COMPUTERIZED DRAWING OF ISOMETRIC PIPING DRAWINGS

FEASIBILITY STUDY ON COMPUTERIZING THE DRAWING OF ISOMETRIC PIPING DRAWINGS

BY

RUDOLF ERNEST PAULS

A Thesis

Submitted To The Faculty of Graduate Studies

in Partial Fulfillment of The Requirements

For The Degree

Master of Engineering

McMaster University

February, 1968

MASTER OF ENGINEERING (1968) McMASTER UNIVERSITY
(Mechanical) Hamilton, Ontario

TITLE: Feasibility Study on Computerizing the Drawing of Isometric Piping Drawings

AUTHOR: Rudolf Ernest Pauls, B.S.A., University of Guelph

SUPERVISOR: Dr. G. Kardos

NUMBER OF PAGES: vi, 103

SCOPE AND CONTENTS: A feasibility study has been made on computerized drafting. In this study a system was developed for computerizing the production of isometric piping drawings. The software developed for this system is described in detail and computer programme listings included.

A comparison between the system developed by the author and a similar system independently developed by the Sun Oil Company has also been made. Recommendations are given for extensions to the system to include the computerized production of material list and parts requisitioning.

The computerized production of isometric piping drawings was found to be competitive in cost with, and as easy to read as manually produced isometric piping drawings.

Table of Contents

APPENDICES

List of Illustrations

List of Tables

Acknowledgments

To Dr. G. Kardos, McMaster University, for his constant help and suggestions throughout the project.

To Prof. W.R. Newcombe, McMaster University, for suggesting the topic of study and subsequent help in obtaining the necessary information.

To Ford Foundation, for the financial assistance without which such a study could not be carried out.

To Mr. E.A. Goodwin, D.H.O. Toronto, for the use of their EAI 3500 plotter.

To Mr. W. Nuss, D.H.O. Toronto, for his help in getting the program operational.

To Mr. A. McKenzie, Beamer & Lathrop, for his help in the technical aspects of isometric piping drafting.

To Mr. J.R. Aicher and Mr. E.L. Peacock, Sun Oil Co., Philadelphia, Pa., for their hospitality and cooperation in showing me their computer drafting operation.

To Miss S. Jackson, C.G.E., for her expertise in typing the manuscript.

I ABSTRACT

A feasibility study has been made on computerized drafting. To yield an insight into the computerized drafting problem, a system has been developed for making isometric piping drawings.

The equipment used consists of an IBM 7040 computer and an EAI 3500 plotter. A programme has been prepared for generating a three-dimensional model of the piping as well as its associated dimensioning and labelling. In this programme, the various symbols used have been preprogrammed and are called for by coded input characters. At the same time the associated dimensioning is automatically performed. The output of this programme, consisting of three-dimensional coordinates of line-segment end-points, is punched out on cards. These data cards are then fed into the computer in conjunction with the GRUMMAN AIRCRAFT rotational programme DRAFT which has been extensively modified for this particular application.

The operation of DRAFT, in this case, is to rotate the three-dimensional model into the isometric orientation and then mathematically project each point onto the picture plane. Instructions are then generated and stored on tape, for plotting this two-dimensional drawing of the piping.

This method for producing isometric piping drawings was found to be competitive in cost with, and as easy to read as manually produced piping drawings.

II INTRODUCTION

During the present century, the computer has evolved to become a most useful scientific and engineering tool. Its high speed and large storage capacity make it readily adaptable to many uses. One of its uses, the one which is explored in this study, is that of controlling an incremental plotter.

The computer controlled incremental plotter has been used for many purposes. Both the Boeing Aircraft Corp. [1] and the Grumman Aircraft Corp. [2] use the computer-plotter in preparing two-dimensional drawings of three-dimensional subjects. Examples of these are perspective views of aircraft landing approaches, aircraft and automobiles. At the Ontario Department of Highways [5], highway cross-sections and elevations of cuts and fills are automatically plotted from the surveyed data.

The purpose of this project is to make a feasibility study on the use of the computer controlled incremental plotter in drafting. Having chosen the area of study, the search for a suitable project was begun.

It was realized that a time consuming and thus costly drafting procedure was that of drawing three-dimensional objects. This field offered a substantial savings in cost, if the drawing of three-dimensional objects could be automated. It was decided to use existing software for producing the two-dimensional drawing of the three-dimensional model.

A necessary feature for keeping the automated drawing competitive in cost with the manual drawing is repetitive drawing procedure. In the automated drafting of three-dimensional objects, one of the major problems is the automatic removal of hidden lines (see Appendix A). The

automated production of isometric piping drawings was found to have repetitive drawing procedure and at the same time by-passed the hidden line problem. This project also presented the general drafting problems of scaling, dimensioning and locating. For these reasons, the field of isometric piping drawings was chosen to study computerized drafting.

III AUTOMATED DRAFTING OF ISOMETRIC PIPING

The rotational programme DRAFT, Grumman Aircraft programme [3], was studied for adaptation to isometric piping drawings. This programme was designed to perform certain preprogrammed operations on three-dimensional data. The operations that could be performed were: (1) rotation about any of the principal axes, (ii) translation along any principal axis and (iii) projection into two dimensions in either perspective or projection. By the use of the proper control cards, any sequence of the above operations could be performed. In the problem of isometric piping drawings, the required operations are rotation (into isometric form) and projection onto the two-dimensional plane. By the use of the proper control cards, these operations could be performed.

Once the output phase had been planned, the remaining problem was to generate the three-dimensional model of the piping drawing. The restrictions on this phase were that the amount of input data should be as small as possible and as simple as possible. This would allow a draftsman to pick the information from orthographic prints in a short length of time, and thus keep the cost of labour down. Piping drawings normally consist of a series of pipes with various fittings inserted between them.

The input data for this type of drawing could consist of: (1) three-dimensional coordinates of end-points of straight lines or (2) the direction and length of lines from which the computer would have to generate the coordinates. The first method requires the operator to manually compute running sums of the coordinates, allowing room for error as

well as increasing the manual labour involved. With the second method, only the direction of the line and its length (as given on the orthographic print) need be specified. The computer can then be programmed to perform the additions and store the running sums of the coordinate directions. For these reasons, the second method was chosen.

The computer programme suggested above was developed by the author and subsequently given the name "Configuration Generator Programme". In this program, the x, y and z coordinates are initially set to (0,0,0); then the line lengths are added sequentially in the appropriate coordinate directions. This results in a generated model which will be scaled to the actual dimensions (except for the standard piping symbols which are all drawn according to a preset scaling factor). After performing a sample run, it was found that the drawing became disproportioned due to the scaling of extremely long lines. This caused the drawing to have most of the detail crowded into one or two areas with a long line across most of the page, adding nothing to the clarity of the drawing. At the same time, short lines, drawn to the same scale, could not be dimensioned without interfering with adjacent lines and fittings could not be inserted in them. For this reason, maximum and minimum lengths were defined, and lengths beyond this range were automatically drawn to the length given by the upper or lower limit.

