TECHNICAL NOTE 400-24

GRAPHPAK I -
A THREE-DIMENSIONAL
MANIPULATION PROGRAM

Michael Adamowicz

September 10, 1965

NEW YORK UNIVERSITY
SCHOOL OF ENGINEERING AND SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
Laboratory for Electroscience Research

University Heights
Bronx, New York 10453

ACKNOWLEDGMENT

The research described in this report was supported in part by the National Science Foundation through its Graduate Cooperative Fellowship program and in part by the National Aeronautics and Space Administration under research grant NGR-33-016-038. Throughout the research period, valuable advice and suggestions were received from my advisor, Professor H. Freeman. I am also indebted to Mr. Paul Brady for instructing me in the use of the PDP-5 computer.

ABSTRACT

15984

This report describes the development and use of GraphPak I, a 3-dimensional manipulation and plotting program. This program was written for a PDP-5 digital computer with a 4K memory and a CALCOMP digital plotter. It enables the user to describe an object in terms of points and straight lines and then obtain a perspective drawing of the object in any desired orientation. In addition, certain kinds of distortions and alterations can be made without having to redescribe the object; however, there is no provision for removing hidden lines.

Author

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Sample Drawings Made With GRAPHPAK I

Flow Charts

I. INTRODUCTION

In the last few years there have been extensive improvements in the performance of digital computers and the types of input-output equipment available. A great deal of work has been done in the area of computer-aided systems of analysis and design. The area of computer graphics is a particularly popular one. "Sketchpad"[1] and "Sketchpad III",[2] developed at the M.I.T. Lincoln Laboratories, are rather sophisticated examples of programs written in this area. These programs permit the user to communicate with the computer by means of a light pen and a series of control switches. The computer's output is in the form of a display on a large scope. With these systems the user can draw, rotate, scale, distort, and move figures around. This report describes a limited version of "Sketchpad," called "GraphPak I," written for a PDP-5 digital computer with a 4K memory capacity and a CALCOMP digital incremental plotter. It is an initial attempt at developing a graphical processing program for this computer and is expected to provide guidelines for the design of a more powerful program at a later time.

During the writing of this graphical manipulation program it was decided to restrict the objects to be handled to those composed of points and straight line segments. No attempt was made to eliminate "hidden lines". As a result, all objects are interpreted as being of the "wire-frame" type. The user of this program communicates with the computer by means of a typewriter. He describes the object by typing in a list of points and a list of the point pairs that form lines. He then specifies the angles of rotation about two given axes and the distance from the object to the point at which the observation takes place. The computer will then calculate a true perspective view of the object and draw it using the plotter. Scale, translation, and certain kinds of editing and distortion options are also available. As a result, the program is quite flexible in its applications.

Figure Axes: X, Y, Z

Image Plane Axes: Y_1, Z_1

Figure Coordinates: x, y, z

Rotated Figure Coordinates: x', y', z'

Image Coordinates: y_1, z_1

Scaled Image Coordinates: y_1', z_1'

Final Image Coordinates: y_f, z_f

d_e = the absolute value of the distance from the eye point to the origin of the figure axes.

d_d = the absolute value of the distance from the image plane to the origin of the figure axes.

d_p = the absolute value of the x-component of the distance from the eye point to any point on the rotated figure.

Figure 1 provides a graphical interpretation of some of the preceeding terms.

B. Equations

1. Representation of the Object

The objects this program will be required to handle are of the "wire-frame" type and are constructed only of points and straight line segments. The problem of representing objects is, therefore, reduced to that of representing straight lines. One way of doing this is to use 3x2 matrices to represent lines. Thus, the matrix:

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & a_3 \end{bmatrix} \qquad \text{(6 pieces of data needed per line)}$$

could be used to represent a line from the point (a_1, a_2, a_3) to the point (b_1, b_2, b_3). This method is rather inefficient. In order to represent a tetrahedron (which has 6 lines) a total of 6x6 = 36 data pieces are needed.

A more efficienct method of representing the object would be in terms of points and point pairs. Each point would be represented as a 3x1 matrix of the form:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \text{(3 pieces of data needed per point)}$$

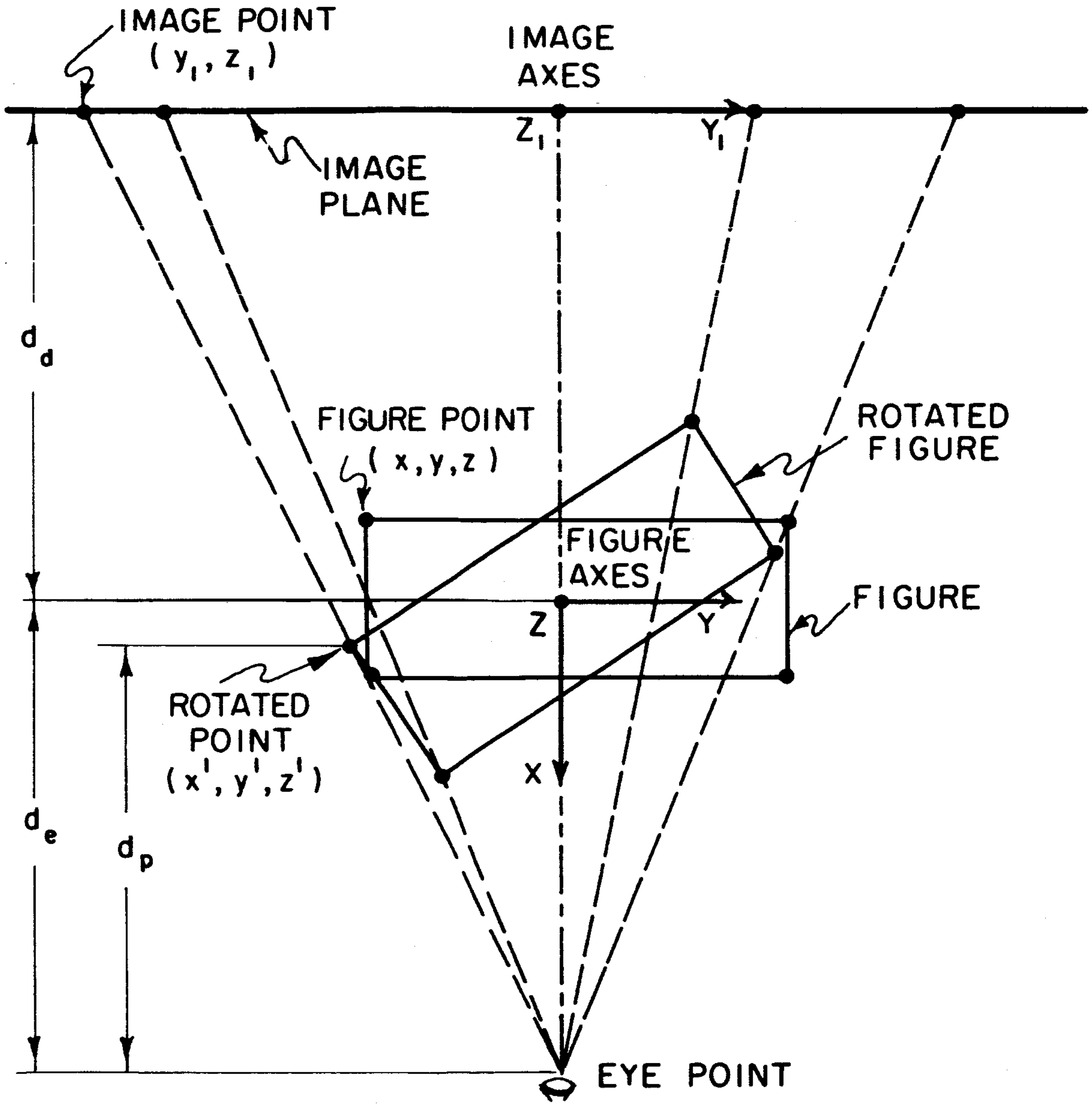


FIG. 1
GRAPHICAL REPRESENTATION OF TERMS

The lines would then be represented as pairs of points: 1-3, 2-5, or 4-1, describing what two points are connected by the line. In this way the above tetrahedron could be represented using only 4x3 + 6x2 = 12 + 12=24 pieces of data. The method used in this program is the points-plus-point-pairs one.

2. Rotation Equations

The operation of rotating the figure in space in order to obtain the corresponding rotated figure can be represented as a matrix equation of the form:

$$[A'] = [R][A] \tag{1}$$

where [A] is the 3x1 matrix of the figure point, [R] is the 3x3 rotation matrix, and [A'] is the 3x1 matrix of the rotated figure point. In general, the rotation matrix, R will be of the form:

$$[R] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2}$$

If the multiplication indicated by Equation (1) were carried out the following would be obtained:

$$\begin{aligned} x' &= r_{11}x + r_{12}y + r_{13}z \\ y' &= r_{21}x + r_{22}y + r_{23}z \\ z' &= r_{31}x + r_{32}y + r_{33}z \end{aligned} \tag{3}$$

The evaluation of the r_{ij} coefficients depends on what rotation technique is used. For this program it was decided to have two axes of rotation, the z-axis and an axis that would start out in line with x-axis but would be moved around by any z-axis roatation. The angle of rotation counterclockwise about the z-axis will be called θ and the counterclockwise rotation about moving axis originally corresponding to the x-axis will be called Φ. At all times the θ rotation will come first and then the Φ rotation. Using this technique of rotation, Equation (3) becomes (see Figure 2):

$$\begin{aligned} x' &= \cos\theta\, x - \cos\Phi \sin\theta\, y + \sin\Phi \sin\theta\, z \\ y' &= \sin\theta\, x + \cos\Phi \cos\theta\, y - \sin\Phi \cos\theta\, z \\ z' &= 0\cdot x + \sin\Phi\, y + \cos\Phi\, z \end{aligned} \qquad (4)$$

The rotation matrix is therefore:

$$[R] = \begin{bmatrix} \cos\theta & -\cos\Phi \sin\theta & \sin\Phi \sin\theta \\ \sin\theta & \cos\Phi \cos\theta & -\sin\Phi \cos\theta \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix} \qquad (5)$$

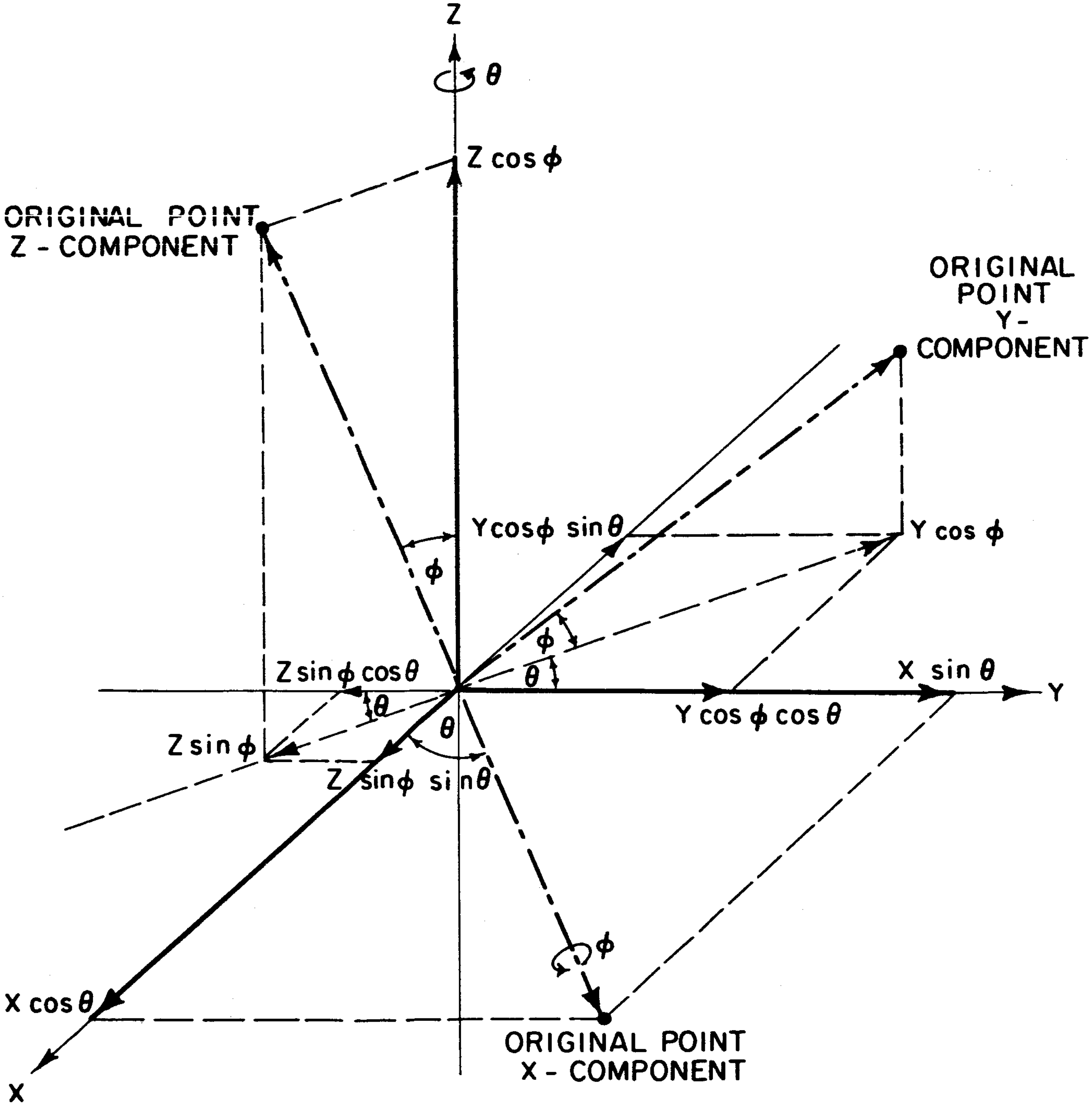
3. Projection

The operation of obtaining the image of the rotated figure can be represented as a matrix equation of the form:

$$[A_1] = [P][A'] \qquad (6)$$

where $[A_1]$ is the 3x1 matrix of the image point and $[P]$ is the 3x3 projection matrix. For the method of approach used in this program the $[P]$ will be of the form:

$$[P] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} \qquad (7)$$



F I G. 2

FIGURE ROTATION

where p is the projection factor. The value of p is a function of the x-component of the rotated point and is computed as follows (refer to Figure 3):

$$y_1 = py' \tag{8}$$

thus

$$p = y_1/y'$$

Referring to Figure 3, triangle EBF and EIP are similar triangles and thus:

$$\frac{y_1}{y'} = \frac{EI}{EF} = \frac{d_d + d_e}{d_p} = \frac{d_e + d_d}{d_e - x'} \tag{9}$$

