (30, XA(I), YA(I), 0, 10, 3, PTMRK, IERR)
```

Note that the plotting symbols are used cyclically, returning to the first one when the array PTMRK is exhausted. If desired, a center plotting dot can be superimposed upon the plotting symbol by repeating the entries to APLØTV with the plotting dot used for PTMRK.

Plotting Individual Data Points: PØINTV

For each entry to PØINTV, one symbol is plotted (with or without a center dot). The coordinates may be specified as floating point data, which PØINTV will convert into raster coordinates. Scale factors must have been established; this can be accomplished by a prior entry to GRID1V.

PØINTV relieves the programmer of the burden of setting up a symbol table. The most suitable plotting symbols derived from Charactron characters are arranged in a table, with the best choices near the beginning of the group. The symbols are selected by using integers from 1 to 48. (Note that these integers do not correspond to the decimal equivalents of the Charactron characters shown in Figure 1-i.) The calling statement is:

CALL PØINTV (X, Y, ±NS)

- X, Y Coordinates of the point to be plotted, stated as floating point data values.
- ±NS Integer which selects a plotting symbol. If NS is minus, there will be no center dot plotted. If NS is positive, a center dot will appear.

Points outside the scaled area will not be plotted. Until the programmer learns how to detect off-scale points (by a method explained in the advanced material), he should be sure that the data coordinates will fall within the limits of the scaled area.

An alternate version of this subprogram allows the programmer to specify position information in raster coordinates. This is particularly useful when the programmer wants to construct a legend in the margin, showing the symbols used and their meaning. The alternate call statement is:

CALL PØINTV (IX, IY, ±NS, ANY)

- IX, IY Raster coordinates of the point to be plotted; fixed point integers.
- ±NS Integer to select a plotting symbol, as stated above.
- ANY The inclusion of a fourth argument signals the subroutine that the position information is specified as raster coordinates. The form of this argument is immaterial; it may be any fixed point or floating point variable or constant.

The available symbols are shown in Figure 1-1. An illustration of the use of PØINTV appears in Figure 1-m.

The Basic Plot-Print Subprogram: PLØTV

Any one of the 64 Charactron characters can be displayed at a specified raster position by using PLØTV, the basic subprogram used as a lower-level module of other routines. However, the programmer may find it useful when other plotting or printing subprograms are not suitable. The call statement is:

CALL PLØTV (IX, IY, NS)

- IX, IY Fixed point raster coordinates at which the character will be displayed.
- NS An integer which selects the character. The selection code (decimal equivalent) of the desired character should be used. (See Figure 1-i.)

NS	SYMBOL								
0	.	10	H	20	Ø	30	(40	F
1	⊙	11	€	21	∕	31	~	41	M
2	x	12	v	22	⊖	32)	42	P
3	⊠	13	f	23	⊖	33	N	43	R
4	v	14	Z	24	x	34	T	44	Y
5	+	15	⊗	25	I	35	=	45	2
6	*	16	⊠	26	G	36	G	46	3
7	⊥	17	⊕	27	S	37	A	47	7
8	W	18	⊠	28	w	38	B	48	8
9	G	19	Ø	29	Ø	39	E		

Figure 1-1

THE SAME DATA PLOTTED USING POINTV INSTEAD OF APLQTV

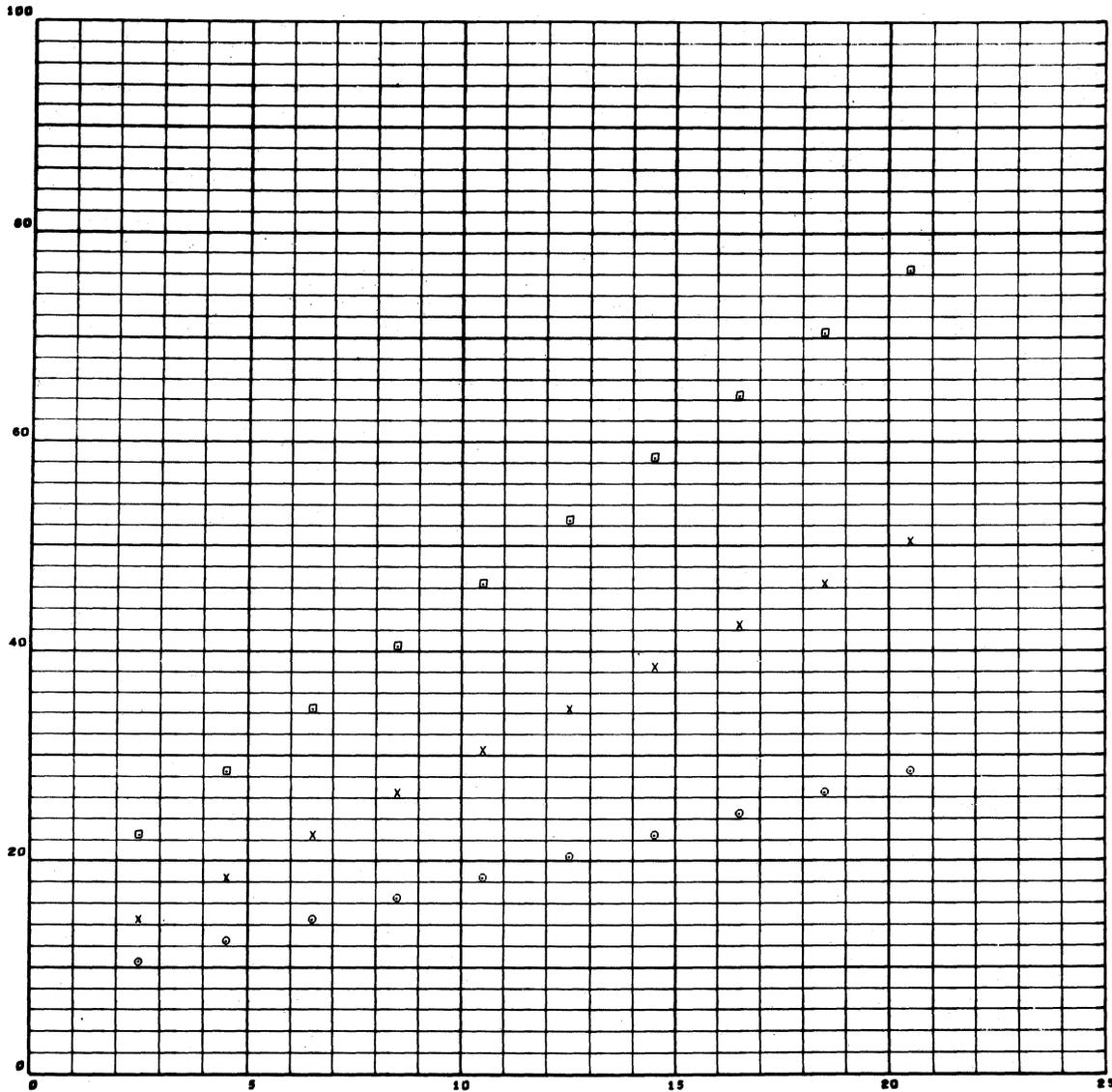
1230-87
022 022

Figure 1-m

POINTV

For this example, the same GRID1V statement and the same data were used as were used for Figure 1-k. The programming statements follow; note that they are somewhat longer than the sequence required for Figure 1-k.

```
DØ 1390 I = 1,3
```

```
K = (I - 1) * 10
```

```
DØ 1390 J = 1,10
```

```
K1 = K + J
```

```
1390 CALL POINTV (XA(J), YA(K1), I)
```

10-15-63

PRINTING AND LABELING SUBPROGRAMS

The printing and labeling subprograms enable the programmer to affix titles and other identifying information to a picture. Three subprograms of this type will be introduced in the First Reader: PRINTV, APRNTV, and LABLV. Other means for printing and labeling will be given in the more advanced material.

For many applications, the positions of titles and labels must be independent of the scale. Therefore, printing and labeling subprograms accept position information in raster coordinates. This contrasts with the plotting routines, which include facilities for the conversion of data into raster counts. (When titles or labels must be positioned relative to data, the conversion functions discussed on page 74.10.25 can be employed.)

Typewriter Printing: PRINTV

To review, the S-C 4020 has a built-in Typewriter Simulator, which prints a series of Charactron characters very rapidly. The typewriter mode allows 8 raster counts for the width of a character space. If "typewriting" carries over to the next row, the next printing will start at the left edge of the frame, 16 raster counts below the preceding row. Writing takes place only in horizontal rows, and the characters are always upright.

In the Typewriter Mode, the machine recognizes octal codes 56 and 12 as special instructions, which are not compatible with the use of PRINTV. If these codes are accidentally used, imperfect pictures will result.

The PRINTV calling statement provides for printing characters read in by the A format, or characters specified in the call statement itself in the nH----- form.

CALL PRINTV (N, BCDTXT, IX, IY)
or CALL PRINTV (-N, nH-----, IX, IY)

N The number of Charactron characters to be printed. Notice that a negative sign on N signals that the second item is a Hollerith argument.

The typewriter mode is used by PRINTV until N characters have been printed.

BCDTXT An array containing the BCD

(A- or Ø-type format) text to be printed.

nH----- A Hollerith argument containing the text to be printed.

IX, IY The raster coordinates for the center of the first character. IX and IY may be any number from 0 to 1023.

Figure 1-n includes a horizontal title written by PRINTV.

Note: If octal code 52 appears as one of the characters in BCDTXT, a carriage return will result. (A carriage return will also result if the position of the next character would otherwise fall to the right of the raster.)

Vertical Titles: APRNTV

This subprogram can be used to display vertical titles composed of Charactron characters. Each individual character will be upright. The call statement provides for printing characters in BCD form (read in by the A format) or characters stored as a Hollerith argument. Since APRNTV does not use the Typewriter Mode, the spacing of the characters is controlled by arguments specified by the programmer.

The call statement for APRNTV is:

CALL APRNTV (INCRX, INCRY, N, BCDTXT,
IX, IY)

or CALL APRNTV (INCRX, INCRY, -N,
nH-----, IX, IY)

INCRX Increments used to space the characters in the X or Y direction, given in raster counts. For vertical titles, INCRX will be zero and INCRY should have a negative value. (It is suggested that INCRY fall in the range between -12 and -18 for vertical titles in most applications.) Figure 1-n shows a vertical title, with INCRY equal to -14. Using a slightly higher value for the Y increment would have spaced the letters farther apart.

The remaining arguments are as specified under PRINTV, above.

EXAMPLES OF FRAMEV, PRINTV, APRNTV, AND LABLV

1230-87
024 024V
E
R
T
I
C
A
L

T
I
T
L
E

U
S
I
N
G

A
P
R
N
T
V

HORIZONTAL TITLES USING PRINTV

MACH NO.= 1.3

150.5
85.7
0.3Figure 1-n

The following FORTRAN statements were used for these examples.

```

260 CALL FRAMEV
CALL PRINTV(- 31, 31HORIZONTAL TITLES USING PRINTV ,200,925)
CALL APRNTV(0,-14,-28,28H VERTICAL TITLE USING APRNTV ,0,900 )
CALL PRINTV(- 9, 9HMACH NO.= , 200,850 )
C ASSUME VALUE IN XMACH IS 1.3 (FLOATING POINT)
CALL LABLV(XMACH,280,850,3,1,1)
C ASSUME D(1), D(2), AND D(3) CONTAIN 150.5, 85.7, AND 0.3
IY=775
DO 410 J=1,3
CALL LABLV(D(J),200,IY,5,1,3)
410 IY=IY-16

```

Display of Non-FORTRAN Characters

The characters in the last column of the table of Characteron characters (Figure 1-i) require special consideration. IBM equipment does not provide for key-punching these characters, so they cannot appear in a Hollerith argument nor on an A-type format card. However, the octal equivalents of the selection codes can be read into storage by the Ø-type format, and APRNTV or PRINTV can be employed to display them.

A single non-FORTRAN character can easily be displayed by the basic plot-print subprogram, PLØTV, or by APLØTV, using a decimal integer as the selection code. If a character is being inserted into information otherwise printed by PRINTV, the fact that PRINTV allots 8 raster counts to each character must be taken into consideration.

Fixed Point Labels: LABLV

LABLV was developed for GRID1V to employ in labeling grid lines, but will prove helpful when values of computed quantities must be printed. The routine performs two chief tasks: it converts a floating point quantity into a number in BCD format, with the decimal point in the proper position, and it displays the BCD number at the raster coordinates specified.

Before the BCD label is displayed, it is right adjusted in accordance with the specified decimal scale. (The number of positions to the left of the decimal point is called the "decimal scale.") During the right adjustment, BCD blanks are moved into the high order positions that are vacated. A specified number of characters (counting leading blanks and decimal point) will then be displayed.

The result of this procedure is demonstrated by the column of numbers: 150.5, 85.7, 0.3 in Figure 1-n. Notice that the three quantities have been truncated to the same number of decimal places, and that the decimal points have been aligned.

The calling statement of LABLV is:

```
CALL LABLV (D, IX, IY, NCHAR, NT,
            NDMAX)
```

- | | |
|--------|--|
| D | The floating point quantity to be printed. |
| IX, IY | The raster coordinates which will position the first character of the label. Note that this first character may be a leading blank. If |

the quantity to be displayed is negative, the minus sign will be displayed one character space to the left of IX, IY.

- | | |
|-------|---|
| NCHAR | Number of characters to be displayed, including leading blanks and the decimal point, if any. NCHAR is limited to 6 (or 7 if one of the characters is the decimal point). |
| NT | The number of times each character is to be displayed (number of over-strikes). Normally this should be 1, but 2 may be chosen if a darker label is desired. |
| NDMAX | Maximum decimal scale; i.e., maximum number of characters to be displayed to the left of the decimal point. |

An integer quantity may be displayed by first changing it to floating point form and then using LABLV.

Further Explanation of Figure 1-n

Examination of Figure 1-n reveals more details of FRAMEV, PRINTV, APRNTV, and LABLV.

For one thing, this frame demonstrates the use of the S-C 4020 when no grid is drawn and no scaling of any kind is done. Position information is stated in raster coordinates in each instance. In all the preceding examples, the frame was advanced by the first argument of GRID1V; in this illustration, a CALL FRAMEV statement was used.

The entries to PRINTV and APRNTV show the use of Hollerith arguments to specify the text to be printed. Note that the argument which gives the number of characters to be printed (N) is written with a minus sign to indicate that the characters are stored in reverse of the normal storage of FORTRAN arrays.

This illustration also shows the Characteron characters may be positioned on the frame boundary lines (raster count 0 to 1023). Although the vertical title shown is positioned at IX = 0, complete Characteron characters are displayed in spite of the fact that portions of them extend beyond the frame boundary.

The line reading "MACH NO. = 1.3" illustrates the insertion of a computed quantity into a printed statement. The "MACH NO. =" was written by

PRINTV and the value "1.3" was inserted by LABLV. The label was positioned 80 raster counts beyond the start of "MACH NO. =" to allow for ten character spaces (8 counts in width). Since no right adjustment of the label value was required in this example, NDMAX was specified as the actual decimal scale of the quantity displayed.

The column of three numbers illustrates the right adjustment feature of LABLV. NCHAR is 5, specifying the number of characters (counting the decimal point) in the largest number, 150.5. The decimal scale (NDMAX) of 3 specifies the largest number of characters required to the left of the decimal point. The quantities are right adjusted to the maximum decimal scale, so that the decimal points are in line.

Scientific Notation Labels: LABLV

An alternate form of LABLV may be used to display labels in scientific notation. The call statement shown on page 74.10.23 is used with the following changes:

- NCHAR Number of significant figures to be displayed. NCHAR may be less than or equal to 6. The negative sign will result in the use of scientific notation.

NDMAX May be any fixed point quantity. Since the right adjustment of these labels will not be necessary, the value of NDMAX will be ignored.

NCHAR will affect the format in the following ways:

<u>NCHAR</u>	<u>Format</u>
1	Y.10±YY
2	Y.Yx10±YY
3	Y.YYx10±YY
4	Y.YYYx10±YY
5	Y.YYYYx10±YY
6	Y.YYYYYx10±YY

Since the space required for these labels will be greater than that required for the fixed point format, the programmer should allow NCHAR + 7 Character character spaces in width and 1-1/2 spaces in height as a minimum; it may be necessary to allow even more to avoid overlapping other images.

The following illustrates a scientific and a fixed point label.

```
CALL LABLV ( 579.5, 625, 1023, 5, 1, 3 ) RESULTS IN 579.5
CALL LABLV ( 579.5, 625, 999, -4, 1, 3 ) RESULTS IN 5.795x10+02
```

CONVERSION: NXV, NYV

For many applications, the placement of printing and labeling is independent of the values of the data being plotted. For this reason, the printing and labeling subprograms have been designed to accept position information specified directly in raster coordinates. In addition, many other S-C 4020 subprograms, particularly the most basic ones, are designed to accept position information specified in raster coordinates. For subprograms of this type, it is possible to position a display relative to data values by making use of the two conversion functions, NXV and NYV.

To convert data value X into raster value IX, and data value Y into raster value IY, the following FORTRAN statements may be utilized:

$$IX = NXV(X)$$

$$IY = NYV(Y)$$

Scale factors must have been established for use by these functions. This can be done by a prior entry to GRIDIV.

If either the NXV or the NYV function determines that the quantity to be converted is off-scale (outside the X or Y limits), the result will be set to zero. For most plotting, it can be assumed that a zero result indicates an error, and the programmer should provide an error procedure. (If further information is needed, see Section 74. 4.)

LINE GENERATION: LINEV, XAXISV, YAXISV

LINEV connects two points by a straight line composed of vectors, joined end-to-end. The arguments for LINEV, which specify the points to be connected, must be given in raster counts. As described above, the programmer may connect two data points by a line if he first uses the functions NXV and NYV to convert the data coordinates into raster coordinates. (If there is a possibility that the data points being converted may be off-scale, the conversion results should be tested for errors before LINEV is executed.)

The calling statement is:

$$CALL \text{LINEV} (IX1, IY1, IX2, IY2)$$

IX1, IY1 Raster coordinates of one end point.

IX2, IY2 Raster coordinates of the other end point.

Figure 1-0 contains an illustration of the use of LINEV. This frame is also a summary of the material in the First Reader; it is discussed in detail at the end of this section.

XAXISV, YAXISV enable the programmer to use the axis line feature of the S-C 4020 to generate horizontal or vertical lines. Horizontal axis lines started at a specified raster position will be swept to the right edge of the frame. Vertical axis lines started at a specified raster position will be swept to the top of the frame. Since the lines cannot be stopped short of the edge of the frame, this feature is not used by GRIDIV.

The call statement for sweeping a horizontal line is

$$CALL \text{XAXISV} (IX, IY)$$

IX, IY The fixed point raster coordinates of the origin of the line. IX may have a value from 0 to 959, and IY may have a value from 0 to 1023.

For a vertical line, the statement is

$$CALL \text{YAXISV} (IX, IY)$$

IX, IY The fixed point raster coordinates of the origin of the line. IX may have a value from 0 to 1023, while IY may have a value from 0 to 959.

ERROR MARKS

Certain types of errors are indicated on the frame by one of the following error marks.

- a. Void Mark. This is a solid circle of approximately 50 raster counts in diameter, which the S-C 4020 places in the margin under certain conditions.

Void Mark

1230 89 L
008 008

When the tape drive unit is stopped because of a parity error, the operator will press the "RESTART" key, which will cause the tape drive unit to go backward to the preceding frame advance on the tape which

signifies the beginning of the frame containing the error. The machine then automatically advances the film and attempts to read the page again. If the error is again encountered, the operator will press the "START" key to enable the machine to continue. When either "RESTART" or "START" is pressed, the "Void" mark is placed in the margin.

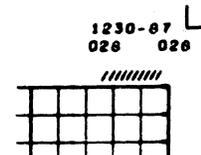
- b. Error Symbol. The Charactron character □ is displayed by the S-C 4020 if a parity error occurs when printing or plotting; that is, the specifications for a symbol are garbled. It is positioned at the coordinates contained in the erroneous word. This error symbol is not displayed if the erroneous word is one which normally does not display a character, for example, an axis or vector generation word. In addition to displaying the error symbol, the tape drive is stopped when a parity error is detected. A frame with such a mark will usually have a Void Mark as well.

Error Symbol

8603085
 8780.09
 8868.60
 8957.29
 9046.86

- c. GRID1V Error Mark. A series of slash marks will be placed in the upper right corner of the frame when the arguments furnished to GRID1V are inadmissible quantities that would result in stopping the machine or causing a loop. When GRID1V encounters such bad data, it places the error mark on the frame and makes some kind of adjustment in data values so the computation can proceed.

GRID1V Error Mark



Although a picture with this mark should be considered unreliable, it may be useful for detecting the cause of the error. Unlike the preceding two marks, this error indication is not a feature of the S-C 4020 hardware, but is created under GRID1V control.

10-15-63

REVIEW OF FIRST READER

Figure 1-0 summarizes the routines that have been discussed in the First Reader. It contains points plotted on a labeled grid, the points are connected by straight lines, titles are printed in the margins, the value of a variable quantity is printed in a title, and information is printed to identify each curve.

Also for this illustration, the construction of the

grid was made dependent upon the input data by letting the program make a search for maximum and minimum values of X and of Y. DXDYV was then used to compute the remaining arguments for GRID1V.

The following program was used to produce Figure 1-0. (This program was written as an exercise rather than as a model of excellence in programming.)

```

DIMENSION  XX(10),YY(30) ,AMRK(3)
C          THE XX ARRAY IS STORED X1,X2...X10          YY ARRAY IS STORED
C          A1,A2...A10,B1,B2...B10,C1,C2...C10
CALL CAMRAV(935)
READ INPUT TAPE 5,1450,AMRK
1450 FORMAT (3A1)
READ INPUT TAPE 5,1490 ,XX
READ INPUT TAPE 5,1490,YY
READ INPUT TAPE 5,1490,THKNS
1490 FORMAT (6F12.6)
XL=XX(1)
XR=XX(1)
DO 1540 J=2,10
XL=MIN1F(XL,XX(J))
1540 XR=MAX1F(XR,XX(J))
YB=YY(1)
YT=YY(1)
DO 1590 J=2,30
YB=MIN1F(YB,YY(J))
1590 YT=MAX1F(YT,YY(J))
CALL DXDYV (1,XL,XR,DX,N,I,NX,10.0,IERR1)
CALL DXDYV (2,YB,YT,DY,M,J,NY,10.0,IERR2)
CALL GRID1V(1,XL,XR,YB,YT,DX,DY, N,M,I,J,NX,NY)
CALL APRNTV(0,-12,-16,16HTEMPERATURE F. , 4,576)
CALL PRINTV (-11,11HTIME (SEC.) ,468,4)
CALL PRINTV (-10,10HTHICKNESS= ,200 ,1015 )
CALL LABLV(THKNS,296,1015,4,1,1)
DO 1670 I=1,10
1670 CALL APL0TV(30,XX(I),YY(I),0,10,3,AMRK,IERR )
DO 1830 J=1,3
K=(J-1)*10
NX1=NXV(XX(1))
NY1=NYV(YY(K+1))
C          NOTE..ZERO GENERALLY INDICATES OFFSCALE POINT
IF(NX1*NY1) 1760 ,1840,1760
C          IN PRACTICAL PROBLEMS A MORE COMPLETE ERROR PROCEDURE IS SUGGESTED
1760 DO 1830 I=2,10
NX2=NXV(XX(I))
K1=K+I
NY2=NYV(YY(K1))
IF(NX2*NY2)1800,1830,1800
1800 CALL LINEV(NX1,NY1,NX2,NY2)
NX1=NX2

```

```
NY1=NY2
1830 CONTINUE
1840 CONTINUE
NX1=NXV(XX(8))
NY1=NYV(YY(8)) -3
NY2=NYV(YY(18))-3
NY3=NYV(YY(28))-3
NXX= NX1+14
CALL PRINTV (- 7, 7HMETAL A   ,NXX,NY1 )
CALL PRINTV (- 7, 7HMETAL B   ,NXX,NY2 )
CALL PRINTV (- 7, 7HMETAL C   ,NXX,NY3 )
CALL FRAMEV
CALL EXIT
END
* DATA
```

In conclusion, the student of the First Reader should remember that the write-ups contained in its pages do not include many of the details which make these routines flexible for use in a variety of situations. In addition, there are many routines available other than those discussed here.

After the student has tried the material in the

First Reader for some plotting problems, he should have enough experience to proceed to the advanced material. Sections 74.2 and 74.3 contain routines that will enable the programmer to produce more elaborate graphs, and will also make it possible to employ the S-C 4020 for extended applications.

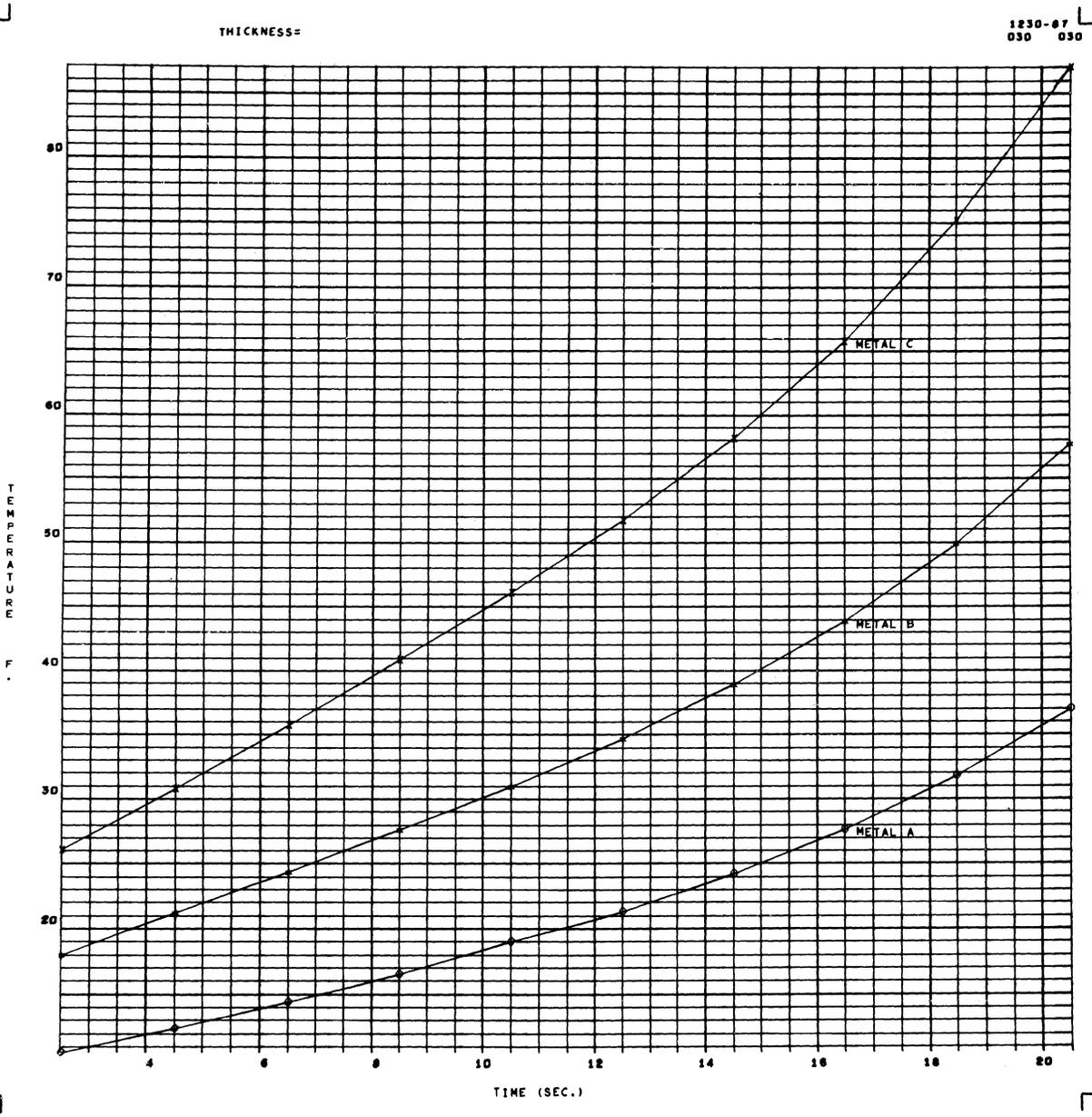


Figure 1-o



Section 74.2SPECIAL PRINTING AND FORM OVERLAY

This section describes special printing of two types: printing of Charactron characters by methods not described in the First Reader, and printing of characters formed from vectors. In addition, provision for displaying a master form is discussed.

The first part of this section describes the SCØUTV system. In its simplest form, SCØUTV is a means for using the S-C 4020 as a printer, producing frames of S-C 4020 output which correspond to pages of printed output on paper. WRITE ØUTPUT TAPE and FØRMAT statements are used for this application. In addition, SCØUTV includes several variations which permit FØRMAT-controlled printing for a number of other applications.

The display of vector characters is described beginning on page 74. 20. 21. These characters have a number of advantages over the Charactron type: they can be turned in four directions (at 90° intervals); they can be varied in size; and there are

many more different characters available for display.

However, they also have certain disadvantages. More computing time is required for the generation of each vector character. Also, the speed of the typewriter mode can not be utilized in printing a sequence of these characters.

APRNTV, which was introduced in the First Reader, can be used to write Charactron characters on different slopes and at different spacings. These applications of this routine appear on page 74. 20. 41.

A type of display different from any described previously is the overlay of a master form projected from a glass slide. Form slides and the subroutine FØRMV are explained on page 74. 20. 51.

Further aspects of S-C 4020 printing are discussed in Section 74.6.

LINE PRINTER SIMULATION ON THE S-C 4020: SCØUTV

Line printing can be simulated on the S-C 4020 by means of the SCØUTV system. Whenever the statement

CALL SCØUTV

appears in the program (or Chain link), any WRITE ØUTPUT TAPE 16 statement will be recognized as a command to produce printed output on the S-C 4020. The output will be determined by the list of the WRITE ØUTPUT TAPE 16 statement and the associated FØRMAT statement; S-C 4020 output will be returned instead of output from the printer. (The SCØUTV mode will not interfere with WRITE ØUTPUT TAPE 6 statements for standard printer output.)

One natural application of the SCØUTV mode is the production of printed output on 35mm film for long-term storage, making it unnecessary to keep large stacks of printed paper. The use of SCØUTV as a printer simulator is discussed on the following pages. Another application, the combination of plotting and printing on one film frame, will be deferred to Section 74.3, Special Grids.

BASIC SPECIFICATIONS

The following list explains most of the differences and likenesses that prevail between ordinary printing and SCØUTV printing, when there have been no programmed alterations to change the normal SCØUTV output.

1. The camera must have been selected by a CALL CAMRAV (N) statement.
2. A CALL SCØUTV statement is necessary to make printer simulation effective. The SCØUTV mode will not carry over from one Chain link to the next; hence a new CALL SCØUTV statement is necessary in each link.
3. A WRITE ØUTPUT TAPE 16 statement is used instead of a WRITE ØUTPUT TAPE 6. The FØRMAT statement and list requirements are the same. Output will correspond line-for-line and page-for-page with the same output printed by a WRITE ØUTPUT TAPE 6 statement using the same list and FØRMAT statement. (FØRMAT statements are as described on page 60.25.09 ff, and lists are as discussed on page 60.25.05 f.)

4. In some cases, it may be desirable to use a WRITE ØUTPUT TAPE N statement, reading N in as a piece of data. At execution time, if N = 6, normal printing will take place; if N = 16, S-C 4020 printing will result.

5. The first character of each BCD unit record is recognized as a carriage control character, as follows:

- | | |
|-------|--|
| + | No space before printing. |
| Blank | Single space before printing. |
| 0 | Double space before printing. |
| - | Triple space before printing. |
| 1 | Frame advance (equivalent to sheet eject). |
| 4 | Four spaces before printing. (This is not an exact equivalent of standard printer output.) |
| 8 | Skip to the bottom line on the frame. |

An incorrect carriage control character is much more critical on the S-C 4020 than on a standard printer. The S-C 4020 operator cannot override the action of these characters, as the operator of a 1401 can. Non-standard characters, or loops involving "1" or "8," can cause great waste of film.

6. The position of print line 1 will be on the Y raster coordinate that is 23 counts from the top of the frame.
7. In the basic SCØUTV mode, it is usually desirable to advance the film as you would eject a sheet of paper, by a carriage control character "1." The first information to be printed on the new frame will then start on print line 1. This is also true if the frame is changed as a result of the line count being exceeded (see paragraph 13).
8. If the frame is changed by a CALL FRAMEV, the first carriage control character of the first FØRMAT statement will affect the position of the information to be displayed in one of the following ways:

10-15-63

A blank will cause printing to start on print line 1.

A "0" will cause printing to start on print line 2 (46 counts from the top).

A "-" will cause printing to start on print line 3. Similarly, a "4" or "8" will cause printing to start on the fourth line or at the bottom of the page, respectively.

A "1" will cause the frame to eject, leaving a wasted frame.

(So what happens if you use a "+" to suppress spacing? Usually you will write on line 0, right at the top of the page. Since this feature is not guaranteed, it is not recommended.)

9. Although the S-C 4020 printing will correspond line-for-line and page-for-page with 1401 printer output, the proportions of length-to-width will not be the same. This is illustrated by the SCØUTV examples on subsequent pages.
10. Each line can contain up to 120 characters. The first character will be positioned with its center at IX = 24, and the last one will be placed at IX = 976.
11. Once a line (of 120 characters or less) has been established, it is displayed by the S-C

4020 typewriter mode. The accidental appearance of the octal codes 12, 52, or 56 in the BCD text could cause an imperfect picture, since these codes have special uses in the typewriter mode.

12. The spacing between lines is not governed by the typewriter mode; each new line will be started 23 raster counts below the previous line, which is more generous than the spacing provided by the typewriter mode.

There are 44 lines permitted per page. An attempt to write beyond the 44th line will cause the film to be advanced. Printing will continue on the new frame; the first line will be 23 raster counts from the top. The carriage control character for that line will have no effect, just as in printed output.

(Actually, the bottom of the frame is fixed at 1012 raster counts from the top of the frame; it is not regulated by the line count. This point will be encountered again in the discussion of variations from basic SCØUTV printing.)

Figures 2-a and 2-b show an exercise in the use of SCØUTV printer simulation, demonstrating some of the carriage controls, Hollerith character printing, integer, fixed, and floating point output, etc. Part of the exercise is a line count, including a count of lines skipped. Note that when a line count of 44 was exceeded, a new frame was begun.

1230-00
002 002

CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	1	1.45	0.14E 01
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	3	3.45	0.34E 01
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	6	6.45	0.64E 01
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	10	10.45	0.10E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	11	11.45	0.11E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	13	13.45	0.13E 02
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	16	16.45	0.16E 02
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	20	20.45	0.20E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	21	21.45	0.21E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	23	23.45	0.23E 02
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	26	26.45	0.26E 02
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	30	30.45	0.30E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	31	31.45	0.31E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	33	33.45	0.33E 02
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	36	36.45	0.36E 02
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	40	40.45	0.40E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	41	41.45	0.41E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	43	43.45	0.43E 02

Figure 2-a

BASIC SCOUTV PRINTING

(This example is continued on Figure 2-b.)

1230-00 L
003 003