As stated previously, one of the aims of this project was to produce computer drawings closely resembling manually produced drawings. In some manually produced isometric piping drawings, arcs are drawn between adjacent perpendicular lines to depict 90° bends. This was attempted with the computer-plotter (see Appendix F). This operation worked

satisfactorily. It required, however, a large amount of storage, increased the plotter time quite considerably, and did not add appreciably to the clarity of the computer produced drawings. For these reasons, this sub-programme was deleted.

Another problem encountered was that of dimensioning the drawings. Since the drawing of dimension lines is a repetitive procedure, the possibility of doing this automatically was explored. A typical dimension line for dimensioning the line AB is shown in Figure 1.

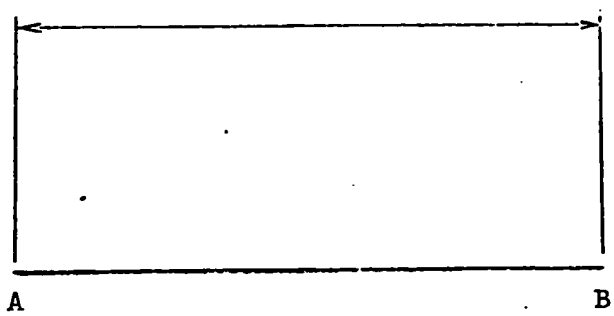


Figure 1. Typical Dimension Line

The generation of these lines can be programmed provided the following information is known: the points A and B which are the starting points of the dimensioning, the plane of the dimensioning and the side of AB on which the dimensioning is to occur. The points A and B are merely the end points of the line or symbol to be dimensioned. The logic has been built into the programme to automatically store the coordinates of

these points; then, if dimensioning is called for, the dimension lines will be automatically generated. The plane of the dimensioning and the side on which the dimensioning is to appear may either be specified by the user or logic built into the program to determine this.

The criterion for placing dimension lines is that they may not interfere with other parts of the drawing. The logic to determine this would have to check all the end point coordinates as well as the lines to be drawn between these coordinates to see if they will interfere with dimensioning. This type of thing is very complicated to programme. For this reason, the method was chosen whereby the user must specify the plane of the dimensioning as well as the side on which it is to appear. This method worked very well and resulted in easy to read drawings with a minimum of input data.

Once the dimension lines were drawn, the labeling had to be transferred to the paper. Logic was built in, to automatically choose the starting point at the proper position relative to the dimension line. Several alternatives for transferring the dimensioning information to the computer-plotter were studied. The EAI 3500 plotter had the capability of printing with the letters in either the vertical or horizontal orientation, or plotting the characters with the pen. An experiment was performed, using the printer head, and printing the characters along the isometric axes. This meant having succeeding characters offset both vertically and horizontally. It was found that the printing was nearly impossible to read and this method was subsequently abandoned (see Figure 2).

Attention was then shifted to plotting the alphameric information with the pen. The capabilities of the EAI 3500 plotter are described in

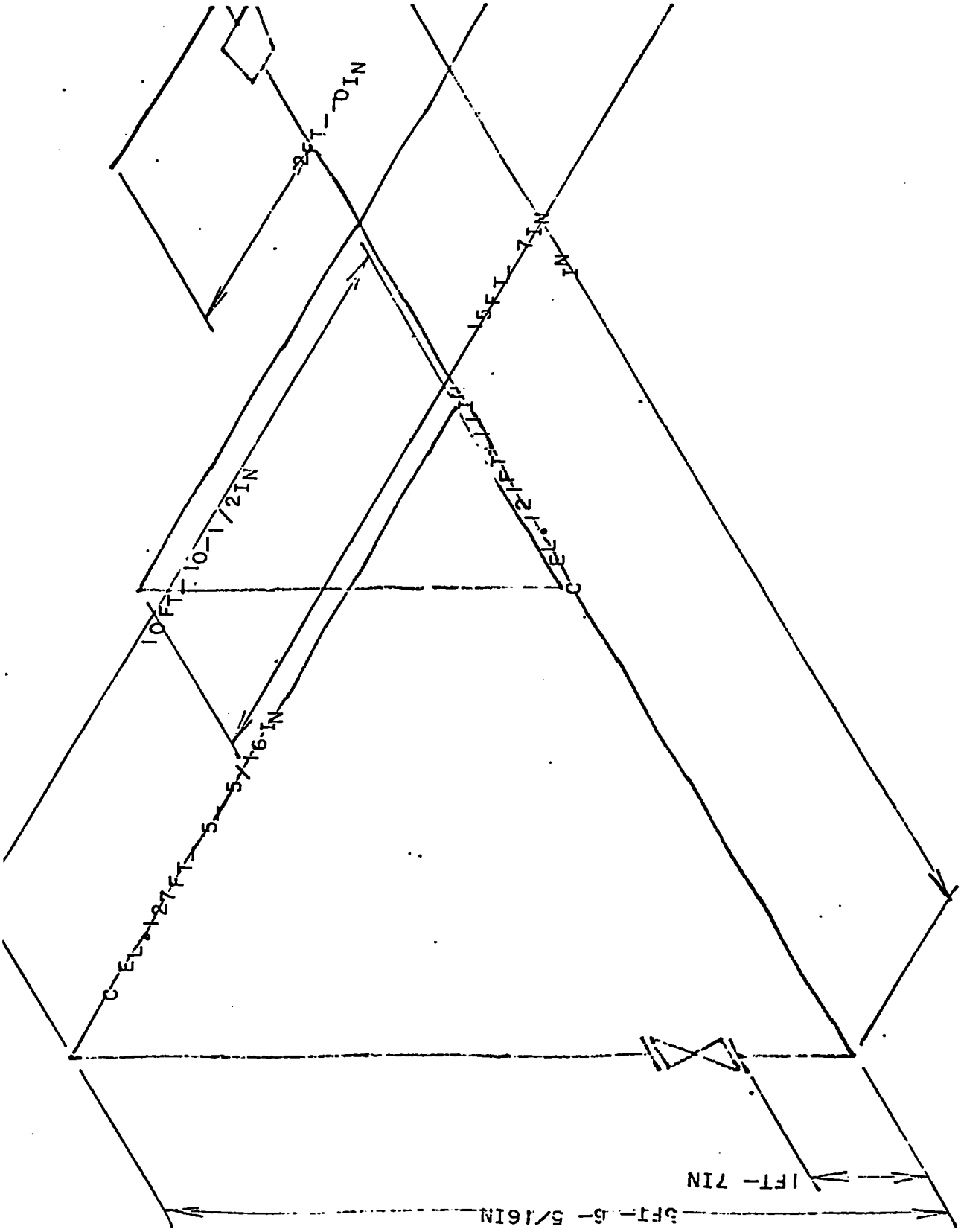


Figure 2. Experiment on Using the Printer in Dimensioning Isometric Piping Drawings

the manual [4]. The subroutine SCRIBE has the capability of plotting the data along any base angle desired and the characters having any slant desired. The user can also specify the height of characters. The necessary logic was written in computer language to automatically set these variables (see Appendix B,6). This method of dimensioning was found to work quite satisfactorily and was used in the computer plot reproduced in Figure 4.