Multiplying numerator and denominator by $1/d_e$ ($d_e \neq 0$):

$$\frac{y_1}{y'} = \frac{1 + d_d/d_e}{1 - x'/d_e} \tag{10}$$

Except for a change of scale, the p factor will remain unchanged if it is multiplied by a positive constant, K:

$$p = \frac{y_1}{y'} = \frac{\frac{1}{2}K(1+d_d/d_e)}{\frac{1}{2} - \frac{1}{2}x'/d_e} \tag{11}$$

The calculations can be simplified by always having $d_d = d_e$. This would also help to control the size of the images, which have a tendency of becoming extremely large as the eye point approaches the rotated figure.

$$p = \frac{K}{\frac{1}{2} - \frac{1}{2}x'/d_e} \qquad (12)$$

This is the basic equation used in the projection section of this program. Because the program uses a fractional number divide subroutine, K was adjusted so that the divisor would always be larger than the dividend. (K was set = to 1/16 = .0625).

4. Scaling

The operation of enlarging or reducing the image can be represented by:

$$[A_1'] = [S][A_1] \qquad (13)$$

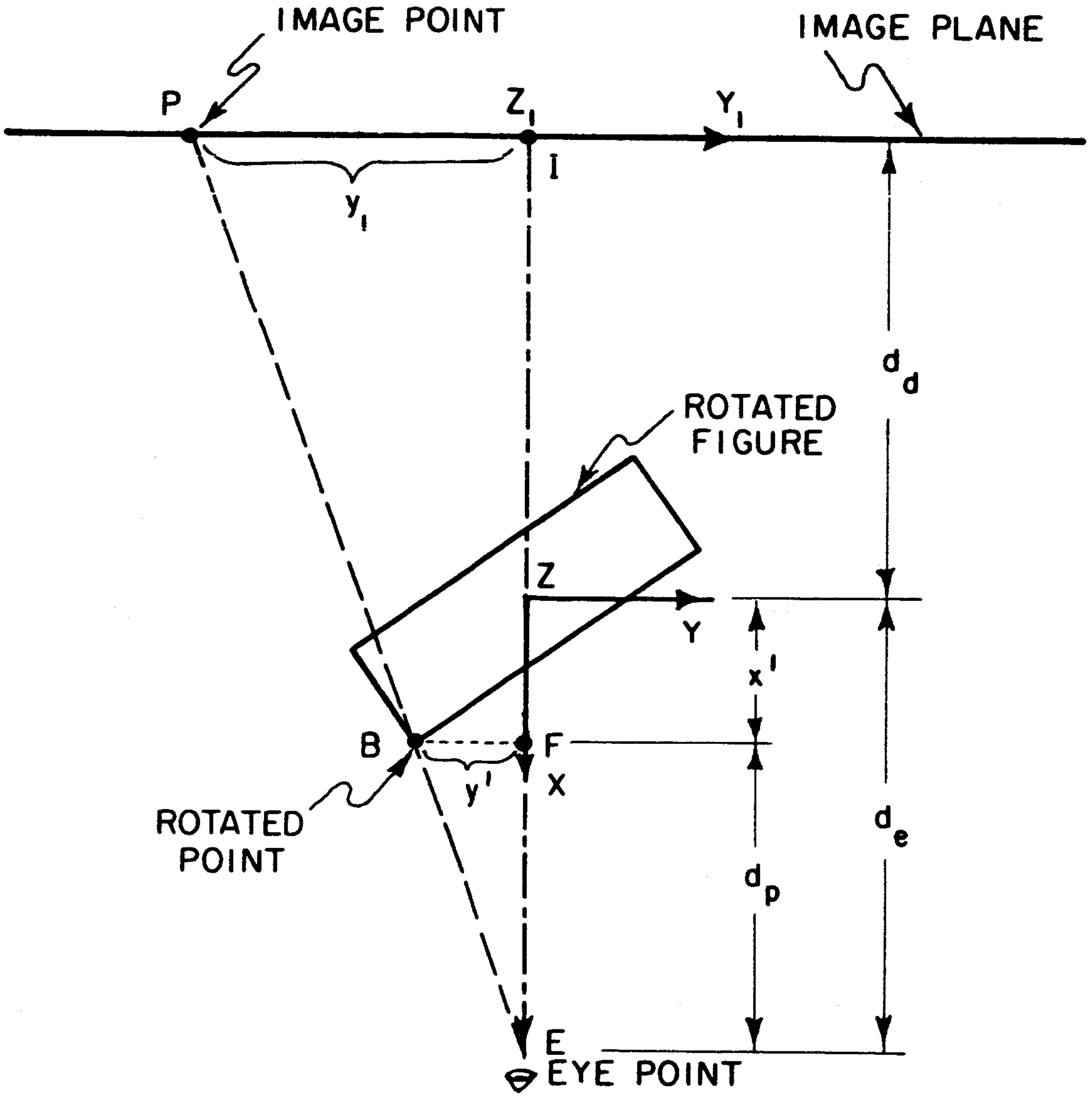
where $[A_1']$ is the 3x1 matrix of the scaled image point and $[S]$ is the 3x3 scalar matrix:

$$[S] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \qquad (14)$$

5. Overall Graphical Manipulation

The overall process[3] of rotating, projecting, and scaling can be represented as follows:

$$[A_1'] = [S][P][R][A] \qquad (15)$$



F I G. 3

PERSPECTIVE PROJECTION

It was decided to perform translation in this program by translating the origin of the image axes on the image plane. As a result, the translation was made entirely independent of the points. One disadvantage of this is that if any part of the image is translated the entire image will also be translated.

III. INSTRUCTIONS ON THE USE OF GRAPHPAK I

The program was written for a PDP-5 digital computer with a 4K memory capacity and a CALCOMP digital incremental plotter. The program is self-contained and occupies registers 0160 to 4600, inclusive, plus 5162 to 5170, 5303 to 5335, and 5341 to 5364.

A. How to Use Program

After the binary tape is read in, place the teletype on LINE, set the switch register to 0513, then push LOAD ADDRESS and then START. "NEXT?" will then be typed out. Whenever this is typed, the user will know that the program is in the central control section and is waiting for a character to be typed in. The program is divided into eight subsections. Each of these can be entered only from the central control by typing the appropriate character code. When the operation being performed by a subsection is completed, control is returned to the central control.

Section Name	Code	Comments
Points	P	Type in number of points and then x, y, z coordinates of the points as signed decimal fractions.
Lines	L	Type in number of lines and then lines as point pairs in decimal.
Rotation	R	Type in z-axis angle and then the moving x-axis angle in signed decimal tenths of degrees (without decimal point) in the range: $-\pi = -1800$ to $1800 = \pi$.
Projection	E	Type in eye-point distance as a positive, decimal, mixed number between 2.0 and 63.9999.

Section Name	Code	Comments
Scale	S	Type in scale factor as a positive, decimal, mixed number.
Translations	T	Type in y and z translation as signed decimal fractions.
Draw	D	Plotter will draw view.
Correction	C	Type P to correct points, L to correct lines, M to replace original points with rotated points.

B. Numbers

Every number can be terminated by typing a space. If a mistake is made and the number has not yet been terminated, type the RUBOUT key and then the desired number.

The fractional number subroutine automatically types the decimal point. If a negative number is desired, the minus sign must be typed after the decimal point.

The mixed numbers are composed of an integer part and a fractional part and both parts must by typed. Any non-numeric character (except RUBOUT) will terminate the integer part. However, to avoid getting the mixed number typed out as 9..37,9 .37, or 9z.37, use the ALT MODE key to terminate the integer part.

Although fractional numbers can have any number of digits, only the left-most $3\frac{1}{2}$ digits are significant.

Integer numbers are interpreted modulo 4096. Thus both 1 and 4097 will be interpreted as 1.

The decimal points for the angles are not typed. Thus if an angle of -π radians (-180.0 degrees) is desired, one would type -1800.

Do not use + to indicate positive numbers. Except for - and RUBOUT, every non-numeric character is interpreted as a terminating character.

C. Points

Points are described by typing their x, y, and z components as signed, fractional numbers. When the correct number of points have been entered, the program automatically returns to the central control.

D. Lines

Lines are described by typing the numbers of the two points that form the line. If, for example, 1 3 or 3 1 are typed, the computer records that a line joining the first and third points of the points list is to be drawn. When the correct number of lines have been entered, control is automatically returned to the central control.

E. Projection

The user types in the distance from the eye to the origin of the original figure axes as a positive, mixed number. A drawing that is close to being an isometric one can be obtained by typing 63.9999, the largest eye point distance permitted.

F. Translation

The signed fractional numbers typed in this section correspond to fractions of 5.12 inches. Thus, y = -.2 will result in a movement of about 1.02 inches in the negative y direction. Translation involves only a movement of the plotter's pen. The values of the points are not altered.

G. Correction

In correcting both points and lines, the user first types the number of the point or line with which the correcting is to begin. The corrected data is entered as before. However, at the end of each corrected line or point, the program waits for an additional character. If an F is now typed, control will be returned to the central control. If any other character is typed the program will wait for corrections in the next point or line. The correction section can thus be used to add points or lines.

In the move (M) part of the correction section, the user types the numbers of the starting and terminating points whose rotated components are to replace those of the original figure.

Example:

```
NEXT?   C
CORRECT?   L
#   9
9   10   F
NEXT?   C
CORRECT?   P
#   6
.0   .3   .2
.1   .3   .5   F
```

Example (continued):

```
NEXT?    C

CORRECT?    M

#   5   TO

#   8

NEXT?       (program waiting in central control)
```

H. The Order of Entering Sections

Normally, one would follow this sequence: Points, Lines, Rotation, Projection, Scaling, Translation, and Drawing. However, the lines can be described at any time before the drawing is made. Translation can be performed at any time. Projection must follow rotation and scaling must follow projection. As a result, different views of the object can be obtained without having to retype the points or lines, but by simply returning to the desired section (Rotation, Projection, Scaling, etc.).

I. Taped Descriptions

If an object is to be used over and over again, a tape can be punched out containing the points and lines data. This tape can then be read in whenever the object is to be studied. (See attached memory map for locations of point and line data: Figure A-2)

J. Errors

The program is able to recognize certain kinds of errors. When such an error is encountered "ER" is typed.

Section	Cause of Error
Central Control	A character other than P,L,R,E,S,T,D, or C was typed.
Rotation	Angle outside -1800 to 1800 range.
Projection	Eye point distant outside 2.0 to 63.9999 range.
Scale	"OV" is typed if multiplying by the scale factor caused a component of a point to overflow. The octal number of the first such point is also typed.
Change	A character other than P, L, or M was typed.