```

CARRIAGE CONTROL CHARACTER IS - THIS IS LINE 46      46.45      0.46E 02

CARRIAGE CONTROL CHARACTER IS 4  THIS IS LINE 50      50.45      0.50E 02

```

Figure 2-b

BASIC SCOUTV PRINTING

Frame "eject" occurred when an attempt was made to write below the 44th line of the frame shown in Figure 2-a. The remaining lines were displayed on the next frame, as shown above.

The following statements were used to produce both frames. Note that, in true printer simulation, a CALL FRAMEV(0) would not be used to eject the frame; the statement was used to simplify the coding for the example.

```

CALL SCOUTV
CALL FRAMEV(0)
L1=1
DO 160 I=1.5
L2=L1+2
L3=L2+3
L4=L3+4
AL1= ELOATF(L1) +0.45
AL2=AL1+2.0
AL3=AL2+3.0
AL4=AL3+4.0
WRITE OUTPUT TAPE 16,125,L1 ,AL1,AL1,L2,AL2,AL2
125 FORMAT(53H CARRIAGE CONTROL CHARACTER IS BLANK THIS IS LINE 12,
18X,F8.2 ,8X,E8.2 /53H0CARRIAGE CONTROL CHARACTER IS 0 THIS I
2S LINE 12,8X,F8.2 ,8X,E8.2 )
WRITE OUTPUT TAPE 16,145,L3, AL3,AL3, L4,AL4,AL4
145 FORMAT(53H-CARRIAGE CONTROL CHARACTER IS - THIS IS LINE 12,
18X,E8.2 ,8X,E8.2 /53H4CARRIAGE CONTROL CHARACTER IS 4 THIS I
1S LINE 12,8X,F8.2 ,8X,E8.2 )
L1=L1+10
160 CONTINUE

```



10-15-63

SCØUTV VARIATIONS

The basic SCØUTV system treats a frame as if it were segmented into 44 print lines, starting on the 23rd raster count from the top of the frame and ending on the 1012th raster count from the top. Each print line is 23 raster counts deep. Printing begins on the first print line, and each line starts on the 24th raster count from the left of the frame.

The following discussion explains how these basic specifications can be altered in a number of ways.

Altering the Starting Line: LINE1

SCØUTV printing may be forced to start on a print line other than the first by using the statement:

```
CALL SCØUTV (LINE1)
```

LINE1 An integer which specifies the print line on which the next printing in the SCØUTV mode will begin.

For example, if the statement CALL SCØUTV (5) is used, the next WRITE ØUTPUT TAPE 16 statement will place printing on the 5th print line (the raster position that is 5*23 counts from the top of the frame). The carriage control character (even the "1" for "eject") will be ignored for that line.

For LINE1 to be in force, this statement must be executed in advance of the associated WRITE ØUTPUT TAPE 16 statement.

If no other alterations are made, LINE1 may have any value from 1 to 44, and the maximum number of lines of printing will be (45 - LINE1).

Since LINE1 will be effective only for one frame, CALL SCØUTV (LINE1) should be restated if printing is to start on the named LINE1 on the next frame.

Altering the Starting Line and Line Spacing
Increments: LINE1, INC

Further alteration of the basic SCØUTV specifica-

tions may be made by using the statement:

```
CALL SCØUTV (LINE1, INC)
```

LINE1 An integer which specifies the print line on which the next printing in the SCØUTV mode will begin. The actual position of the first line on the frame will depend upon the value of the second argument, INC.

INC An increment of line spacing. The frame will be considered to be segmented into print lines that are INC raster counts deep (as opposed to the basic 23 raster counts). The maximum number of print lines will be 1012/INC, since the basic limit for the bottom line is 1012 raster counts below the top of the frame.

The carriage control character for the first line of information to be printed following this call statement will be ignored. (Note that this also applies to the character for "eject.") Therefore, printing will start on the raster coordinate that is LINE1*INC raster counts from the top of the frame.

INC will remain in effect for subsequent SCØUTV mode printing in the program (or chain link) until altered by another CALL SCØUTV (LINE1, INC).⁽¹⁾

For LINE1 or INC to be effective, the CALL SCØUTV (LINE1, INC) statement must be executed before the associated WRITE ØUTPUT TAPE 16 statement.

Note: A CALL SCØUTV (with or without arguments) placed anywhere in a program (or chain link) will cause the SCØUTV mode to be used for all WRITE ØUTPUT TAPE 16 statements in the program or chain link, even those which precede the CALL SCØUTV. That is, the basic SCØUTV mode will be in effect until a CALL SCØUTV with arguments is actually executed.

(1) At the present time, although INC will remain effective for subsequent frames, it will not be used in computing the position of the first line of these frames. After the last line on a frame has been printed under the control of CALL SCØUTV (LINE1, INC), printing on the next frame will begin 23 raster counts from the top, just as if the basic SCØUTV were in effect. The remaining lines on the new frame will be spaced under the control of INC. (It is probable that this procedure will be changed, so that the starting position of each subsequent frame will be determined by LINE1 and INC.)

1230-00 L
003 003

CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	1	1.45	0.14E 01
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	3	3.45	0.34E 01
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	6	6.45	0.64E 01
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	10	10.45	0.10E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	11	11.45	0.11E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	13	13.45	0.13E 02
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	16	16.45	0.16E 02
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	20	20.45	0.20E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	21	21.45	0.21E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	23	23.45	0.23E 02
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	26	26.45	0.26E 02
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	30	30.45	0.30E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	31	31.45	0.31E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	33	33.45	0.33E 02
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	36	36.45	0.36E 02
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	40	40.45	0.40E 02
CARRIAGE CONTROL CHARACTER IS BLANK	THIS IS LINE	41	41.45	0.41E 02
CARRIAGE CONTROL CHARACTER IS 0	THIS IS LINE	43	43.45	0.43E 02
CARRIAGE CONTROL CHARACTER IS -	THIS IS LINE	46	46.45	0.46E 02
CARRIAGE CONTROL CHARACTER IS 4	THIS IS LINE	50	50.45	0.50E 02

Figure 2-c

SCØUTV WITH ALTERED STARTING LINE AND LINE INCREMENTS

The coding for the above example is on the following page .

```

163 CALL FRAMEV (0)
    CALL SCOUTV ( 4 ,16)
    L1=1
    DO 160 I=1,5
    L2=L1+2
    L3=L2+3
    L4=L3+4
    AL1= FLOATF(L1) +0.45
    AL2=AL1+2.0
    AL3=AL2+3.0
    AL4=AL3+4.0
    WRITE OUTPUT TAPE 16,125,L1 ,AL1,AL1,L2,AL2,AL2
125 FORMAT(53H CARRIAGE CONTROL CHARACTER IS BLANK THIS IS LINE 12,
    18X,F8.2 ,8X,E8.2 /53H0CARRIAGE CONTROL CHARACTER IS 0 THIS I
    2S LINE 12,8X,F8.2 ,8X,E8.2 )
    WRITE OUTPUT TAPE 16,145,L3, AL3,AL3, L4,AL4,AL4
145 FORMAT(53H-CARRIAGE CONTROL CHARACTER IS - THIS IS LINE 12,
    18X,F8.2 ,8X,E8.2 /53H4CARRIAGE CONTROL CHARACTER IS 4 THIS I
    1S LINE 12,8X,F8.2 ,8X,E8.2 )
    L1=L1+10
160 CONTINUE

```

Figure 2-c illustrates the use of SCOUTV with arguments that alter the basic specifications. Note the use of INC = 16, which caused the lines to be closer than normal, enabling the same 50 lines shown on Figures 2-a and 2-b to be printed on one frame. Since LINE1 = 4 and INC = 16, the first line displayed is actually 4*16 raster counts below the top of the frame.

Observe also that the CALL FRAMEV(0) precedes the CALL SCOUTV (LINE1, INC). Had the CALL FRAMEV(0) followed the SCOUTV statement, the value of LINE1 would have been reset, so that the first line of printing would not have been at the point desired.



Precise Control of Raster Positions for SCØUTV
Printing: LØCSTV

In the preceding discussion, SCØUTV was presented as a means for simulating printer output on the S-C 4020. For such simulation, vertical positioning is controlled by a line count and the increment between lines. Lateral positioning is dependent upon the built-in left margin and the number of character spaces provided by the associated FØRMAT statement.

The range of SCØUTV applications has been greatly extended by means of routines which permit precise control of raster positions, with the printing still under FØRMAT control. Two routines, LØCSTV and LØCSAV, which permit setting and retrieval of raster positions, add the following capabilities to the SCØUTV system:

1. Printing can be done below $IY = 11$. (The basic SCØUTV does not permit writing below the 1012th raster position, counted from the top of the frame.)
2. Printing can be done to the left of $IX = 24$. (This area is inaccessible to the basic SCØUTV system, which uses a fixed left margin of $IX = 24$.)
3. Printing can extend to the right of $IX = 976$, by starting the line at $IX \geq 25$. (This area is inaccessible to the basic system, since the maximum (120 character) SCØUTV line starts at $IX = 24$ in the basic SCØUTV, and thus ends at $IX = 976$.)
4. Printing can be started at a point which is independent of a line count, line increment, or number of character spaces. (For the basic system, which simulates line printing, printing can only be positioned in units of character widths and of print line depth.)

The LØCSTV provides these facilities by means of the call statement:

CALL LØCSTV (IX, IY, IB)

- IX The X-raster position to be used as the starting point of the next line of printing. Stated in another way, IX is the new left margin, which will be effective for subsequent lines and frames until another CALL LØCSTV is encountered.

The programmer must ensure that information to be printed on subsequent

lines can be accommodated between IX and the right edge of the frame. IX should be specified such that

$$1023 - IX \geq 8 * (\text{number of characters in the line to be printed})$$

- IY A Y-raster position counted from the BOTTOM of the frame, to be used as a reference for subsequent lines of printing. Since the position IX, IY will be referenced as if it were the starting point of the previous line of printing, the next line will start at IX, IY only if the carriage control character + (to suppress spacing) is used.

The position of subsequent lines of SCØUTV mode printing relative to IY will depend upon the carriage control characters used and upon the value of the current line increment, INC (page 74. 20. 07).

IY is effective only for subsequent lines on the same frame. It will be changed (reset to the top of the frame) if the frame is advanced, or if one of the forms of CALL SCØUTV intervenes between a CALL LØCSTV and the affected WRITE ØUTPUT TAPE 16 statement.

- IB Limit below which printing will not take place, expressed in raster counts from the BOTTOM of the frame. It will be effective for subsequent lines and frames until another CALL LØCSTV is encountered.

To return the indicators to the basic SCØUTV specifications, use CALL LØCSTV (24, 1000, 11).

Note particularly that, unlike IX and IB, the IY reference point will not carry over from frame to frame, nor will it survive execution of a CALL SCØUTV. If the IY reference is to be used on a new frame, the CALL LØCSTV statement should be repeated.

One use of LØCSTV is illustrated in Figure 2-d.

An illustration of how SCØUTV mode printing can be combined with a graph is shown in Figure 3-d, Section 74.3. That example also illustrates another way to print tabular data using SCØUTV and LØCSTV.

EXERCISE IN THE USE OF LOCSTV WITH SCOUTV IN SPECIAL PRINTING APPLICATIONS

1230-03 L
000 000

ENGINEER J. MATTEL
 DEPT.-GROUP 202-072
 EXT. 2651
 JOB. NO. 230-30

PROJECT SABRE - SUB
 PART PERISCOPE
 MATERIAL PLASTIC

MODEL NO. C 8 IN
 TEST NO. 33T
 DATE 10/ 5

7125.00	7488.45	7870.43	8271.90	8693.85
7196.25	7563.33	7949.14	8354.62	8780.79
7268.21	7638.96	8028.63	8438.17	8868.60
7340.89	7715.35	8108.91	8522.55	8957.29
7414.30	7792.51	8190.00	8607.78	9046.86

Figure 2-d

USE OF LOCSTV FOR SPECIAL SCOUTV PRINTING

It is assumed that the necessary data are in storage. In addition to printing the test identification information, it is desired to print five columns of test results, with DATA(1) to DATA(5) in column 1, DATA(6) to DATA(10) in column 2, etc.

The coding for this frame is on the following page.

10-15-63

```

CALL SCOUTV
DIMENSION NAME(4) ,DEPT(2) ,      LTAB(5) ,PART(2),PROJ(4),DATA(25)
1 , MATRL(2)
READ INPUT TAPE 5,185,NAME,DEPT, EXT,JOB,PROJ,PART,MATRL,MODEL
1,TEST,DATE
185 FORMAT (12A6)

      LTAB(1)=10
      DO 193 I=2,5
193 LTAB(I)=LTAB(I-1)+190
      CALL FRAMEV
      WRITE OUTPUT TAPE 16,220
220 FORMAT (1H113X,75HEXERCISE IN THE USE OF LOCSTV WITH SCOUTV IN SPE
1CIAL PRINTING APPLICATIONS      )
      CALL LOCSTV (700,950, 0)
      WRITE OUTPUT TAPE 16,255,NAME,DEPT, EXT, JOB
255 FORMAT (12H+ENGINEER  4A6 /15H DEPT.--GROUP  2A6 / 8H EXT.
1A6 /12H JOB. NO.  2A6      )
      CALL LOCSTV (0,800, 0)
      WRITE OUTPUT TAPE 16, 280, PROJ, PART, MATRL
280 FORMAT (11H+PROJECT  4A6/ 6H PART  2A6 /10H MATERIAL  2A6 )
      CALL LOCSTV (400,700, 0)
      WRITE OUTPUT TAPE 16,310,MODEL , TEST, DATE
310 FORMAT (13H+MODEL NO.  A6 /13H TEST NO.  A6 / 9H DATE  A6 )
      J=1
      DO 350 I=1,5
      CALL LOCSTV (LTAB(I), 500,0)
      DO 345 K=1,5
      WRITE OUTPUT TAPE 16,340, DATA(J)
340 FORMAT( 1H F8.2)
345 J=J+1
350 CONTINUE

```



10-15-63

Preservation (or Retrieval) of Raster Positions
Used by SCØUTV: LØCSAV

The raster coordinates of the starting point of the line most recently printed in the SCØUTV mode can be stored in ISX, ISY by using the statement:

CALL LØCSAV (ISX, ISY)

The programmer can use this information to relate other types of S-C 4020 output to the location of the last line of print.

LØCSAV will also aid in a special situation that occurs with interrupted printing. If SCØUTV printing is interrupted on one camera in order to display one or more frames on the alternate camera, the SCØUTV printing and film advance for the alternate camera will alter the line reference information. On return to the original frame on the first camera, the programmer will be unable to resume SCØUTV printing at the next line unless he uses LØCSAV and LØCSTV. The following sequence shows the procedure that might be utilized.

```

.
.
.
CALL SCØUTV
.
.
CALL CAMRAV (35)

```

CALL FRAMEV

```

.
.
.

```

Write a portion of this 35mm frame, using the appropriate WRITE ØUTPUT TAPE 16 and FØRMAT statements.

```

.
.
.

```

CALL LØCSAV (M1X, M1Y)

CALL CAMRAV (9)

CALL FRAMEV

```

.
.
.

```

Write one or more frames on the 9-inch camera, using the appropriate WRITE ØUTPUT TAPE 16 and FØRMAT statements.

```

.
.
.

```

CALL LØCSTV (M1X, M1Y, 11)

CALL CAMRAV (35)

```

.
.
.

```

Continue writing on 35mm film as if no interruption had taken place.

```

.
.
.

```



PRINTING WITH PROGRAM-GENERATED CHARACTERS:

RITE2V, VCHARV, CHSIZV, RITSTV

THE VECTOR CHARACTER WRITING
SUBROUTINE: RITE2V

The RITE2V subprogram provides a means for printing variable-size characters in one of four different orientations. It uses, on a lower level, the vector character-generating subprogram, VCHARV.

The call statement can be either of the following:

CALL RITE2V (IX, IY, LIMIT, K, INT,
NTOTAL, NTH, BCDTXT,
NLAST)

CALL RITE2V (IX, IY, LIMIT, K, INT,
NTOTAL, -NTH, nH-----,
NLAST)

IX, IY The raster coordinates used for positioning the center of the first character to be printed.

LIMIT A raster position which will determine the extent of a line of writing. When this limiting position is reached, writing is halted and a new line is started below the starting position of the previous line.

K A code number which selects the direction of writing and the orientation of characters within the written text.

K = 90 Characters will be upright, as A

Writing will be horizontal, left to right, as AND

K = 180 Characters will be rotated left, as A

Writing will be vertical, south toward north, as AND

K = 270 Characters will be upside down, as V

Writing will be horizontal, right to left, as DNV

K = 0 Characters will be (or 360) rotated right, as A

Writing will be vertical, north toward south, as AND

INT An integer which controls intensity of the characters by controlling the number of times each vector is displayed. If "overstriking" for a darker character is not required, INT should be 1.

NTOTAL An integer which specifies the number of characters to be printed.

±NTH A character position count that specifies the location within a BCDTXT where writing is to begin. (Usually this is 1, but this argument allows printing to begin with some character in the BCDTXT other than the first.)

The sign indicates the type of storage used for the next argument. NTH should be positive if the 8th argument is a FORTRAN array. A negative value should be used if it is a Hollerith argument.

BCDTXT The name of an array of BCD information to be written.

nH----- A Hollerith argument. If used, a negative sign must be used with NTH to signal the subroutine that the information is in this form.

NLAST The name of an error location (fixed point). NLAST will be set to zero if all the NTOTAL characters have been displayed. If writing must be stopped because the end of the frame has been reached, NLAST will be set to the character count where writing was stopped.

The standard character size is approximately 12 raster counts in width by 18 counts in height. The standard spacing allowance for adjacent characters is 18 raster counts, and for adjacent rows is 26 raster counts.

The programmer can alter the size of characters and spacing by means of other subroutines, discussed later in this section. He will not need to use them if the standard specifications are satisfactory for his purposes.

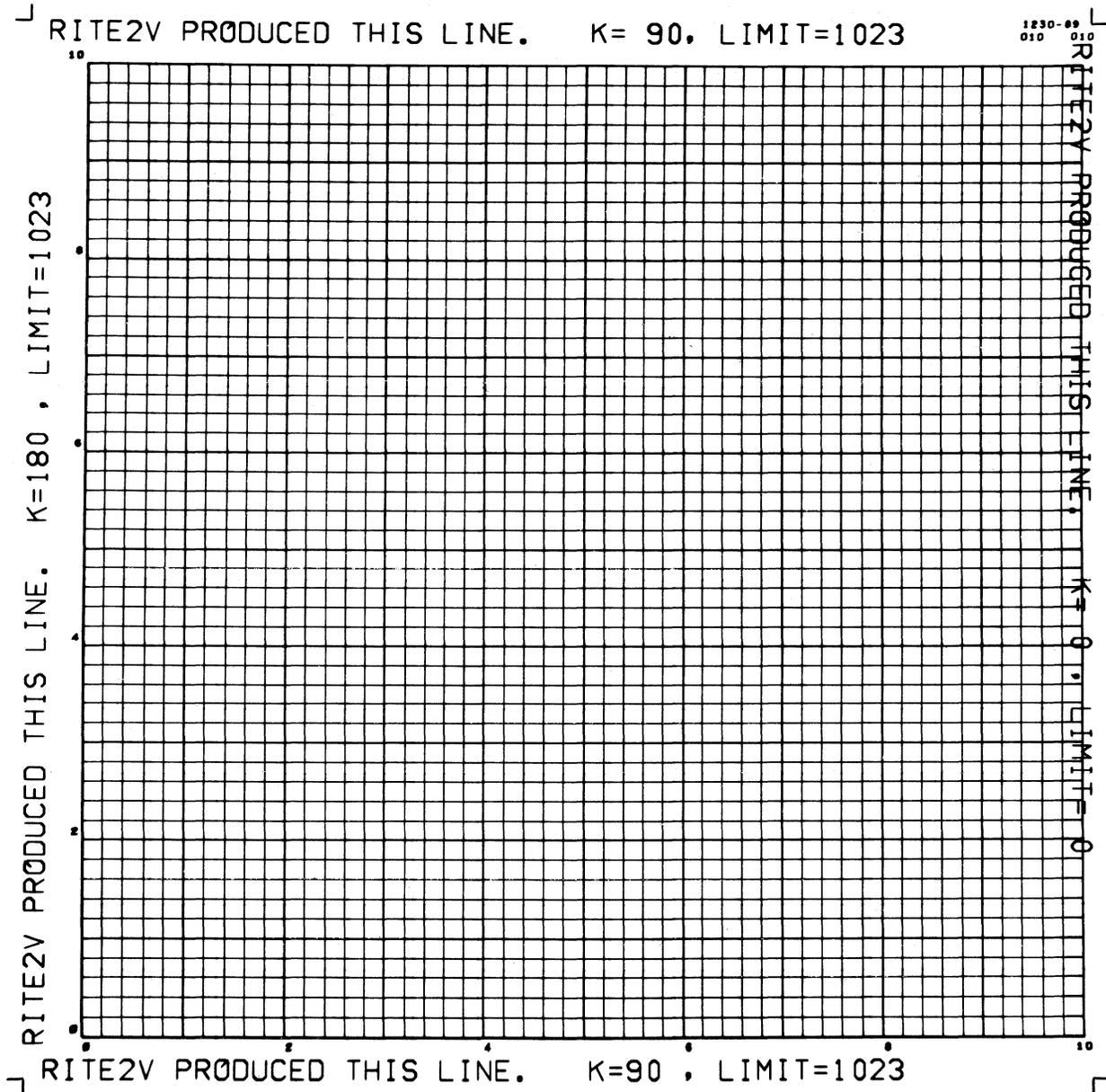


Figure 2-e

RITE2V TITLES IN GRID MARGINS