As stated previously, a real saving can be realized with the computer-plotter when the drawing procedure is repetitive. Isometric piping drawings consist of pipes and simple symbols representing pipe fittings which appear again and again. Logic for generating these piping symbols was built into the Configuration Generator Programme. This logic was written so the piping symbols could be generated in any plane and direction desired. The user's input required to generate the piping symbol consists of: coded characters specifying a particular piping symbol, its direction and its plane. The programme then automatically generates the piping symbol in the desired orientation and plane.

Several alternatives existed for generating the piping symbol in its proper location. Two of these are described below: (1) call for generation of the main line, then the piping symbol and then continue the main line to its end, and (2) generate all the main lines first and then insert the piping symbols in the appropriate lines. With the first method, the coordinates of the piping main lines as well as the coordinates of piping symbol lines would be stored in the same array. Because different symbols require different numbers of straight line segments it would be impossible to choose a particular line by its storage location. With the second method, however, each line requires two rows of storage,

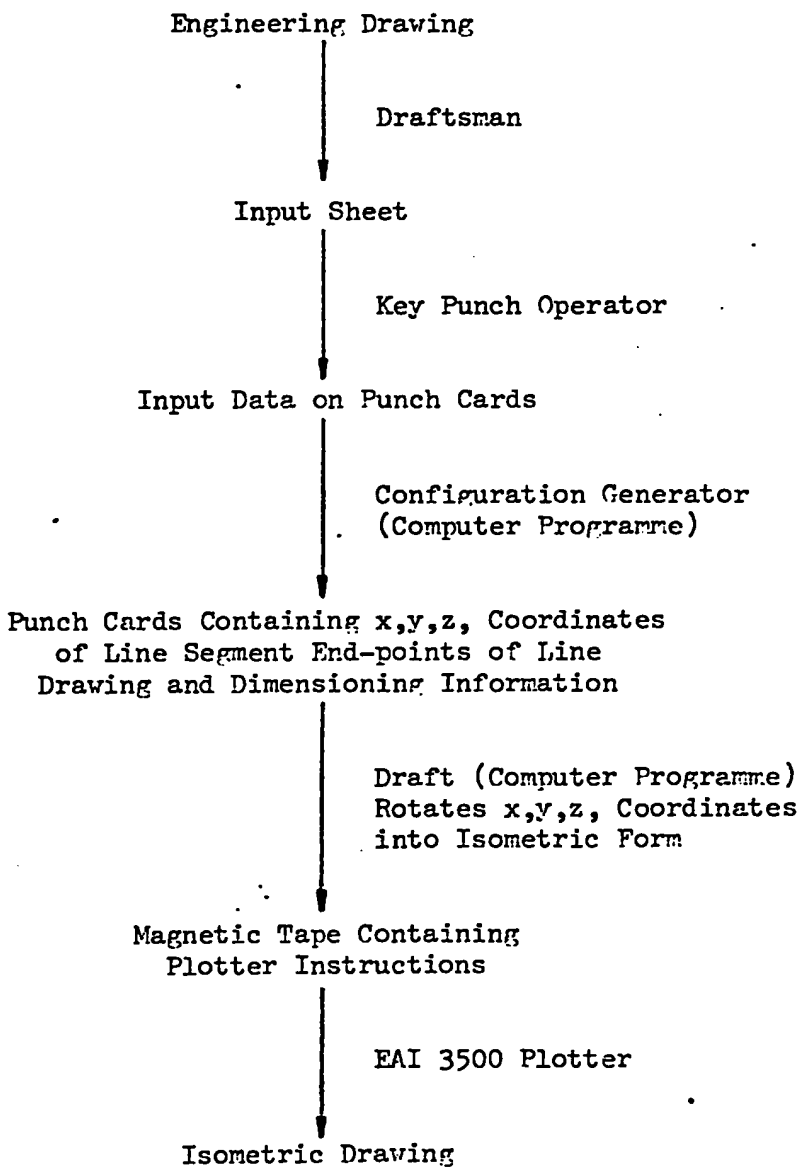
making it possible to choose the coordinates of a particular point by its index. The index method would then also be used to choose the proper line for the insertion of a piping symbol. The piping line would then have to be broken, the symbol inserted and the line continued to its end. A further advantage of the second method is that the drawing would not become unproportioned due to the drawing of unscaled piping symbols. Due to these advantages, the second method was chosen. This logic is more fully described in Appendix B.4.

After the Configuration Generator Programme became operational, the first trial run was performed. The output data from the Configuration Generator Programme, along with appropriate control cards, was used as input for the modified programme DRAFT. At this point, various other plotter instructions like scaling, location of origin and plot size were punched in manually. The output of this programme was on magnetic tape, which was then used directly with the EAI 3500 plotter. Upon completion of the plotting, it was realized that the scaling and origin location had been chosen quite arbitrarily and that they were far from optimum. A subprogramme was written which would scan the data after rotation into isometric form and compute the best scale factors and origin coordinates (see Appendix C.2).

Figure 1 shows the operations necessary for producing isometric piping drawings by computer graphics. The cost involved will depend largely on the experience of the operators. A typical piping drawing, as shown in Figure 4, required approximately the time and cost shown in Table 1. The input data is shown in Table 2.

Figure 3

Flow Chart for Generating Isometric Piping Drawings by Computer Graphics



IV CONFIGURATION GENERATOR PROGRAMME

The purpose of the Configuration Generator Programme is to generate a three-dimensional model of the piping drawing, complete with dimension lines and labeling instructions. The programme may be divided into four different sections, according to the operation performed. Their description and operation is given below:

(i) Input Section - In this section, the coded input information is read in. The coded characters are translated into computer instructions and control of the programme transferred to the proper section.

(ii) Catalogue - This section of the programme consists of preprogrammed instructions for generating standard piping symbols. Coded information, transferred from the input section, instructs the computer to place this symbol in a certain location and orientation as desired by the programmer.

(iii) Dimensioning and Labeling - When the programme exits from the generation of a piping symbol, control is transferred to the dimensioning and labeling section. Logic has been programmed into the computer to automatically generate the dimension lines and arrow heads (the only input required here is a code saying on which side of the piping symbol dimensioning is to appear). At the same time, instructions are generated for writing the length immediately above or below the dimension lines, as determined by the computer logic.