K. Example

Suppose one wanted to study the rectangular solid shown in Figure 4. The procedure would be as follows:

```
NEXT? P
POINTS
# 8
.Ø  .Ø  .Ø        (Note:  Ø is computer's way of representing 0)
.5  .Ø  .Ø
.5  .5  .Ø
.Ø  .5  .Ø
.Ø  .Ø  .75
.5  .Ø  .75
.5  .5  .75
.Ø  .5  .75
NEXT? L
LINES
#  12
1  2
2  3
3  4
4  1
5  6
6  7
7  8
8  5
1  5
6  2
```

```
3  7
8  4
NEXT? R
ANGLE
Ø
ANGLE
Ø
NEXT? E
EYE PT
63.Ø
NEXT? S
SCALE
2.Ø
NEXT? D
DRAW
END
NEXT? T
TRANS
Y=.3  Z=.Ø
NEXT? E
EYE PT
2.Ø
NEXT? S
SCALE
3.Ø
NEXT? D
DRAW
END
NEXT? R
ANGLE
45Ø
ANGLE
3ØØ
NEXT? E
EYE PT
63.Ø
NEXT? S
SCALE
3.Ø
NEXT? D
DRAW
END
NEXT?  (Computer waiting in control section.)
```

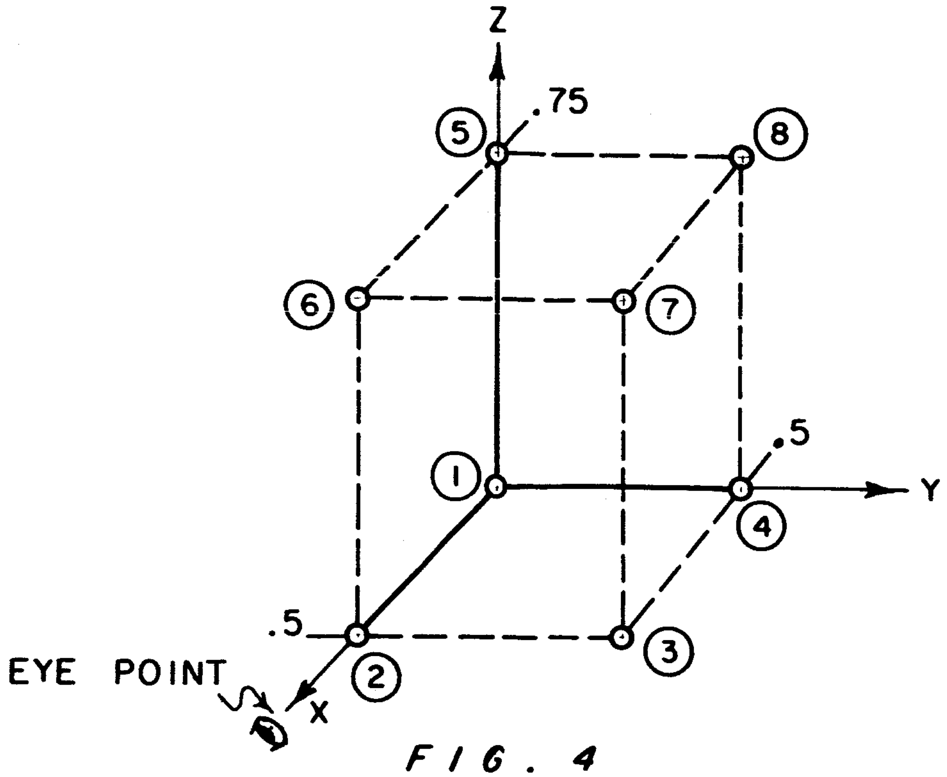


FIG. 4
RECTANGULAR SOLID

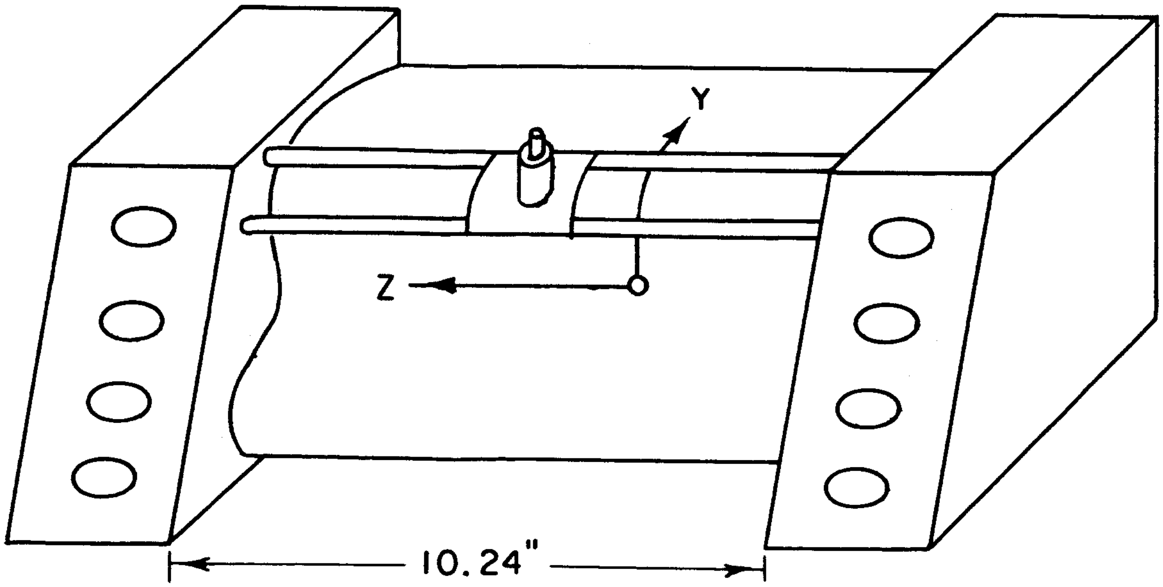


FIG. 5
PLOTTER

IV. APPLICATIONS

The uses to which GraphPak I can be put are really best left to the imagination of the user. There are certain features and limitations that would certainly determine to some extent the possible applications, but most of these have already been mentioned in the section of this report dealing with instructions on the use of the program. The purpose of this section is to provide examples of drawings made with GraphPak I and to mention some of the not too obvious ways of using it.

The program can be used as an "on line" designing aid. That is, a basic object can be described and studied in any orientation and perspective in space. Additional points and lines can be added or existing points and lines changed and the effects of such changes can be immediately studied. When the desired design is achieved the program can be used to produce a "blueprint" by drawing the top, front and side of the object using isometric projections.

Because certain kinds of distortions are available, topics in topology involving the deformation of shapes can also be studied. For example, one can demonstrate how a square can be deformed first into a rectangle, then a pyramid, and then a tetrahedron. One could also show that a torus (doughnut) can not be deformed into a square.

In general, problems involving geometry, topology, or cartography are particularly good ones to study with GraphPak I. A set of typical graphical studies is illustrated in the Figures 6 through 10.

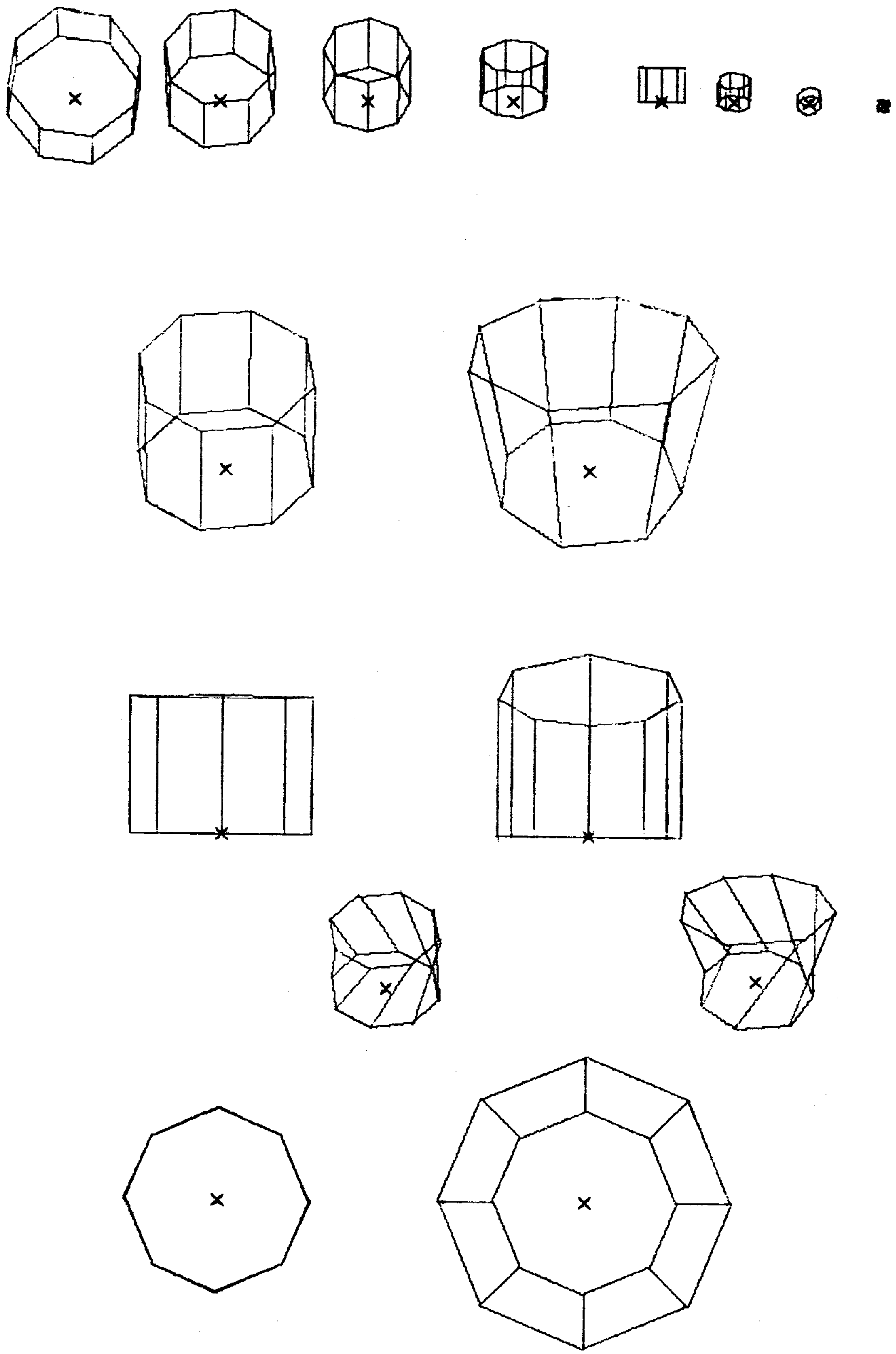
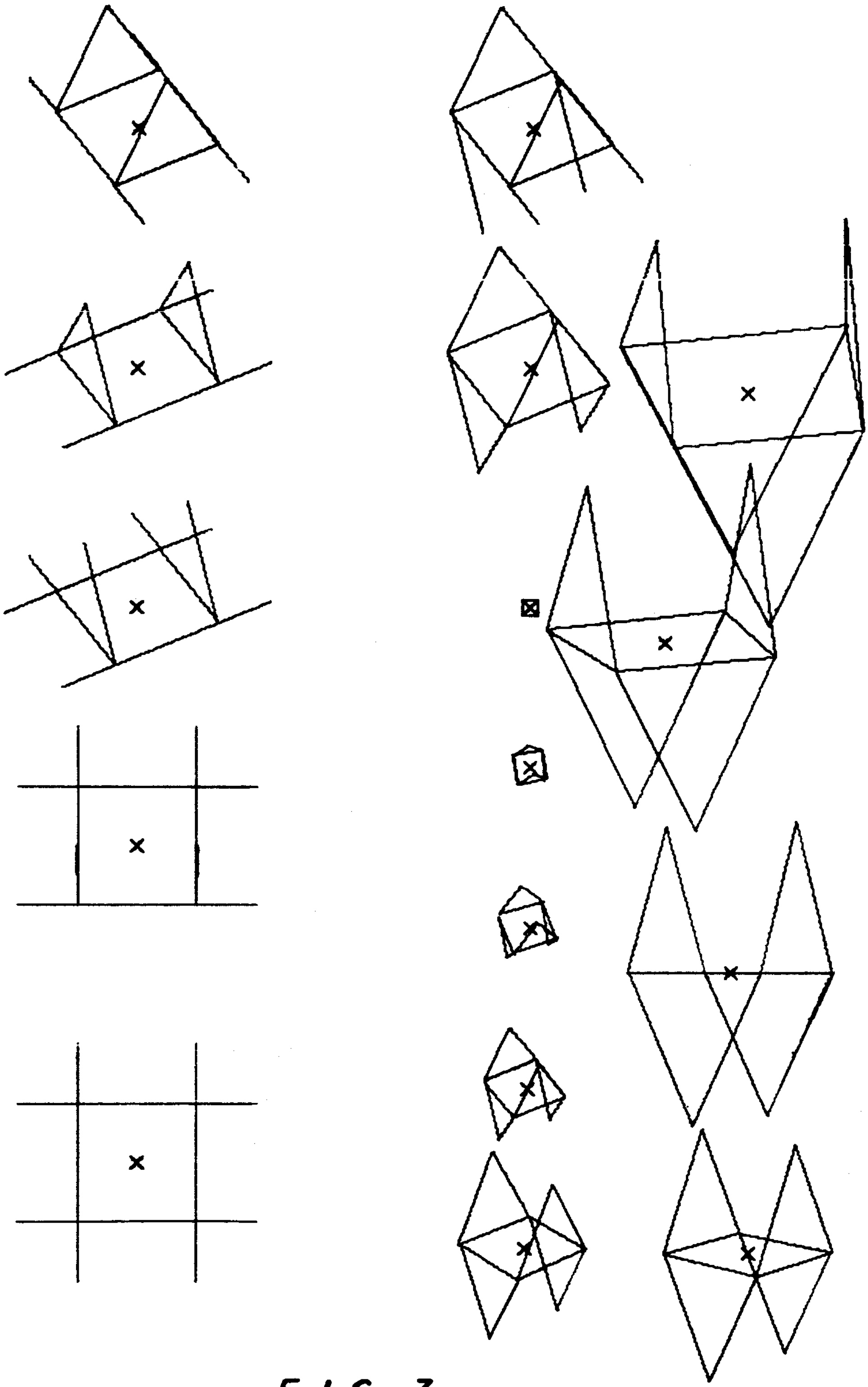
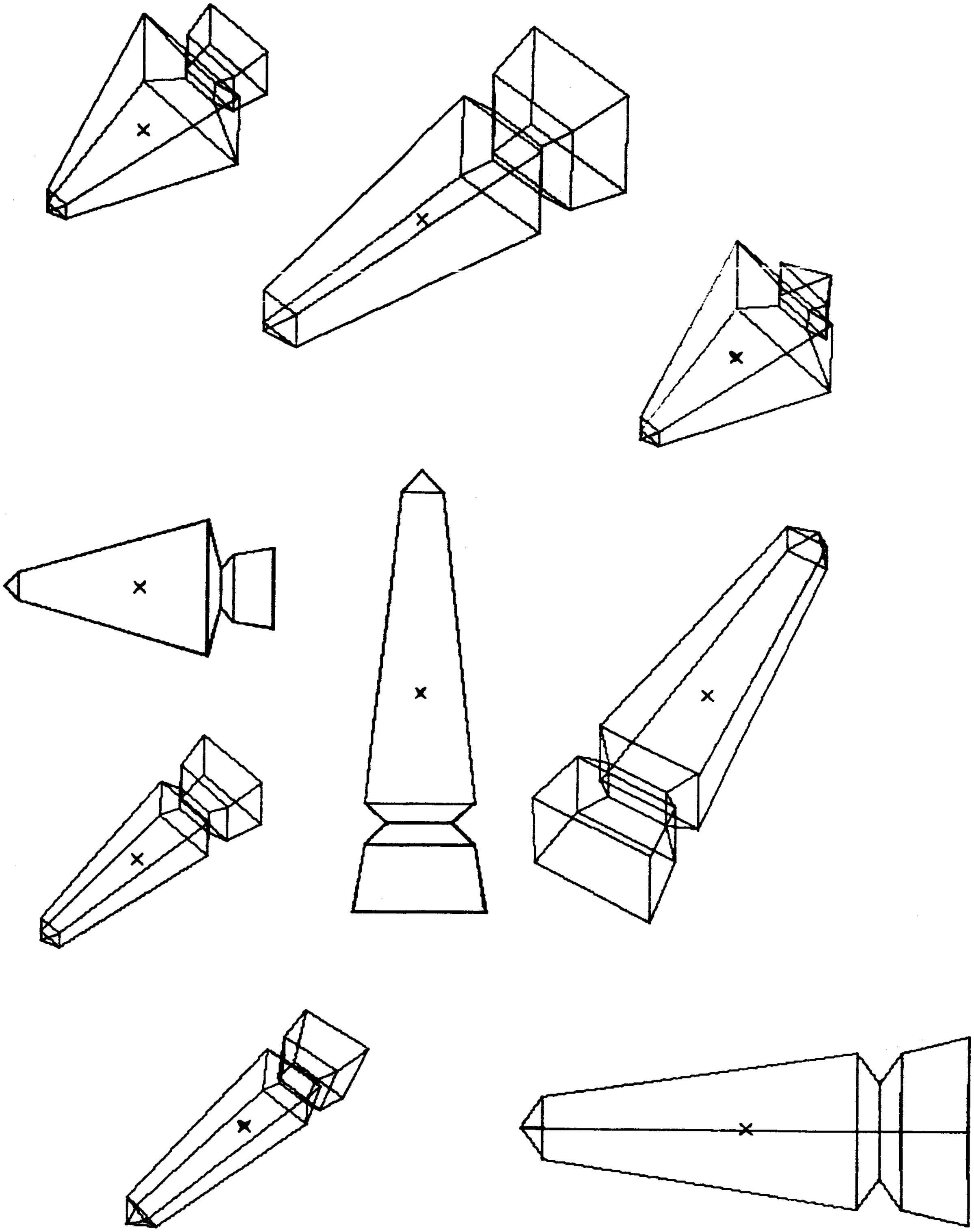