```
CALL GRID1V(1,0.0,10.0,0.0,10.0, 0.2,0.2,5,5,10,10,2,2 )
CALL RITE2V (25 ,1014,1023, 90,1, 48, - 1,48HRITE2V PRODUCED THIS
1LINE. K= 90, LIMIT=1023 ,NL1)
CALL RITE2V ( 9,40,1023,180,1,46,-1,46HRITE2V PRODUCED THIS LINE.
1 K=180 , LIMIT=1023 ,NL2)
CALL RITE2V (25, 9,1023, 90,1,46,-1,46HRITE2V PRODUCED THIS LINE.
1 K=90 , LIMIT=1023 ,NL3 )
CALL RITE2V (1014,997,0,0,1,46,-1,46HRITE2V PRODUCED THIS LINE.
1K= 0 , LIMIT= 0 ,NL4)
```

10-15-63

An illustration of titles written in grid margins by RITE2V is shown in Figure 2-e. Note that the standard-size vector characters fit into the margins normally reserved by GRID1V at the top, left and bottom of the grid, and also that GRID1V does not normally reserve space for titles at the right of the grid.

The orientation of characters and the direction of writing are controlled by the values specified for K. Notice that the value specified for LIMIT is an upper limit when K is 90 or 180, and a lower limit when K is 0 (or 360). When K is 270, LIMIT is also a lower limit, as will be shown in the next example.

1230-00 L
012 012

WRITTEN BY RITE2V
COMPARE THIS WITH
CALL STATEMENT (A)

WRITTEN BY RITE2V
COMPARE THIS WITH
CALL STATEMENT (B)

WRITTEN BY RITE2V
COMPARE THIS WITH
CALL STATEMENT (C)

WRITTEN BY RITE2V
COMPARE THIS WITH
CALL STATEMENT (D)

Figure 2-f

EFFECT OF LIMIT

The portion of a frame above illustrates the use of LIMIT as a "carriage return" control. When the LIMIT is reached, writing is halted and a new line is started, just below the starting point of the preceding line.

C	CALL STATEMENT (A)	
	CALL RITE2V(350,1000,674, 90,1,54,-1,54HWRITTEN BY RITE2V COMPARE	
	1THIS WITH CALL STATEMENT (A) ,NLAST)	
C	CALL STATEMENT (B)	
	CALL RITE2V(270,596,920,180,1,54,-1, 54HWRITTEN BY RITE2V COMPARE	
	1THIS WITH CALL STATEMENT (B) ,NLAST)	
C	CALL STATEMENT (C)	
	CALL RITE2V(656,516,332,270,1,54,-1, 54HWRITTEN BY RITE2V COMPARE	
	1THIS WITH CALL STATEMENT (C) ,NLAST)	
C	CALL STATEMENT (D)	
	CALL RITE2V(776,902,578,0,1,54, -1, 54,WRITTEN BY RITE2V COMPARE	
	1THIS WITH CALL STATEMENT (D) ,NLAST)	

SHAPE CHANGED BY USING CALL CHSIZV (3,5) BEFORE RITE2V
 WIDTH AND ISPACE CHANGED BY USE OF
 CHSIZV(5,3) AND RITSTV
 ALMOST MAXIMUM SIZE

ALMOST MINIMUM SIZE

Figure 2-g

EFFECTS OF CHSIZV AND RITSTV

	CALL CHSIZV (3,5)
	CALL RITE2V (26,450,1023,90,1,54,-1,54)SHAPE CHANGED BY USING CALL CHSIZV (3,5) BEFORE RITE2V (,NLAST)
	CALL CHSIZV(5,3)
F	TABLIV
	CALL RITSTV (28,26,TABLIV)
	CALL RITE2V (26,400,1023,90,1,60,-1,60)WIDTH AND ISPACE CHANGED BY USE OF CHSIZV(5,3) AND RITSTV (,NLAST)
	CALL CHSIZV (9,9)
	CALL RITSTV (48,66,TABLIV)
	CALL RITE2V (104,314,1023,90,1,19,-1,19)ALMOST MAXIMUM SIZE (,NL)
	CALL CHSIZV(2,2)
	CALL RITSTV(12,18,TABLIV)
	CALL RITE2V(104, 200,1023,90,1,19,-1,19)ALMOST MINIMUM SIZE,(NLAST)
	CALL CHSIZV(3,3)

10-15-63

Variation in Size of Vector Characters: CHSIZV

The width and height of vector characters can be altered by the subroutine CHSIZV ("Change Size V"). The call statement is:

CALL CHSIZV (LVW, LVH)

LVW An integer, from 2 to 15, that controls the width of characters.

LVH An integer, from 2 to 9, that controls the height of characters.

Obviously, LVW and LVH need not be equal, but any great disparity will result in the formation of characters that are elongated in one direction.

For the usual applications of RITE2V, the character size can be considered to be 4*LVW in width, and 6*LVW in height. (These dimensions do not include the space required to prevent overlapping of adjacent characters.) Examples of different proportions are shown in Figure 2-g.

If CHSIZV is never called, the standard values of LVW = 3 and LVH = 3 will be used, resulting in a character size of 12 x 18 raster counts.

When CHSIZV is used to alter the standard character size, it will probably also be necessary to adjust the character spacing. The subroutine RITSTV was designed for this purpose.

Changing Vector Character Spacing; Selecting Alternate Character Sets: RITSTV

RITSTV can be used to alter spacing used by RITE2V for characters and rows. In addition, RITSTV can be used to select an alternate character set to be displayed by RITE2V. The call statement is:

CALL RITSTV (ISPACE, IRØW, TABL_V)

ISPACE An integer that specifies the number of raster counts allowed for adjacent characters. This should be somewhat larger than the width of each character, to allow a gap between them.

IRØW An integer that specifies, in ras-

ter counts, the depth of adjacent rows. This should be somewhat larger than the height of a character.

TABL_V The name of a table of character patterns to be used. The most frequently used table, containing the alphanumeric characters, is named TABL1V. The available tables are described on page 74. 20. 35.

WHEN RITSTV IS USED, THE TABLE NAME GIVEN AS THE THIRD ARGUMENT MUST ALSO BE GIVEN ON AN 'F' CARD. (See page 60. 42. 05.)

If RITSTV is called, all three arguments must be specifically stated, even though not all of them are changed, and the table name must appear on an "F" card.

If RITSTV is never called, RITE2V will use the standard values of ISPACE and IRØW (18 and 26, respectively), and the standard table (TABL1V).

The following formulas are a general guide for the assignment of minimum values of ISPACE and IRØW.

$$\text{ISPACE} \approx 5 * \text{LVW} + 3$$

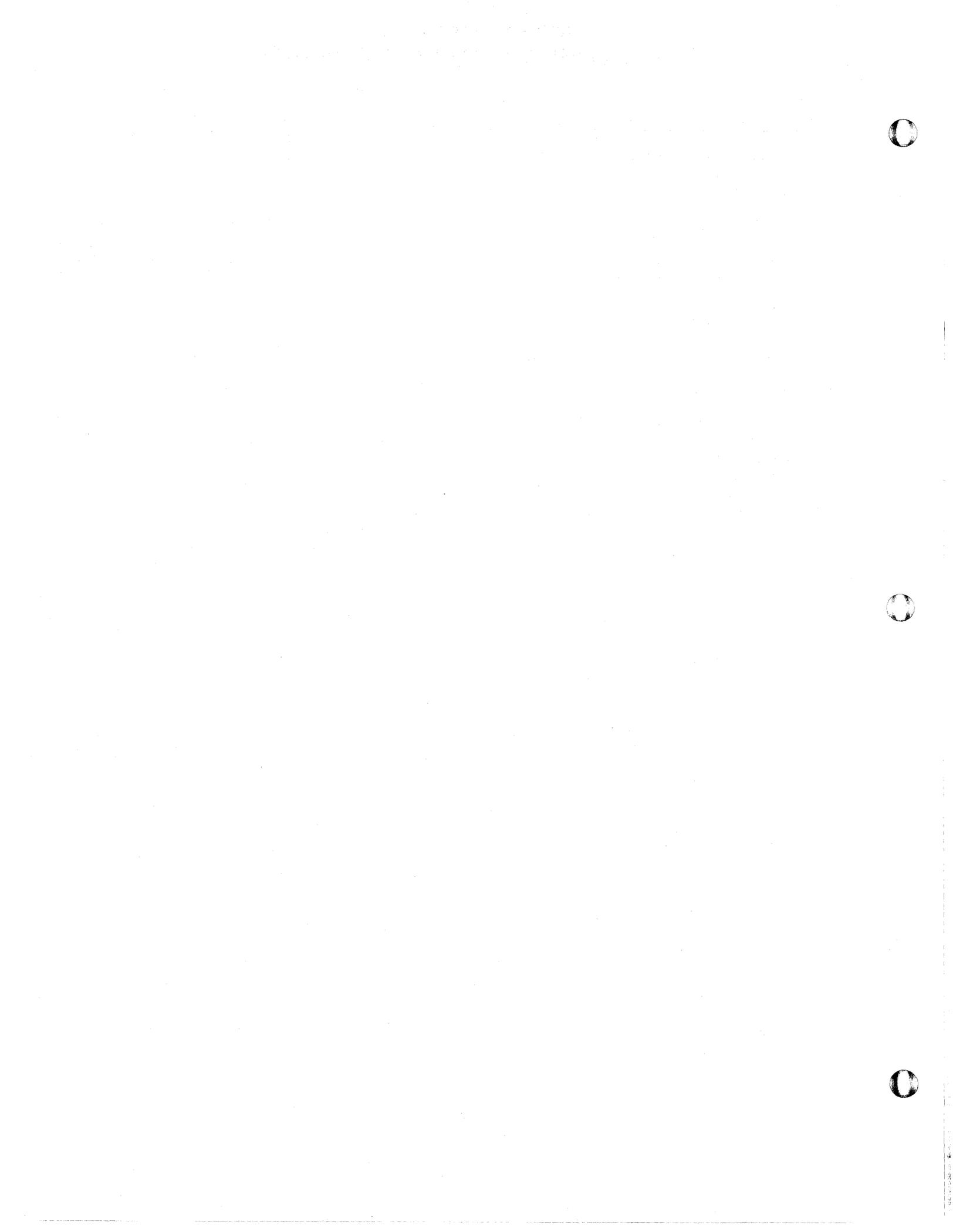
$$\text{IRØW} \approx 7 * \text{LVH} + 5$$

Retrieval of RITE2V Terminal Coordinates: RITXYV

At the time RITE2V terminates, the coordinates for the next character are set up. RITXYV retrieves these coordinates so that they can be used to determine the correct position of a special character to be inserted into a text written by RITE2V. The call statement is:

CALL RITXYV (LX, LY)

LX, LY Names of the locations in which RITXYV will store the raster coordinates (IX, IY) that RITE2V has computed as the location of the next character.



DISPLAY AN INDIVIDUAL VECTOR CHARACTER:
VCHARV

VCHARV is used as a lower-level subroutine by RITE2V to generate each vector character. When a single character is to be displayed, a direct call to VCHARV by the programmer will often be more convenient than a call to RITE2V, especially if the character to be displayed is a non-Hollerith or a special character.

The call statement is:

CALL VCHARV (K, INT, ILX, ILY, NS,
 TABL_V)

- K** Code number which selects the orientation of the character:
- K = 90 Character will be upright, as A
 - K = 180 Character will be rotated left, as ◀
 - K = 270 Character will be upside down, as ∇
 - K = 0 (or 360) Character will be rotated right, as ▶

INT An integer which controls the intensity of the character by controlling the number of times each vector is repeated. For normal intensity, this should be 1, but may be greater if a darker character is desired.

ILX, ILY Integers which specify the raster coordinates to be used to position the character. This position will be used as the lower left corner of the character pattern. (Note that this is different from the center position specified for RITE2V.)

NS An integer which selects the character to be generated from a table of patterns. The actual character produced will depend upon the table being used; the available tables are discussed

on page 74. 20. 35.

(If a Hollerith character is desired, and if TABL1V is used for the last argument, a properly stated Hollerith argument can be employed for NS. For example, the argument 3H001 is equivalent to an integer 1, 3H00A is equivalent to the integer 17 that will select A, etc.)

TABL_V The name of a table of character patterns to be used. The name of the table containing the most frequently used characters is TABL1V. Further information on available tables is given on page 74. 20. 35.

WHEN VCHARV IS CALLED DIRECTLY, THE TABLE NAME GIVEN MUST ALSO APPEAR ON AN "F" CARD.

RITE2V uses the specified coordinates as the center of the first character to be displayed, but VCHARV uses the specified coordinates as the lower left corner of the character. Thus, it may be necessary to translate the coordinates when direct calls to VCHARV are used to insert characters into a line of information displayed by RITE2V. Without going into great detail on the structure of vector characters at this point, the formulas given below show the relation between the position arguments for VCHARV and for RITE2V. In these formulas, LVW and LVH are the arguments used in CHSIZV. If CHSIVZ has not been used to alter these quantities, LVW = LVH = 3.

- For K = 90 $ILX = IX - 2*LVW$
 $ILY = IY - 3*LVH$
- K = 180 $ILX = IX + 3*LVH$
 $ILY = IY - 2*LVW$
- K = 270 $ILX = IX + 2*LVW$
 $ILY = IY + 3*LVH$
- K = 0 (or 360) $ILX = IX - 3*LVH$
 $ILY = IY + 2*LVW$

Three tables are currently available for vector character generation; they will be described in the next section.



Available Vector Character Tables: TABL1V, TABL2V, and TABL3V

Of the three tables of character patterns now available, TABL1V is an alphanumeric table, TABL2V contains lower case Greek characters, and TABL3V contains the upper case Greek characters.

The first 64 characters in TABL1V are arranged in an order compatible with the Charactron character table. Any code which will select one of the Charactron characters will select the equivalent character from TABL1V. However, TABL1V is extended to include some special characters, selected by codes numerically greater than those permissible for the Charactron characters. Since codes greater than 63 can not be expressed as BCD (or Hollerith) characters, RITE2V cannot be used to display them. However, VCHARV can be employed for any special characters that have selection codes greater than 63.

Usually, VCHARV will be utilized whenever a character from TABL2V or TABL3V is to be displayed. However, it is of interest to note that the

position of 0, 1, 2, . . . , etc., in TABL1V are equivalent to the positions of α , β , γ , , etc., in TABL2V. Therefore, if the Hollerith argument 3H012 were to be used in RITE2V, and if TABL2V had been specified by RITSTV, the characters α β γ would be displayed.

REMEMBER, WHEN ONE OF THE TABL V NAMES IS USED BY THE PROGRAMMER AS AN ARGUMENT, IT MUST ALSO BE NAMED ON AN "F" CARD.

Figure 2-h shows the three vector character tables TABL1V, TABL2V, and TABL3V. The programming for this illustration is shown because it illustrates the use of the tables and of VCHARV, as well as providing an example of the use of APRNTV in a special situation. The statements which resulted in the display of TABL3V are not listed because of their similarity to those used for TABL2V.

Note the use of APRNTV to produce the headings for the rows and columns. Utilization of APRNTV for this application permits the required special spacing of Charactron characters.

TABL1V, TABL2V, TABL3V --NOTE USE OF VCHARV TO DISPLAY THE CHARACTERS

1230-88 L
 014 014

TABL1V,

	0	1	2	3	4	5	6	7	8	9
00	0	1	2	3	4	5	6	7	8	9
10	∂	=	"	'	δ	α	+	A	B	C
20	D	E	F	G	H	I	π	.)	β
30	±	?	-	J	K	L	M	N	O	P
40	Q	R	•	\$	*	∞	~	∂	/	
50	S	T	U	V	W	X	Y	Z	°	,
60	(∫	Σ	□	&	%	€	[]	{
70	}	>	<	≥	≤	∞				

TABL2V

	0	1	2	3	4	5	6	7	8	9
00	α	β	γ	δ	ε	ζ	η	θ	ι	κ
10	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ
20	φ	χ	ψ	ω						

TABL3V

	0	1	2	3	4	5	6	7	8	9
00	A	B	Γ	Δ	E	Z	H	Θ	I	K
10	Λ	M	N	Ξ	O	Π	P	Σ	T	Υ
20	Φ	X	Ψ	Ω						

Figure 2-h

AVAILABLE VECTOR CHARACTER TABLES

A prior CALL FRAMEV statement is assumed.

A portion of the coding used to produce this frame appears on the following page. It demonstrates the use of VCHARV and the tables, TABL1V, TABL2V. In addition, it shows a special application of APRNTV.

NORTH AMERICAN AVIATION, INC.
ENGINEER'S COMPUTING MANUAL

Code For Figure 2-h

```

F      TABL1V ,TABL2V,TABL3V
CALL APRNTV (30,0,-11,11H 0123456789      ,353,888)
CALL APRNTV (0,-30,- 9, 9H 01234567      , 341,883)
CALL APRNTV (0,-30,- 9, 9H 00000000      ,349,883)
CALL PRINTV(- 6, 6HTABL1V ,492,920 )
IX1=377
IY=844
J=0
DO 2450 I=1,8
IX2=IX1
DO 2440 K=1,10
CALL VCHARV (90,1,IX2,IY,J,TABL1V )
J=J+1
IF(J-75) 2440,2440,2460
2440 IX2=IX2+30
2450 IY=IY-30
2460 CALL APRNTV (30,0,-11,11H 0123456789      , 33,438)
CALL APRNTV (0,-30,-4, 4H 012      , 21 ,433 )
CALL APRNTV (0,-30,-4, 4H 000      , 29, 433 )
CALL PRINTV (-6, 6HTABL2V,172 ,470 )
IX1= 57
IY= 394
J=0
DO 2600 I=1,3
IX2=IX1
DO 2590 K=1,10
CALL VCHARV (90,1,IX2,IY,J,TABL2V )
J=J+1
IF (J-23) 2590,2590,2610
2590 IX2= IX2+30
2600 IY=IY-30

```



SPECIAL USES OF APRNTV

The subroutine APRNTV was introduced in the First Reader (page 74.10.21) as a means of writing vertical titles. Since APRNTV allows the programmer to specify increments of character spacing in both the X and Y directions, it is useful when special spacing of Charactertron characters is required.

The illustration on page 74.20.36 shows an application of APRNTV for special horizontal and vertical Charactertron character spacing. The following example shows that APRNTV can also be used to print Charactertron characters along certain slopes.

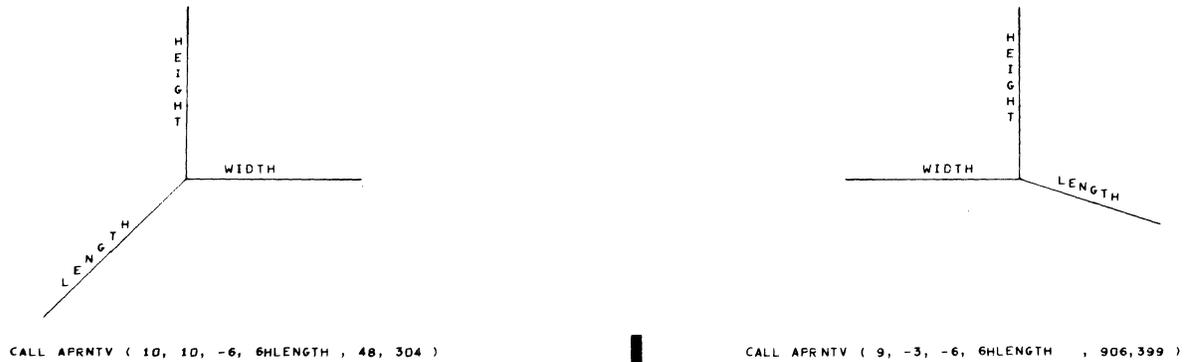


Figure 2-i

APRNTV WRITING ON SLOPES

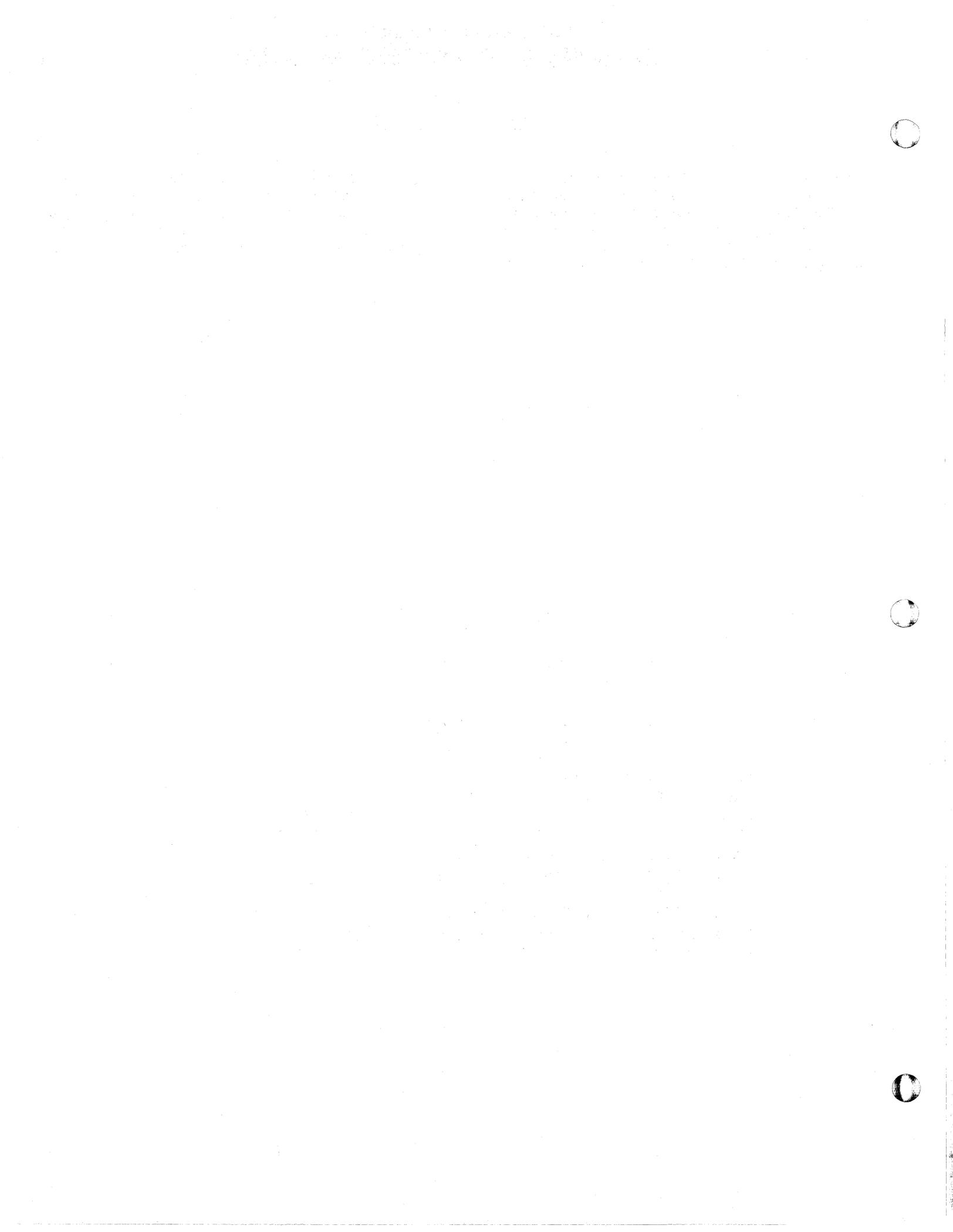
The following statements were used to produce the examples on the portion of a frame shown above.

```

CALL LINEV (150,400,150,550)
CALL LINEV (150,400,300,400)
-----
CALL LINEV (150,400,30,280)
CALL APRNTV(10,10,-6,6HLENGTH,50,310)
CALL APRNTV(0,-14,-6,6HHEIGHT,145,520)
CALL APRNTV(9,0,-5,5HWIDTH,186,409)
-----

CALL LINEV (870,400,720,400)
CALL LINEV (870,400,870,550)
-----
CALL LINEV (870,400,990,360)
CALL APRNTV (9,-3,-6,6HLENGTH,906,398)
CALL APRNTV (0,-14,-6,6HHEIGHT,865,524)
CALL APRNTV (9,0,-5,5HWIDTH,790,405)

```



USE OF THE FORM SLIDE

A Form Projector unit in the S-C 4020 allows an image from a specially prepared slide to be projected, in registration with the tube output. A form slide is a glass slide approximately 5 x 5 inches (not a 35mm slide), constructed to fit into the form projector.

Since standard procedures for the preparation and handling of form slides have not been established, anyone who plans to exploit this feature must discuss his application with his Divisional Computing Coordinator. The following considerations must be taken into account:

1. The form must be prepared according to strict specifications if it is to be aligned accurately.

2. Preparation of the form and slide are costly. Therefore, a slide is justified only for a form that cannot be program generated, or one that is used so frequently that the computer time saved will pay for the expense.

3. Although projection is under program control, the slide is inserted manually by the S-C 4020 operator. Since this can be done only at the beginning of a job, there is no choice of different slides within a job.

The following call statement will cause the form to be "flashed" (projected):

CALL FØRMV



Section 74.3

GRID VARIATIONS

The First Reader introduced the basic methods of preparing a grid and plotting on it. This section presents other features of the system that provide greatly increased flexibility for a variety of applications.

The first part of this section presents some of the external control subprograms that can be used with GRID1V to obtain variations of the basic grid. Among these is a routine which alters the margin specifications, permitting the display of more than one grid on a frame, or the combination of a graph with printed information.

The second part of this section describes how GRID1V and the basic plotting subprograms can be used to produce log and semi-log graphs. This is made possible by a single routine that can alter the scale mode in either or both directions.

Two other routines, LINRV and NØNLNV, are introduced briefly. Since the understanding of their operation depends upon a knowledge of scaling and conversion, which will be discussed in Section 74.4, a more detailed discussion of these two subprograms has been deferred to Section 74.5, Further Aspects of Grid Construction.

The information on the basic linear GRID1V is not repeated here. The programmer should be familiar with the First Reader before studying this material.

GRID1V CONTROLS

Certain features of the basic linear GRID1V can be altered by subprograms that control its internal operation. The subprograms can be classified as "set" and "retrieve" routines since they permit information to be set by the programmer and retrieved during execution of GRID1V.

The routines that furnish values different from those normally employed by GRID1V are:

SETMIV, which allows the programmer to make nonstandard margin assignments. The companion routine called by GRID1V to retrieve margin values is SETMØV.

SETCIV, which makes it possible to provide extra space for grid line labels. The companion routine is SETCØV.

Routines that furnish indicators recognized by GRID1V as signals to execute alternate branches are:

HØLDIV, which assists in holding margins from graph to graph. HOLDØV is called to retrieve the indicators.

SMXYV, which enables the programmer to select log or semi-log mode of operation. The companion routine is MSXYV. These two routines are described under "Log and Semi-log Plotting," page 74.30.21.

Grid Margin Variation: SETMIV, SETMØV

As discussed in the First Reader, GRID1V normally reserves a strip, 24 raster counts in width, at the top, left, and bottom of the grid, for the display of titles. For the many applications which require special margin widths, the subprogram SETMIV can be called to change the basic specifications.

One obvious application of SETMIV is to provide margin space for multiple lines of printed titles and headings. In addition, and perhaps even more important, SETMIV makes it possible to display more than one graph on a frame, or to display a graph with its accompanying text.

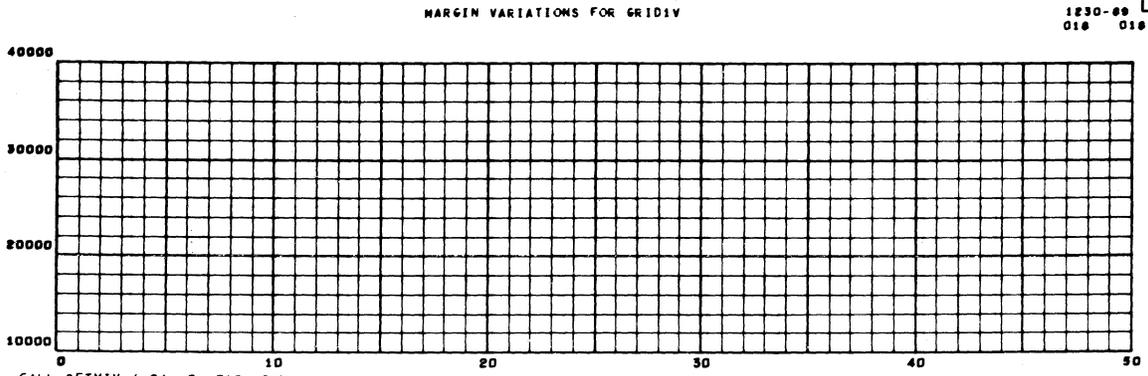
The standard GRID1V margin specifications can be altered by the statement:

CALL SETMIV (MTL, MTR, MTB, MTT)

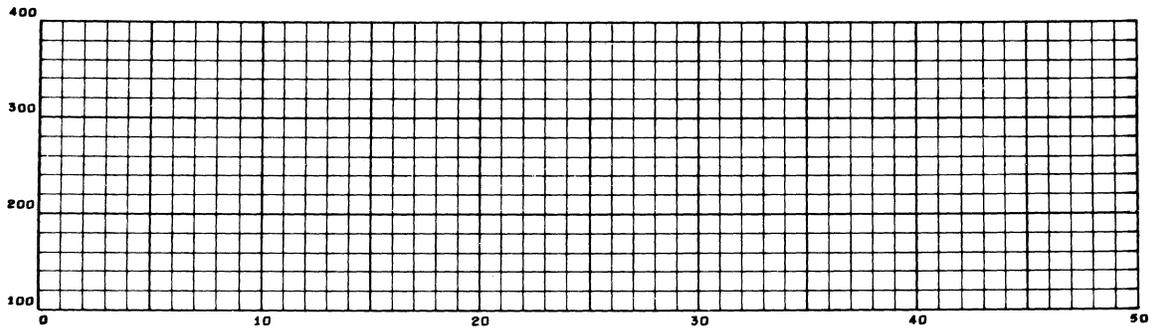
Each argument is an integer which specifies, in raster counts, the width of one area to be reserved for a margin.

MTL	Width of area for left margin.
MTR	Width of area for right margin.
MTB	Width of area for bottom margin.
MTT	Width of area for top margin.

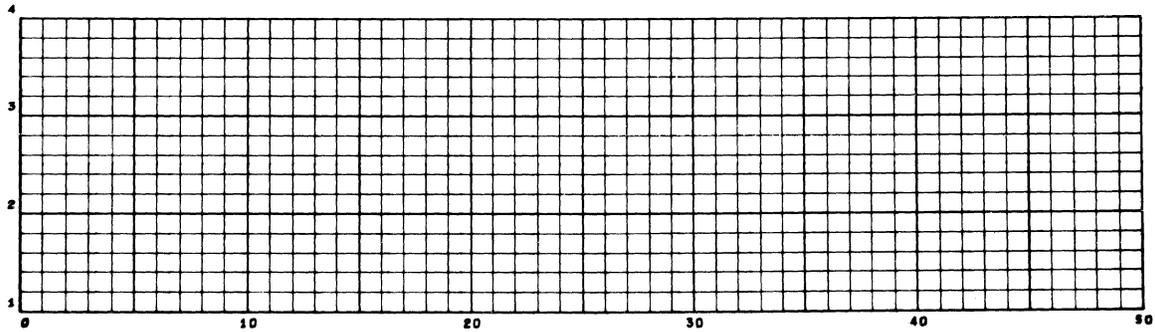
GRID1V does not necessarily use these exact values for the upper and lower limits of X and Y. It guarantees that the reserved space will not be overlapped, assigning additional space if required for label margins. After the total margin space has been reserved, the remaining area will be used for the grid.



CALL SETMIV (24, 0, 712, 28)
CALL GRID1V (2, 0.0, 50.0, 10000.0, 40000.0, 1.0, 2000., 5, 5, 10, 5, 2, 5)



CALL SETMIV (24, 0, 368, 371)
CALL GRID1V (2, 0.0, 50.0, 100.0, 400.0, 1.0, 20.0, 5, 5, 10, 5, 2, 3)



CALL SETMIV (24, 0, 24, 715)
CALL GRID1V (1, 0.0, 50.0, 1.0, 4.0, 1.0, 0.2, 5, 5, 10, 5, 2, 1)

Figure 3-a

THREE GRAPHS ON ONE FRAME

10-15-63

If SETMIV is never called, GRID1V will use the values 24, 0, 24, 24 as MTL, MTR, MTB, and MTT, respectively. To return to a standard grid after the margins have been altered, restore the standard margin values by

```
CALL SETMIV (24, 0, 24, 24)
```

The current values of MTL, MTR, MTB, and MTT can be retrieved by using the statement