All this information is stored and control returned to the input section. Other symbols or lines may follow. After the last item has been generated, control is transferred to the output section of the programme.

(iv) Output Section - During generation of the main lines, symbols, dimension lines and alphameric data, the computer was set up to allow separate storage for each type of drawing operation. This involved allotting six storage regions (see Appendix B,5 for a fuller description of the storage). During output, all the stored data is punched out on cards, one section at a time, with the necessary plotter control cards automatically included.

This system of storage and output was chosen to keep the number of pen and plotter mode changes to a minimum during the drafting operation. For a full description of the output phase of the Configuration Generator Programme, see Appendix B,7.

V MODIFICATIONS TO THE PROGRAMME DRAFT

The programme DRAFT was an IBM 7094 programme written by the Grumman Aircraft Corporation [3] for producing perspective drawings with a CALCOMP digital plotter. This programme had to be modified: (1) to be compatible with the McMaster University IBM 7040 computer, (2) for producing drawings with an EAI 3500 digital plotter, and (3) for producing projected isometric drawings rather than perspective drawings.

The logic used in the programme DRAFT was to read in three-dimensional coordinates representing a three-dimensional object, rotate the object through any angle or angles desired, and project it onto two dimensions. These two-dimensional points were then plotted with straight lines drawn between them to represent a two-dimensional picture of the three-dimensional object.

The modifications which were made to the programme DRAFT are described in Appendix C.2.

VI COMPARISON WITH ANOTHER AUTOMATED SYSTEM FOR PRODUCING ISOMETRIC PIPING DRAWINGS

In the fall of 1966, after the first isometric piping drawings had been produced, using the above described system, the author became aware of another computerized drafting operation under development at the Sun Oil Company [2]. In November 1966, the author had the pleasure of visiting the Sun Oil Company office in Philadelphia, Pa. and of having discussions with the people who originated the computerized drafting operation. Although the company policy did not allow any of the actual programming or theory involved to be divulged, the visit turned out to be very interesting and rewarding.

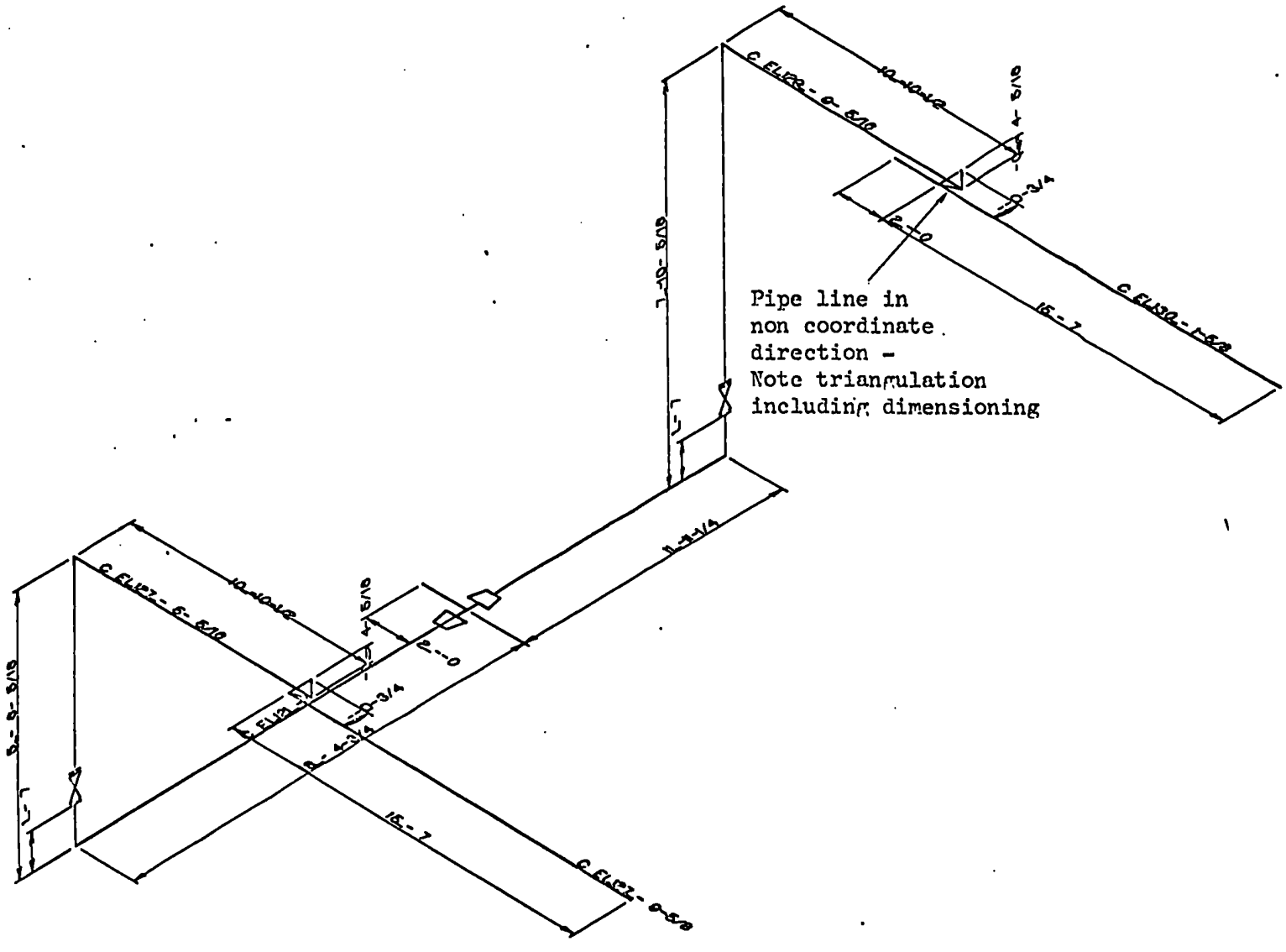
A brief description of Sunoco's computerized drafting syster follows (for a complete description, see Ref. 2). The approach used by Sunoco is to store instructions for plotting isometric views of all the piping symbols on a random access disk. Then, during running time, the computer merely chooses the correct view from this canned information and feeds the plotting instructions to the plotter. In preparation of the disk storage, a planar view of the symbol is drawn manually and 48 different isometric views of the symbol generated by a rotational programme. The plotter instructions for plotting these views of all the symbols are stored on a random access disk.