FIG. 6
PERSPECTIVE AND ISOMETRIC VIEWS



F I G. 7
WIRE - FRAME FIGURE DISTORTION



F I G . 8
3-DIMENSIONAL OBJECT IN SPACE

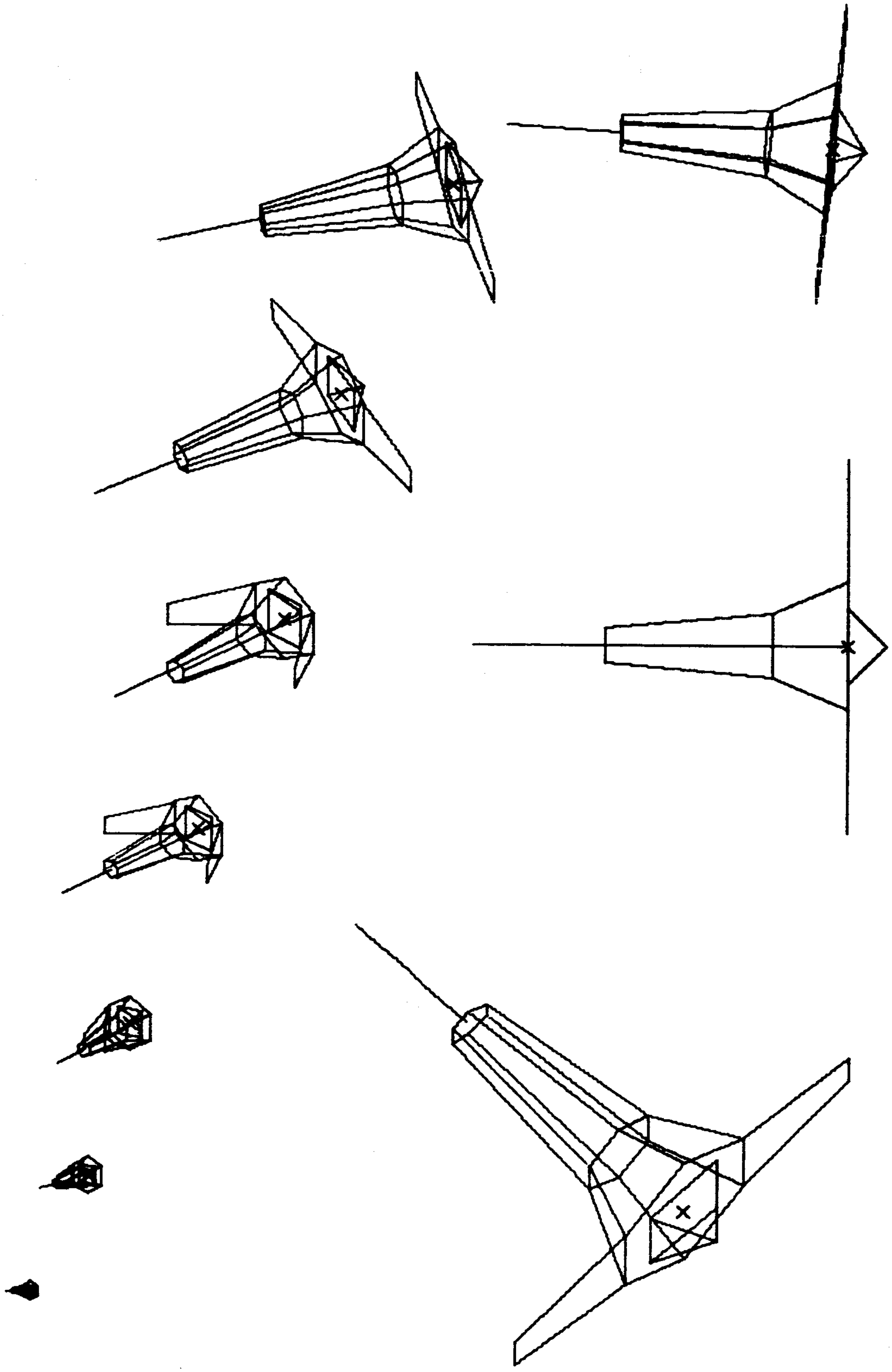


FIG. 9
AN APPROACHING SPACE PROPE

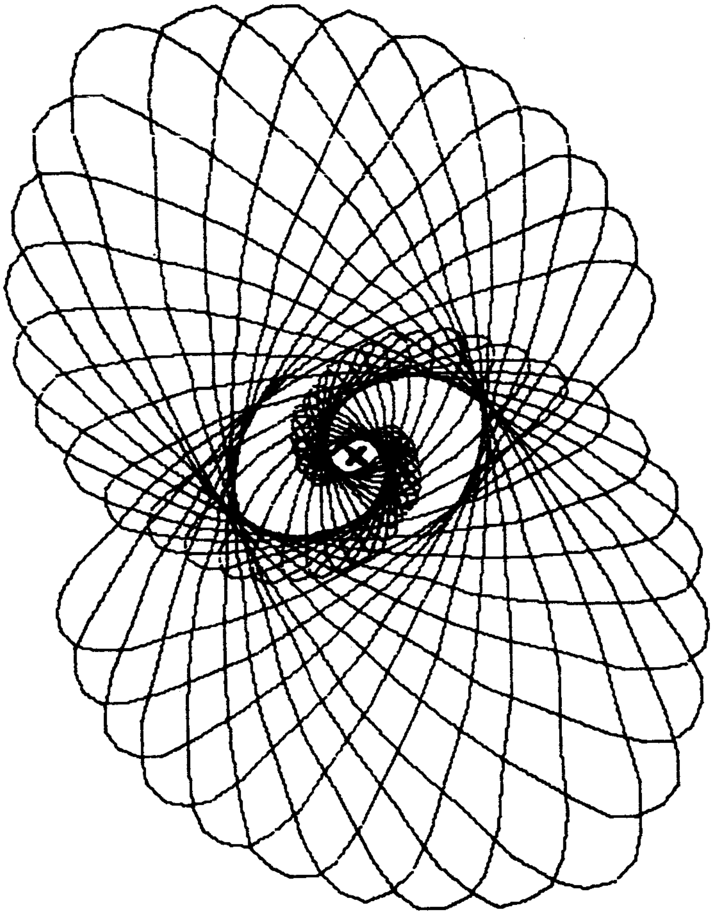


FIG. 10

ROTATING ELLIPSE DESIGN

V. SUGGESTIONS FOR EXTENDING GRAPHPAK I

There are many features that would be desirable additions to GRAPHPAK I. For example, if an inversion section were added, the program could be used to study all types of bilinear transformations. As the objects become more complex the removal of the "hidden lines" becomes more and more important. Thus, it would be very desirable to have a section that removes "hidden lines". It might be desirable to add a section that produces stereoscopic drawings of objects. The usefulness of the program as a design aid would be considerably extended if a section that enabled one to translate single points or groups of points were added. Another possible addition is the ability to work with any functional transformations. The addition of a section enabling the user to impose constraints upon the length of lines, the postition of points, or the angles between lines would also be an extremely useful addition. It is also possible to rewrite the program so that it can be controlled by another program. In this way one could, for example, automatically obtain drawings corresponding to what one would see as he moved over a terrain in an airplane. Some of these possibilities are more desirable than others. The problem of the limited memory capacity of the computer will also influence any decisions to expand GRAPHPAK I.

VI. SUMMARY

This report describes the development and use of GRAPHPAK I, a three-dimensional manipulation and plotting program. In as much as it is the first attempt at writing such a program it has several limitations. One of the most important of these is the inability to remove "hidden lines". It is still, however, a rather flexible graphical tool. With the addition of inversion, "hidden lines" removal, and translation of sets of points sections, the program could become a powerful research and design tool.

REFERENCES

1. Sutherland, I. E., "Sketchpad: A Man-Machine Graphical Communication System," M.I.T. Lincoln Laboratory, Technical Report No.296, January 1963.

2. Johnson, T.E., "Sketchpad III: Three-Dimensional Graphical Communication with a Digital Computer," M.I.T. Electronic Systems Laboratory, Department of Electrical Engineering, May 1963.

3. Roberts, L. G., "Machine Perception of Three-Dimensional Solids," M.I.T. Lincoln Laboratory, Technical Report No.315, May 1963.