```
CALL SETMOV (MTLL, MTRL, MTBL, M TTL)
```

where the arguments are variables (never constants) to which SETMOV is to assign the current margin values. SETMOV was designed for use by GRID1V to retrieve current margin values; the programmer will rarely have reason to call it.

Examples

Figure 3-a shows three grids with the SETMIV and GRID1V call statements used to produce them. The

grid at the bottom was the first one displayed; a 1 was used as the first argument of the first GRID1V statement executed, in order to change the film frame. The other two GRID1V statements include a 2 as the first argument, to inhibit the film advance.

Note particularly the variation in the raster locations assigned by GRID1V to XL in each of the grids. This effect is caused primarily by the differences in the specification of NY (the last argument), which gives the number of characters to be displayed in the labels of horizontal lines. In each case, NY has been assigned a value just large enough to satisfy the needs of the grid. For the bottom grid, NY = 1; for the middle grid, NY = 3; and for the top grid, NY = 5. Since margin space was reserved for labels of different lengths, the positions of the left limits, and of the corresponding values of X, vary noticeably. Such a nonalignment is often of no importance, but if it does matter, the programmer may have to make special provisions to force alignment.

Figure 3-b is a similar illustration, except that NY = 5 on all three grids, permitting the left limits to be in line. If XL, NX, and NY are equal from

grid to grid, the desired alignment will usually be achieved.

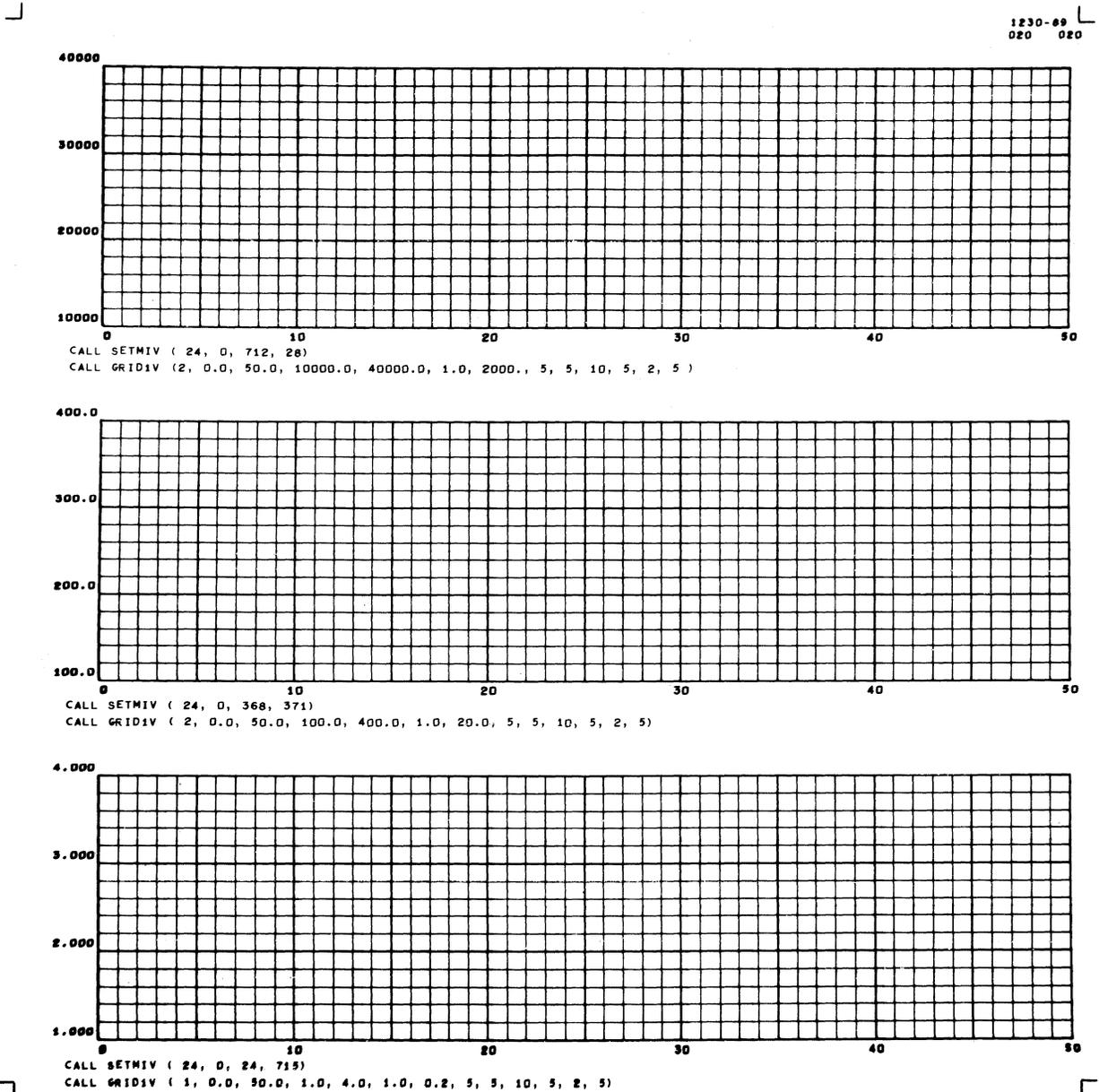


Figure 3-b

GRAPH SHOWING FORCED ALIGNMENT OF LEFT LIMITS

Figure 3-c shows that four graphs may be practical on a single frame. The SETMIV statements used to produce the margins for each grid are shown. It is

obvious that only one CALL GRIDIV was allowed to advance the film.

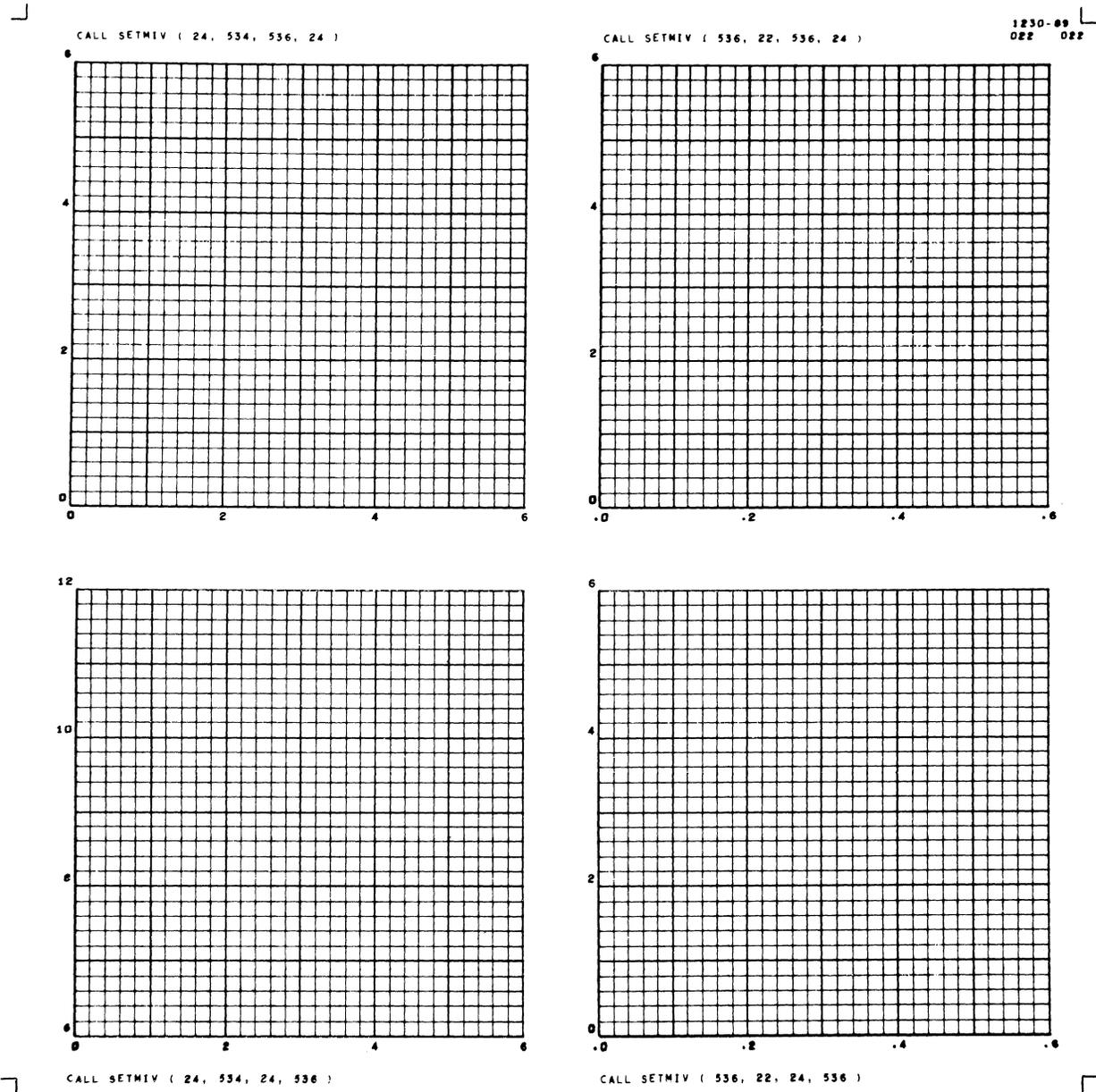


Figure 3-c

FOUR GRAPHS ON ONE FRAME

Combining a Graph and Printed Information on One Frame

Figure 3-d shows an additional exercise combining the use of SETMIV, SCOUTV, and LOCSTV. It is assumed that the values of Y to be plotted are stored in DATA; the values DATA(1) through DATA(5) are to be printed in column 1, DATA(6)

through DATA(10) in column 2, etc.

SETMIV is called to enlarge the top margin to allow for more than one line of printing, and to leave about half the frame free at the bottom for tabular data. SCOUTV is used to print the table, and also the comment "Y = 8190.00 AT X = 15.5" on the graph itself.

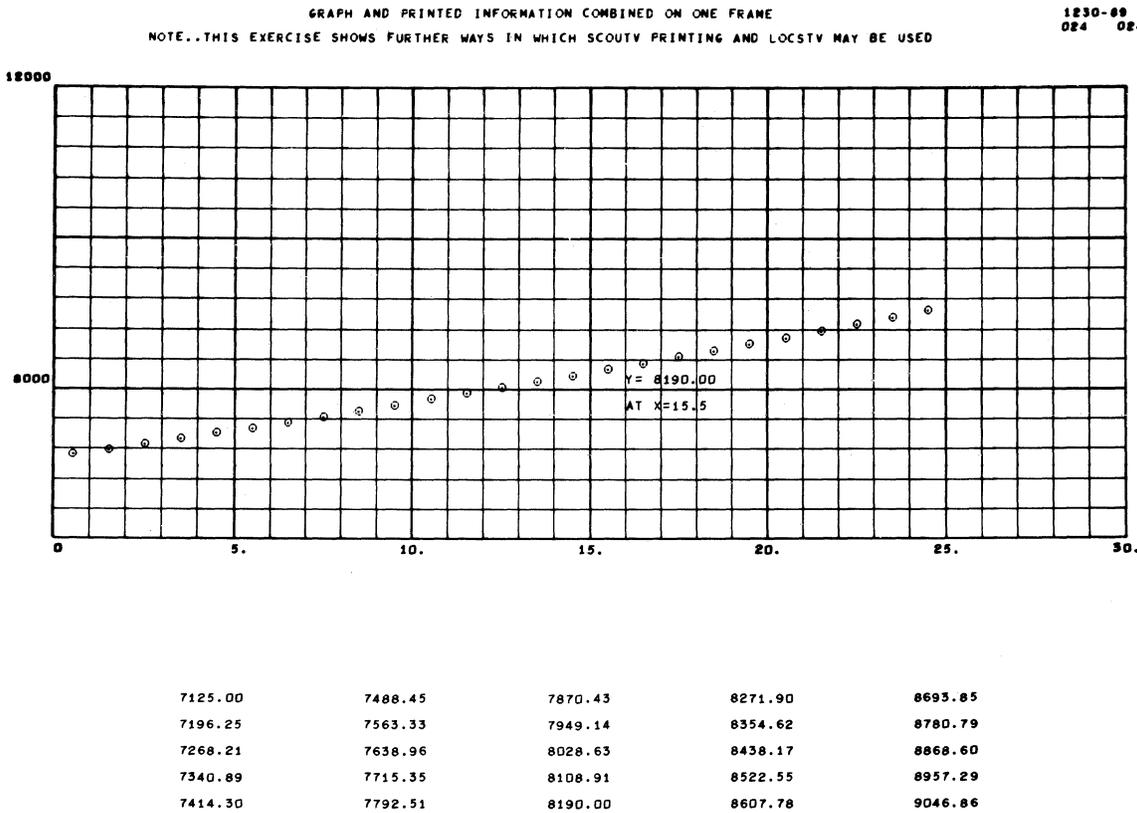


Figure 3-d

GRAPH AND PRINTED INFORMATION ON ONE FRAME

10-15-63

In the following list of statements used for Figure 3-d, note the use of NXV, NYV, and LOCSTV to

set the comment on the graph in the proper position.

```

CALL SCOUTV
CALL SETMIV (24,0,550,48)
CALL GRID1V(1,0,0,30,0,6000,0,12000,0,1,0, 400,0,5,5, 5,10,3,5)
X=0.5
DO 1660 I=1,25
CALL POINTV (X,DATA(I),1)
1660 X=X+1.0
NXL=NXV(15.5) +19
NYL=NYV(DATA(15)) -3
CALL LOCSTV (NXL,NYL,0)
WRITE OUTPUT TAPE 16,1700, DATA(15)
1700 FORMAT( 3H+Y= F8.2 / 10H AT X=15.5 )
CALL LOCSTV (118,450,0)
DO 1750 J=1,5
WRITE OUTPUT TAPE 16,1740,DATA(J),DATA(J+5),DATA(J+10),DATA(J+15),
1DATA(J+20)
1740 FORMAT(1H 8X ,F8.2,12X,F8.2,12X,F8.2,12X,F8.2, 12X,F8.2 )
1750 CONTINUE

```

Another exercise in the use of SCOUTV to print tabular data appears in Figure 2-d.



10-15-63

Providing for Special Label Characters: SETCIV, SETCØV

GRID1V computes the starting location of each label, taking into consideration the size of the characters used. If the labels are to be placed outside the grid, GRID1V assigns space for them, again taking the character size into consideration. Normally, labeling is done in Charactron characters (via LABLV). If the programmer substitutes a non-system labeling routine for LABLV, it may be necessary to furnish adjusted character dimensions to GRID1V.

To state the dimensions of nonstandard label characters, use

CALL SETCIV (IW, IH)

- IW An integer which specifies, in raster counts, the allowance needed for the width of each label character.
- IH An integer which specifies, in raster counts, the allowance for the height of a label character.

If SETCIV is never called, the indicator table contains IW = 8 and IH = 10. Obviously, if it is called, the arguments must be compatible with the size of the characters employed by the LABLV subprogram used.

GRID1V retrieves the values of the indicators by using:

CALL SETCØV (IWL, IHL)

The width will be retrieved from the table and stored in the fixed point variable location IWL, and the height will be similarly stored in IHL. (The arguments must not be constants.) Note that GRID1V uses this information to control the space that will be reserved for labels; it does not control the size of the label characters themselves in any way.

Holding Margins from Graph to Graph: HØLDIV, HØLDØV

For a large graph, it may be necessary for the programmer to display segments of the graph in separate frames, and join the segments "tile fashion" to form the complete plot. If the graphs are to

have the same scale, certain equalities should exist.

1. The range of X in each segment should be equal, and the range of Y should be equal. In other words, the quantity (XR - XL) should be the same in each segment, and, similarly, the quantity (YT - YB).
2. The dimensions of the scaled area should be the same from segment to segment; that is, the dimensions of the space between margins should be equal.

The programmer can easily provide for equality in the ranges of X and Y, but he cannot so readily ensure equality in the scaled areas. Since GRID1V computes label margins (and, therefore, total margins) to suit the needs of each graph, the dimensions of the scaled area may vary.

One method that will usually give equality of scaled areas is to specify the option that forces labels to be placed outside the grid, and to always request the same number of label characters (NX, NY) for each segment. If this is not practical, a "holding" feature is provided.

GRID1V can be instructed to hold the label margin spaces used for the preceding grid and use them in computing total margins for the next grid. The statement to be used is

CALL HØLDIV (NH)

If NH \neq 0, the label margins from the preceding grid will be used again. If NH = 0 (as is the case if HØLDIV is never called), label margins will be computed in the normal manner.

The status of this indicator is tested in GRID1V by using

CALL HØLDØV (NHL)

The value of the indicator will be retrieved and stored in the location named in the argument.

The "hold" may be released by executing the statement

CALL HØLDIV (0)



8-20-64

LOG AND SEMI-LOG PLOTTING

The examples in the S-C 4020 First Reader used only linear scaling and conversion. The modal subroutine SMXYV can be used to alter the scaling mode such that scaling and conversions will be made in the logarithmic mode.

The call statement for establishing the logarithmic mode is:

CALL SMXYV (MX, MY)

where MX and MY are scale mode indicators that designate whether the logarithmic or linear mode is to be used:

If $MX \neq 0$, $MY \neq 0$ Log in X, log in Y
 $MX \neq 0$, $MY = 0$ Log in X, linear in Y
 $MX = 0$, $MY \neq 0$ Linear in X, log in Y
 $MX = 0$, $MY = 0$ Linear in both X and Y (to restore linear mode)

At the beginning of each job, MX and MY are zero, so that linear scaling and conversion result if SMXYV is never called.

If the programmer wants to generate a log or semi-log plot, he must call SMXYV to set the logarithmic mode before using any S-C 4020 subroutines that involve scaling or conversion.

Once the log-log or semi-log mode has been set by SMXYV and the scale factors have been established, the function statements NXV and NYV can be used to convert data coordinates into raster coordinates, in the same manner as shown in the First Reader for linear scaling (page 74.10.25).

The contents of the scale mode indicators may be retrieved by using the following statement:

CALL MSXYV (MXL, MYL)

The indicator for the X scale mode will be stored in MXL, and for the Y scale mode in MYL.

GRID1V uses this statement to determine what scale mode has been selected by the programmer.

Restrictions on Logarithmic Mode

In general, once SMXYV has been called, the pro-

grammer can use any of the routines introduced in the First Reader to generate a logarithmic display. (One exception: DXDYV should not be used to generate arguments for GRID1V in the direction in which logarithmic scaling is being used.)

However, some of the GRID1V arguments are restricted in the logarithmic mode. Following is a list of the 13 arguments, with notations as to the arguments affected if the mode is logarithmic in the direction affected by each.

CALL GRID1V (L, XL, XR, YB, YT, DX,
DY, ±N, ±M, ±I, ±J, ±NX,
±NY)

L	Controls film advance; see page 74.10.05.
XL, XR	Left and right limits of the grid. May not be negative or zero.
YB, YT	Bottom and top limits of the grid. May not be negative or zero.
DX, DY	Should be set to 1.0. If greater, only the cycle lines will be displayed. If less, no lines will be drawn.
N, M	Will be ignored; however, an argument must be present. A negative sign on N or M will force the grid to be square.
I, J	Will be ignored; however, an argument must be present.
NX, NY	Number of characters to be displayed in the labels; see page 74.10.05.

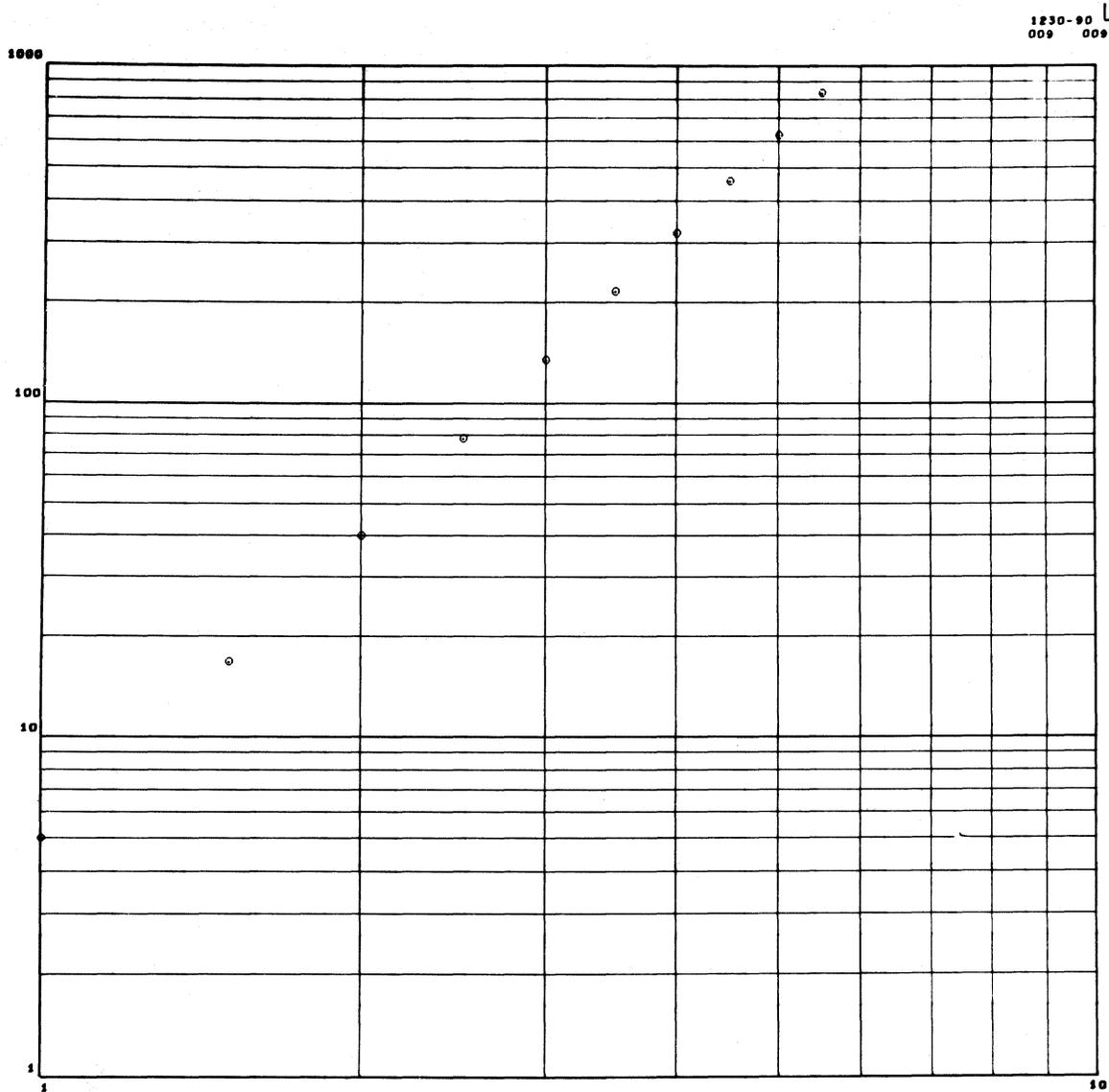
Generally, line labels will be placed only at the cycle lines. If, however, the grid spans less than one complete cycle, each grid line will be labeled.

No more than 10 log cycles are permitted in the GRID1V system.

Examples

Figure 3-e illustrates a plot that is logarithmic in both the X and Y directions. Note the arguments for SMXYV and GRID1V in the coding.

Figure 3-f illustrates a semi-log grid, with plotting of the data used in figure 3-e.



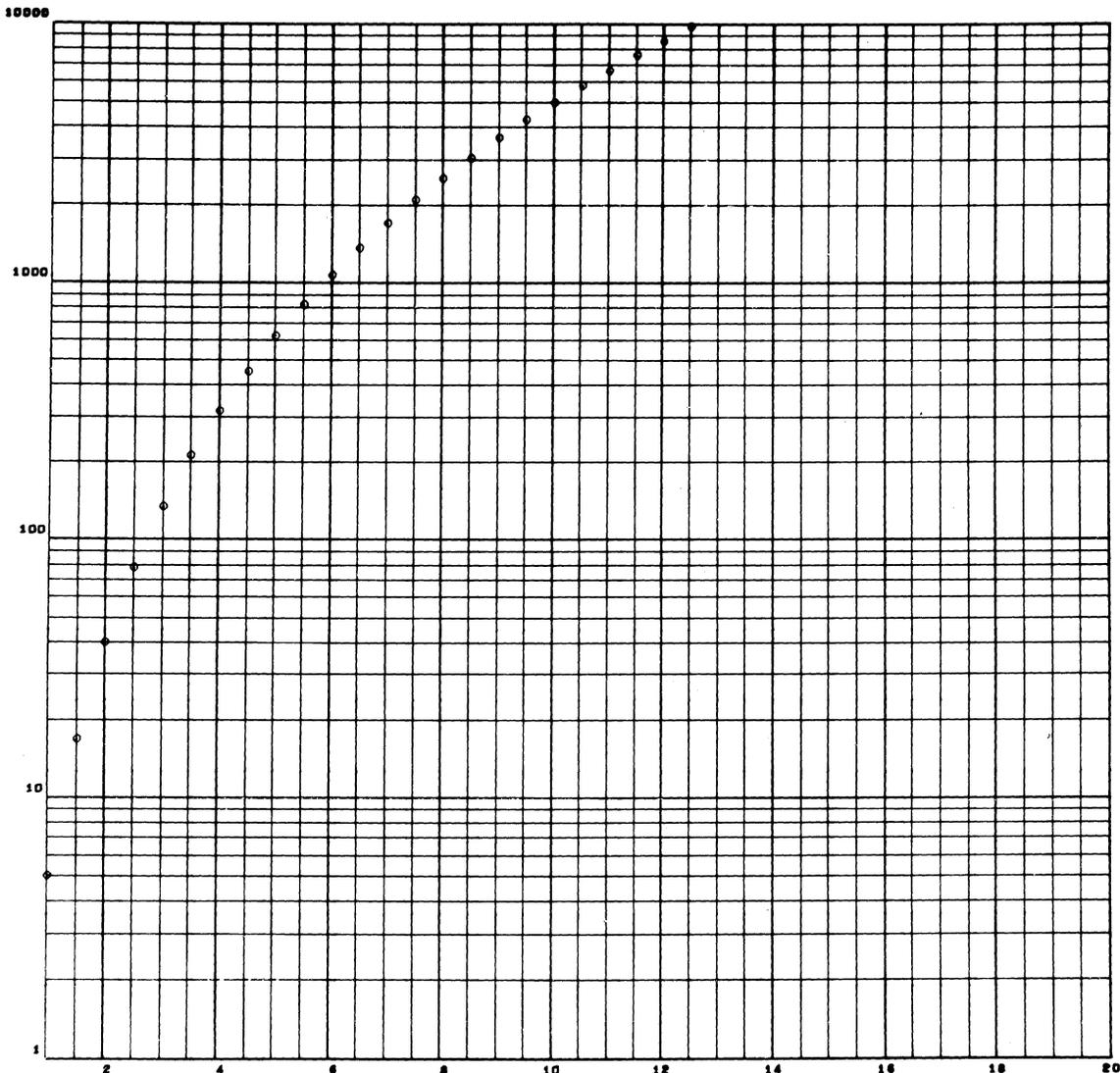
```
CALL SMXYV ( 1, 1 )
CALL GRID1V ( 1, 1.0, 10.0, 1.0, 1000.0, 1.0, 1.0, 1, 1, 1, 1, 2, 4 )
POINTS PLOTTED BY USING CALL POINTV ( X,Y, 1 ) IN A LOOP
```

Figure 3-e

LOGARITHMIC GRID

```
CALL SMXYV ( 1, 1 )
CALL GRID1V ( 1, 1.0, 10.0, 1.0, 1000.0, 1.0, 1.0, 1, 1, 1, 1, 2, 4 )
X=1.0
DO 2000 I=1,18
Y= 5.0*X**3.0
CALL POINTV ( X,Y,1 )
2000 X=X+0.5
```

1230-09
 034 034



```
CALL SMXYV ( 0, 1 )
CALL DXDYV ( 1, 1.0, 20.0, DX, N, I, NX, 12.0, IERR )
CALL GRID1V ( 1, 1.0, 20.0, 1.0, 10000.0, DX, 1.0, N, 1, I, 1, NX, 5 )
```

Figure 3-f

SEMI-LOG GRID

This plot is linear in the X direction and logarithmic in the Y direction, initiated by a CALL SMXYV (0,1). Note the use of DXDYV for the linear direction only, and the effect on the GRID1V arguments for each direction.

```
CALL SMXYV ( 0, 1 )
CALL DXDYV ( 1, 1.0, 20.0, DX, N, I, NX, 12.0, IERR )
CALL GRID1V ( 1, 1.0, 20.0, 1.0, 10000.0, DX, 1.0, N, 1, I, 1, NX, 5 )
X=1.0
DO 2090 K=1,37
Y= 5.0*X**3.0
CALL POINTV ( X, Y, 1 )
2090 X=X+0.5
```

The data points could have been connected by lines in either of the preceding examples. The following coding shows how this could have been done for Figure 3-e. (Note that the procedure is the same as that shown on page 74. 10. 25 for linear mode.)

$$X = 1.0$$

$$Y = 5.0 * X ** 3.0$$

$$DØ 2000 K = 2, 18$$

$$X2 = X + 5.0$$

$$Y2 = 5.0 * X2 ** 3.0$$

$$NX1 = NXV(X)$$

$$NX2 = NXV(X2)$$

$$NY1 = NYV(Y)$$

$$NY2 = NYV(Y2)$$

CALL LINEV (NX1, NY1, NX2, NY2)

$$X = Y2$$

$$2000 Y = Y2$$

Labels in Scientific Notation

Labels in scientific notation can be obtained on a logarithmic grid, in the way shown for linear grids on page 74. 10. 24. Figures 3-g and 3-h demonstrate that positive values of NX and NY will result in fixed point labels, while negative values will produce scientific labels.

For scientific labels of logarithmic grids, it is only necessary to use -1 or -2 for the specification, because labels will normally be powers of 10, written as $1 \times 10^{\pm XX}$ or $1.0 \times 10^{\pm XX}$.

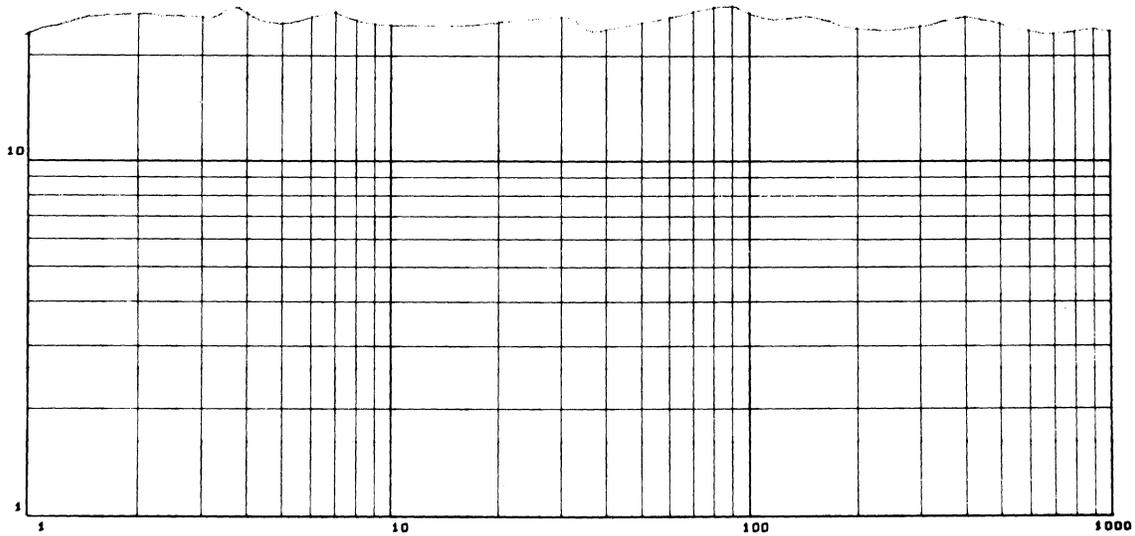


Figure 3-g

LOG GRAPH WITH FIXED POINT LABELS

CALL SMXYV (1, 1)

CALL GRIDIV (1, 1.0, 1000.0, 1.0, 1000.0, 1.0, 1.0, 1, 1, 1, 1, 4, 4)

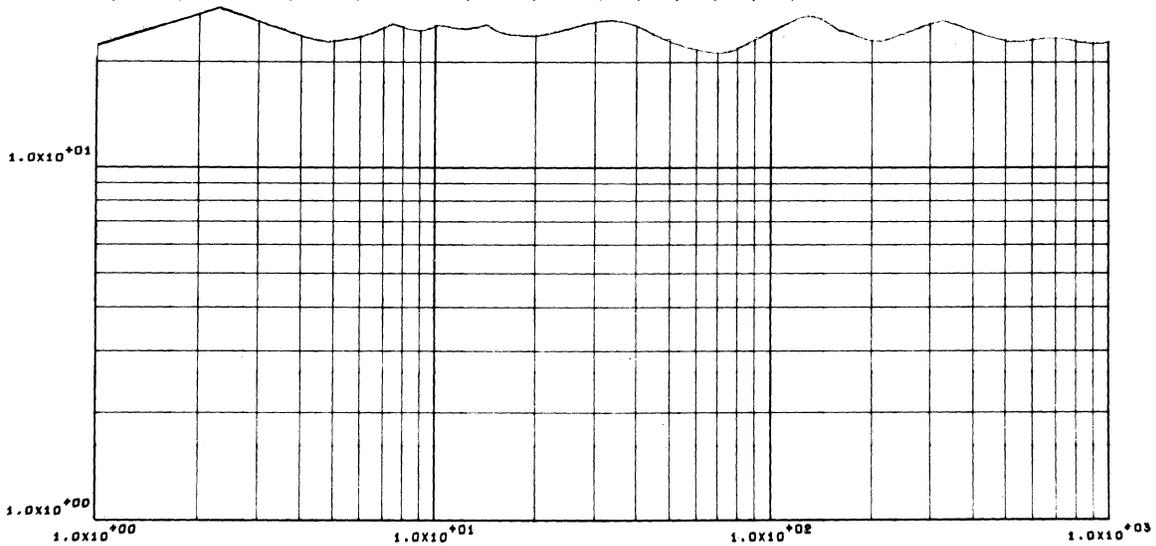


Figure 3-h

LOG GRAPHS WITH SCIENTIFIC LABELS

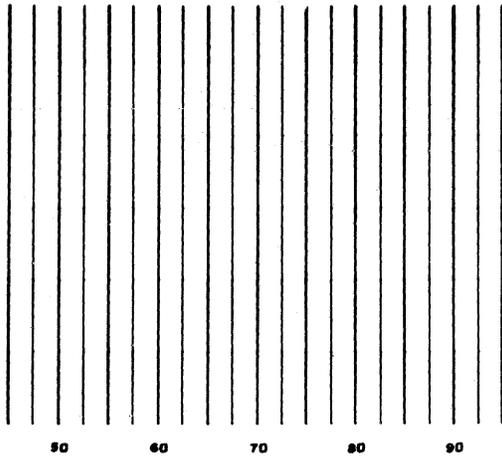
CALL SMXYV (1, 1)

CALL GRIDIV (1, 1.0, 1000.0, 1.0, 1000.0, 1.0, 1.0, 1, 1, 1, 1, -2, -2)

Note that the use of -2 is adequate for scientific labels of log grids because all the labels will be of the type "1.0X10^{±XX}."

EXAMPLES OF USE OF LINRV

1230-89
040 040