At the time of the author's trip to Sunoco, the equipment used was an IBM 1620 computer and an IBM 1627 plotter. The plotter was run on-line, requiring a fair amount of computer stand-by time while waiting for the plotter. They were, however, in the process of changing to an

IBM 360 computer and the same IBM 1627 plotter. Here the plotter would still be on-line, but through multiprogramming (simultaneous peripheral operations on-line) other programmes could be run co-currently. The main limitation of the IBM 1627 plotter compared to the EAI 3500 plotter is the one-pen output capability. With the EAI 3500 plotter, a choice of eight pens is available upon user's command as well as a 48 character printer and a more versatile subroutine (SCRIBE) for plotting alphameric information.

The main difference in operation between the two systems is that with the Sun Oil system the operator must choose the particular isometric view to be used for each symbol and manually choose each line to be drawn in dimensioning. With the system described in this report, however, the programme automatically performs the dimensioning and rotates the model after it has been generated. Also, all the labeling performed on the Sun Oil drawings could only be done either horizontally or vertically, while, with our system, the labeling could be done in the isometric planes.

Inherent in the two systems, the Sun Oil Company system will be faster, thus less costly in computer and plotter time, while our system will require less input data and produce more easily read drawings (compare Figures 4 and 5).



Produced by System Described in this Report

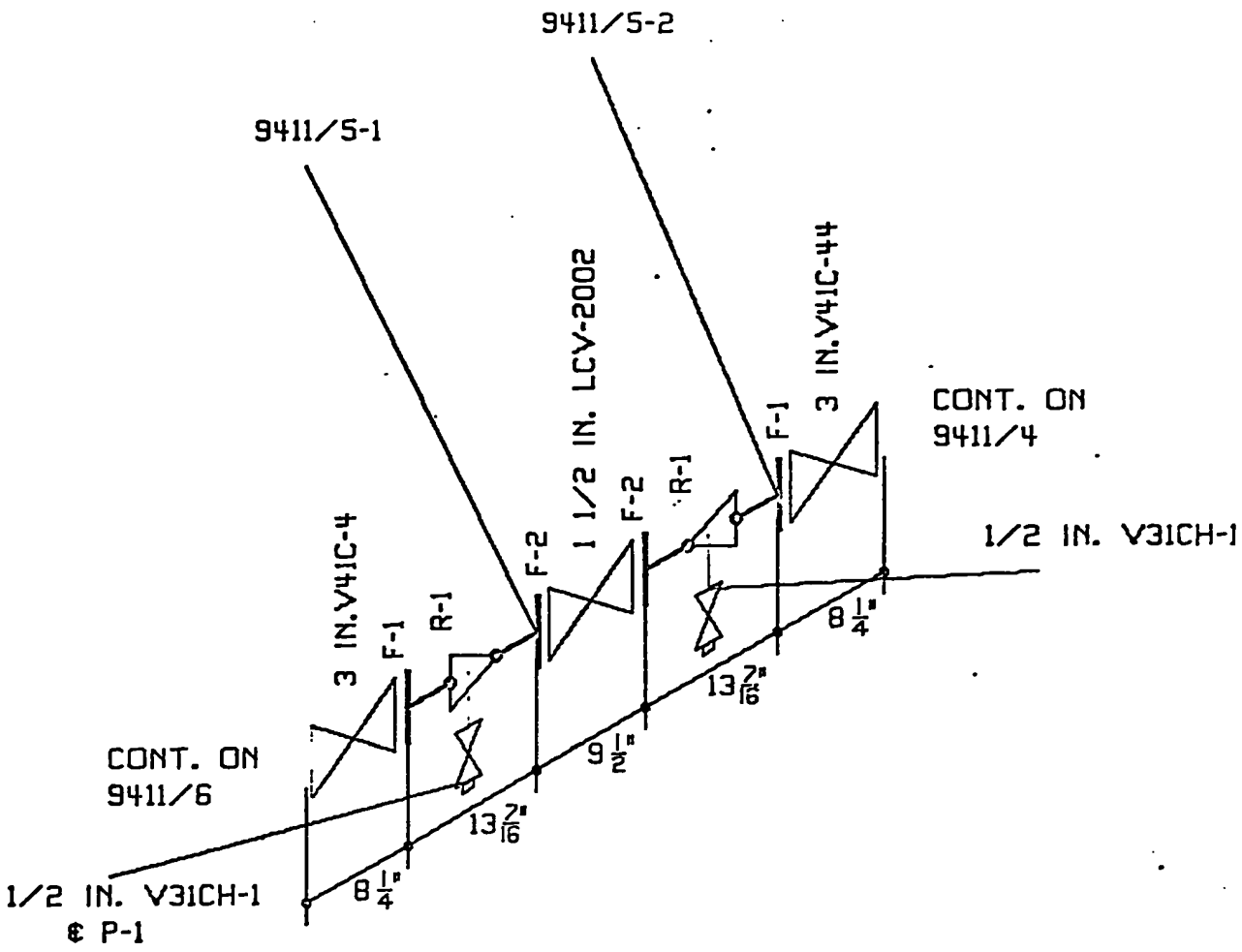


Figure 5. Isometric Piping Drawing Produced by the Sun Oil Co., Automated Drafting System

VII ESTIMATED COST

The input data necessary to produce Figure 4 is shown on page 20. The draftsman's time involved in preparing the input data will vary considerably with his experience and familiarity with the programme. However an estimate of the time involved can be made from the author's experience. Approximately half an hour should suffice, depending on how complicated the drawing is. In the production of Figure 4 approximately 40 seconds of computer time were required by the Configuration Generator Programme and 30 seconds by the programme DRAFT. The EAI 3500 plotter required approximately 15 minutes.

As seen in Table 1, the cost of producing isometric piping drawings by computer-plotter is approximately $16 per drawing. This is based on a typical drawing as shown in Figure 4. The cost of manually producing isometric piping drawings is approximately $25 per drawing. This is an average cost based on the experience of McKenzie [7] at Beamer and Lathrop. Thus a quite favourable cost advantage may be realized by computerizing the production of isometric piping drawings.

Description of Operation	Time	Approximate Cost Rate	Cost, $
Draftsman's Preparation of Input Sheet	$\frac{1}{2}$ hr	$ 4 per hr	2.00
Key Punch Operator	10 min	$ 3 per hr	.50
Computer Time (IBM 7040)		$320 per hr	
Configuration Generator	40 sec		3.56
DRAFT	30 sec		2.66
EAI 3500 Dataplotter	15 min	$ 28 per hr	7.00
Total Cost			15.72