APPENDIX A: Flow Charts

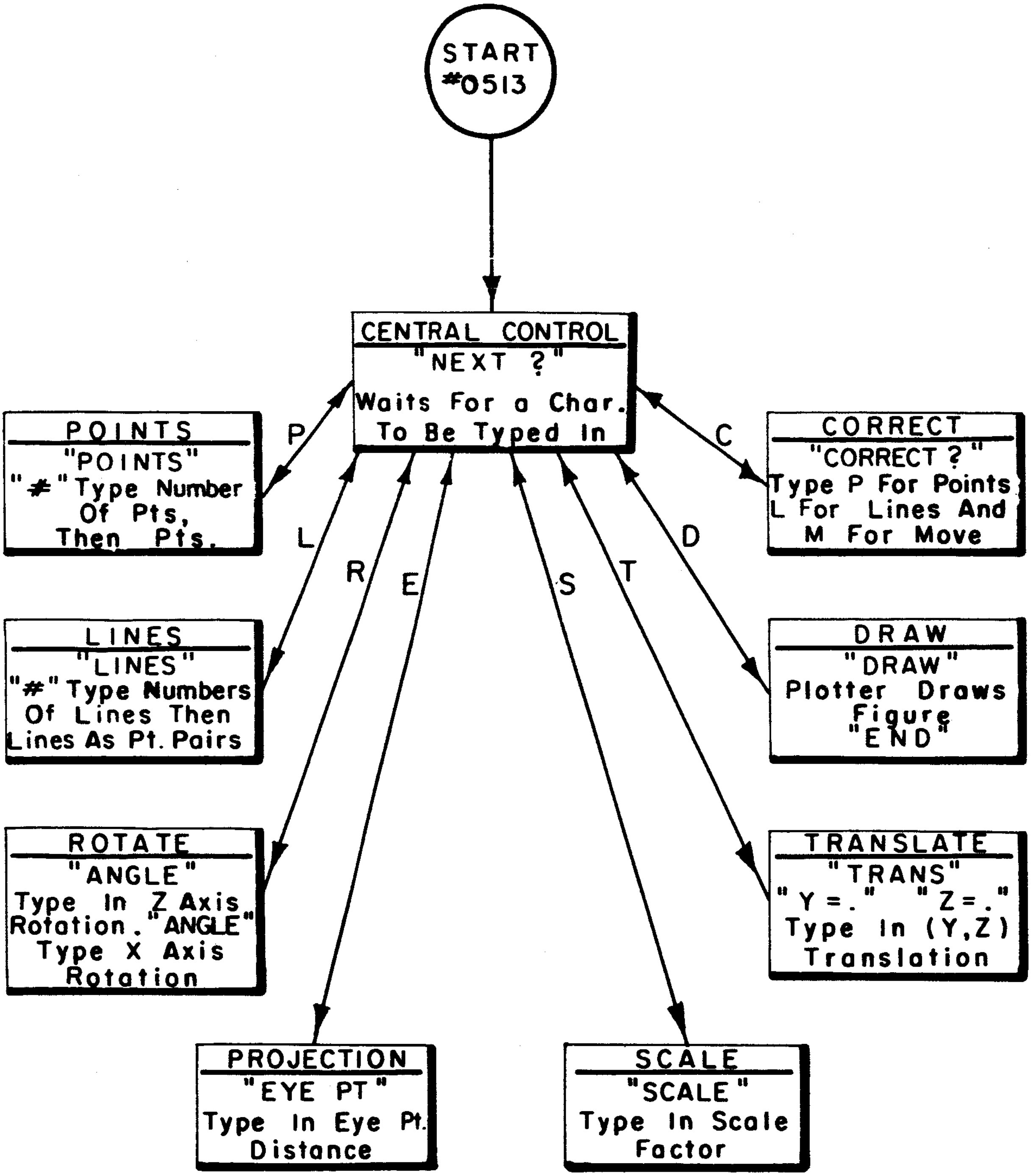


FIG. A-1

GENERAL FLOW DIAGRAM

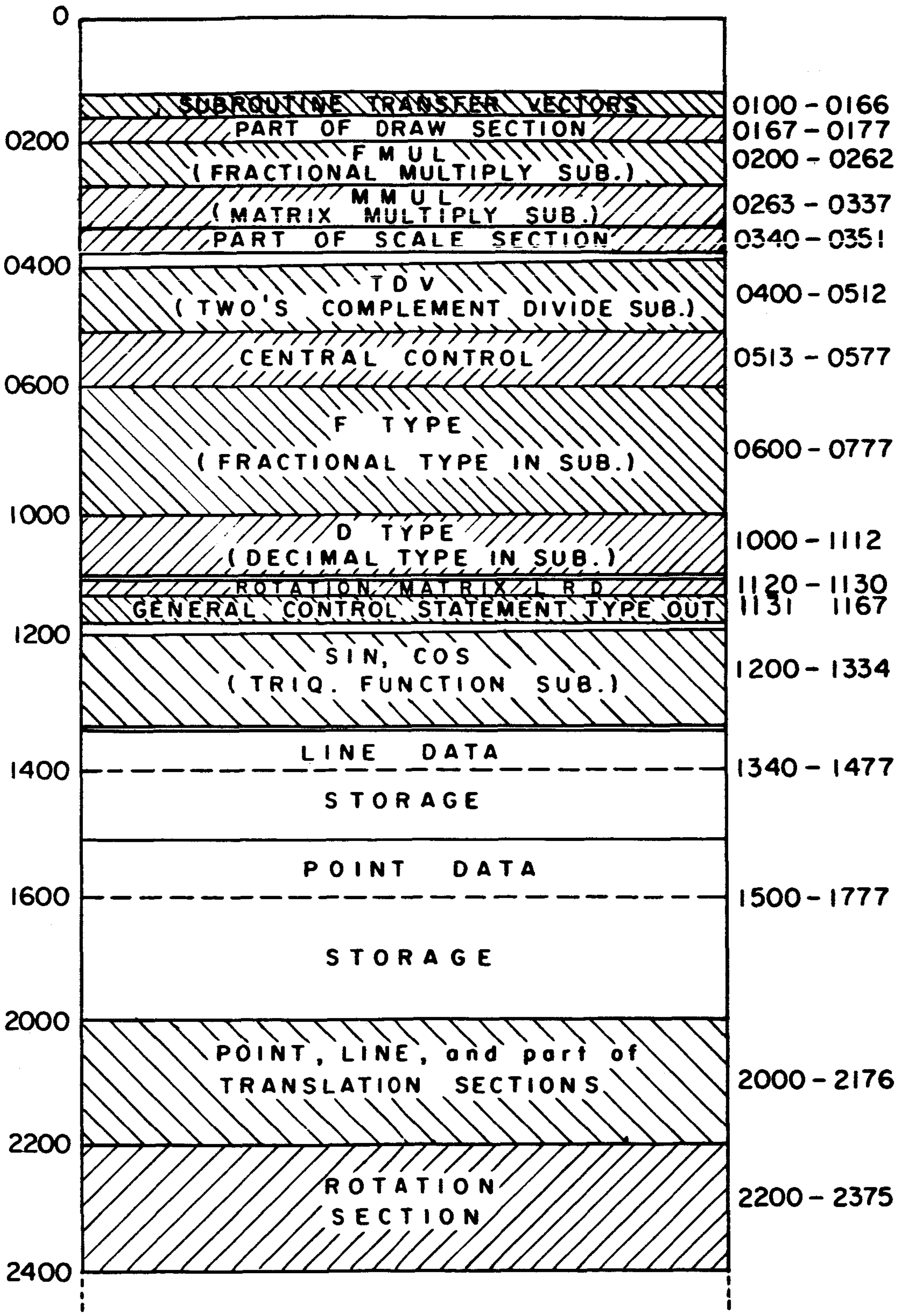


FIG. A-2

MEMORY MAP

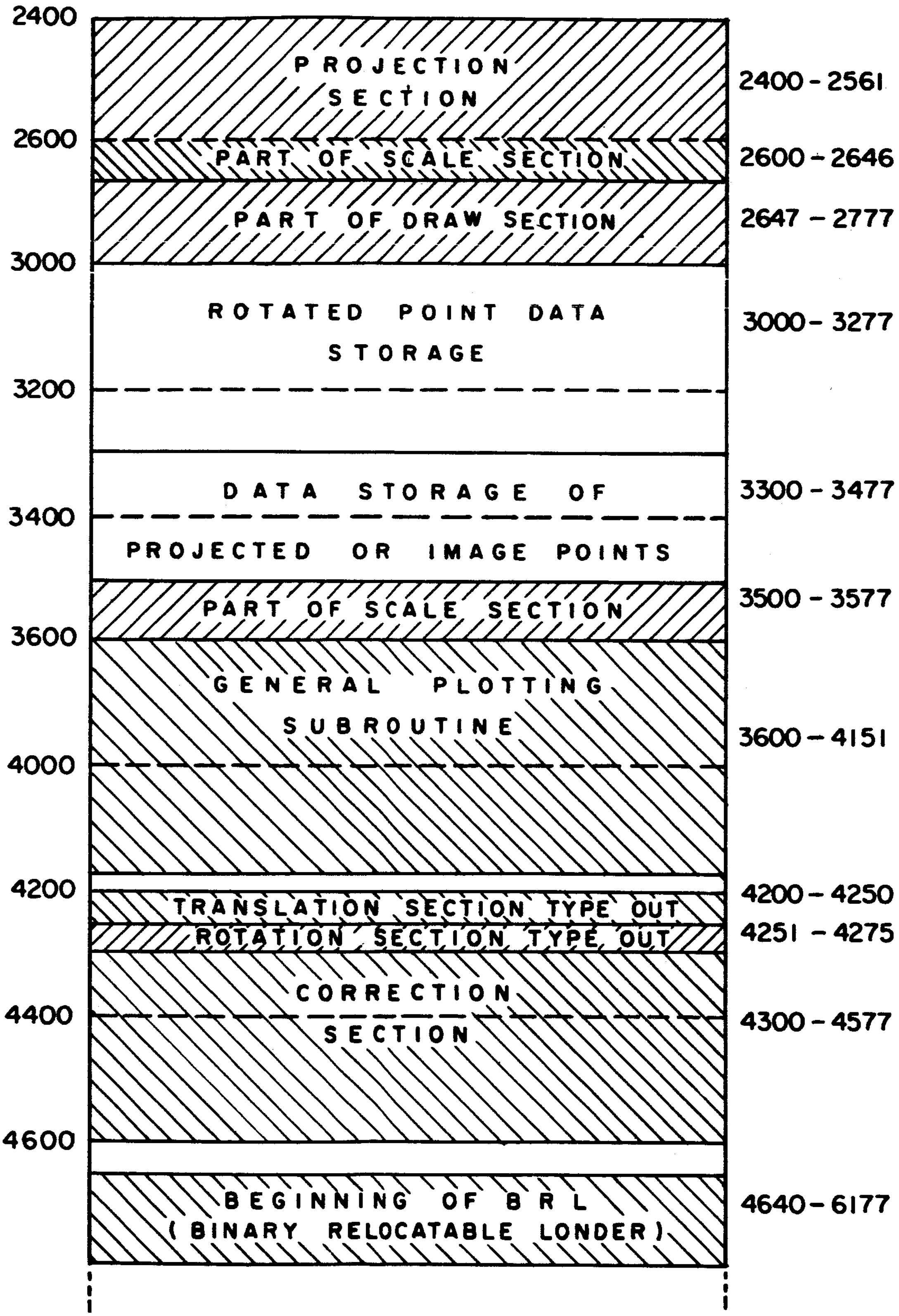
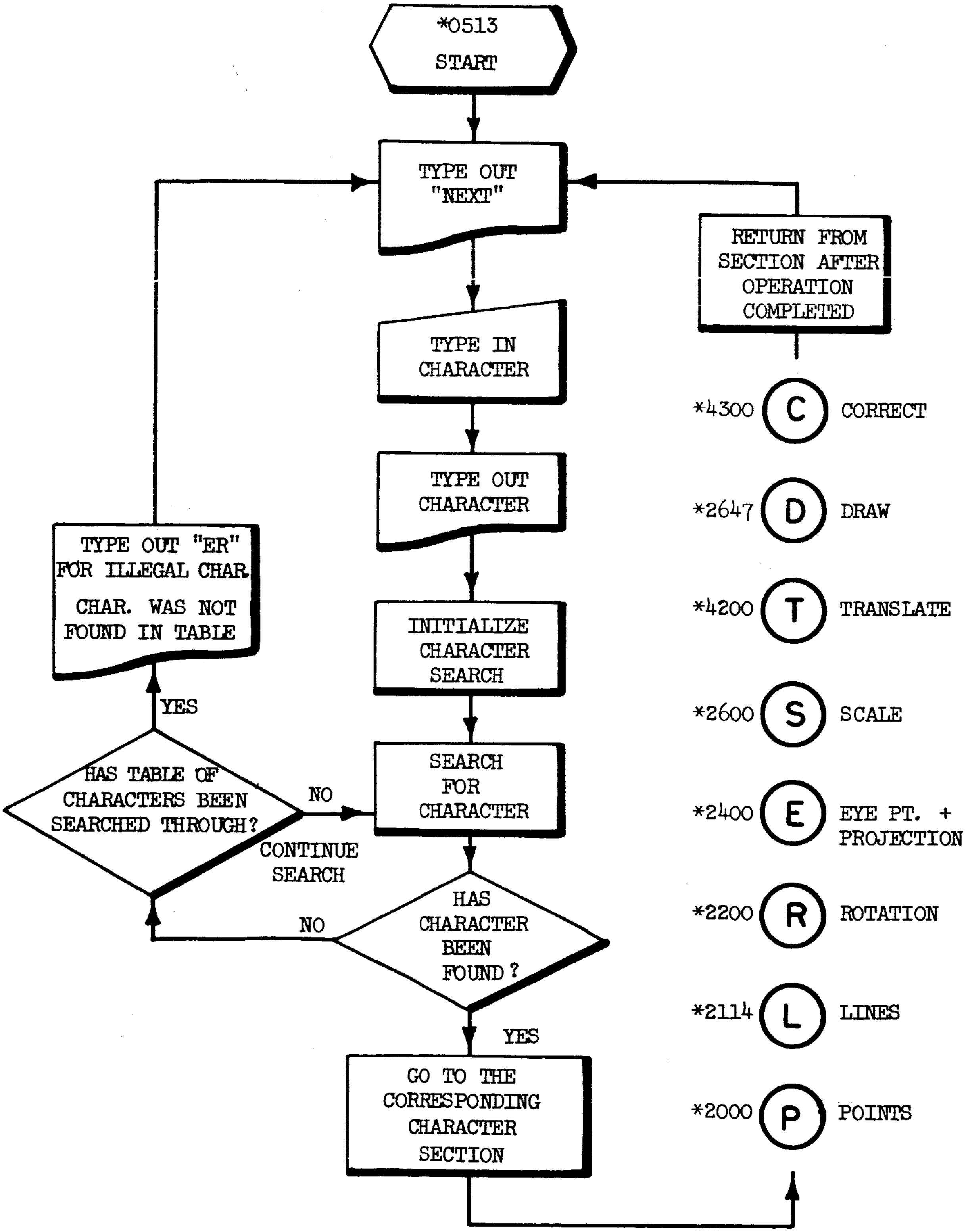


FIG. A-2

MEMORY MAP CONT.