```
CALL FRAMEV
CALL XSCALV ( 45.0, 95.0, 50, 544 )
CALL LINRV ( 1, 515, 535, 900, 45.0, 95.0, 2.5, 2, 4, 2, 0 )
```

```
CALL SMXYV ( 0, 1 )
CALL YSCALV ( 1.0, 10.0, 20, 575 )
CALL NONLNV ( 2, 520, 550, 1023, 1.0, 10.0, 1.0, 1, 1, 2, 0 )
```

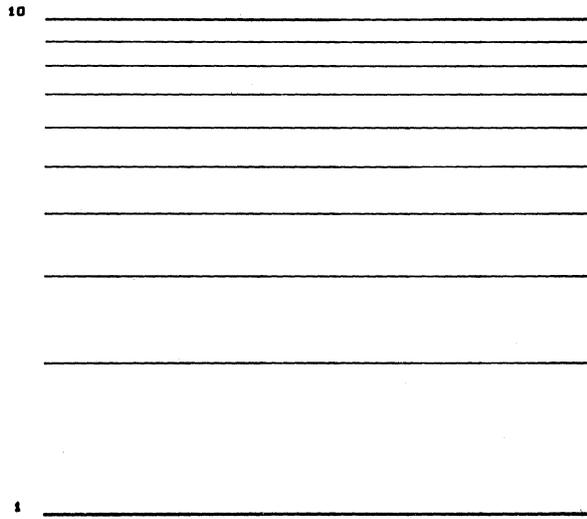


Figure 3-i

EXAMPLES OF NONLNV AND LINRV OUTPUT

ADDITIONAL MODULES FOR SPECIAL GRIDS:
LINRV, NØNLNV

The modules LINRV and NØNLNV, which are used at a lower level by GRID1V, can be employed by the programmer as building blocks for the creation of unusual grids. The top half of Figure 3-i shows how LINRV might produce only the vertical portion of a linear grid. It can also produce only the horizontal portion of a linear grid.

The bottom half of Figure 3-i illustrates the horizontal portion of a log grid displayed by NØNLNV. If desired, just the vertical grid lines can be drawn.

These modules are particularly helpful in the following situations:

1. The programmer needs a standard grid in one dimension and a grid of his own design in the other dimension.
2. The programmer must control the length of the grid lines.

A special advantage is the fact that DXDYV may be used with LINRV in much the same way as it is used with GRID1V.

Since additional knowledge of scaling routines is needed before the programmer attempts to use LINRV and NØNLNV, the particulars of the latter subprograms has been deferred to Section 74.5, which follows the section on Scaling and Conversion.



Section 74.4

SCALING AND CONVERSION

In earlier sections, scaling and conversion problems have been considered principally in connection with higher-level routines. However, much of the actual computation is done in lower-level modules. This section describes these and some associated modules that provide additional tools for some scaling and conversion problems.

The descriptions may also be useful in clarifying the operation of higher-level routines. For example, GRIDIV uses XSCALV, YSCALV as a lower-level routine to do scaling. Consequently, the comments in this section concerning scaling and conversion equations, scale factors, and retrieving and resetting scale factors also apply when the programmer uses GRIDIV to control scaling.

Methods of operation in both the linear and nonlinear modes are also discussed. The nonlinear mode built into the system is the logarithmic mode, but the possibility of substitution of other nonlinear modes is mentioned.

BASIC SCALING SUBPROGRAMS: XSCALV, YSCALV

XSCALV, YSCALV will compute the scale factors for a specified display and store them in an internal table for later use by those functions which convert data. The calling statements are:

```
CALL XSCALV (XL, XR, ML, MR)
```

```
CALL YSCALV (YB, YT, MB, MT)
```

XL, XR	Floating point values of X for the leftmost and rightmost limits of the scaled plotting area.
ML, MR	The amount of margin space to be reserved to the left and right of the scaled area, expressed in raster counts (fixed point integers).
YB, YT	Floating point values of Y for the bottom and top limits of the scaled plotting area.
MB, MT	The amount of margin space to be reserved below and above the scaled area, expressed in raster counts (fixed point integers).

XSCALV, YSCALV contains a test for nonlinear mode. If this mode is indicated, XL, XR and/or YB, YT will be transformed before the scale factors are computed by the basic scaling equations. (The indicators and the transformation will be discussed on page 74.40.07.)

Example

Figure 4-a illustrates the relationship of the arguments. The margin specifications are: ML = 170, MR = 192, MB = 340, MT = 128.

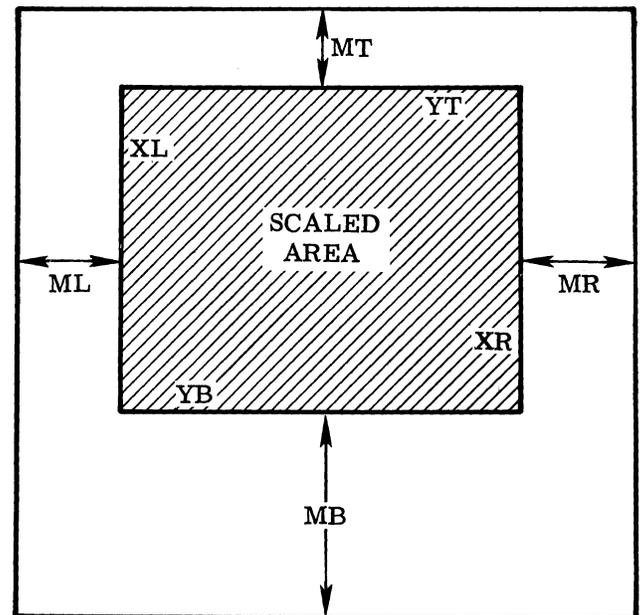


Figure 4-a

XSCALV will assign XL to raster location IX = 170, and XR to raster location IX = 831 (i.e., 1023 - 192). YSCALV will assign YB to raster location IY = 340, and YT to raster location IY = 895 (i.e., 1023 - 128). The scaled area will then be the rectangle from IX = 170 to IX = 831, and from IY = 340 to IY = 895.

Basic Equations for Computing Scale Factors

In the following equations, "A" and "B" represent the scale factors computed and stored by XSCALV, and "C" and "D" are the factors computed and stored by YSCALV. (Since the computation is done in floating point arithmetic, the floating point variable names FML, FMR, FMB, and FMT are used

USE OF NXV, NYV WITH ERROR TESTING

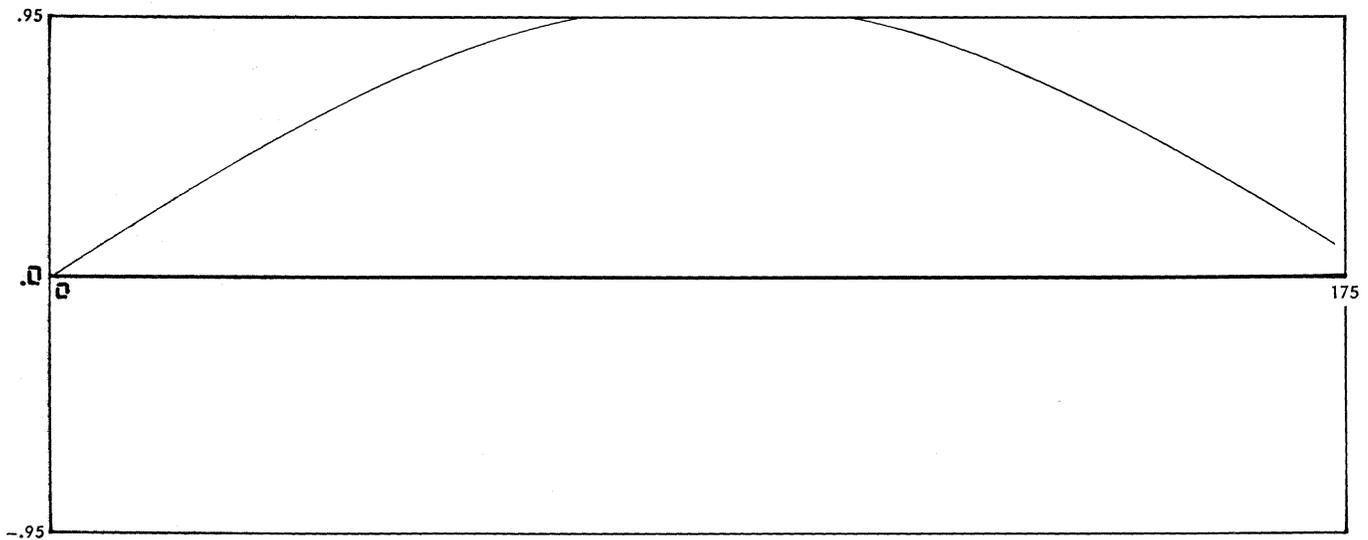


Figure 4-b

This graph illustrates how error testing of the results of NXV, NYV can be used to by-pass plotting of points that are off-scale. NXV and NYV were used to convert the points along the curve into raster positions, and LINEV was employed to

connect the points. Since the results of NXV and NYV were tested for zeros, and plotting was by-passed whenever a point was off-scale, the curve stopped at the top and right limits of the scaled area (outlined). The pertinent statements are:

```
CALL GRIDIV (1.0.0.175.0,-0.95,0.95,175.0,0.95 .1.1.1.1.3.3 )
X=-2.0
DO 4348 I=1.92
Y=SINDF(X)
IX=NXV(X)
IY=NYV(Y)
IF(I-1) 4344,4344,4340
4340 IF (IX*IX0*IY*IY0) 4344,4344,4342
4342 CALL LINEV (IX0,IY0,IX,IY)
4344 IX0=IX
      IY0=IY
4348 X=X+2.0
```

10-15-63

to represent the floating point equivalents of the margin values ML, MR, MB, and MT.)

$$A = \frac{(1023. - FMR) - FML}{XR - XL}$$

$$B = FML - A * XL$$

$$C = \frac{(1023. - FMT) - FMB}{YT - YB}$$

$$D = FMB - C * YB$$

CONVERSION OF DATA: NXV, NYV, IXV, IYV

Four function subprograms, NXV, NYV, IXV, and IYV, are provided to convert data coordinates into raster coordinates. The argument for each of the functions must be a floating point quantity; the result will be an integer quantity. (This means that the mode configuration of the functions is "floating argument, fixed function," as specified on page 60.23.07.)

The following FORTRAN statements show how these functions may be used to convert data coordinates X (or Y) into raster coordinates IX (or IY):

$$IX = NXV(X)$$

$$IY = NYV(Y)$$

$$IX = IXV(X)$$

$$IY = IYV(Y)$$

These four functions are similar in that they all convert data by means of the basic equations for data conversion discussed below. They are dissimilar in the way they handle off-scale data (that is, data which falls outside the limits XL, XR or YB, YT.)

The functions NXV and NYV check for off-scale data values. The result IX (or IY) will be set to zero if the argument X (or Y) is outside the limits that were used to establish the scale. In addition, an error indication is set, as discussed on page 74.40.09.

The functions IXV and IYV do not test for off-scale data values. The resulting position can be outside the plotting area, or even outside the frame, but the value will be properly scaled relative to the plotting area. (However, no test is made for the possibility that the result is greater than 131071; integer bits above the 17th will be lost.)

The two examples shown in Figures 4-b and 4-c illustrate how the differences between NXV, NYV and IXV, IYV may affect plotting.

All four functions test for nonlinear mode. If indicated, X (or Y) will be transformed before it is converted by one of the basic conversion equations.

Basic Equations for Data Conversion

The following equations show how the conversion functions convert data coordinates X and Y into raster coordinates IX and IY:

$$IX = A * X + B$$

$$IY = C * Y + D$$

The scale factors A, B, C, and D are those derived from the equations at the beginning of this page.

Generally speaking, the programmer should use the conversion functions rather than writing statements of his own containing these equations. The functions offer the following advantages:

- They have direct access to the internal table in which the scale factors A, B, C, and D are stored.
- They check the scale mode, and use a nonlinear conversion if that mode is indicated. (See page 74.40.07.)
- NXV, NYV contains a test for off-scale points.

If it is ever necessary to use these equations directly, the programmer can retrieve the scale factors (A, B, C, D) by employing the routine SCLSAV (page 74.40.05).

The Inverse of Conversion: UXV, UYV

The functions UXV, UYV allow the programmer to obtain the coordinates of a specified raster location in terms of his data. The following statements show how these functions may be used:

$$X = UXV(IX)$$

$$Y = UYV(IY)$$

Although UXV, UYV represent the inverse of IX = IXV(X) and IY = IYV(Y), the results are approximate (because truncation occurs in the IXV and IYV functions).

USE OF NXV, NYV, IXV, IYV WITHOUT ERROR TESTING

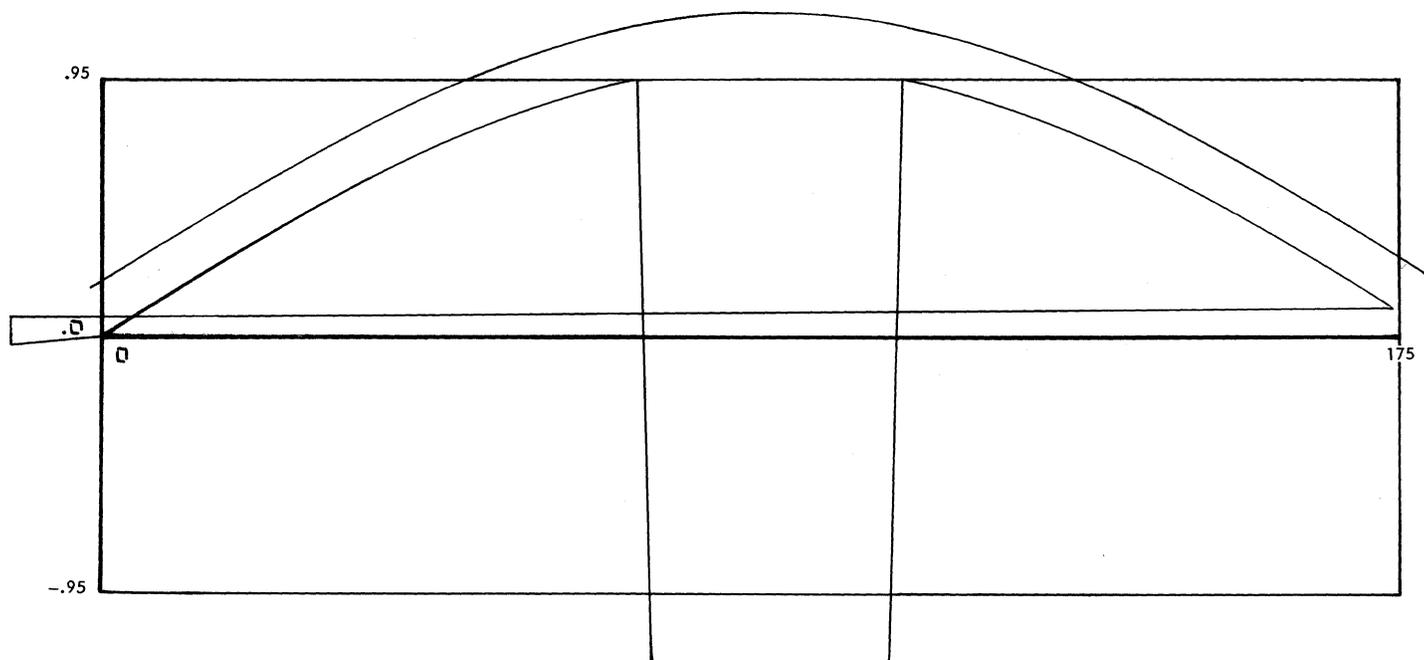


Figure 4-c

Two sine curves are shown, one plotted after using NXV, NYV to do the conversion, and the other after IXV, IYV were used. As in Figure 4-b, LINEV was used to connect the points. NO ERROR TESTS WERE MADE.

The lower curve shows the line going to zero when off-scale values were encountered by NXV, NYV. Note that this curve drops to the bottom of the frame (IY = 0) when values that were off-scale in Y were encountered. Also note that the off-scale initial and last values of X caused the curve to

start and end at the left edge of the frame (IX = 0).

The higher curve was drawn after IXV, IYV were used to convert the points. The curve continued past the boundaries of the scaled area when off-scale values were encountered. (When IXV, IYV are used, the programmer must decide what action should be taken when the result of IXV and/or IYV is < 0 or > 1023.)

The following statements were used for the figure above:

```

CALL GRIDIV(2.0,0,175.0,-0.95,0.95,175.0,0.95,1,1,1,1,3.3 )
X=-2.0
DO 4500 I=1,92
Y=SINDF(X)
IF(I-1) 4480,4480,4470
4470 CALL LINEV (NXV(X), NYV(Y), NXV(X0), NYV(Y0) )
4480 X0=X
Y0=Y
4500 X=X+2.0
C CURVE USING IXV,IYV
X=-2.0
DO 4590 I=1,92
Y= SINDF(X)+.20
IF(I-1) 4570,4570,4560
4560 CALL LINEV (IXV(X),IYV(Y),IXV(X0),IYV(Y0) )
4570 X0=X
Y0=Y
4590 X=X+2.0

```

10-15-63

NOTE

UXV, UYV CANNOT BE USED
IN THE NONLINEAR MODE.

The equations used by UXV, UYV are the inverse
of the equations for data conversion:

$$X = \frac{IX - B}{A}$$

$$Y = \frac{IY - D}{C}$$

(The computation is performed in floating point
arithmetic.)

RETRIEVAL OF SCALING INFORMATION: SCLSAV

The subroutine SCLSAV will retrieve scale factors
and other scaling information from an internal
table and store them in an array named by the pro-
grammer. Although SCLSAV makes scaling infor-
mation available for special-purpose conversions,
certain limit tests, etc., its principal value is that
it permits saving scaling information from one
chain link to another. During one chain link, the
array containing this information can be stored in
COMMON. When a new link is entered, another
routine (RESCLV, page 74.40.05) can be called to
restore the scaling information in the internal
table, where it is accessible to NXV, NYV, IXV,
or IYV. (Note that the latter routines communicate
with the internal table; they cannot use the array
named by the programmer.)

The calling statement to retrieve the scaling
information is:

CALL SCLSAV(R)

R The name of a ten-cell array, named
and dimensioned by the programmer

The storage locations in the block of cells, R, are
assigned as follows:

R(10)	Minimum IY	
R(9)	Minimum IX	
R(8)	Maximum IY	
R(7)	Maximum IX	
R(6)	D	
R(5)	B	Scale factors
R(4)	C	(Page 74.40.03)
R(3)	A	
R(2)	Scale mode indicator for Y	
R(1)	Scale mode indicator for X	

Resetting Scaling Information: RESCLV

If the scaling information has been stored in
COMMON by SCLSAV, it can be reset into the
internal table when a new chain link is entered.
The statement is:

CALL RESCLV(R)

where R is the ten-cell array described under
SCLSAV.



NONLINEAR SCALING AND CONVERSION

The only nonlinear capability built into the system is logarithmic scale mode, introduced in Section 74.3 in connection with log grids. However, the system design allows the programmer to incorporate some special nonlinear scale mode, by the substitution of one module of his own. For this reason, the more inclusive term, "nonlinear," is used in the following discussion, instead of "logarithmic."

Linear-Nonlinear Scale Mode Indicators: SMXYV, MSXYV

The subprogram SMXYV allows the programmer to set scale mode indicators; it must be called if nonlinear scaling is desired. These indicators are tested within other subprograms (GRID1V, IXV, IYV, NXV, NYV, XSCALV, YSCALV). If nonlinear mode is indicated, additional steps will be taken to handle the selected mode. (The programmer may also retrieve these indicators, using the subprogram MSXYV.)

The statement which sets the scale mode indicators is:

CALL SMXYV (MX, MY)

MX \neq 0, MY \neq 0	X nonlinear, Y nonlinear
MX \neq 0, MY = 0	X nonlinear, Y linear
MX = 0, MY \neq 0	X linear, Y nonlinear
MX = 0, MY = 0	X linear, Y linear

To reset the indicators for linear-linear, use CALL SMXYV (0,0). If SMXYV is never called, the scale mode will be linear in X and in Y, and the internal scale mode indicators will be set as if CALL SMXYV (0,0) had been executed.

For chain jobs, note that the values of MX and MY do not carry over from link to link. If a scale mode other than linear-linear is desired, it will be necessary to restate the SMXYV statement in each chain link.

FAP subprograms use two other references in connection with these indicators, causing the names (XXXX) and (YYYY) to appear on the load map.

The following statement will retrieve the scale mode indicators for testing:

CALL MSXYV (MXL, MYL)

MXL, MYL Locations in which the quantities furnished as arguments in the last SMXYV statement will be stored.

Nonlinear Transformation: XMØDV, YMØDV

The scaling and conversion equations shown on page 74.40.03 not only apply to the linear mode, but also can be adapted to the nonlinear mode if transformed arguments are used. The functions XMØDV, YMØDV are provided to perform such transformation; they can be used in statements of the following type:

XPRIME = XMØDV(X)

YPRIME = YMØDV(Y)

The scaling and conversion subprograms (XSCALV, YSCALV, IXV, IYV, NXV, NYV) test the scale mode indicators to see if they have been set to nonlinear mode (by a CALL SMXYV). If nonlinear mode is indicated, each of these subprograms will use XMØDV (or YMØDV) to perform a nonlinear transformation on X (or Y) before the remainder of the scaling or conversion takes place. Since the XMØDV, YMØDV functions in the system compute the log of X or Y, the system nonlinear mode is synonymous with log mode.

For a problem requiring special nonlinear transformation, the programmer can substitute subprograms of his own named XMØDV and YMØDV, with one argument and one result. Also, since the system XMØDV and YMØDV are physically contained in one subprogram, both must be replaced if a substitution is made for either. Obviously, such substitute functions must meet the nonlinear scaling and conversion requirements of the entire program (or chain link), since they will replace the system functions.

System subprograms that use XMØDV and YMØDV (either directly or indirectly) are: GRID1V, IXV, IYV, LINRV, NONLNV, NXV, NYV, XSCALV, YSCALV, APLØTV, PØINTV.

WARNING: GRID1V WAS DESIGNED ONLY FOR THE LINEAR, LOG, AND SEMI-LOG OPTIONS. SUBSTITUTION OF A SPECIAL TRANSFORMATION FUNCTION IS DISCOURAGED WHENEVER GRID1V IS BEING USED IN THE NONLINEAR MODE.



8-20-64

OFF-SCALE ERROR DETECTION

Whenever there is a possibility that off-scale data points might be encountered, error tests should be made by the programmer. For example, APLØTV sets an error indicator which should be tested. A zero result from NXV or NYV nearly always indicates an error; a test should be made for this condition.

There are additional situations which demand special error detection procedures. For one thing, if no left and/or bottom margin space is reserved, the conversion of XL and/or YB can produce a legitimate zero result from NXV, NYV. More important, the programmer may be using NXV or NYV indirectly, via other modules, and thus be unable to test the results. For these reasons, additional subprograms are provided for detailed analysis of conversion errors resulting from off-scale points.

Keep in mind that the special procedures which follow are designed for unusual situations, in which normal error testing does not suffice.

Set Conversion Error Indicators: SCERRV

Two internal cells are used by NXV and NYV to store indications of successful or unsuccessful data conversion. The subroutine SCERRV allows the programmer to assign two cells which NXV, NYV will use in place of the internal error cells. In this way, the programmer can name error indicator locations that are accessible to his program. The call statement is:

```
CALL SCERRV (KX, KY)
```

When NXV converts a quantity successfully, it will place a "0" in KX; when unsuccessful, it will store a "1" in KX. NYV will use the cell KY in the same way. The cells named will be used in each subsequent execution of NXV, NYV (including execution via other subprograms) until new cells are named by another call to SCERRV (or the internal cells are reset at the beginning of a new link of a chain job). If possible, tests of these error cells should be made as soon after execution of NXV, NYV as possible, to avoid any possibility that they might be altered by subsequent executions of NXV, NYV.

The following practical examples show how SCERRV can provide error indicator cells in connection with PØINTV. Since PØINTV uses NXV and NYV only once, the named error cells will still contain indications of off-scale errors when control is returned from PØINTV to the calling program. (The examples assume that the scale factors have already been established.)

Example 1

```

.
.
.
CALL SCERRV (KX, KY)

DØ 300 I = 1, N

CALL PØINTV (X, Y, NS)

IF (KX) 700, 200, 700      Test for non-
                             zero KX and/or
200 IF (KY) 700, 300, 700  KY

300 CØNTINUE

.
.
.
700 CALL DUMP              Dump when an
                             off-scale point
                             is encountered

```

Example 2

```

.
.
.
CALL SCERRV (KX, KY)

M1 = 0

M2 = 0

DØ 300 I = 1, N

CALL PØINTV (X, Y, NS)

M1 = M1 + KX

300 M2 = M2 + KY

.
.
.

```

In example 2, M1 and M2 will contain the total number of points that are off-scale. The contents of these locations may be printed out at the end of the job or after all points have been plotted on each frame.

Saving and Resetting Error Indicator Cells:
SERSAV, SERREV

As has been pointed out, the locations assigned by SCERRV will be used as error indicator cells until new ones are named (or a new link of a chain job is loaded). As a result, subsequent executions of NXV, NYV, whether on the same level or on a lower or higher one, will alter the contents of the error cells (setting them to 1 for an unsuccessful conversion or to 0 for a successful conversion).

Two problems can occur. The obvious one is that indications of error may be masked by a subsequent execution of NXV, NYV. This can usually be avoided if the programmer completes error tests before there is any possibility that the contents of the error cells might be changed.

The less obvious problem is that the error cells named in one level of a program will not be accessible for error testing in other levels unless special action is taken. One way to solve this problem is to carry the error cell names in the call statements of the subprograms, not always a desirable solution.

The modules SERSAV and SERREV, used in conjunction with SCERRV, offer a convenient means for avoiding these problems in multi-level jobs. The call statements are:

CALL SERSAV (LØCX, LØCY)

This subprogram saves, in LØCX and LØCY, the locations of the cells that are currently being used for off-scale indicators.

CALL SERREV (LØCX, LØCY)

This subroutine resets the off-scale error cell locations that were saved by SERSAV.

The arguments, LØCX and LØCY, must be variable names (either fixed or floating point) which are not used for any other purpose between the execution of SERSAV and of SERREV.

Page 74. 40. 11 shows how the programmer can use these modules to insure that errors occurring in a subprogram will not mask out errors indicated but not yet tested in a main program.

ENGINEER'S COMPUTING MANUAL

10-15-63

Main Program

Subprogram

·
·
CALL SCERRV (LX1, LY1) ·

Names two error indicator cells.

·
·
IX = NXV(X)
IY = NYV(Y)

Error indications will be set in LX1, LY1.

·
·
CALL SUBR → SUBR

·
·
CALL SERSAV (LØCX, LØCY)
CALL SCERRV (LX2, LY2)

·
·
IX = NXV(X)
IY = NYV(Y)

Error indications will be set in LX2, LY2.

·
·
Test LX2, LY2 for error indications.

·
·
CALL SERREV (LØCX, LØCY)

Resets error cells LX1, LY1, using the information stored in LØCX, LØCY by SERSAV.

LX1, LY1 will again be used as the ← RETURN error indicator cells.

·
·
·



Section 74.5

FURTHER ASPECTS OF GRID CONSTRUCTION

OPERATIONAL DETAILS OF GRID1V

GRID1V is, in many respects, an executive routine. It examines the information furnished by the argument list and by certain external subprograms, makes decisions based on this information, and then calls other subprograms to advance the film, compute scale factors, generate the grid, etc.

Initially, GRID1V uses STØPTV to ensure that the typewriter mode is "off," BRITV to ensure that the bright intensity mode is "on," and FRAMEV to advance the film and affix the job number and frame count (unless an option is used that inhibits this).

GRID1V then checks certain internal locations to obtain basic information, by calling the following subprograms:

- SETMØV (Retrieves margin assignments. The programmer may have changed the standard specifications by a CALL SETMIV, page 74. 30. 01.)
- SETCØV (Retrieves character size specifications that programmer may have changed by a CALL SETCIV, page 74. 30. 09.)
- MSXYV (Determines linear or nonlinear mode. Indicators may have been set to nonlinear by a CALL SMXYV, page 74. 30. 21.)
- HØLDØV (Determines whether margin specifications were held over from preceding graph. Indicator may have been specified by a CALL HØLDIV, page 74. 30. 09.)

Computations are made to determine the raster positions to be used as grid boundaries. ERRNLV and/or ERRNLV are called to check the boundaries and the data limits to see if a grid can be produced from this information. (See GRID1V Error Procedure, page 74. 50. 02.) Finally, XSCALV and YSCALV are called to compute scale factors, and LINRV (in the linear mode) and/or NØNLNV (in the logarithmic mode) are called to generate the grid. (LINRV and NØNLNV are described on page 74. 50. 21.)

Determining Grid Boundary Positions

GRID1V sets aside the margin space specified by a prior call to SETMIV (or the standard margin space if SETMIV has not been called). With the exception of the small area in the upper right corner used for frame identification, GRID1V does no writing in these basic margins.

Then GRID1V tests to see if labels are to be placed (or may extend) outside the grid area. Space required for such labels is computed, taking into consideration the type of label (fixed point or scientific), the number of label characters specified in GRID1V arguments plus space for a sign, and the height and width of the label characters.

If any space is needed for labels at the left, right, bottom, or top of the grid, it is added to the basic margins (specified by SETMIV table), to produce the total margins. (HØLDIV, discussed on page 74. 30. 09, can alter this procedure by causing label spaces "held" from a previous grid to be added when computing the total margins.)

If the option for a square grid is specified, the right, or top, total margin will be adjusted so the remaining area will be square.

If the total margins and the grid limits (XL, XR, YV, and YT) meet certain error tests, they are used as arguments of XSCALV and YSCALV in computing the scale factors required for generating the grid and plotting on it.

Since so many items influence the margin assignments, the grid boundaries will rarely fall at the exact raster positions that the programmer might have estimated. If the precise raster positions of the boundaries are required in a program, they should be derived by converting the limits into raster counts. For example:

$$IXL = NXV(XL)$$

$$IXR = NXV(XR) \quad (\text{The right total margin is } 1023 - IXR.)$$

$$IYB = NYV(YB)$$

$$IYT = NXV(YT) \quad (\text{The top total margin is } 1023 - IYT.)$$

GRID1V Error Procedure

Even if bad input data is used, GRID1V will attempt to produce a grid. The philosophy is that some useful information may be revealed, even if the grid is inaccurate.

The grid limits and total margins are tested by lower-level subprograms, ERRNLV (for linear mode) and/or ERRNLV (for log mode). If these tests show that a grid cannot be produced from the given information, some data values are manufactured so that the program can continue. The manufactured quantities are used only internally; data values in the main program will not be affected. (Since ERRNLV and ERRNLV are meant to be used only by GRID1V, the call statements are not given.)

When artificial quantities are used, an error mark (//////) is placed in the upper right corner of the frame by ERMRKV. Although ERMRKV was designed for use by GRID1V, the programmer can place this mark on the frame if he uses the statement.

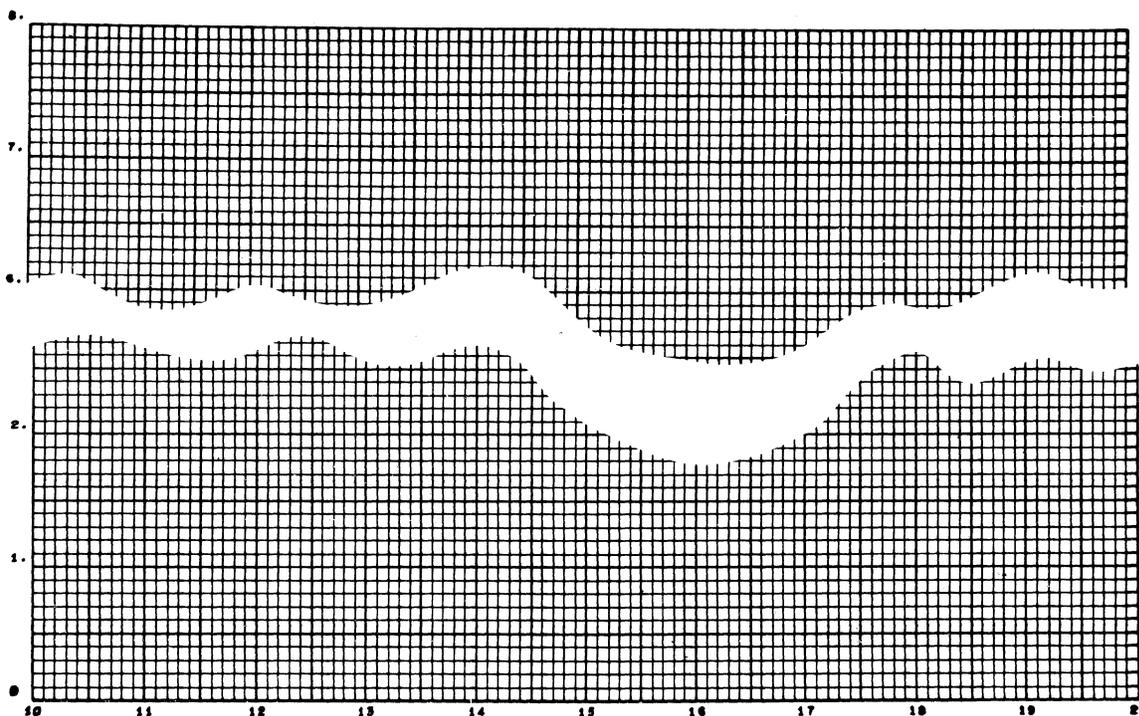
CALL ERMRKV

An error mark from GRID1V may indicate that one or more of the following errors has been found.

1. Equal values have been specified for grid limits; that is, $XL = XR$ and/or $YB = YT$. See Figure 5-a.
2. The specified margins are too wide. In other words, $MR + ML \geq 1023$, and/or $MB + MT \geq 1023$. See Figure 5-b.
3. In a linear grid the specified values of DX and/or DY would result in grid lines spaced closer than 3 raster counts.
4. In a log grid the value of one or more of the limits is either zero or negative.
5. In the log mode there are more than 10 cycles in either the X or Y direction.

GRID1V ERROR HANDLING

SPECIAL GRID1V EXAMPLES

1230-88
004 004
//////

CALL GRID1V (1, 10.0, 10.0, 0.0, 8.0, 0.1, 0.1, 5, 5, 10, 10, 2, 2)

Figure 5-a

CALL GRID1V (1, 10.0, 10.0, 0.0, 8.0, 0.1, 0.1, 5, 5, 10, 10, 2, 2)

The second and third arguments, XL and XR , are equal. GRID1V provided a value of 20.0 in place of XR , and put an error mark in the upper right corner. Scale factors would be computed on the basis of the adjusted grid limits.

NORTH AMERICAN AVIATION, INC. ENGINEER'S COMPUTING MANUAL

GRID1V ERROR HANDLING

SPECIAL GRID1V EXAMPLES

1230-00
000 000