Table 1. Cost of Producing Isometric Piping Drawings via Computer-plotter

Operations Performed	Number of Line	Direction		Scaled Length				Plane		Symbol for Welded Joint	Unscaled Length	Triangulation Signal
NN	KVAL	I	ASIGN	FEET	AINCH	ANUM	DENOM	J	SIG	DOT	DIST	MTRI
1	1	1	+1	2				2	+1			
1	2	2	+1	18	4	3	4	1	+1			
1	3	3	+1	5	6	5	16	3	+1			
1	4	1	+1	10	10	1	2	2	-1			
1	5	3	+1		4	15	16	2	-1			
1		2	+1			3	4	1	+1			1
1	6	1	+1	15	7			2	+1			
1	1	2	-1	11	11	1	4	1	+1			
1	8	3	+1	7	10	5	16	2	+1			
1	9	1	+1	10	10	1	2	2	-1			
1		3	+1		4	15	16	2	-1			
1	10	2	+1			3	4	1	+1			1
1	11	1	+1	15	7			2	+1			
1	10	1	-1	2				2	+1			
4	2	2	+1					1				
4	7	2	-1					1				
3	3	3	+1	1	7			2	+1			
3	8	3	+1	1	7			2	+1			
2	2	2	+1	121	11			1			4	
2	4	1	+1	127	5	5	16	2	-1		1	
2	6	1	+1	127	9	5	8	2	+1		12	
2	9	1	+1	129	9	5	16	2	-1			
2	11	1	+1	130	1	5	8	2	+1		8	

Table 2. Input Required for Producing Figure 4.

VIII CONCLUSIONS

The method of producing computerized isometric piping drawings, described in this thesis, was found to be quite attractive from an economic point of view (see Table 1). The drawings produced, as shown in Figure 4, are nearly as easy to read as manually produced drawings [7]. With the present situation of shortage of skilled labour, the computerized production of isometric piping drawings is a quite attractive proposition. A fuller potential can be realized by computerizing the production of the material list and of the requisitioning, thus eliminating more time consuming, error prone human operations.

The feasibility study has shown that it is possible to economically produce isometric piping drawings via the computer-plotter.

IX RECOMMENDATIONS

The computerized drafting system for producing isometric piping drawings, described in this report, is by no means complete. As stated in the introduction, operations which are repetitive in nature are easily adapted to automated procedure and inherently have the greatest potential for being economically justified.

Two such operations, which the author has not included due to lack of time, are those of automating the production of the material list and parts requisitioning. These could easily be incorporated in the present Configuration Generator Programme and printed out on the final drawing. A fuller description is given in Appendix D.

Another development which would increase the versatility of the Configuration Generator Programme would be a rotational subroutine for generating the built-in symbols and dimensioning in other than coordinate directions. At the present stage of development, the programme can generate main lines in other than coordinate directions by triangulation. If whole sections of piping appear in non-coordinate directions, they cannot be handled by the present system. A fuller description of this subroutine is given in Appendix E.

APPENDIX A

Experiment on Automatic Removal of Hidden Lines

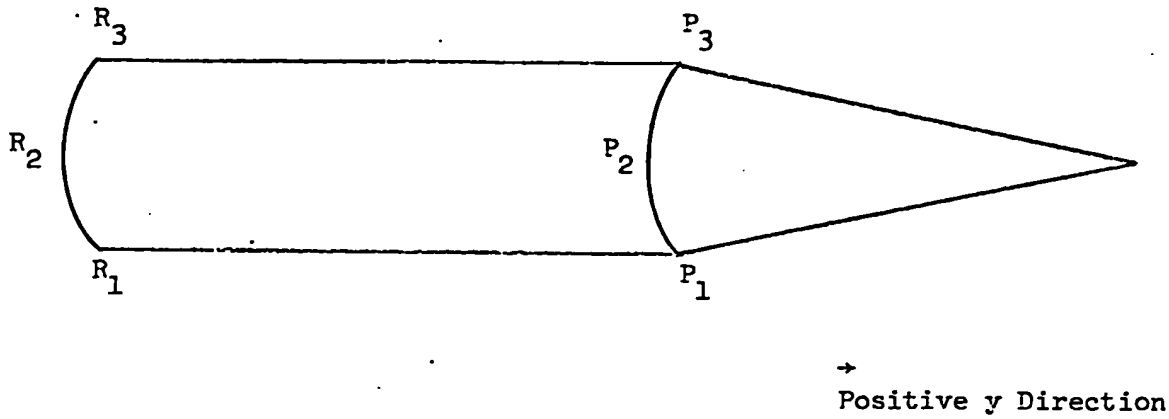
In order to become familiar with the problems involved in the automatic removal of hidden lines, the following experiment was performed. The experiment was suggested by Dr. G. Kardos, of McMaster University, on the basis of his experience with the project HARP (High Altitude Research Project) at McGill University. This project involves the projecting of rockets from a large gun. Each rocket is equipped with a control system for controlling the rocket's attitude. A computer analysis of the rocket's behaviour, and from this plots of pitch angle versus yaw angle with time as the independent variable, were performed by McGill University.

From these plots, the author attempted to produce an animated movie of the rocket's behaviour with the use of the computer-plotter. To this end, a simple rocket model was defined in three-dimensional co-ordinates and read in by the modified programme DRAFT. Pitch and yaw axes were defined about the rocket's centre point. Then a series of pitch and yaw angles was read in by DRAFT, the rocket rotated about the proper axes, and drawn in projection. The next step was to expose these plots successively on movie film and thus produce an animated sequence portraying the motion of the rocket.

At this point, it was realized that it was impossible to differentiate between angles into and out of the picture plane (due to hidden lines being drawn). To overcome this defect of the automated drafting, it was necessary to automatically remove the hidden lines (see Figure 6).

A characteristic of the drawing procedure, because of the sequence of input of the data, was that the circumferential lines P_1, P_2, P_3, and R_1, R_2, R_3 and R_4 were always drawn in the same direction. As seen from Figure 6(i), the line R_3, R_4, R_1 is hidden and should therefore be removed when P_2 has a smaller y-coordinate than a straight line between P_1 and P_3. This logic was chosen for the automatic removal of the hidden line.

This experiment on hidden line removal worked quite well in this simple case, but at the same time it pointed out the complexity involved in the automatic removal of hidden lines in the general case. For this reason a project was chosen where the automatic removal of hidden lines was not required.



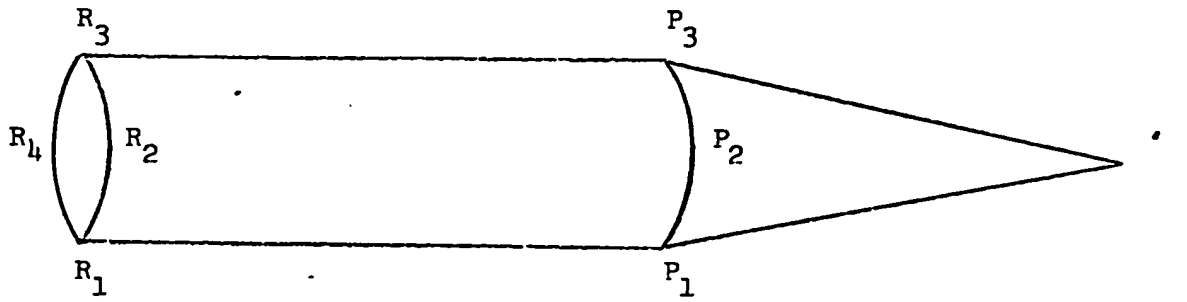


Figure 6. Experiment on Automatic Removal of Hidden Lines (i) Rocket at Angle Out of Paper and (ii) Rocket at Angle into Paper

APPENDIX B

Configuration Generator Programme (Detailed Description)

In Section IV, a general description of the Configuration Generator Programme has been presented, broken down by function. Since the present discussion is a more detailed, computer oriented description, the programme will be broken down by sections, as they appear in the actual computer programme. Since the logic involved is quite complicated and difficult to describe by flow charts, only simple flow charts are given (see Figure 10), and the logic is more fully described in words and figures. The programme listing, included in Section 10 of this appendix, will serve as the most inclusive reference to the logic used in the programmes.