*0513
START
TYPE OUT "NEXT"
TYPE IN CHARACTER
TYPE OUT CHARACTER
INITIALIZE CHARACTER SEARCH
SEARCH FOR CHARACTER
HAS CHARACTER BEEN FOUND ?
YES
NO
GO TO THE CORRESPONDING CHARACTER SECTION
HAS TABLE OF CHARACTERS BEEN SEARCHED THROUGH?
NO
CONTINUE SEARCH
YES
TYPE OUT "ER" FOR ILLEGAL CHAR.
CHAR. WAS NOT FOUND IN TABLE
RETURN FROM SECTION AFTER OPERATION COMPLETED
*4300 C CORRECT
*2647 D DRAW
*4200 T TRANSLATE
*2600 S SCALE
*2400 E EYE PT. + PROJECTION
*2200 R ROTATION
*2114 L LINES
*2000 P POINTS

F I G. A - 3
CENTRAL CONTROL

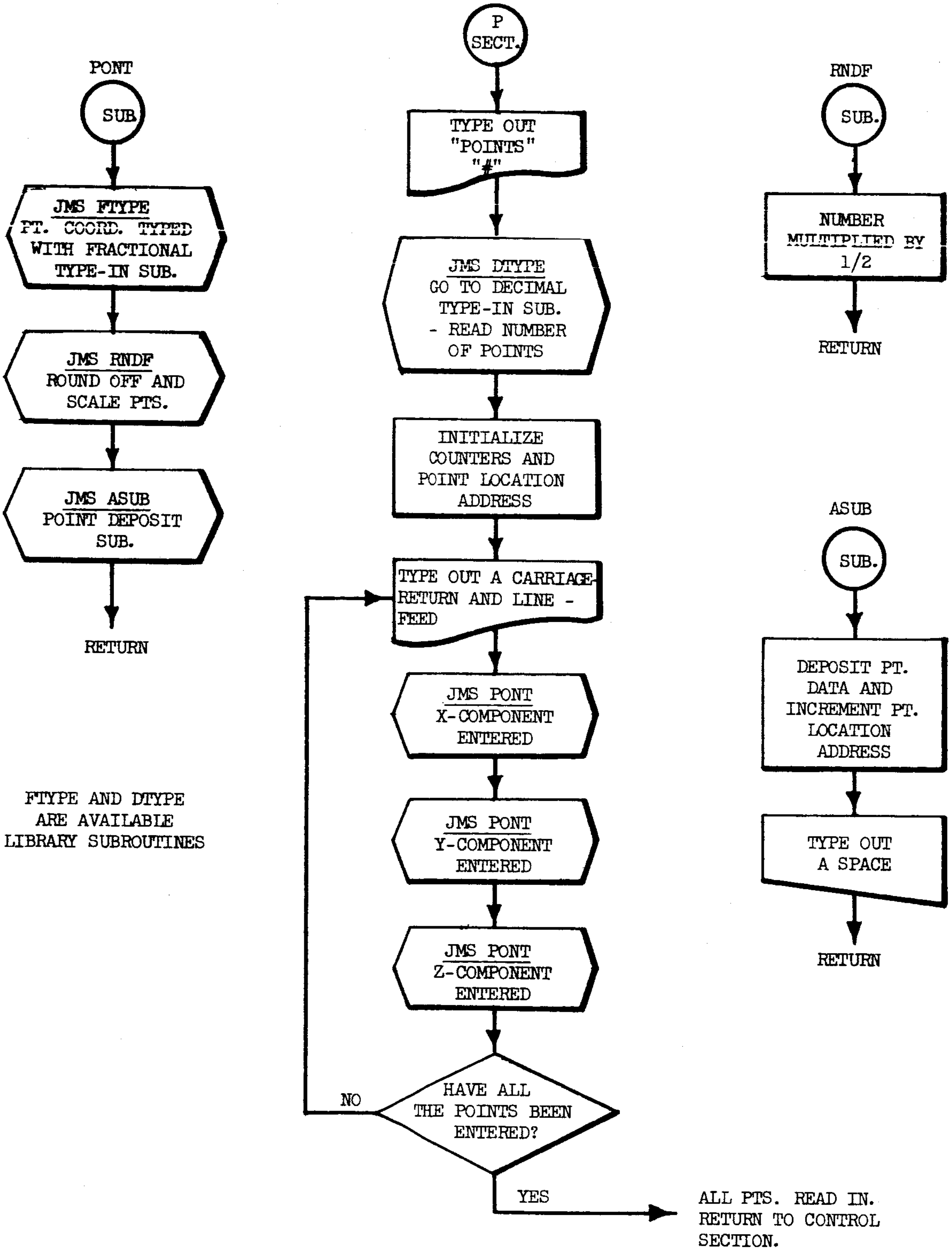
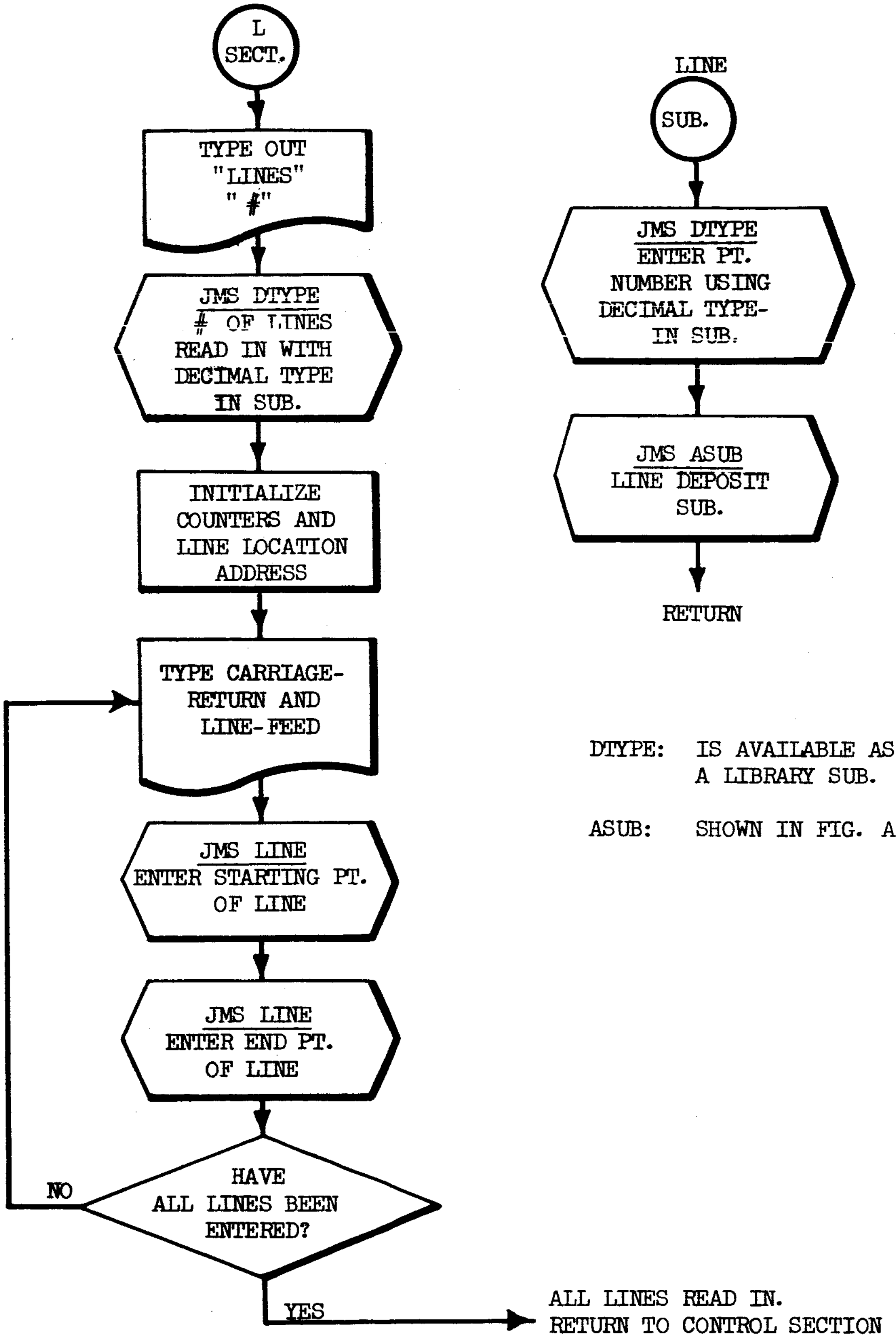

P SECT.
TYPE OUT "POINTS" "#"
JMS DTYPE
GO TO DECIMAL TYPE-IN SUB.
- READ NUMBER OF POINTS
INITIALIZE COUNTERS AND POINT LOCATION ADDRESS
TYPE OUT A CARRIAGE-RETURN AND LINE - FEED
JMS PONT
X-COMPONENT ENTERED
JMS PONT
Y-COMPONENT ENTERED
JMS PONT
Z-COMPONENT ENTERED
HAVE ALL THE POINTS BEEN ENTERED?
NO
YES
ALL PTS. READ IN. RETURN TO CONTROL SECTION.
PONT
SUB
JMS FTYPE
PT. COORD. TYPED WITH FRACTIONAL TYPE-IN SUB.
JMS RNDF
ROUND OFF AND SCALE PTS.
JMS ASUB
POINT DEPOSIT SUB.
RETURN
FTYPE AND DTYPE ARE AVAILABLE LIBRARY SUBROUTINES
RNDF
SUB.
NUMBER MULTIPLIED BY 1/2
RETURN
ASUB
SUB.
DEPOSIT PT. DATA AND INCREMENT PT. LOCATION ADDRESS
TYPE OUT A SPACE
RETURN

FIG. A-4

POINTS SECTION

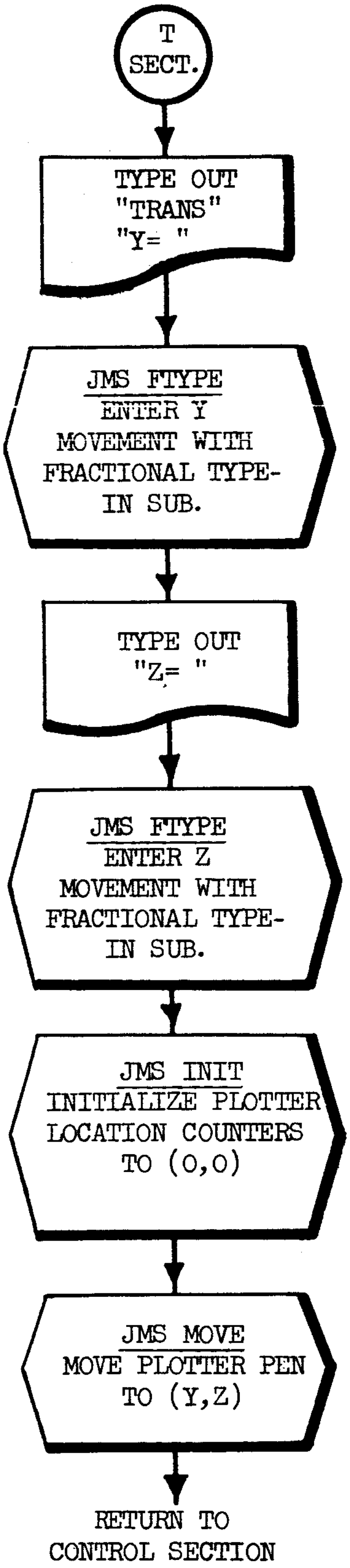


DTYPE: IS AVAILABLE AS A LIBRARY SUB.

ASUB: SHOWN IN FIG. A-4

FIG. A-5

LINES SECTION



FTYPE, INIT, AND MOVE
ARE AVAILABLE
AS LIBRARY SUB.