```
CALL SETMIV ( 600, 600, 48, 48 )
CALL GRID1V ( 1, 0.0, 10.0, 0.0, 10.0, 1.0, 1.0, 5, 5, 10, 10, 2, 2 )
```

Figure 5-b

CALL SETMIV (600, 600, 48, 48)

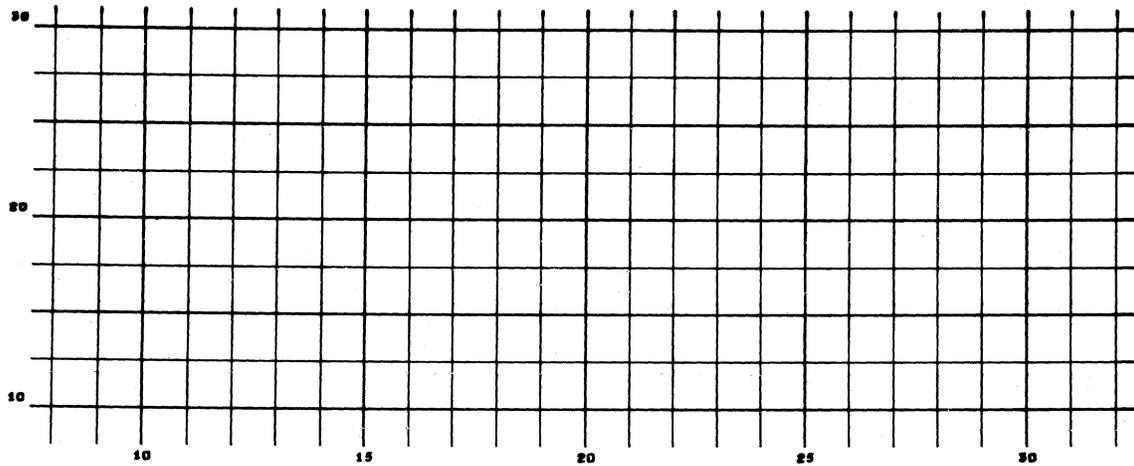
CALL GRID1V (1, 0.0, 10.0, 0.0, 10.0, 1.0, 1.0, 5, 5, 10, 10, 2, 2)

Since $(MTL + MTR) > 1023$, GRID1V manufactured new margin values. This permitted some scale factors to be computed, and allowed the program to proceed. Although this grid is largely meaningless, it reveals the type of error that occurred.

SPECIAL GRIDIV EXAMPLES

1230-00
000 000

EFFECTS OF STEPPING OFF DELTA INCREMENTS FROM ZERO

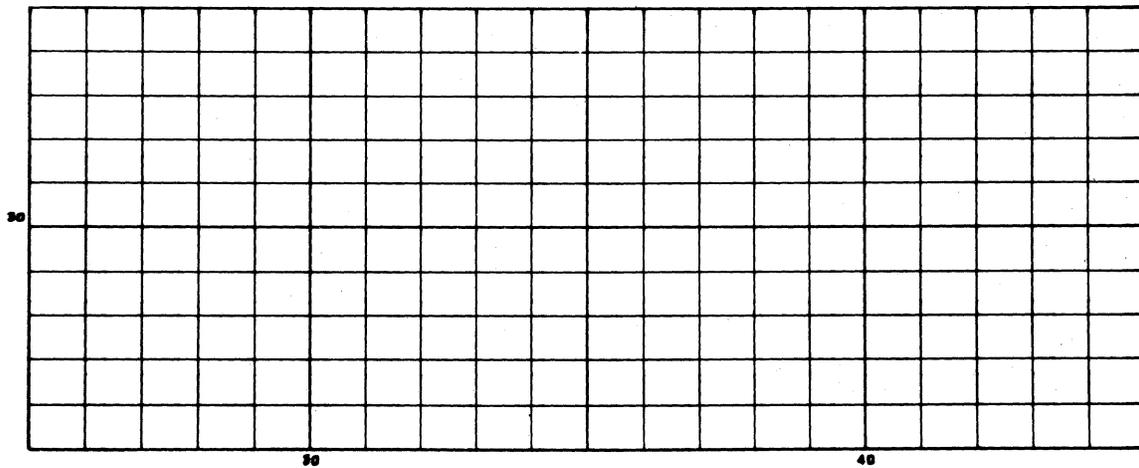


CALL GRIDIV (1, 7.5, 32.5, 8.0, 31.0, 1.0, 2.5, 5, 2, 5, 4, 2, 2)

Figure 5-c

(Example 1)

CALL GRIDIV (1, 7.5, 32.5, 8.0, 31.0, 1.0, 2.5, 5, 2, 5, 4, 2, 2)



CALL GRIDIV (2, 25.0, 45.0, 25.0, 35.0, 1.0, 1.0, 5, 5, 10, 10, 2, 2)

Figure 5-d

(Example 2)

CALL GRIDIV (2, 25.0, 45.0, 25.0, 35.0, 1.0, 1.0, 5, 5, 10, 10, 2, 2)

10-15-63

Examples of GRID1V Characteristics

Certain special characteristics of the grid generated by GRID1V are shown in the following examples. The argument names used are the same as those in the discussion of GRID1V in the First Reader, Section 74.1.

Example 1

During construction of a linear grid, vertical lines are displayed at intervals "stepped off" from $X = 0$ in DX increments until the maximum limit of the grid is exceeded. Then they are "stepped off" from $X = 0$ in the negative direction until the minimum limit of the grid is passed. Similarly, horizontal lines are "stepped off" from $Y = 0$ in DY increments. Because of this method of construction, lines will be generated at the grid limits only if XL and XR are integer multiples of DX , and if YB and YT are integer multiples of DY . Figure 5-c shows a grid without lines at the grid limits for this reason.

Absence of lines along the border of a grid does not indicate an error. If border lines are preferred, the programmer can provide for them by specifying that the GRID1V arguments XL and XR (or YB and YT) are integer multiples of DX (or DY).

Example 2

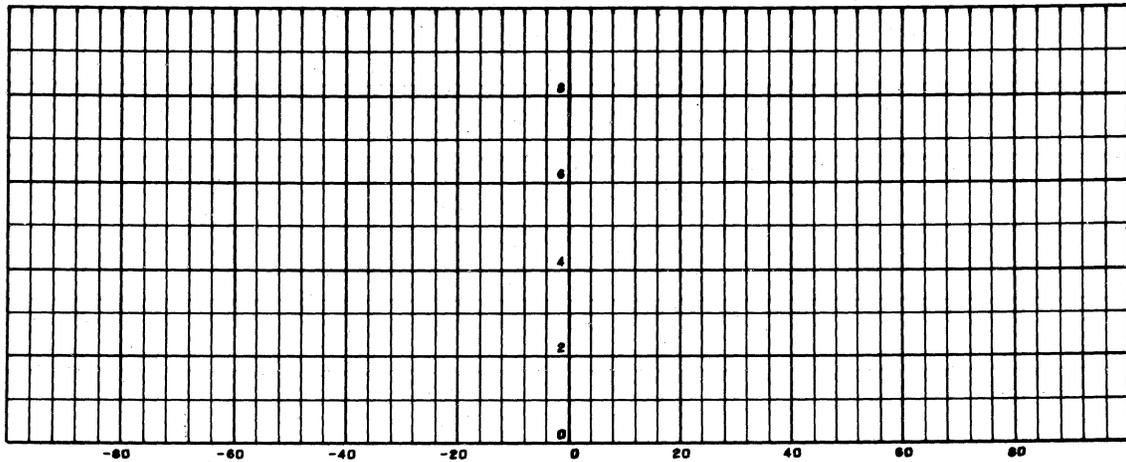
Vertical lines to be emphasized are "stepped off" from $X = 0$ by increments of $N \cdot DX$, and vertical lines to be labeled are "stepped off" by increments of $I \cdot DX$. Similarly, horizontal lines to be emphasized are "stepped off" from $Y = 0$ by increments of $M \cdot DY$, and those to be labeled by increments of $J \cdot DY$. This procedure means that labels and/or emphasized lines will not necessarily occur at the grid limits. Lines at XL and XR will be emphasized only if XL and XR are integer multiples of $N \cdot DX$, and labeled only if XL and XR are integer multiples of $I \cdot DX$. In the same way, lines at YB and YT will be emphasized only if YB and YT are integer multiples of $M \cdot DY$, and labeled only if they are integer multiples of $J \cdot DY$.

This method of determining the positions of line labels is responsible for the absence of labels on the limit lines in Figure 5-d. Notice in the example that the labels for vertical grid lines are adequate; there is no real need to provide labels on the lines at XL and XR . However, the labels for the horizontal lines in this illustration are not adequate; they demonstrate another factor to be considered in planning for line labels. To be meaningful, labels should appear on at least two lines. In the example, the use of $J = 5$, instead of $J = 10$, would cause labeling of the lines at $Y = 25.0$, 30.0 , and 35.0 , giving a scale that can be read easily.

SPECIAL GRID1V EXAMPLES

1230-00
012 012

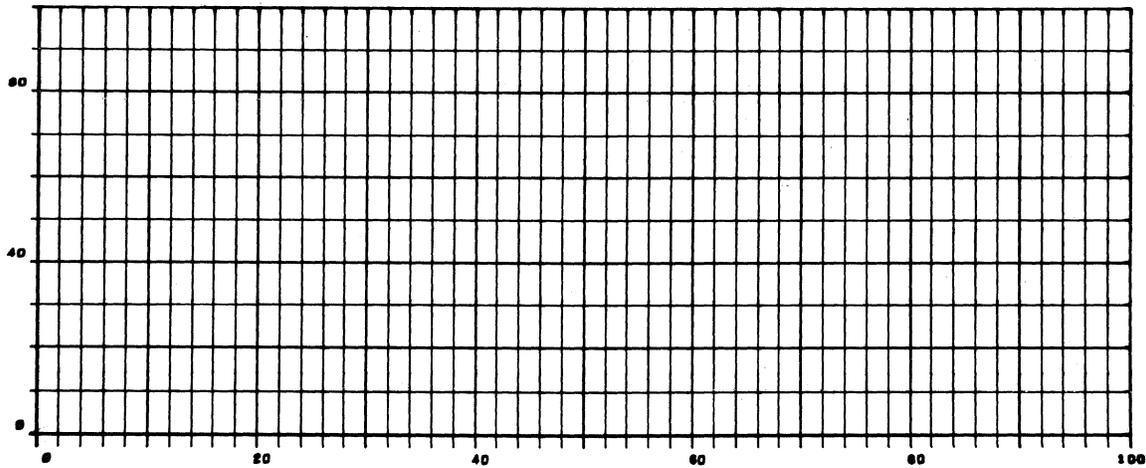
SPECIAL CASES AFFECTING GRID LABELS



CALL GRID1V (1, -99.9999 , 99.9999 , 0.0, 9.9999, 4.0, 1.0, 5, 2, 5, 2, 2, 1)

Figure 5-e
(Example 3)

CALL GRID1V (1, -99.9999, 99.9999, 0.0, 9.9999, 4.0, 1.0, 5, 2, 5, 2, 2, 1)



CALL GRID1V (2, -0.7, 100.0, -3.0, 100.0, 2.0, 10.0, 5, 2, 10, 4, 3, 3)

Figure 5-f
(Example 4)

CALL GRID1V (2, -0.7, 100.0, -3.0, 100.0, 2.0, 10.0, 5, 2, 10, 4, 3, 3)

10-15-63

Example 3

Under certain conditions, GRID1V purposely omits labels. Figure 5-e shows how this can happen. The limits of X used for the grid were -99.9999 and 99.9999, and the limits of Y were 0.0 and 9.9999. During scaling, these limits were rounded, causing them to be treated as -100.0, 100.0 and 0.0, 10.0.

This process caused the decimal scales of the grid limits to be larger than the initial limits, i.e., the values of NX and NY specified by the programmer were not compatible with the new limits. To avoid erroneous labeling, label values will not be displayed if the rounding process causes the decimal scale to be larger than the initial value.

The system does not adequately provide for all problems of this type. Such complications can be avoided if the programmer considers the rounded values of XL and XR, and YB and YT, when specifying NX and NY.

Example 4

When $X = 0$ (and/or $Y = 0$) lies within the grid limits, and I (and/or J) is positive, GRID1V places labels along the $X = 0$ (or $Y = 0$) line. If there isn't enough space for the label between $X = 0$ and the left limit of the grid (or between $Y = 0$ and the bottom of the grid), GRID1V will provide space outside the grid area for labels, just as if negative values of I (and/or J) had been specified. An illustration is shown in Figure 5-f.

Example 5

The number of label characters, NX (or NY), specified for fixed point labels should satisfy the largest and smallest label values to be displayed. Normally, the quantity should be the sum of: (1) the decimal scale of the largest value of X (or Y), (2) the number of fractional positions required in the smallest value, and (3) one more position to provide for the decimal point, if required. The total quantity may not exceed 6 (or 7 if the decimal point is included).

Figure 5-g illustrates how trouble can occur when NX and/or NY are not specified properly. Note that the values of $NX = 1$ and $NY = 1$ are adequate for the largest values of X and Y, but do not allow for the fractional values required in some of the labels. In this example, 3 should have been used for the values of NX and NY, in the top graph.

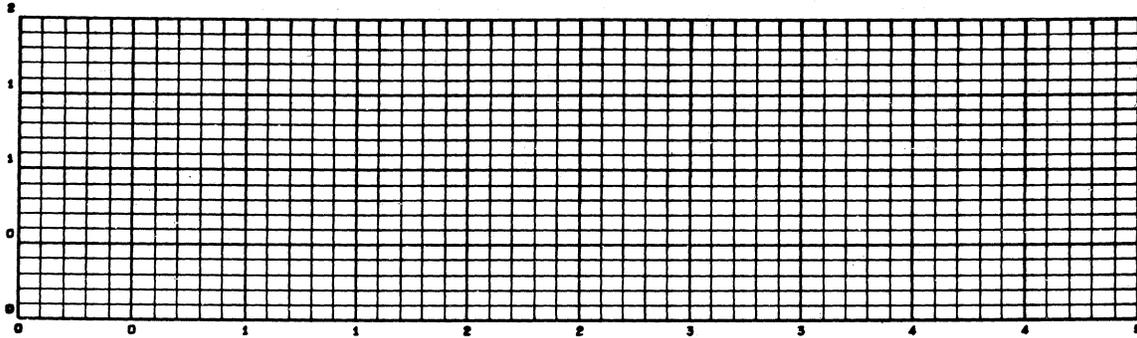
For some applications, it may be advantageous to specify values of NX and/or NY less than the decimal scale of the largest label value. In the middle grid of Figure 5-g, the two low-order positions in the labels have been dropped intentionally, by using values of NX and NY that are less than the decimal scales of the largest values of X and Y. In this case, the resulting graph has been effectively rescaled.

This procedure must be used with care. In the bottom grid, Figure 5-g, the values of NX and NY are so small that the increment between labeled lines is not reflected in the labels that are displayed.

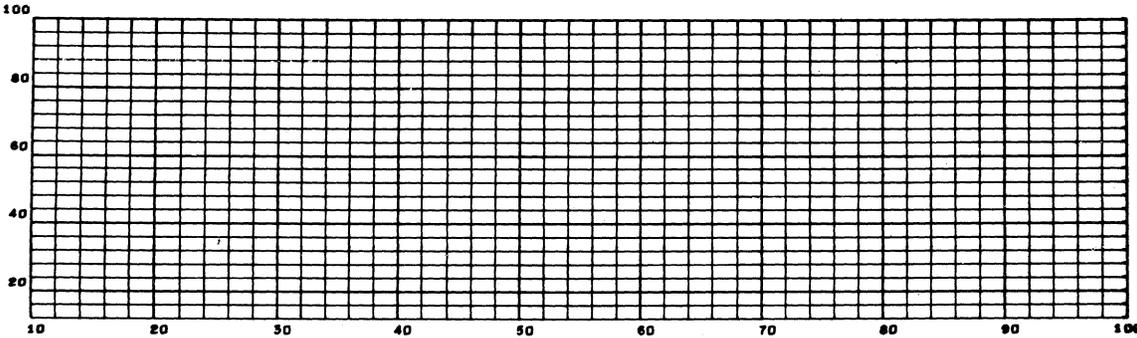
SPECIAL GRIDIV EXAMPLES

FURTHER GRID LABEL CONSIDERATIONS

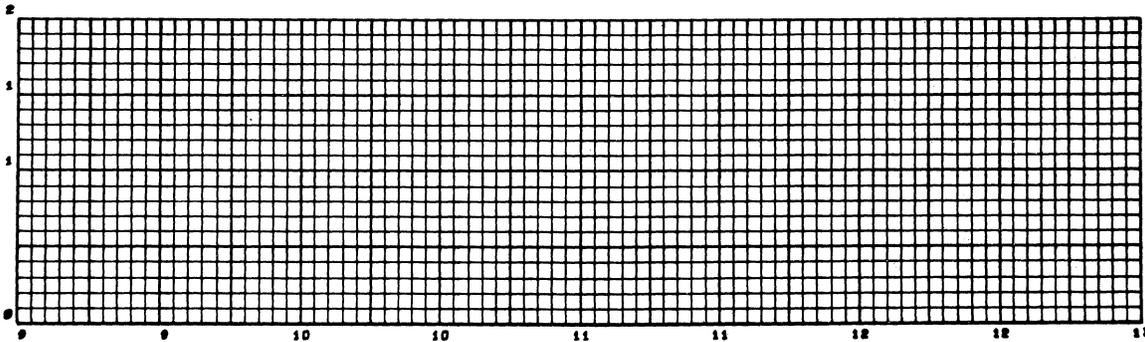
1230-00
014 014



CALL GRIDIV (1, 0.0, 5.0, 0.0, 2.0, 0.1, 0.1, 5, 5, 5, 5, 1, 1)



CALL GRIDIV (2, 1000.0, 10000.0, 1200.0, 10000.0, 200.0, 400.0, 5, 5, 5, 5, 3, 3)



CALL GRIDIV (2, 900.0, 1300.0, 0.0, 2000.0, 5.0, 100.0, 5, 5, 10, 5, 2, 1)

Figure 5-g

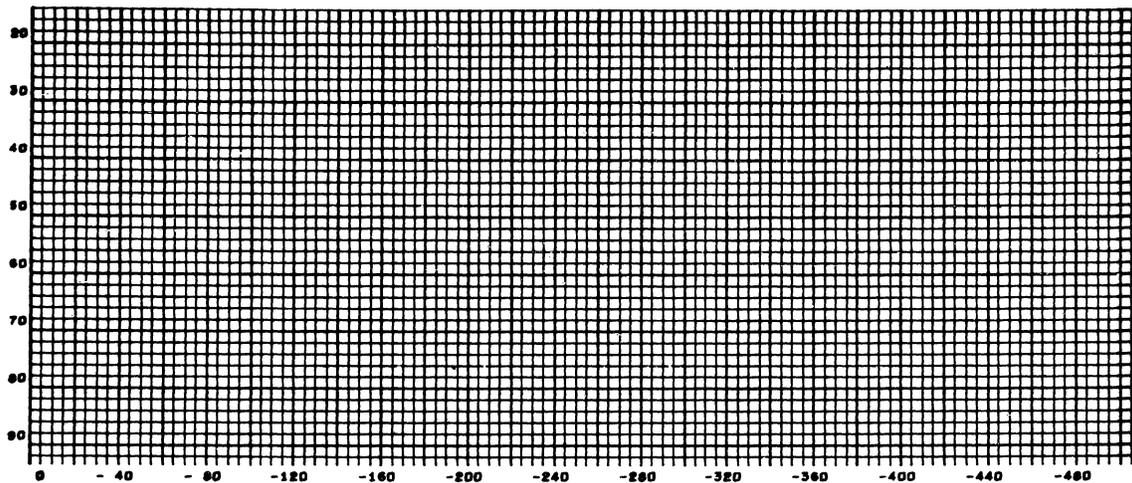
(Example 5)

CALL GRIDIV (1, 0.0, 5.0, 0.0, 2.0, 0.1, 0.1, 5, 5, 5, 5, 1, 1)

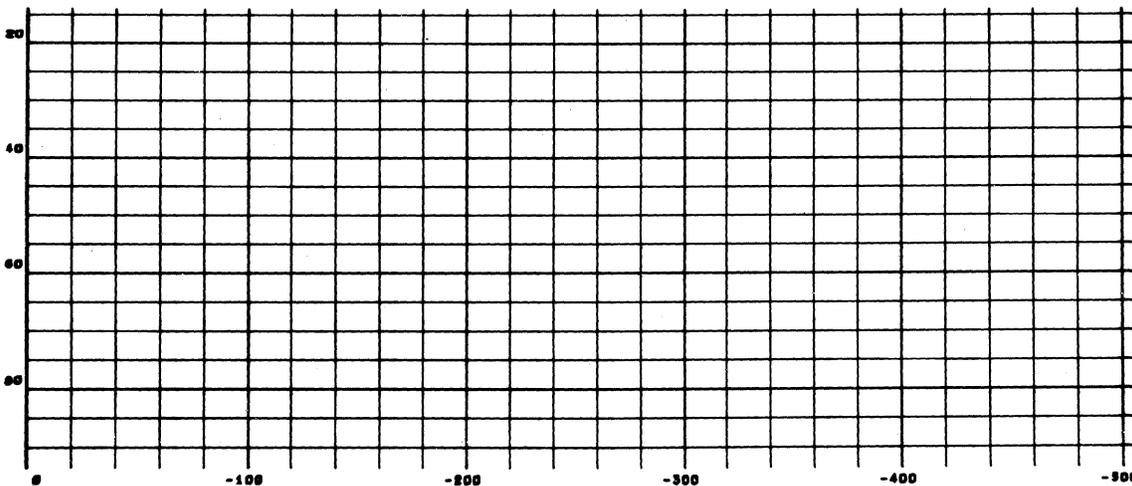
CALL GRIDIV (2, 1000.0, 10000.0, 1200.0, 10000.0, 200.0, 400.0, 5, 5, 5, 5, 3, 3)

CALL GRIDIV (2, 900.0, 1300.0, 0.0, 2000.0, 5.0, 100.0, 5, 5, 10, 5, 2, 1)

1230-07
016 016



```
CALL DXDYV ( 1, 0.0, -505.0, DX, N, I, NX, 8.0, IERR )
CALL DXDYV ( 2, 93.5, 14.0, DY, M, J, NY, 8.0, IERR )
CALL GRIDIV ( 1, 0.0, -505.0, 93.5, 14.0, DX, DY, N, M, I, J, NX, NY )
```



```
CALL DXDYV ( 1, 0.0, -505.0, DX, N, I, NX, 20.0, IERR )
CALL DXDYV ( 2, 93.5, 14.0, DY, M, J, NY, 20.0, IERR )
CALL GRIDIV ( 2, 0.0, -505.0, 93.5, 14.0, DX, DY, N, M, I, J, NX, NY )
```

Figure 5-h

(Example 6)

This frame shows the influence upon the grid of the density factor used by DXDYV. For the case at the top of the figure, a density factor of 8.0 was specified as the 8th argument of DXDYV. A larger factor, 20.0, caused DXDYV to derive values of DX and DY such that the grid at the bottom is less dense.



BUILDING SPECIAL GRIDS: LINRV, NØNLNV

Two of the lower-level modules employed by GRID1V are useful as building blocks for building special grids. LINRV may be used to generate only the vertical portion or only the horizontal portion of a linear grid. NØNLNV can generate only the vertical or only the horizontal portion of a log grid.

In addition to generating the vertical and horizontal portions of a grid in separate operations, these modules offer other special capabilities:

1. The programmer can control the length of the grid lines. For example, the grid lines can be as short as "time tics."
2. The programmer can exert greater control over the position of line labels by specifying of a label reference location.
3. Selected grid lines can be emphasized and/or labeled in the same way as with GRID1V.
4. DXDYV can be used with LINRV in much the same way it is used with GRID1V.

Certain subprograms must be executed prior to the use of LINRV or NØNLNV. The frame must have been advanced, and XSCALV and/or YSCALV must have established scale factors. If a change in scale mode is required, SMXYV must have been called.

The call statement for using LINRV to generate a vertical grid is

```
CALL LINRV (1, LYREFR, IYMIN, IYMAX,
            XL, XR, DX, N, I, ±NX, IW)
```

For a horizontal grid, the statement is

```
CALL LINRV (2, LXREFR, IXMIN, IXMAX,
            YB, YT, DY, M, J, ±NY, IH)
```

LYREFR Label reference locations.
 LXREFR For a vertical grid, LYREFR is the Y raster coordinate that will be used to position the labels of the vertical grid lines. The X raster coordinate will vary, and will be computed by the subprogram.

For a horizontal grid, LXREFR is the X raster coordinate that will be used to position the labels of the horizontal grid lines. The Y

raster coordinates will vary, and will be computed by the subprogram.

The character dimensions, IW and IH (below), should be considered in assigning LYREFR and LXREFR. For example, LYREFR should be chosen to allow at least one character height (IH) below the origin of vertical grid lines (IYMIN). LXREFR should be a raster position that is at least IW* (NY + 1) raster counts to the left of the origin of the horizontal grid lines (IXMIN). In both instances, some additional space should be added to prevent overlapping.

If scientific labels are selected (-NX and/or -NY), provision must be made for 7 additional label characters in the computation of LXREFR. About 5 raster counts extra must be added to LYREFR to allow for the raised exponent.

NOTE: IW*(NY + 1) allows space for a sign.

IYMIN, IYMAX
 IXMIN, IXMAX

Raster positions that determine the origin and end of each line. For a vertical grid, these integers are raster counts in the Y direction. For a horizontal grid, these integers are raster counts in the X direction.

IW, IH

The dimension to be allowed for each label character: IW for the width, and IH for the height. Since LINRV uses LABLV to display the labels, and since the system LABLV uses Charactron characters, the dimensions should be IW = 8 and IH = 10. (If the standard values have not been altered, these values can be obtained by calling SETCØV.)

The remainder of the arguments have the same definitions given for these terms in the discussion of GRID1V, page 74. 10. 04.

The vertical portion of a log grid can be generated by the statement