B.1 Input

Because the digital computer is a numerical input device, the description of the drawing must be coded into numbers. Since all symbols and other operations performed on the various drawings are similar, it became possible to use the same list of arguments for each item to be drawn. This simplified the input section considerably and allowed an input sheet to be drawn up.

Following is the list of arguments, in the order in which they are read in by the input section of the Configuration Generator Programme. The whole list of arguments pertaining to one symbol appears on one line of the input sheet, as shown in Table 2.

NN - A control variable coded to the various symbols and other operations which may be performed.

KVAL - A coding number associated with each line or symbol for identification purposes. All lines are numbered sequentially in the order in which they are generated. The coordinates of the starting point of branch lines and piping symbols are then chosen by the value of KVAL.

I - An integer value of 1, 2 or 3 which specifies the axial direction of the line or symbol with directions as shown in Figure 7.

ASIGN - A variable referring to 'I'. It may have a value of +1. or -1. which yield the positive and negative directions, respectively, as defined in Figure 7.

FEET, AINCH, ANUM, DENOM
- These variables specify the length in feet, inches and fraction of an inch, respectively. In the dimensioning section, these values appear in the labeled length.

J - An integer value of 1, 2 or 3 which specifies the plane in which, the piping symbol or dimensioning is to appear with directions as shown in Figure 7.

SIG - A variable referring to 'J'. It may have a value of +1. or -1. which place the dimensioning on the positive or negative side of the line or piping symbol.

DOT - A control variable to instruct the computer to generate the symbol for a welded joint hereafter called a welding dot. In the case of main lines, the welding dot will appear 0.4 feet from the end or ends of the line governed by the following values for DOT:

Blank - no welding dot will appear

2. - welding dots placed near both ends of the line

+1. - welding dot will be placed near last end of line

-1. - welding dot will be placed near beginning of line

For the case of the piping symbol, welding dots will appear at both ends of the symbol if DOT = 2.; no welding dots will appear if DOT is blank.

DIST - If the line is desired to be scaled, DIST is left blank; otherwise the line will be drawn the length in feet specified in DIST. For the case of symbols, they will be inserted the length specified in DIST from the beginning of the line.

MTRI - If a main line occurs in a direction other than a coordinate direction, MTRI is given a value of 1. This suppresses storage of the previous end point and combines it with the present line. Triangulation and dimensioning of the triangulation lines is possible.

B.2 Operations and Symbols Available

At the present time, the Configuration Generator Programme has a catalogue of several piping symbols and various other operations necessary in an isometric piping drawing procedure. As stated in Section 1 of Appendix B, the variable 'NN' is a control variable coded to the various symbols and operations. The functions presently available and their coded relationship to the variable 'NN' are given below:

1 - The variable 'NN' taking on the value 1 specifies the generation of a main line. This line length will be generated at the end of the previous line or at the end of the line specified by KVAL. The dimension lines and dimensioning information related to this line will also be generated if called for by the variable SIG.

2 - Instructions will be generated to write C ELEV FEET'AINCH-ANUM/DENOM

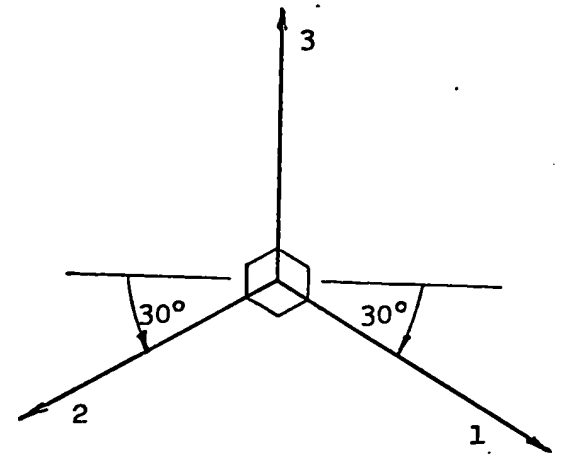


Figure 7. Positive Isometric Directions

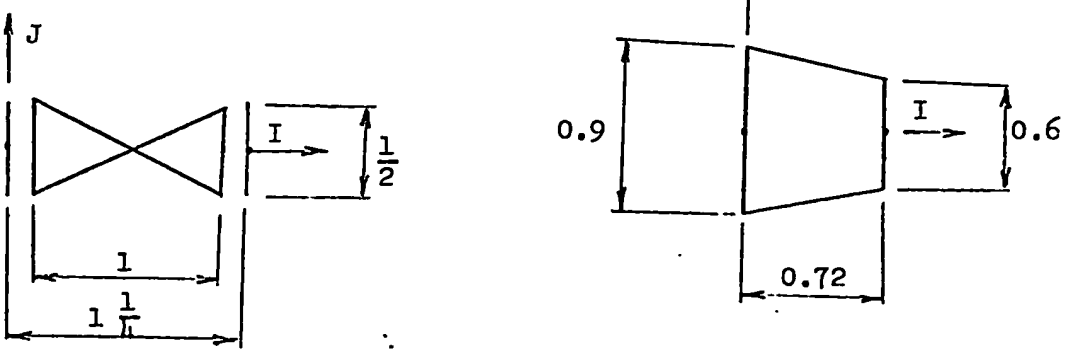


Figure 8. Two Typical Piping Symbols Encountered Showing their Relative Dimensions. The Diagrams show a Flanged Globe Valve and a Reducer, Respectively.

where the values for FEET, AINCH, ANUM and DENOM and read in according to the input sheet. This information will be written immediately above the line specified by KVAL, in the direction and plane specified by 'I' and 'J', respectively. The length in feet from the beginning of the line to the start of the written material is specified by DIST.

3 - Instructions will be generated to insert the symbol for a flanged globe valve (see Figure 8) in the line specified by KVAL. The symbol will appear in the direction and plane specified by 'I' and 'J'. The symbol will be inserted the length given by DIST from the beginning of the line and this length will be dimensioned, if called for by the variable SIG.