FIG. A-6

TRANSLATION SECTION

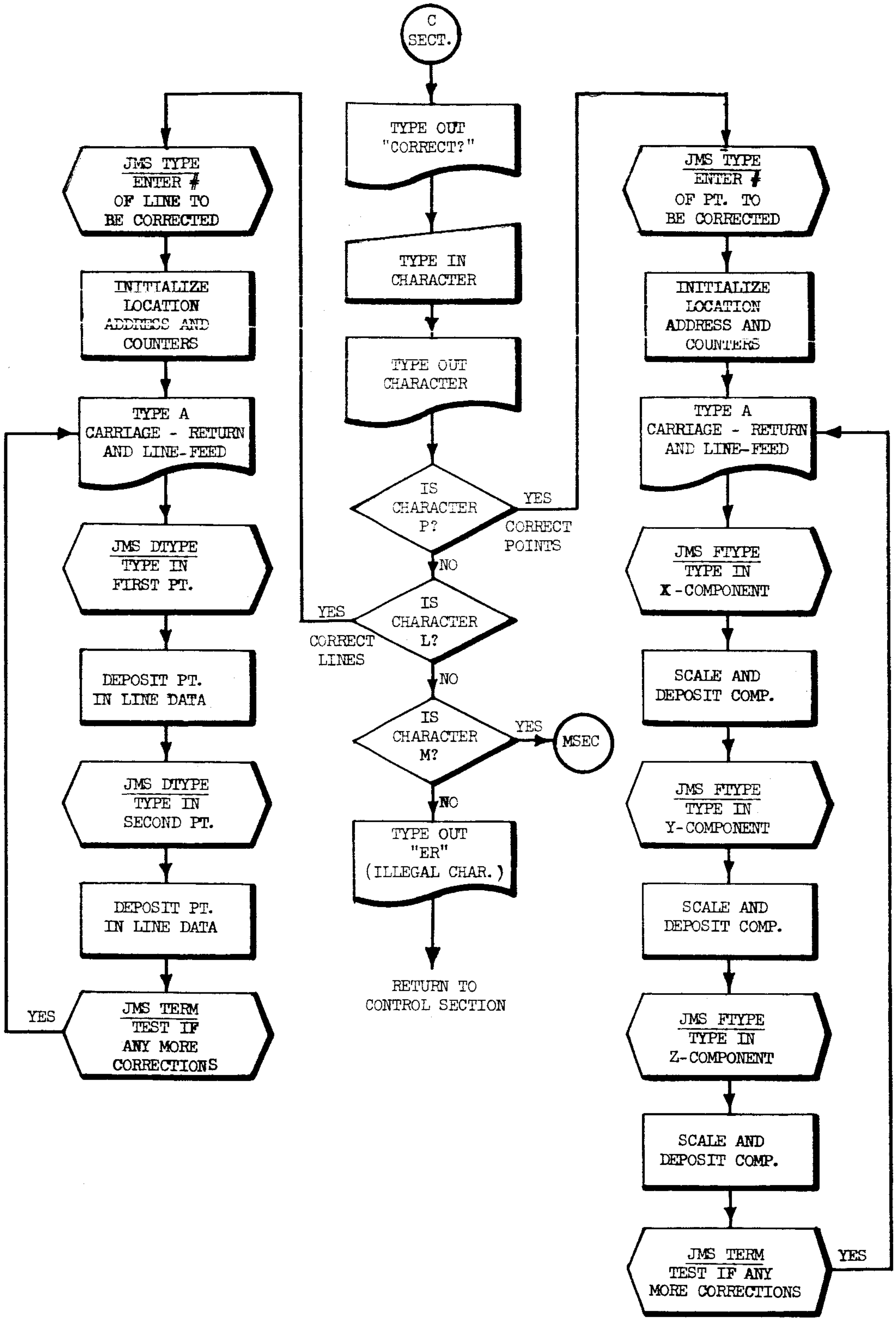
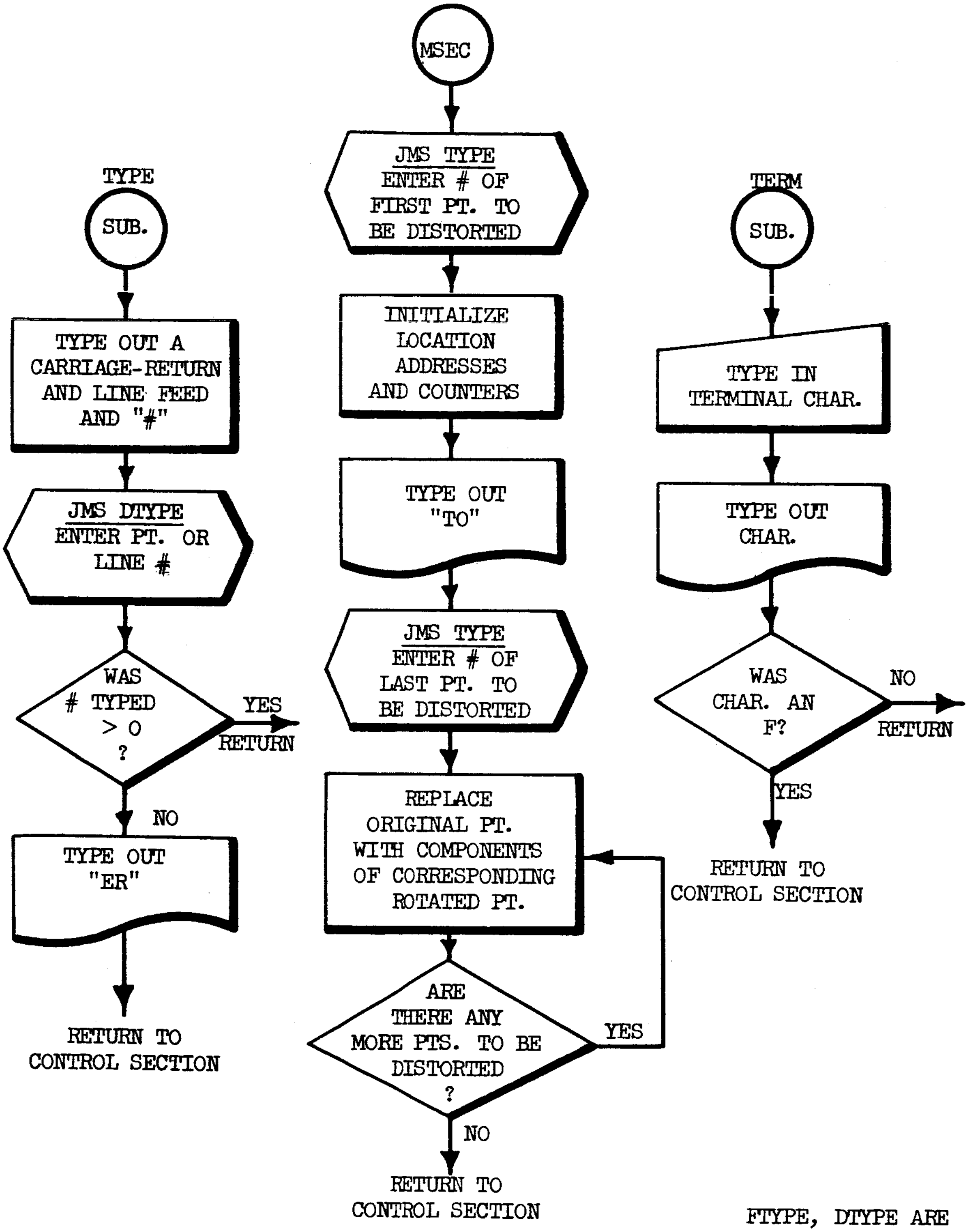


FIG. A-7

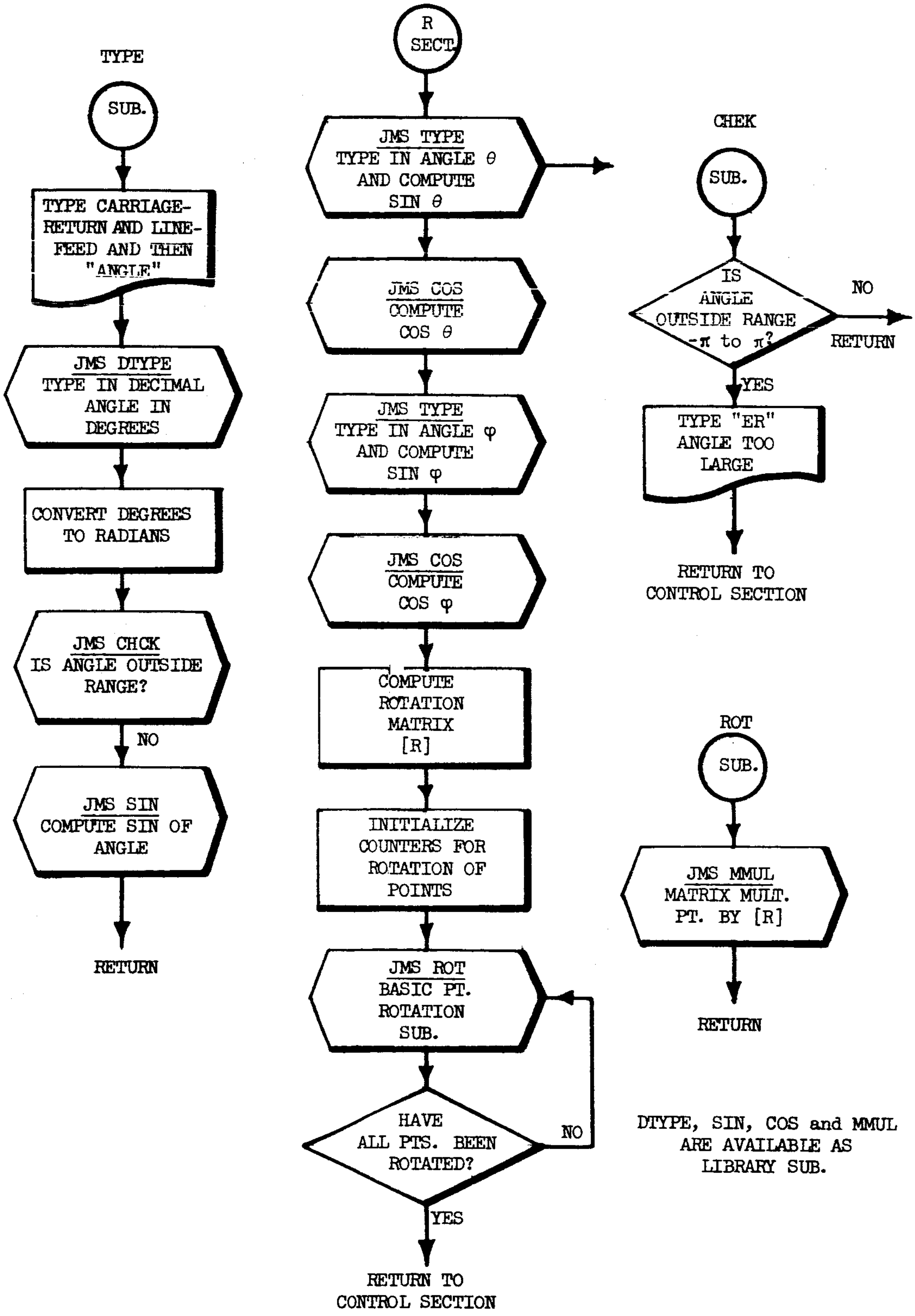
CORRECTION SECTION


MSEC
JMS TYPE
ENTER # OF FIRST PT. TO BE DISTORTED
INITIALIZE LOCATION ADDRESSES AND COUNTERS
TYPE OUT "TO"
JMS TYPE
ENTER # OF LAST PT. TO BE DISTORTED
REPLACE ORIGINAL PT. WITH COMPONENTS OF CORRESPONDING ROTATED PT.
ARE THERE ANY MORE PTS. TO BE DISTORTED ?
YES
NO
RETURN TO CONTROL SECTION
TYPE
SUB.
TYPE OUT A CARRIAGE-RETURN AND LINE FEED AND "#"
JMS DTYPE
ENTER PT. OR LINE #
WAS # TYPED > 0 ?
YES
RETURN
NO
TYPE OUT "ER"
RETURN TO CONTROL SECTION
TERM
SUB.
TYPE IN TERMINAL CHAR.
TYPE OUT CHAR.
WAS CHAR. AN F?
NO
RETURN
YES
RETURN TO CONTROL SECTION

FTYPE, DTYPE ARE AVAILABLE AS LIBRARY SUBROUTINES

F I G . A - 7

CORRECTION SECTION CONT.


TYPE
SUB.
TYPE CARRIAGE-RETURN AND LINE-FEED AND THEN "ANGLE"
JMS DTYPE
TYPE IN DECIMAL ANGLE IN DEGREES
CONVERT DEGREES TO RADIANS
JMS CHCK
IS ANGLE OUTSIDE RANGE?
NO
JMS SIN
COMPUTE SIN OF ANGLE
RETURN
R SECT.
JMS TYPE
TYPE IN ANGLE θ AND COMPUTE SIN θ
JMS COS
COMPUTE COS θ
JMS TYPE
TYPE IN ANGLE φ AND COMPUTE SIN φ
JMS COS
COMPUTE COS φ
COMPUTE ROTATION MATRIX [R]
INITIALIZE COUNTERS FOR ROTATION OF POINTS
JMS ROT
BASIC PT. ROTATION SUB.
HAVE ALL PTS. BEEN ROTATED?
NO
YES
RETURN TO CONTROL SECTION
CHEK
SUB.
IS ANGLE OUTSIDE RANGE -π to π?
NO
RETURN
YES
TYPE "ER" ANGLE TOO LARGE
RETURN TO CONTROL SECTION
ROT
SUB.
JMS MMUL
MATRIX MULT. PT. BY [R]
RETURN

DTYPE, SIN, COS and MMUL
ARE AVAILABLE AS
LIBRARY SUB.