```
CALL NØNLNV (1, LYREFR, IYMIN, IYMAX,  
            XL, XR, DX, N, I, NX, IW)
```

For a horizontal log grid, the statement is

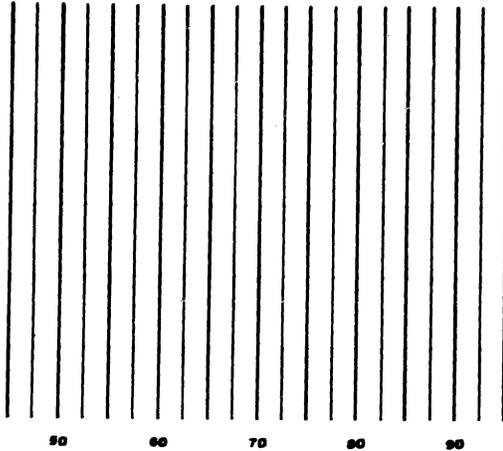
```
CALL NØNLNV (2, LXREFR, IXMIN, IXMAX,  
            YB, YT, DY, M, J, NY, IH)
```

The arguments XL, XR, DX, N, I, NX, and YB, YT, DY, M, J, NY have the same definitions (and restrictions) given for these terms in the discussion of GRID1V for the log mode, page 74.30.21. The remaining arguments have the definitions given for LINRV.

Figures 5-m and 5-n show examples of portions of grids created by LINRV and NØNLNV, respectively. Figure 5-o shows a practical application of LINRV.

EXAMPLES OF USE OF LINRV

1030-00
010 010



CALL FRAMEV
 CALL XSCALV (45.0, 95.0, 50, 544)
 CALL LINRV (1, 515, 535, 900, 45.0, 95.0, 2.5, 2, 4, 2, 8)

CALL YSCALV (15.0, 65.0, 20, 575)
 CALL LINRV (2, 520, 550, 1023, 15.0, 65.0, 2.5, 2, 4, 2, 12)

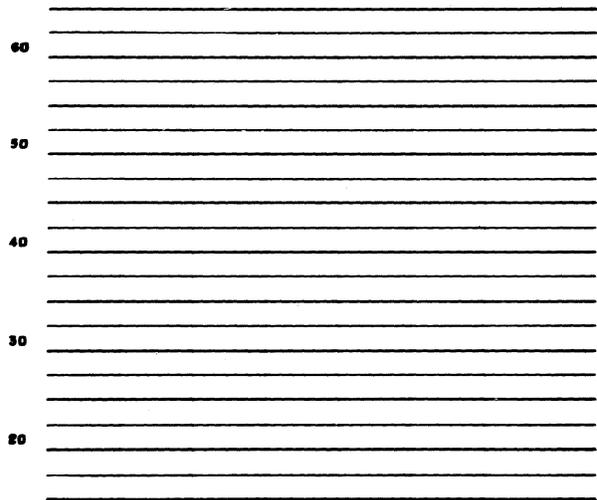
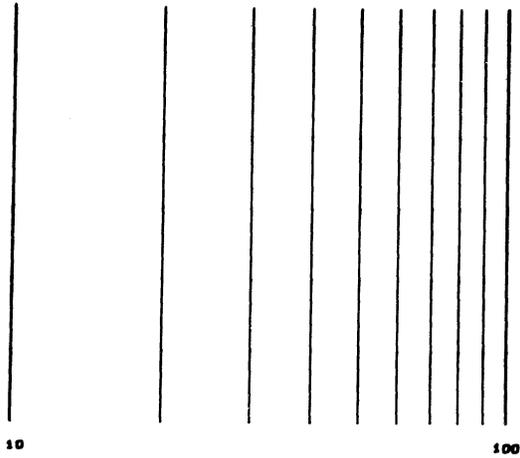


Figure 5-m

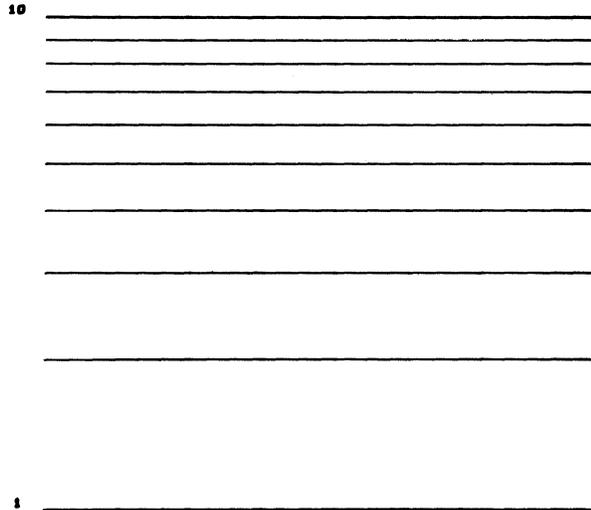
CALL FRAMEV
 CALL XSCALV (45.0, 95.0, 50, 544)
 CALL LINRV (1, 515, 535, 900, 45.0, 95.0, 2.5, 2, 4, 2, 8)
 CALL YSCALV (15.0, 65.0, 20, 575)
 CALL LINRV (2, 520, 550, 1023, 15.0, 65.0, 2.5, 2, 4, 2, 12)

EXAMPLES OF USE OF NONLNV

1230-00 L
010 010



```
CALL FRAMEV
CALL SMXYV ( 1, 0)
CALL XSCALV ( 10.0, 100.0, 50, 544 )
CALL NONLNV ( 1, 515, 535, 900, 10.0, 100.0, 1.0, 1, 1, 3, 8 )
```



```
CALL SMXYV ( 0, 1)
CALL YSCALV ( 1.0, 10.0, 20, 575 )
CALL NONLNV ( 2, 520, 550, 1023, 1.0, 10.0, 1.0, 1, 1, 2, 8 )
```

Figure 5-n

```
CALL FRAMEV
CALL SMXYV (1, 0)
CALL XSCALV (10.0, 100.0, 50, 544)
CALL NONLNV (1, 515, 535, 900, 10.0, 100.0, 1.0, 1, 1, 3, 8)
CALL SMXYV (0, 1)
CALL YSCALV (1.0, 10.0, 20, 575)
CALL NONLNV (2, 520, 550, 1023, 1.0, 10.0, 1.0, 1, 1, 2, 8)
```

SAMPLE CASE FOR AN AC LOAD ANALYSIS FOR AN ARBITRARY AIR/SPACE VEHICLE

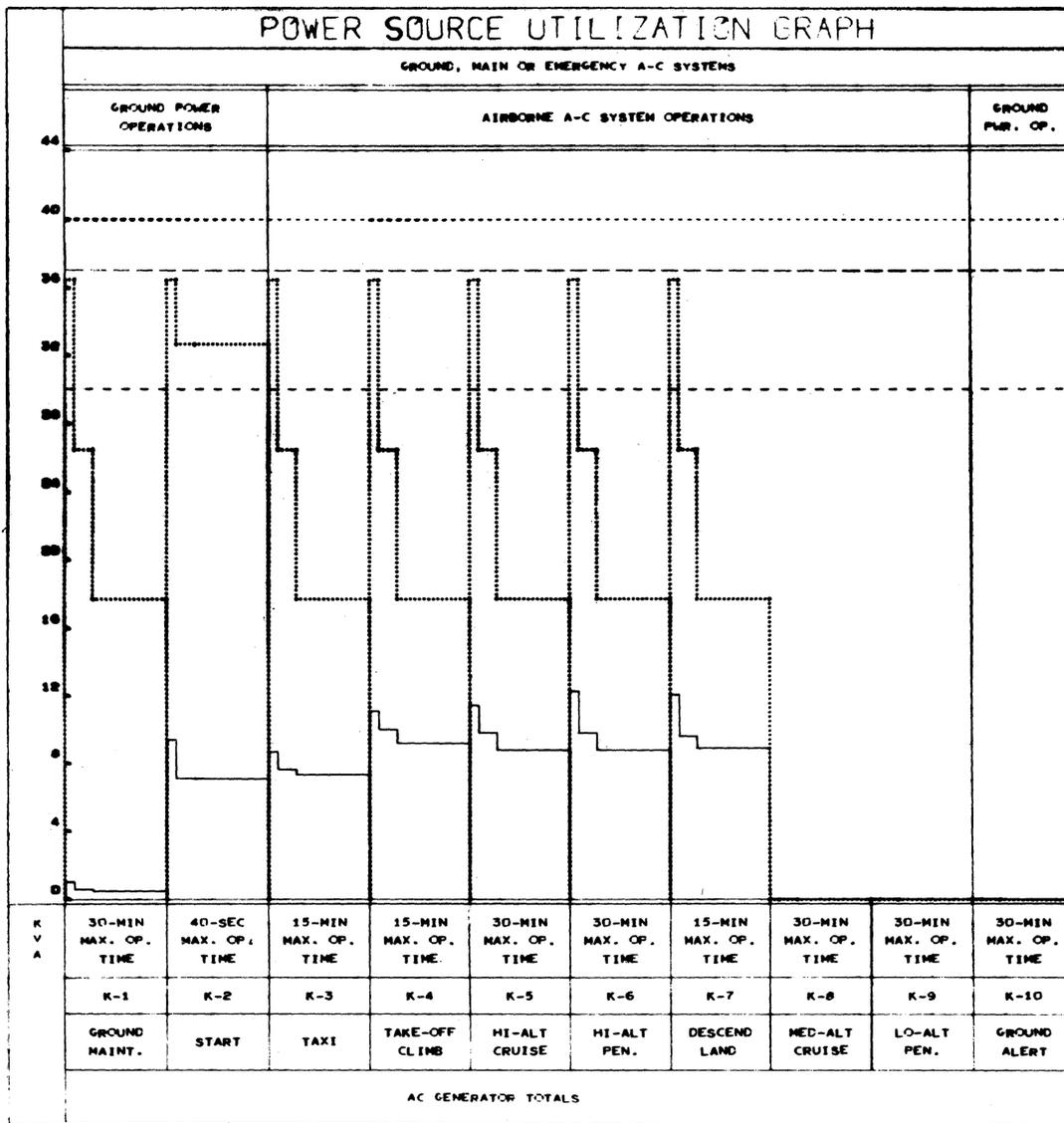


Figure 5-0

The programmer did not want a grid in the X direction, but took advantage of LINRV (and DXDYV) to produce and label reference points in the Y direction.



Section 74.6FURTHER ASPECTS OF PRINTING

This section supplements the information on printing, plotting, and labeling presented in the First Reader and the material in Section 74.2 on special printing.

The first part contains some considerations in the selection and placement of individual Charactron characters. Next, some of the characteristics of the typewriter mode are listed, and comments upon their influence on the output of certain printing and labeling subprograms are given. Additional subroutines for use in conjunction with the typewriter mode are presented. These include **STOPTV** and **RESETV**, which can turn off the typewriter mode, as well as special forms of **PRINTV**, **BRITEV**, and **FAINTV**, which provide for use of the "current point" feature of this mode.

Two modules are described that are useful building blocks for the creation of special printing and labeling routines. **HOLLV** makes it possible to pick out the Nth character code from an array of BCD characters or from a Hollerith argument. Given a floating point quantity, **BNBCDV** will obtain the six most significant figures (in BCD form) and the decimal scale of the quantity.

Details on vector character construction and use are given in the last part of this section. The material will enable the programmer to make up a pattern for a special character or to make up tables of patterns.

DETAILS OF CHARACTRON CHARACTERS

When a Charactron character is plotted or printed at a specified coordinate, it will lie in a rectangle that is centered on the specified coordinate. This rectangle is approximately 6 raster counts wide by 9 raster counts high (not counting space for separation of characters).

The "plotting dot" will be the Charactron character most frequently used to plot a point. Since it is centered within its character rectangle, it will be displayed exactly on the specified coordinate. Any of the Charactron characters may be used,

however, to mark a point. The best ones for this purpose are those which are symmetrical; for example, O, +, X, etc.

Experimentation has shown that the degree and period can also serve as plotting dots if allowances are made for the fact that they are off center. To center the degree, use IX + 3 and IY - 2 (or IY - 3). To center the period, use IY + 3 (or IY + 4) and IX (or IX - 1). These characters will be positioned with reasonable accuracy, although not so accurately as the plotting dot.

TYPEWRITER MODE CONSIDERATIONS

The typewriter mode, a "built-in" feature of the S-C 4020, has the following characteristics:

1. Character characters are displayed.
2. Once positioned and started in the typewriter mode, the S-C 4020 prints character after character, automatically spacing 8 raster counts to the right as each character is printed.
3. The commands for the typewriter mode (except for those that start and stop typing) are simply the 6-character BCD words to be typed, transmitted without alteration. This mode, therefore, is very economical of 7090-7094 time, and the length of the tape used for words in the typewriter mode is held to a minimum. Since the S-C 4020 can display 6 characters per command, this mode is also economical of S-C 4020 time.
4. When a line of typing reaches the right edge of the frame, an automatic carriage return takes place. Typing halts at this point and resumes at the left edge of the frame, 16 raster counts below the preceding line.
5. If typing reaches the bottom of the frame, a return to the top of the frame occurs automatically. No automatic film advance takes place, so it is possible for typing to overlay previously displayed information.
6. Three of the 64 Character characters serve special purposes when they are encountered in the words of the text being typed.

42 (Decimal)	A carriage return (described in item 4) will occur when this character is encountered.
10 (Decimal)	"Stop type." A return is made to the plotting mode. Any characters remaining in the word will be ignored. The next word on tape will be interpreted by the S-C 4020 in the standard (plotting) mode.
46 (Decimal)	"Reset" is executed. This means that typing is stopped, the plotting mode is restored, the film is advanced, and the bright intensity mode is turned on. Any characters

remaining in the word will be ignored. The next word on tape will be interpreted in the standard (plotting) mode.

Because 42, 10, and 46 are used for these special purposes, the characters ∂ , \cdot (plotting dot), and \sim are unavailable in the typewriter mode.

7. Typing may be started at either a specified point or at the "current point." The "current point" is defined as the point specified by the last operation performed; i.e., either the last point plotted or typed, or the origin of the last vector or axis drawn.

System Subprograms and the Typewriter Mode

The subprograms PRINTV and LABLV, and the SCØUTV system, make use of the typewriter mode. Therefore, the output of these subprograms will be influenced by the characteristics of this mode.

Since it was not practical to require the programmer to insert a "stop type" code, other methods of stopping typing are used. PRINTV and LABLV insert the "stop type" code after the specified N characters. The SCØUTV system inserts the code in accordance with the FØRMAT statement used.

It should be apparent that an incompatibility would occur if a "stop type" or "reset" code were encountered within the text being typed by PRINTV, LABLV, or SCØUTV. Any information following the special code would not be interpreted as typewriter output.

If PRINTV, LABLV, or SCØUTV printing encounters a "carriage return" character or the right edge of the frame, a carriage return will be made. This would interfere with the normal line advance provided in the SCØUTV system. It can be used to advantage, however, with PRINTV.

Except for the method used to stop typing, PRINTV can be considered a basic typewriter mode routine. It is a suitable lower-level module for other Character character typing routines; in fact, it is used in this way by LABLV.

"Stop Type" Routines: STØPTV, RESETV

It is remotely possible that some unusual use of the subroutines or of the S-C 4020 could cause the typewriter mode to be in effect when the standard

(plotting) mode is needed. Two subroutines are provided which ensure that the typewriter mode is no longer on.

CALL STOPTV

Turns off the typewriter mode. This has no effect on the mode if the typewriter mode is already off. However, since this statement writes a word on tape, it should be used with discretion.

GRID1V uses STOPTV to ensure that the plotting mode is in effect at the start of a new grid display.

CALL RESETV or CALL RESETV (NF)

Turns off the typewriter mode, advances the film and establishes the bright intensity mode. When this statement has either no argument or a zero argument, the job number and frame count are placed at the upper right corner of the frame. This display is inhibited if the argument $NF > 0$.

SETTING THE CURRENT POINT

The current point feature of the typewriter mode can be utilized if a special form of PRINTV is

employed. Typing may be started at the last point plotted or printed, or at the origin of the last vector or axis line drawn, by giving one of the following statements:

CALL PRINTV (N, BCDTXT)

CALL PRINTV (-N, nH-----)

Except for the use of the current point instead of specified coordinates, this alternate form of PRINTV functions in the usual manner. (See page 74.10.21.)

If the programmer wants to set a new current point, he may plot or print a blank at a specified position. Another method, which includes both setting the current point and establishing the "Brite" or "Faint" mode, is by one of the following statements:

CALL BRITEV (NX, NY)

CALL FAINTV (NX, NY)

NX, NY The raster coordinates which will be set as the current point.

Only a blank will be plotted at NX, NY by either of these statements.

SPECIAL PRINTING AND LABELING AIDS

HØLLV and BNBCDV were designed as lower-level modules for the system printing and labeling subprograms. However, they can be called directly, and can sometimes be useful building blocks for the programmer's character printing and labeling subprograms.

Isolating One Character from a BCD Array:HØLLV

HØLLV selects the Nth BCD character from BCD words that are either contained in a one-dimensional FORTRAN array or given as a Hollerith argument, and stores the numerical equivalent of the selected character in a specified location. PLØTV or VCHARV can then be used to display the character. The call statement is one of the following:

CALL HØLLV (N, BCDTXT, NS)

CALL HØLLV (-N, nH-----, NS)

±N The position of the desired character within the block of BCD characters, counting from 1. If the block of characters is a FORTRAN array, N should be positive. If the BCD characters are in a Hollerith argument, the sign of N should be negative.

BCDTXT A one-dimensional FORTRAN array that contains the BCD characters.

nH----- A Hollerith argument.

NS The name of a location in which the numerical equivalent of the selected character will be stored. For example, if the Nth BCD character is R, the stored result will be the same as if the statement "NS = 41" had been used.

A practical example of HØLLV usage is illustrated

by the way APRNTV uses it. The latter subprogram loops through a sequence of BCD characters, using HØLLV to select a character and PLØTV to print it. APRNTV then advances the print position and returns to the beginning of the loop for the next character.

Obtaining BCD Equivalent of a Floating Point Number: BNBCDV

BNBCDV is used by LABLV as a lower-level module to obtain the BCD equivalent of the floating point number displayed in the label. Since it converts a floating point number so as to yield the six most significant digits and the decimal scale, it can be useful also in programs that do not involve the S-C 4020. The call statement is:

CALL BNBCDV (D, BCDWD, NDS)

D	Floating point number to be converted into BCD.
BCDWD	Location in which the 6-character (rounded) BCD equivalent will be stored.
NDS	Location in which the decimal scale (number of digits to the left of the decimal point) will be stored.

The sign of D is ignored, and must be handled separately by the programmer.

Examples

D = 12.345678

The BCD equivalent of 123457 will be placed in BCDWD and the decimal scale of 2 will be stored in NDS.

D = -.01234567

The BCD equivalent of 123457 will be placed in BCDWD, and the decimal scale of -1 in NDS.

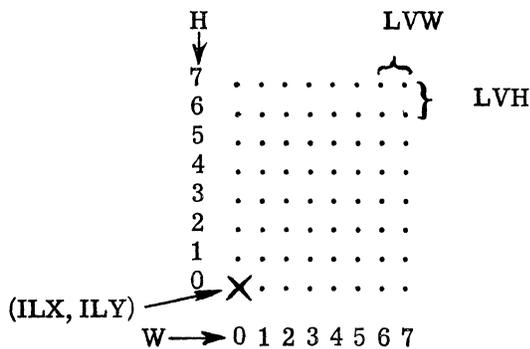


VECTOR CHARACTER CONSTRUCTION AND USE

In a preceding section, page 74. 20. 35, the tables of vector characters which are available in the system were presented. This section explains how vector characters are generated and how the programmer can create additional characters to be displayed by VCHARV and RITE2V.

Vector Character Formation

Given the proper specifications, VCHARV will generate any design that can be constructed by joining points on an 8 x 8 lattice with straight lines. A coordinate system in width (W) and height (H) is used to refer to positions on the lattice. Patterns are specified by giving coordinates of the positions on the lattice that are to be connected by vectors. As shown on the diagram, the W and H coordinates each can have values from 0 to 7.



ILX, ILY are the coordinates used by VCHARV to position the character. LVW, LVH represent the distances (expressed in raster counts) from one lattice position to an adjacent position.

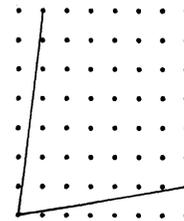
Character Size

The size of a character (or other design) depends upon the amount of space between adjacent positions on the lattice. The number of raster counts between positions may be specified by calling the subprogram CHSIZV (page 74. 20. 25). The arguments LVW and LVH indicate the desired number of raster counts between positions in width and height, respectively. If CHSIZV is never called, the

values $LVW = LVH = 3$ raster counts will be used.

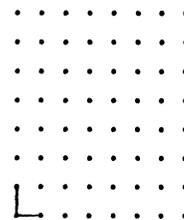
Although it is possible to assign a value as small as 1 to LVW and/or LVH, this is not practical for the majority of characters. The upper limit depends upon the particular character, and upon the restriction that no vectors in the pattern may exceed the resultant of 63 counts in the direction W, and 63 counts in the direction H. For example, if a particular pattern requires joining a position on the bottom row of the lattice with one on the top row, the maximum permissible value of LVH is 9, since 9 (raster counts) multiplied by 7 (lattice intervals) = 63 counts.

To draw these two vectors:



The maximum permissible LVW, LVH = 9 counts

To draw these two vectors:



The maximum permissible LVW, LVH = 63 counts

Character Orientation and Placement: VCHARV

VCHARV permits a selection of four possible orientations, illustrated on page 74. 60. 34. It rotates a character into the selected orientation by the way in which it applies the increments LVW and LVH.

The following table shows the direction of application of LVW and LVH for the four orientations.

Orientation Selection	Directions in Which Increments are Applied		
	K	LVW	LVH
90 (Upright)		Positive in X	Positive in Y
180		Positive in Y	Negative in X
270		Negative in X	Negative in Y
0 (or 360)		Negative in Y	Positive in X

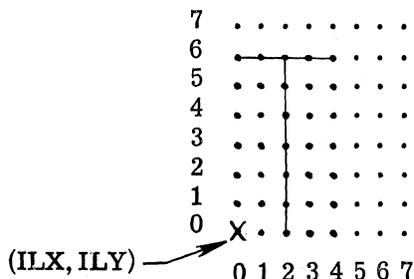
To prevent characters from overlapping the edge of the frame, the orientation and the specifications of the position coordinates (ILX, ILY) must be considered. The following table shows the restrictions upon ILX, ILY for each of the four orientations. In the table, N is one less than the number of lattice positions in width used by the character, and M is one less than the number, in height.

K	ILX	ILY
90 (Upright)	$0 \leq ILX \leq 1023 - N*LVW$	$0 \leq ILY \leq 1023 - M*LVH$
180	$M*LVH \leq ILX \leq 1023$	$0 \leq ILY \leq 1023 - N*LVW$
270	$N*LVW \leq ILX \leq 1023$	$M*LVH \leq ILY \leq 1023$
0 (or 360)	$0 \leq ILX \leq 1023 - M*LVH$	$N*LVW \leq ILY \leq 1023$

Pattern Specification

A pattern specifies the points on the lattice to be joined by vectors. The pattern consists of a sequence of lattice coordinates arranged in the order W1, W2, H1, H2, W3, W4, H3, H4, . . . , etc, where (W1, H1) and (W2, H2) are the coordinates of the end points of the first vector in the character, (W3, H3), and (W4, H4) are coordinates of the end points of the second vector, etc. The end of a pattern is signaled by the presence of at least 12 consecutive zero bits.

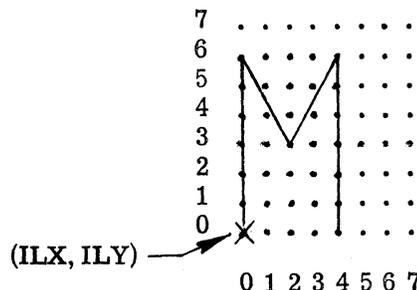
Since values of W and H will not be larger than 7, the values can be packed into 7090/94 words, allotting 3 bits per value. As many words as are needed to include all coordinates may be used. The character T appears on the lattice as follows:



The instructions for this pattern could be read in as the single octal word:

```
PATRN1
0 4 6 6 2 2 0 6 0 0 0 0
W1 W2 H1 H2 W3 W4 H3 H4 (End of pattern)
```

The pattern for M requires more than one word. It could be read in as an octal array, with a dimension of 2, in order to appear in storage in proper FORTRAN array order.



The instructions for this pattern, in octal, are:

PATRN2(1)

0 0 0 6 0 2 6 3 2 4 3 6

W1 W2 H1 H2 W3 W4 H3 H4 W5 W6 H5 H6

PATRN2(2)

4 4 0 6 0 0 0 0 0 0 0 0

W7 W8 H7 H8 (End of Pattern)

The preceding examples make use of only a 5 x 7 portion of the 8 x 8 lattice, as do all the characters in TABL1V, TABL2V, and TABL3V. (Exception: The β sign extends to the top row, using a 5 x 8 portion of the lattice.)

Furnishing a Single Pattern to VCHARV

A special form of VCHARV allows a single pattern to be furnished as a FORTRAN array. The following call statement may be used:

CALL VCHARV (K, NT, ILX, ILY, 0, A, 0)

- A A one-dimensional array in which the pattern must be stored. The fifth and seventh arguments must be zero. The remainder of the arguments are as described on page 74. 20. 31. "A" must not be named on an "F" card.

Table Structure

If the programmer has need for a number of special characters not in the system tables, he can build a table, in the FAP language, similar to TABL1V, TABL2V, and TABL3V. As with these three, he must use an "F" card carrying the table name.

Normally, the ECM does not carry information involving the FAP language, and there is no intention to teach it here. Programmers who are familiar with this language will find the following information helpful in constructing such tables. Those who are not acquainted with it should draw up the specifications for the characters they need, and contact their computing coordinator for assistance in the actual programming.

Since the number of words in each pattern varies, a list of entries is employed in the search for a specific character pattern. The location TABL_V+1 must contain, in its address, the location of the list of entries. Each table is compiled as if it were a FAP subprogram for FORTRAN use. Storage is FORTRAN-array type, i. e., descending storage for the table and for each individual pattern within it. Note that each individual character ends with 12 zeros. For the statement "VFD N/M, N/M," etc, N is the number of bits allotted to M, which in turn is the quantity to be used for W (or H). (The patterns illustrated are the numeric characters 0 through 5.)

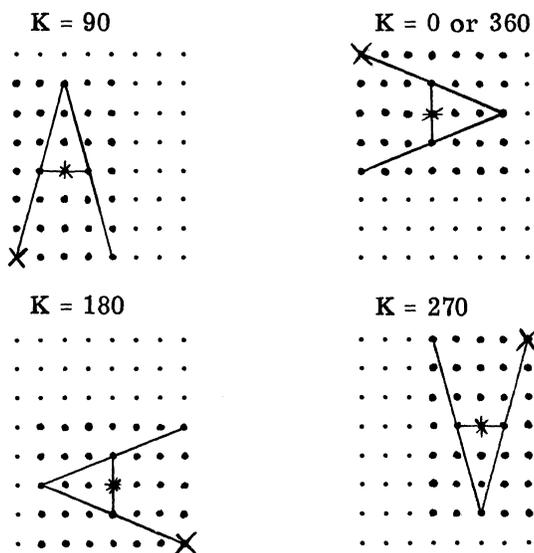
*	FAP COUNT ENTRY	25 TABL9V
	VFD	9/12, 9/14, 18/0
ENTR9V	VFD	9/0, 9/3, 9/5, 9/8
	VFD	3/0, 3/0, 3/4, 3/6, 3/0, 3/4, 3/6, 3/6, 12/0
	VFD	3/4, 3/4, 3/1, 3/3, 3/4, 3/3, 3/3, 3/4, 3/3, 3/0, 3/4, 3/4
N5	VFD	3/0, 3/1, 3/1, 3/0, 3/1, 3/3, 6/0, 3/3, 3/4, 3/0, 3/1
	VFD	3/0, 3/4, 3/2, 3/2, 24/0
N4	VFD	3/2, 3/4, 6/0, 3/3, 3/3, 3/0, 3/6, 3/3, 3/0, 3/6, 3/2
	VFD	3/3, 3/1, 3/6, 3/6, 3/1, 3/0, 3/6, 3/5, 12/0
	VFD	3/3, 3/4, 3/3, 3/4, 3/4, 3/4, 3/4, 3/5, 3/4, 3/3, 3/5, 3/6
	VFD	3/4, 3/4, 3/1, 3/2, 3/4, 3/3, 3/2, 3/3, 3/3, 3/2, 3/3, 3/3
N3	VFD	3/0, 3/1, 3/1, 3/0, 3/1, 3/3, 6/0, 3/3, 3/4, 3/0, 3/1
	PZE	0
	VFD	3/4, 3/4, 3/5, 3/4, 3/4, 3/0, 3/4, 3/0, 3/0, 3/4, 6/0
N2	VFD	3/0, 3/1, 3/5, 3/6, 3/1, 3/3, 3/6, 3/6, 3/3, 3/4, 3/6, 3/5
	PZE	0
N1	VFD	3/1, 3/3, 6/0, 3/2, 3/2, 3/0, 3/6, 3/1, 3/2, 3/5, 3/6
	VFD	3/3, 3/2, 3/0, 3/0, 3/2, 3/1, 3/0, 3/1, 12/0
	VFD	3/3, 3/4, 3/6, 3/5, 3/4, 3/4, 3/5, 3/1, 3/4, 3/3, 3/1, 3/0
TABL9V	VFD	3/1, 3/1, 3/1, 3/5, 3/1, 3/2, 3/5, 3/6, 3/2, 3/3, 3/6, 3/6
	PZE	ENTR9V
	END	

WRITING MORE THAN ONE VECTOR CHARACTER: RITE2V

If only one character is to be displayed, VCHARV is suitable, but for several characters, RITE2V can be employed more easily. Since RITE2V uses VCHARV to generate characters, much of the preceding discussion applies. However, unlike VCHARV, which uses the lower left corner of the lattice as its position reference point, RITE2V uses the center of the 5 x 7 portion of the 8 x 8 lattice.

Character Orientation and Placement: RITE2V

The following diagrams show the relation between the character lattice as it is used in RITE2V and in VCHARV. The 5 x 7 lattice is shown superimposed upon the maximum 8 x 8 lattice. The asterisk indicates the reference point (IX, IY) as used by RITE2V, while the X indicates the reference point (ILX, ILY) as used by VCHARV.



Since the center is used as the character reference point by RITE2V, and since a 5 x 7 lattice is normally used, the restrictions upon the position coordinates differ slightly from those shown in the table for VCHARV on page 74. 60. 32. The restrictions are:

(One more LVH increment must be considered for the § sign.)

The character space size and row size to be used by RITE2V depend upon the character size and the space required to prevent overlapping of characters. Since the characters in the tables normally used by RITE2V are based on a 5 x 7 lattice, the character size is, usually, 4*LVW by 6*LVH. (For the § sign, it is 4*LVW by 7*LVH.) Also, since LVW = LVH = 3, the character size is 12 by 18, unless changed by CHSIZV. The standard quantities ISPACE and IRØW used by RITE2V are 18 and 24, respectively, allowing sufficient space to prevent overlapping of characters.

On page 74. 20. 25, the description of RITSTV shows how the programmer may redefine ISPACE, IRØW, and also may name a table other than TABL1V (which is normally used). The table named may be one of the alternate ones provided by the system (TABL2V, TABL3V), or one of the programmer's own design.

Furnishing a Single Pattern to RITE2V

An alternate form of RITSTV allows the programmer to furnish a single character pattern in an array. The call statement is:

```
CALL RITSTV (ISPACE, IRØW, A, 0)
```

- A A FORTRAN array in which the programmer has stored one character pattern. "A" must not be named on an "F" card.

The last argument must be zero, to indicate to the subprogram that A is not a table. ISPACE and IRØW are as defined on page 74. 20. 25.

As shown on page 74. 60. 33, VCHARV can also be employed to generate a single character from a pattern supplied as an array. Although VCHARV is more economical of machine time, the programmer may find it less awkward to use RITSTV to insert a character into printed information being written by RITE2V.

K	IX	IY
90	$2*LVW \leq IX \leq 1023 - 2*LVW$	$3*LVH \leq IY \leq 1023 - 3*LVH$
180	$3*LVH \leq IX \leq 1023 - 3*LVH$	$2*LVW \leq IY \leq 1023 - 2*LVW$
270	$2*LVW \leq IX \leq 1023 - 2*LVW$	$3*LVH \leq IY \leq 1023 - 3*LVH$
0 (or 360)	$3*LVH \leq IX \leq 1023 - 3*LVH$	$2*LVW \leq IY \leq 1023 - 2*LVW$

TRANSMISSION OF COMMANDS

The (PLOT) subroutine and an entry point within it, (CLEAN, are the only entries provided by the system that transmit commands to the output tape. Since these entries can only be utilized by subprograms in the FAP language, they are normally used directly only by systems programmers. However, some of the characteristics itemized below may be of interest to others.

1. An S-C 4020 command can be transported to the S-C 4020 output tape by (logically) putting the command in the accumulator and using the FAP instruction:

TSX (PLOT), 4

When (PLOT) is entered, the word in the accumulator must be in the exact format recognized as a command by the S-C 4020.

2. If a command must be followed by an end-of-record gap, the subprogram must use (CLEAN to write it. This entry is called by the instruction:

TSX (CLEAN, 4

3. (PLOT) includes (and supervises) two output tape buffers. Each buffer accommodates 167 words. (This buffer size allows the tape records to be within the length that can be accepted by the NAA microwave equipment.) If (CLEAN is called, it will write the current (useful) contents of the

buffer on tape and write an end-of-record gap; the buffer usage will be alternated.

4. The NAA convention is to treat the lower left corner of the raster as the (0,0) coordinate, and the upper right corner as the (1023,1023) coordinate. The machine commands must use the raster coordinate system built into the S-C 4020, with the (0,0) point in the upper left corner and the (1023,1023) point in the lower right corner. In the system subprograms at NAA, one of the final steps taken to prepare an S-C 4020 command for transmission is to subtract the Y raster coordinate from 1023, thus converting the NAA raster orientation to S-C 4020 specifications.
5. When the first S-C 4020 command in a job is sent to (PLOT) for transmission, it will be held up automatically while (PLOT) transmits the ID frame. Currently, (PLOT) selects both cameras prior to the ID frame commands, and selects the 9-inch camera only following the ID frame transmittal.
6. During writing of the ID frame, (PLOT) communicates with several cells in the NAA monitor area in order to obtain identification information derived from the Account card. These locations are (TES), (CTJB), (CTES), (JØBN), and (IDCD).

After the ID frames have been written, the area containing the instructions for writing them is reassigned as part of the buffer area.



S-C 4020 Call Statement Summary

THE FOLLOWING LIST OF S-C 4020 SYSTEM SUBPROGRAMS IS INTENDED AS A QUICK REFERENCE. THE SUBPROGRAMS ARE GROUPED ACCORDING TO THEIR FUNCTIONS. THE CALL STATEMENTS ARE SHOWN BUT NO DETAILS ARE GIVEN. ARGUMENTS THAT REPRESENT OUTPUT FROM THE SUBPROGRAM ARE UNDERLINED.

CLASS	SUB-CLASS	CALL AND FUNCTION STATEMENTS	PAGES
CAMERA SELECT		CALL CAMRAV (NCAM)	74.10.01
FRAME ADVANCE	ADVANCE	CALL FRAMEV OR CALL FRAMEV (NF)	74.10.03
	RESET	CALL RESETV OR CALL RESETV (NF)	74.60.03
COUNT FRAMES		CALL NOFRV (<u>N9</u> , <u>N35</u>)	74.10.04
GRIDIV CONTROL	MARGIN SET	CALL SETMIV (MTL , MTR , MTB , MTT)	74.30.01
	MARGIN RETRIEVE	CALL SETMOV (<u>MTLL</u> , <u>MTRL</u> , <u>MTBL</u> , <u>MTTL</u>)	74.30.01
	LABEL CHARACTER SIZE SET	CALL SETCIV (IW, IH)	74.30.09
	SIZE RETRIEVE	CALL SETCOV (<u>IWL</u> , <u>IHL</u>)	74.30.09
	SET HOLD	CALL HOLDIV (NH)	74.30.09
	RETRIEVE HOLD	CALL HOLDOV (<u>NHL</u>)	74.30.09
			ALSO SEE SCALE MODE SELECTION
SCALE MODE	SET	CALL SMXYV (MX , MY)	74.30.21, 74.40.07
	RETRIEVE	CALL MSXYV (<u>MXL</u> , <u>MYL</u>)	74.30.21, 74.40.07
FIND ARGUMENTS FOR GRIDIV	X-DIRECTION	CALL DXDYV(1, XL, XR, <u>DX</u> , <u>N</u> , <u>I</u> , <u>NX</u> , DC, <u>IERR</u>)	74.10.13, 74.50.09
	Y-DIRECTION	CALL DXDYV(2, YB, YT, <u>DY</u> , <u>M</u> , <u>J</u> , <u>NY</u> , DC, <u>IERR</u>)	74.10.13, 74.50.09
GRID. CONSTRUCT	BASIC	CALL GRIDIV (L, XL, XR, YB, YT, DX, DY, <u>±N</u> , <u>±M</u> , <u>±I</u> , <u>±J</u> , <u>±NX</u> , <u>±NY</u>) SEE FOOTNOTE	74.10.04, 74.30.21, 74.50.01
	VERTICAL LINEAR	CALL LINRV (1, LYREFR, IYMIN, IYMAX, XL, XR, DX, N, I, <u>±NX</u> , IW)	74.50.21, 74.30.27
	HORIZ. LINEAR	CALL LINRV (2, LXREFR, IXMIN, IXMAX, YB, YT, DY, M, J, <u>±NY</u> , IH)	74.50.21, 74.30.27

	LOG VERTICAL	CALL NONLNV (1, LYREFR, IYMIN, IYMAX, XL, XR, DX, N, I, \pm NX, IW) SEE FOOTNOTE	74.50.21, 74.30.27
	LOG HORIZONTAL	CALL NONLNV (2, LXREFR, IXMIN, IXMAX, YB, YT, DY, M, J, \pm NY, IH) SEE FOOTNOTE	74.50.21, 74.30.27
SCALE FACTORS	COMPUTE FOR X	CALL XSCALV (XL, XR, ML, MR)	74.40.01
	COMPUTE FOR Y	CALL YSCALV (YB, YT, MB, MT)	74.40.01
	SAVE FACTORS	CALL SCLSAV (R)	74.40.05
	RESET FACTORS	CALL RESCLV (R)	74.40.05
CONVERSION WITH/WITHOUT ERROR INDICATE	WITH INDICATION	NXV(X)	74.40.03
	WITH INDICATION	NYV(Y)	74.40.03
	WITHOUT	IXV(X)	74.40.03
	WITHOUT	IYV(Y)	74.40.03
		ALSO SEE POINTV , APLOTV	
INVERSE OF CONVERSION		UXV(IX)	74.40.03
		UYV(IY)	74.40.03
		CAN ONLY BE USED IN LINEAR MODE	
TRANSFORMATION		XMODV(X)	74.40.07
		YMODV(Y)	74.40.07
OFF-SCALE ERROR INDICATORS	INDICATOR CELLS	CALL SCERRV (<u>KX</u> , <u>KY</u>)	74.40.09
	SAVE CELL LOC.	CALL SERSAV (<u>KSX</u> , <u>KSY</u>)	74.40.10
	RESET LOCATIONS	CALL SERREV (KSX, KSY)	74.40.10
INTENSITY	BRIGHT	CALL BRITEV OR CALL BRITEV (IX, IY) LATTER ALSO SETS CURRENT POINT	74.10.13, 74.60.04
	FAINT	CALL FAINTV OR CALL FAINTV (IX, IY) LATTER ALSO SETS CURRENT POINT	74.10.13, 74.60.04
POINT PLOTTING	SINGLE POINT	CALL POINTV (X, Y, \pm NS)	74.10.19
	SPECIAL FORM	CALL POINTV (IX, IY, \pm NS, ANY)	74.10.19
	ARRAYS OF DATA	CALL APLOTV (\pm NK, XARRAY, YARRAY, JX, JY, \pm NC, MARKPT, IERR)	74.10.15
	LOWEST FORM	CALL PLOTV (IX, IY, NS)	74.10.19

LINES	VECTOR LINES	CALL LINEV (IX1, IY1, IX2, IY2)	74.10.25
	AXIS VERTICAL	CALL YAXISV (IX, IY)	74.10.25
	AXIS HORIZONTAL	CALL XAXISV (IX, IY)	74.10.25
LABELS	CHARACTRON CHAR	CALL LABLV (D, IX, IY, ±NCHAR, NT, NDMAX)	74.10.23
		ALSO SEE BNBCDV AND HOLLV	74.60.11
PRINTING AND RELATED SUBRS.	ONE CHARACTRON	CALL PLOTV (IX, IY, NS)	74.10.19
	MULTIPLE	CALL PRINTV (±N, BCDTXT, IX, IY)	74.10.21
	SPECIAL SPACING	CALL APRNTV (INCRX, INCRY, ±N, BCDTXT, IX, IY)	74.10.21, 74.20.41, 74.20.36
	AS PRINTER	CALL SCOUTV	74.20.02
	SPECIAL FORM	CALL SCOUTV (LINE1)	74.20.07
	SPECIAL FORM	CALL SCOUTV (LINE1, INC)	74.20.07
	SET SCOUTV LOC.	CALL LOCSTV (IX, IY, IB)	74.20.11
	SAVE LOCATIONS	CALL LOCSAV (<u>ISX</u> , <u>ISY</u>)	74.20.15
	WRITE IN VECTOR CHARACTERS	CALL RITE2V (IX, IY, LIMIT, K, NT, NTOTAL, NTH, BCDTXT, NLAST)	74.20.21, 74.60.34
	SET UP RITE2V	CALL RITSTV (ISPACE, IROW, TABL V)	74.20.25, 74.60.34
	SPECIAL FORM	CALL RITSTV (ISPACE, IROW, A, 0)	74.60.34
	RETRIEVE LOC.	CALL RITXYV (<u>LX</u> , <u>LY</u>)	74.20.25
	ONE VECTOR CHAR	CALL VCHARV (K, NT, ILX, ILY, NS, TABL V)	74.20.31, 74.60.31
	SPECIAL FORM	CALL VCHARV (K, NT, ILX, ILY, 0, A, 0)	74.60.31
	FORM OVERLAY	CALL FORMV	74.20.51
ERROR MARK		CALL ERMKV	74.50.02

OTHER S-C4020 SUBR. NAMES THAT MAY APPEAR ON THE LOAD MAP ARE

ERRNLV (PAGE 74.50.02), ERRNLV (PAGE 74.50.02)
 (SCRTV, (CTAPE, (STH), (CNTV), (P..1), (P..2), (P..3) (P..4), (PLOT), (CLEAN,
 (4020), (BUMP, (YTOP), (YREG), (XXXX), (YYYY), (SMXV), (SMYV), STOPV

AND SOME THAT ARE NOT S-C 4020 SUBRS. , BUT ARE CALLED BY THEM

(IOS), (SDL), (WRS), (WTC), (RCH), (TES), (WER), (CTJB),
 (CTES), (JOBN), (IDCD), (TCO), (IOH), (LOUT), (WDCT)

** FOOTNOTE: IN LOG MODE CERTAIN RE-
 STRICTIONS APPLY TO DX, DY, N, M, I, J



NORTH AMERICAN AVIATION, INC.
ENGINEER'S COMPUTING MANUAL

ALPHABETICAL INDEX, S-C 4020 TABLES AND SUBPROGRAMS

(WITH CALLING SEQUENCES)

NAME	PAGE NO.
APLOTV	74.10.15
CALL APLOTV (N,XARRAY,YARRAY,JX,JY,NC,MRKPT,IERR)	
APRNTV	74.10.21
CALL APRNTV (INCRX,INCRY,N,BCDXT,IX,IY)	
CALL APRNTV (INCRX,INCRY,-N,NH-----,IX,IY)	
PRINTING ON SLOPES	74.20.41
EXAMPLE OF USE FOR SPECIAL SPACING	74.20.36
BNBCDV	74.60.11
CALL BNBCDV (D,BCDWD,NDS)	
BRITEV	74.10.13
CALL BRITEV	74.10.13
CALL BRITEV (NX,NY)	74.60.04
USE TO SET CURRENT POINT	74.60.04
CAMRAV	74.10.01
CALL CAMRAV (N)	
CHARACTRON CHARACTERS, TABLE OF	74.10.14
CHSIZV	74.20.25
CALL CHSIZV (LVW,LVH)	
ADDITIONAL DETAILS	74.60.31
(CLEAN	74.90.11
DXDYV	74.10.13
CALL DXDYV (1,XL,XR,DX,N,I,NX,DC,IERR)	
CALL DXDYV (2,YB,YT,DY,M,J,NY,DC,IERR)	
DENSITY FACTOR, ILLUSTRATION OF	74.50.09
NOT USABLE IN LOG MODE	74.30.21
ERMRKV	74.50.02
CALL ERMKV	
ERRLNV	74.50.02
ERRNLV	74.50.02
FAINTV	74.10.13
CALL FAINTV	74.10.13
CALL FAINTV (IX,IY)	74.60.04
USE TO SET CURRENT POINT	74.60.04

FORMV	74.20.51
CALL FORMV	
FRAMEV	74.10.03
CALL FRAMEV	
CALL FRAMEV (N)	
EFFECT ON SCOUTV	74.20.02
GRIDIV	74.10.04
CALL GRIDIV (L,XL,XR,YB,YT,DX,DY,N,M,I,J,NX,NY)	
LOGARITHMIC MODE	74.30.21
OPERATIONAL DETAILS OF	74.50.01
SEE ALSO SETMIV,SETCIV,HOLDIV,SMXYV	
HOLDIV	74.30.09
CALL HOLDIV (NH)	
HOLDOV	74.30.09
CALL HOLDOV (NHL)	
HOLLV	74.60.11
CALL HOLLV (N,BCDXT,NS)	
CALL HOLLV (-N,NH-----,NS)	
IXV	74.40.03
IX = IXV(X)	
IYV	74.40.03
IY = IYV(Y)	
LABLV	74.10.23
CALL LABLV (D,IX,IY,NCHAR,NT,NDMAX)	
USE OF TYPEWRITER MODE	74.60.03
LINEV	74.10.25
CALL LINEV (IX1,IY1,IX2,IY2)	
LINRV	74.50.21
CALL LINRV (1,LYREFR,IYMIN,IYMAX,XL,XR,DX,N,I,NX,IW)	
CALL LINRV (2,LXREFR,IXMIN,IXMAX,YB,YT,DY,M,J,NY,IH)	
PRELIMINARY INFORMATION	74.30.27
LOCSAV	74.20.15
CALL LOCSAV (ISX,ISY)	
LOCSTV	74.20.11
CALL LOCSTV (IX,IY,IB)	
ADDITIONAL ILLUSTRATION	74.30.06

NORTH AMERICAN AVIATION, INC.
ENGINEER'S COMPUTING MANUAL

MSXYV	74.30.21
CALL MSXYV (MXL,MYL)	
ADDITIONAL DETAILS	74.40.07
NOFRV	74.10.04
CALL NOFRV (N9,N35)	
NONLNV	74.50.21
CALL NONLNV (1,LYREFR,IYMIN,IYMAX,XL,XR,DX,N,I,NX,IW)	
CALL NONLNV (2,LXREFR,IXMIN,IXMAX,YB,YT,DY,M,J,NY,IH)	
PRELIMINARY INFORMATION	74.30.27
	74.10.25
NXV	74.40.03
IX = NXV(X)	
NYV	74.40.03
	74.10.25
IY = NYV(Y)	
PRELIMINARY INFORMATION	74.10.25
(PLOT)	74.90.11
PLOTV	74.10.19
CALL PLOTV (IX,IY,NS)	
POINTV	74.10.19
CALL POINTV (X,Y,NS)	
CALL POINTV (IX,IY,NS,ANY)	
PRINTV	74.10.21
CALL PRINTV (N,BCDXTX,IX,IY)	
CALL PRINTV (-N,NH-----,IX,IY)	
CALL PRINTV (N,BCDXTX)	74.60.04
CALL PRINTV (N,NH-----)	74.60.04
USE OF TYPEWRITER MODE	74.60.03
UTILIZATION OF CURRENT POINT	74.60.04
RESCLV	74.40.05
CALL RESCLV (R)	
RESETV	74.60.03
CALL RESETV	
CALL RESETV (NF)	
RITE2V	74.20.21
CALL RITE2V (IX,IY,LIMIT,K,INT,NTOTAL,NTH,BCDXTX,NLAST)	
CALL RITE2V (IX,IY,LIMIT,K,INT,NTOTAL,-NTH,NH-----,NLAST)	
ADDITIONAL DETAILS, RESTRICTIONS	74.60.34
EFFECT OF LIMIT	74.20.23

RITSTV	74.20.25
CALL RITSTV (ISPACE,IROW,TABL-V)	74.20.25
CALL RITSTV (ISPACE,IROW,A,0)	74.60.34
FURNISHING SINGLE PATTERN	74.60.34
RITXYV	74.20.25
CALL RITXYV (LX,LY)	
SCERRV	74.40.09
CALL SCERRV (KX,KY)	
SCLSAV	74.40.05
CALL SCLSAV (R)	
SCOUTV	74.20.02
CALL SCOUTV	74.20.02
CALL SCOUTV (LINE1)	74.20.07
CALL SCOUTV (LINE1,INC)	74.20.07
GRAPH AND PRINTED INFORMATION ON ONE FRAME	74.30.06
USE OF TYPEWRITER MODE	74.60.03
SERREV	74.40.10
CALL SERREV (LOCX,LOCY)	
SERSAV	74.40.10
CALL SERSAV (LOCX,LOCY)	
SETCIV	74.30.09
CALL SETCIV (IW,IH)	
SETCOV	74.30.09
CALL SETCOV (IWL,IHL)	
SETMIV	74.30.01
CALL SETMIV (MTL,MTR,MTB,MTT)	
SETMOV	74.30.01
CALL SETMOV (MTLL,MTRL,MTBL,MTTL)	
SMXYV	74.30.21
CALL SMXYV (MX,MY)	
ADDITIONAL DETAILS	74.40.07
STOPTV	74.60.03
CALL STOPTV	
TABL-V	74.20.35
UXV	74.40.03
X = UXV(IX)	

UYV	74.40.03
Y = UYV(IY)	
VCHARV	74.20.31
CALL VCHARV (K,NT,ILX,ILY,NS,TABL-V)	74.20.31
CALL VCHARV (K,NT,ILX,ILY,0,A,0)	74.60.33
ADDITIONAL DETAILS	74.60.31
VECTOR CHARACTERS	74.60.31
SYSTEM TABLES OF	74.20.35
SEE ALSO RITE2V,VCHARV,CHS12V,RITSTV	
XAXISV	74.10.25
CALL XAXISV (IX,IY)	
XMODV	74.40.07
XPRIME = XMODV(X)	
XSCALV	74.40.01
CALL XSCALV (XL,XR,ML,MR)	
(XXXX)	74.40.07
YAXISV	74.10.25
CALL YAXISV (IX,IY)	
YMODV	74.40.07
YPRIME = YMODV(Y)	
YSCALV	74.40.01
CALL YSCALV (YB,YT,MB,MT)	
(YYYY)	74.40.07



ENGINEER'S COMPUTING MANUAL

EDPM AUXILIARY OPERATIONS REQUEST

EDPM AUXILIARY OPERATIONS REQUEST					
JOB NUMBER _____				GROUP _____	DATE SUBMITTED _____
PRIORITY _____	NAME _____	DEPT. _____	GROUP _____	DATE SUBMITTED _____	
TYPE _____				PHONE _____	TIME SUBMITTED _____
SIGNATURE _____	FACILITY _____	BLDG. _____	FLOOR _____	PHONE _____	TIME SUBMITTED _____

GROUP _____	DATE SUBMITTED _____
PHONE _____	TIME SUBMITTED _____
TIME _____	BY _____

BOX	DATE	TIME	BY	DEFERRED
Received				DATE _____ TIME _____ BY _____
Keypunched				REASON _____
Verified				
Complete				

KEYPUNCH _____ Sheets

SEE ADDITIONAL INSTRUCTIONS ON REVERSE SIDE

SEE ADDITIONAL INSTRUCTIONS ATTACHED

REMARKS: _____

DECKS TO BE USED: Keypunch _____ From _____
 (Show Deck Number) Enclosed _____ files _____

**INDICATE THE ORDER IN WHICH WORK IS TO BE DONE BY NUMBERING THE SQUARES
 INDICATE CHOICE OF SPECIFICATIONS BY PUTTING X'S IN THE CIRCLES**

STANDARD OPERATIONS	ADDITIONAL SPECIFICATIONS <small>(When non-standard)</small>						
<input type="checkbox"/> REPRODUCE _____ Times, Deck No. _____	Leave Col. _____ blank						
<input type="checkbox"/> PUNCH Identification, _____, in reproduced deck. (Cols. 73-77 Standard)	Columns _____ through _____						
<input type="checkbox"/> PUNCH Sequence Number Starting with Increment	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">73</td> <td style="width: 50%; text-align: center;">80</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> </tr> <tr> <td style="text-align: center;">73</td> <td style="text-align: center;">80</td> </tr> </table>	73	80			73	80
73	80						
73	80						
<input type="checkbox"/> INTERPRET Deck(s) No. _____ <small>(Cols. 1-52 and 73-80 Standard)</small>							
<input type="checkbox"/> SORT Separately Deck No. _____ Deck No. _____ <small>(Numeric sort on Cols. 73-80 Standard)</small>	Columns _____ through _____ <input type="radio"/> Sort all cards (together) <input type="radio"/> Sort alphanumerically						
<input type="checkbox"/> MERGE Deck No. _____ into Deck No. _____ <small>SECONDARY PRIMARY</small> <small>(Standard: numeric merge, columns 73-80, duplicates removed from primary deck)</small>	Columns _____ through _____ <input type="radio"/> Duplicates allowed <input type="radio"/> Flag Duplicates						
<input type="checkbox"/> SEQUENCE CHECK Deck No. _____ <small>(Standard: numeric sequence check, columns 73-80, duplicates flagged)</small>	Columns _____ through _____ <input type="radio"/> Duplicates allowed						
<input type="checkbox"/> LIST _____ <small>DECK NO.</small>	<input type="radio"/> Single Space <input type="radio"/> Double Space						
<input type="checkbox"/> ENTER into production status, Deck _____							

NORTH AMERICAN AVIATION, INC. ENGINEER'S COMPUTING MANUAL

BOX	DATE	TIME	BY	
Received				DEFERRED _____ DATE _____ TIME _____ BY _____
Keypunched				REASON _____
Verified				_____
Complete				

The above portion of the form is reserved for the use of the EDPM group to record the progress of the job and to explain any reason which may cause deferment.

<input type="checkbox"/>	KEYPUNCH _____ Sheets
<input type="checkbox"/>	SEE ADDITIONAL INSTRUCTIONS ON REVERSE SIDE
<input type="checkbox"/>	SEE ADDITIONAL INSTRUCTIONS ATTACHED
<input type="checkbox"/>	REMARKS: _____

DECKS TO BE USED: Keypunch _____ (Show Deck Number) Enclosed _____		From { _____ files } _____
---	--	-------------------------------

Item	Meaning	When Must It Be Filled In?	Reference
Keypunch _____ Sheets	Self-explanatory.	Whenever keypunching required.	
See Additional Instructions On Reverse Side	Self-explanatory.	As needed.	
See Additional Instructions Attached	When a special form or page must be attached because this form will not cover the situation.	As needed. Attach the additional instruction sheet with a staple.	
Remarks: _____	Self-explanatory.	As needed for brief additional information.	
Decks To Be Used:	Self-explanatory.	Always, unless the request is for keypunching <u>only</u> .	

The major portion of the form contains instructions for auxiliary operations other than keypunching. The order in which the work is to be done is indicated by numbering the squares.

Certain operations are considered standard; they appear on the left side of the sheet, following the square boxes. To indicate standard operations, number the boxes and fill in the blanks on the left side of the sheet.

If nonstandard operations are desired, the boxes are numbered, but additional specifications are given on the right side of the form by filling in the blanks and/or putting "X's" in the circles. It is also permissible, on the nonstandard side, to include additional information or a note to "See REMARKS Above." (This is not to imply that lengthy notes in microscopic handwriting will be accepted - if the job is that complicated, the instructions should be put on the back of the form or on an attached sheet.)

TYPICAL JOB REQUEST

FOR FIRST WORK DONE ON A PROGRAM

EDPM AUXILIARY OPERATIONS REQUEST						
<u>3471-01</u> <small>JOB NUMBER</small>	<u>W. HAIG</u> <small>NAME</small>		<u>291</u> <small>DEPT.</small>	<u>072</u> <small>GROUP</small>	<u>3-1-65</u> <small>DATE SUBMITTED</small>	
PRIORITY	<u>L. A.</u> <small>FACILITY</small>	<u>2-15</u> <small>BLDG.</small>	<u>2</u> <small>FLOOR</small>	<u>8432</u> <small>PHONE</small>	<u>3:00 PM</u> <small>TIME SUBMITTED</small>	
TYPE						
SIGNATURE						
BOX	DATE	TIME	BY	DEFERRED _____ <small>DATE TIME BY</small>		
Received				REASON _____		
Keypunched				_____		
Verified				_____		
Complete				_____		
<input checked="" type="checkbox"/> KEYPUNCH <u>12</u> Sheets <input type="checkbox"/> SEE ADDITIONAL INSTRUCTIONS ON REVERSE SIDE <input type="checkbox"/> SEE ADDITIONAL INSTRUCTIONS ATTACHED <input type="checkbox"/> REMARKS: _____						
DECKS TO BE USED: Keypunch <u>5T 795</u> <small>(Show Deck Number) Enclosed</small>				From } _____ files } _____		
INDICATE THE ORDER IN WHICH WORK IS TO BE DONE BY NUMBERING THE SQUARES INDICATE CHOICE OF SPECIFICATIONS BY PUTTING X'S IN THE CIRCLES						
STANDARD OPERATIONS				ADDITIONAL SPECIFICATIONS <small>(When non-standard)</small>		
<input type="checkbox"/> REPRODUCE _____ Times, Deck No. _____				Leave Col. _____ blank		
<input type="checkbox"/> PUNCH Identification, _____, in reproduced deck. (Cols. 73-77 Standard)				Columns _____ through _____		
<input type="checkbox"/> PUNCH Sequence Number Starting with _____ Increment _____ <small>73 80 73 80</small>						
<input checked="" type="checkbox"/> 2 INTERPRET Deck(s) No. <u>5T 795</u> <small>(Cols. 1-52 and 73-80 Standard)</small>						
<input checked="" type="checkbox"/> 3 SORT Separately Deck No. <u>5T 795</u> Deck No. _____ <small>(Numeric sort on Cols. 73-80 Standard)</small>				Columns _____ through _____ <input type="radio"/> Sort all cards (together) <input type="radio"/> Sort alphanumerically		
<input type="checkbox"/> MERGE Deck No. _____ into Deck No. _____ <small>SECONDARY PRIMARY</small> <small>(Standard: numeric merge, columns 73-80, duplicates removed from primary deck)</small>				Columns _____ through _____ <input type="radio"/> Duplicates allowed <input type="radio"/> Flag Duplicates		
<input checked="" type="checkbox"/> 4 SEQUENCE CHECK Deck No. <u>5T 795</u> <small>(Standard: numeric sequence check, columns 73-80, duplicates flagged)</small>				Columns _____ through _____ <input type="radio"/> Duplicates allowed		
<input checked="" type="checkbox"/> 5 LIST <u>5T 795</u> <small>DECK NO.</small>				<input type="radio"/> Single Space <input checked="" type="checkbox"/> Double Space		
<input type="checkbox"/> ENTER into production status, Deck _____						

REPRODUCE _____ Times, Deck No. _____ | Leave Col. _____ blank

The required number of copies of the specified deck will be punched. If certain columns are to be left blank, that fact should be noted on the right-hand side.

PUNCH Identification, _____, in | Columns _____ through _____
reproduced deck. (Cols. 73-77 Standard)

The deck will be identified as specified; the standard program deck identification consists of five alphanumeric characters punched in columns 73-77. Provision is made for identification punching in other columns, if desired, on the right side of the sheet.

PUNCH Sequence Number Starting with _____ 73 _____ 80
Increment _____ 73 _____ 80

The first sequence number desired should be written in the top spaces. Note that if an identification has been punched in columns 73-77, only columns 78-80 should be filled with the desired sequence number. The increment to be used should be specified in the bottom spaces.

INTERPRET Deck(s) No. _____
(Cols. 1-52 and 73-80 Standard)

When a deck is interpreted, the information punched in card columns 1 through 52 (or 1-50 on some interpreters) and 73 through 80 is printed at the top of the card. For binary decks, only columns 73 through 80 can be interpreted.

SORT Separately Deck No. _____ | Columns _____ through _____
Deck No. _____
(Numeric sort on Cols. 73-80 Standard) Sort all cards (together)
 Sort alphanumerically

The standard sort is numeric on columns 73-80. Provision is made for sorting decks separately or together. If the "Sort all cards (together)" circle is X'd, it will be assumed to refer to all decks listed under "Decks to be used" at the top of the form, unless specific deck numbers are listed in this box on the left. (If, for example, two decks out of a possible three are to be sorted together, the numbers of the two to be sorted may be listed on the left side and the "Sort all cards (together)" circle can be X'd. In this case, the word "Separately" on the left side may be crossed out to emphasize exactly what is desired.)

When an alphanumeric sort is requested, numeric characters will appear after alphabetic (e.g., "0" will follow "Z"). If special characters (*, (, =, etc.) are encountered, they will be interspersed among the alphabetic characters according to the zone punch contained on the card.

 LIST _____
DECK NO. Single Space Double Space

Decks in source or symbolic language, or data decks, will be listed, either single or double spaced according to the circle marked.

 ENTER into production status, Deck _____

When the requester indicates that the deck submitted is to be processed for production status, the EDPM group will:

1. Reproduce the deck indicated, columns 1 through 72.
2. Identify the deck with the identification specified, which should consist of the unique group identification and deck number as explained on page 90.90.01.
3. File in the EDPM files. The original deck or decks will be returned to the requester.
4. Reproduce copies of the file deck as needed to meet the demand.

If the deck includes a main program, ask your EDPM Group whether they require a \$DATA (or *DATA) card filed at the end of the deck.