FIG. A-8

ROTATION SECTION

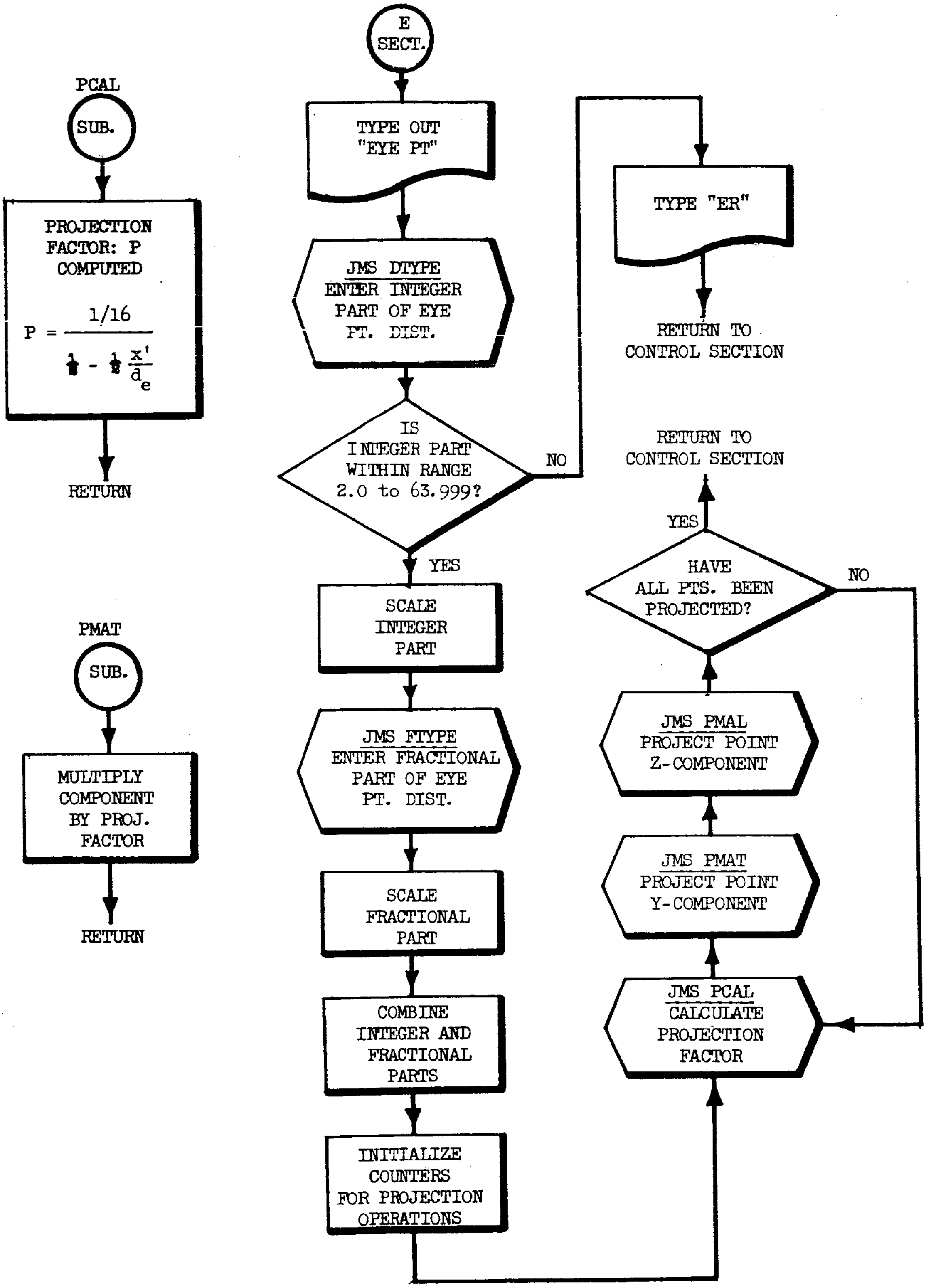

PCAL
SUB.
PROJECTION FACTOR: P COMPUTED
P = 1/16
1/[illegible] - 1/8 x'/d_e
RETURN
PMAT
SUB.
MULTIPLY COMPONENT BY PROJ. FACTOR
RETURN
E SECT.
TYPE OUT "EYE PT"
JMS DTYPE
ENTER INTEGER PART OF EYE PT. DIST.
IS INTEGER PART WITHIN RANGE 2.0 to 63.999?
NO
YES
SCALE INTEGER PART
JMS FTYPE
ENTER FRACTIONAL PART OF EYE PT. DIST.
SCALE FRACTIONAL PART
COMBINE INTEGER AND FRACTIONAL PARTS
INITIALIZE COUNTERS FOR PROJECTION OPERATIONS
TYPE "ER"
RETURN TO CONTROL SECTION
RETURN TO CONTROL SECTION
YES
HAVE ALL PTS. BEEN PROJECTED?
NO
JMS PMAL
PROJECT POINT Z-COMPONENT
JMS PMAT
PROJECT POINT Y-COMPONENT
JMS PCAL
CALCULATE PROJECTION FACTOR

FIG. A-9

PROJECTION SECTION

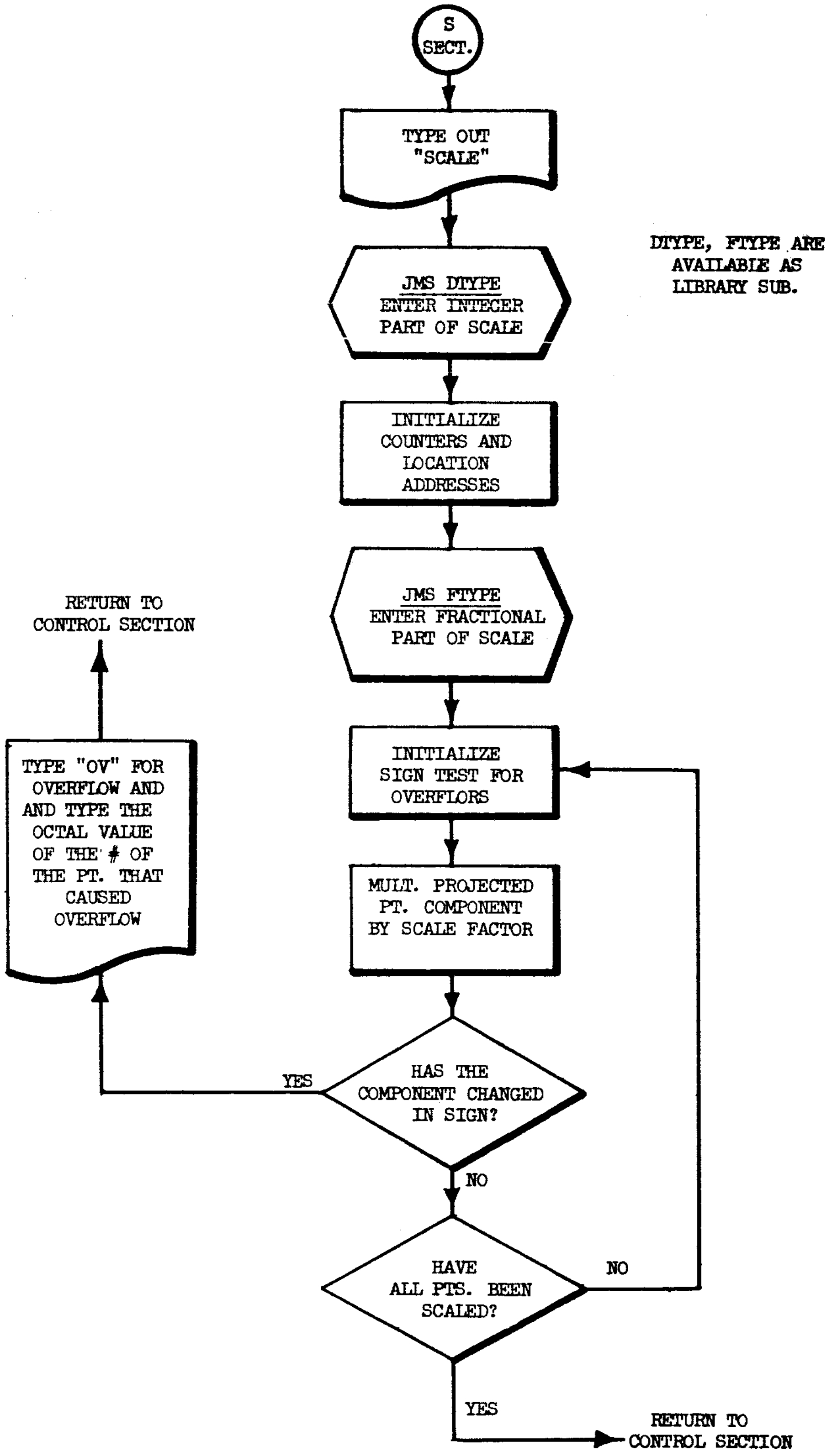

S SECT.
TYPE OUT "SCALE"
JMS DTYPE
ENTER INTEGER PART OF SCALE
DTYPE, FTYPE ARE AVAILABLE AS LIBRARY SUB.
INITIALIZE COUNTERS AND LOCATION ADDRESSES
JMS FTYPE
ENTER FRACTIONAL PART OF SCALE
RETURN TO CONTROL SECTION
INITIALIZE SIGN TEST FOR OVERFLORS
TYPE "OV" FOR OVERFLOW AND AND TYPE THE OCTAL VALUE OF THE # OF THE PT. THAT CAUSED OVERFLOW
MULT. PROJECTED PT. COMPONENT BY SCALE FACTOR
YES
HAS THE COMPONENT CHANGED IN SIGN?
NO
HAVE ALL PTS. BEEN SCALED?
NO
YES
RETURN TO CONTROL SECTION

FIG. A-10
SCALE SECTION

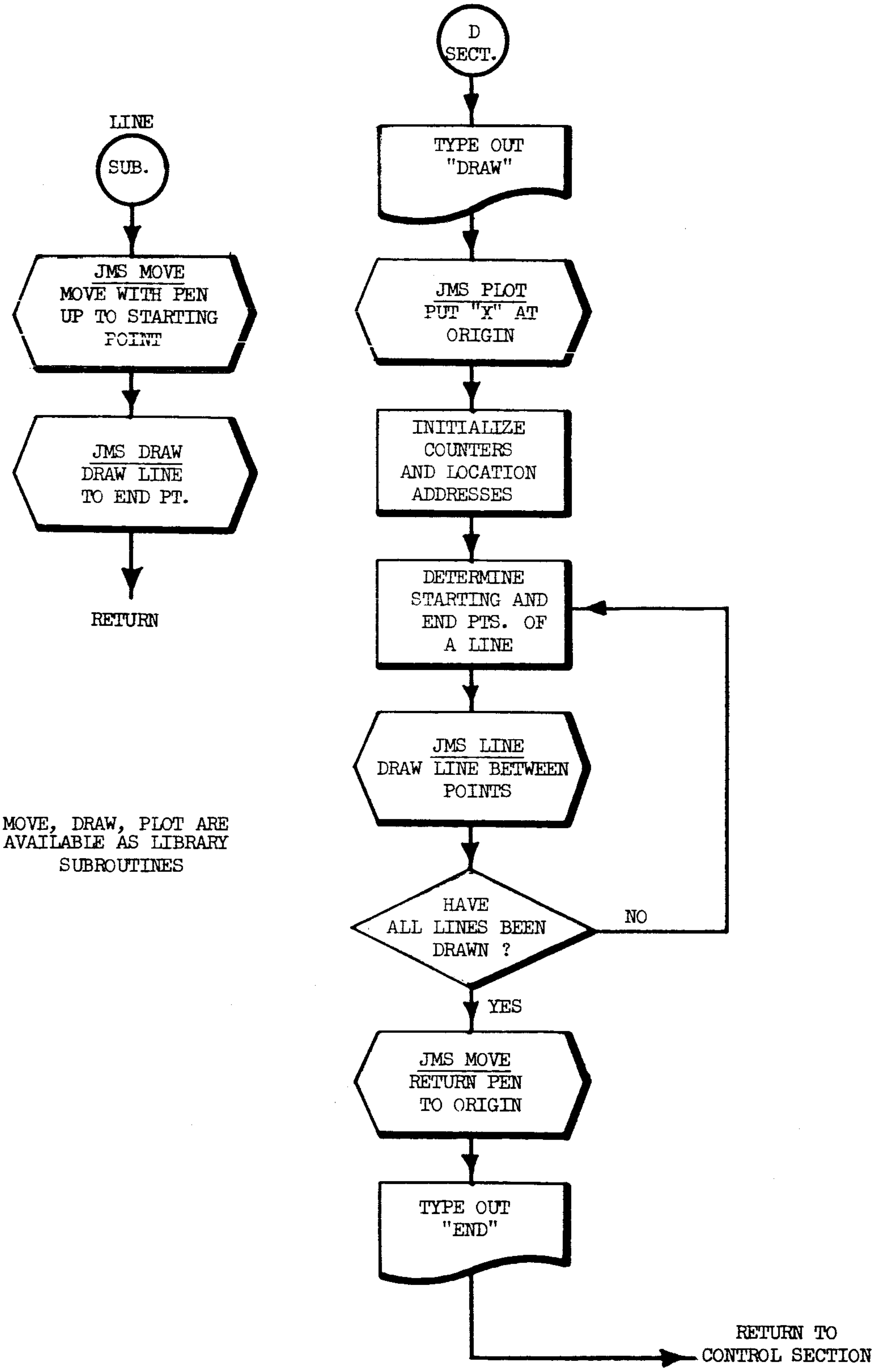

LINE
SUB.
JMS MOVE
MOVE WITH PEN UP TO STARTING POINT
JMS DRAW
DRAW LINE TO END PT.
RETURN
MOVE, DRAW, PLOT ARE AVAILABLE AS LIBRARY SUBROUTINES
D SECT.
TYPE OUT "DRAW"
JMS PLOT
PUT "X" AT ORIGIN
INITIALIZE COUNTERS AND LOCATION ADDRESSES
DETERMINE STARTING AND END PTS. OF A LINE
JMS LINE
DRAW LINE BETWEEN POINTS
HAVE ALL LINES BEEN DRAWN ?
NO
YES
JMS MOVE
RETURN PEN TO ORIGIN
TYPE OUT "END"
RETURN TO CONTROL SECTION

FIG. A-11
DRAW SECTION