4 - Instructions will be generated to insert the symbol for a reducer (see Figure 8) in the line specified by KVAL. The positive direction of the reducer is defined as the large diameter occurring first, in the direction in which the line is drawn. The reducer may be drawn in the negative direction by making KVAL the negative value of the line number. The other variables have the meaning as defined previously.

5 - A flange will be drawn at the end of the line specified by KVAL.

6 - An arrow showing direction of flow will be generated at the centre of the line specified by KVAL. 'I' and 'J' indicate the direction of the line and the plane of the arrow, as before. ASIGN does not refer to 'I' in this case but specifies the direction of the arrow and must have a value of +1. or -1.

7 - The symbol for an unflanged globe valve will be generated in the line called for by the variable KVAL. The instructions for the other var-

iables are as given above.

8 - The variable 'NN' taking on the value '8' indicates that the next series of cards will contain lines of alphameric information to be printed by the printer head. The variables which have to be defined for each line of alphameric information are: KVAL, I, ASIGN, DISTRI, J, SIG, DISTJ, ORIENT, V, W, X, Y, and Z. These variables are read in by FORMAT (2I5, 2F5.0, I5, 3F5.0, 5A6). KVAL indicates the line from whose end the offset is measured. The offset is the distance DISTI in the ASIGN direction of 'I', and the distance DISTJ in the SIG direction of 'J'. ORIENT signals the printer to print in the horizontal or vertical orientation by the values '0' and '1', respectively. The last five variables contain alphameric data, up to thirty characters in length which will be printed exactly as they appear. To signal the end of information to be printed, the variable KVAL takes on a value of 99999.

9 - After all the lines and symbols desired have been called, the variable 'NN' must take on the value of 9. This will initiate the output process. All the stored information will be punched out, along with appropriate control cards to instruct the programme DRAFT and the EAI plotting package.

B.3 Main Line Generation

In main line generation, the Configuration Generator Programme generates three-dimensional coordinates of the end-points of the lines. These coordinates, x_1, y_1, z_1, and x_2, y_2, z_2, are stored in pairs of rows in the A-array. Thus each row of three elements defines a point and

each pair of rows defines a line. The starting point of the first line is given the coordinates (0, 0, 0). The first line length is then added to these coordinates by the statements:

```
A(I,K) = A(I,K-1)+ASIGN*(FEET+(AINCH+ANUM/DENOM)/12.)
A(J,K) = A(J,K-1)
A(L,K) = A(L,K-1)
```

The end point of this line is then taken as the starting point of the next line unless a different line is indicated by KVAL. Because two end points define a line, the line's index will be twice KVAL. The following statements code this logic in Fortran:

```
      IF (KVAL.LT.1) GO TO 80
      N = KVAL*2
      A(1,K-1) = A(1,N)
      A(2,K-1) = A(2,N)
      A(3,K-1) = A(3,N)
      GO TO 90
   80 IF (K.EQ.2) GO TO 90
      A(1,K-1) = A(1,K-2)
      A(2,K-1) = A(2,K-2)
      A(3,K-1) = A(3,K-2)
   90 CONTINUE
```

The length of the line is then added to the proper coordinate direction of the starting point to define its end point. This procedure is repeated for all the main lines generating the skeleton of the drawing.

After the complete skeleton of the piping has been generated, the various piping symbols can be inserted in the proper lines.

It is to be noted that only lines in one of the coordinate directions can be generated by the method described above. If the line occurs in other than a coordinate direction, this is indicated by the variable MTRI taking on a value of '1'. This signals the Configuration Generator Programme to suppress storage of the previous line's end-point and this line's starting point. In this way, the line will be the sum of the triangulation lengths read in. At the same time, the dimensioning is generated showing the triangulation lengths.

At the present time, it is not possible to draw symbols or write alphameric information (SCRIBE subroutine) in other than isometric (coordinate) directions.

B.4 Symbol Generation

The main part of the Configuration Generator Programme consists of instructions for generating the various piping symbols required in making piping drawings. A computed GO TO statement transfers control to the proper section of the programme as coded by the variable 'NN'.

The symbols are written in a general way, with the axial direction of the symbol labeled as 'I' and the plane of the symbol as 'J'. By letting 'I' and 'J' take on the appropriate values (see Figure 7), it is possible to generate the symbol in any coordinate direction and plane desired. The negative directions of 'I' and 'J' are specified by '-1.' appearing in ASIGN and SIG, respectively. The third coordinate direction, which will be perpendicular to both the 'I' and 'J' directions, is labeled by 'L' and its value automatically set by the statements:

```
IF (I.NE.1.AND.J.NE.1) L = 1
IF (I.NE.2.AND.J.NE.2) L = 2
IF (I.NE.3.AND.J.NE.3) L = 3
```

After all the main lines have been generated and stored in the A-array, the various symbols can be inserted in these lines. The lines are labeled sequentially in the order they are called, and thus in the order they are generated. Each line is defined by its two end-points and requires two lines of storage in the A-array. Thus when the number of the line is called by KVAL, the index 'N' in the A-array is computed by:

```
N = KVAL*2
```

Thus the end-points of the line in question will have the indices 'N' and 'N-1'. In order to insert the symbol, the line is first shortened to the length specified in DIST. Then a space, depending on the length necessary for the piping symbol, is left and the line continued to the previous end-point. Because the line occurs in the 'I' direction and the piping symbol should be drawn in the 'I' direction, this is the only coordinate to be affected in this step.

```
A(I,K)   = A(I,N)
A(I,N)   = A(I,N-1)+ASIGN*DIST
A(I,K-1) = A(I,N)+ASIGN*(piping symbol length)
```

This is shown graphically in Figure 9.

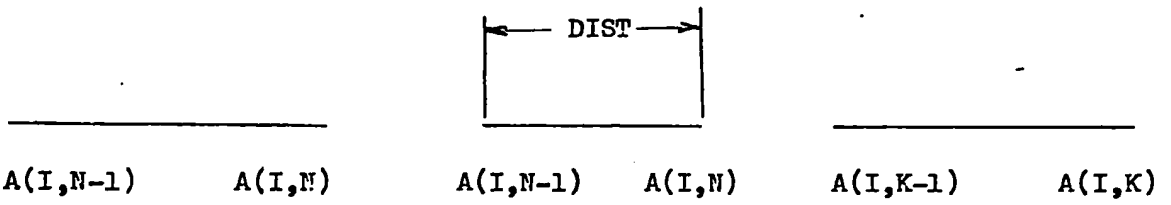


Figure 9. Removal of Section of Main Line for Insertion of Piping Symbol

After this step, the piping symbol is generated, beginning at the point A(I,N), A(J,N), A(L,N).

B.5 Storage in Configuration Generator Programme

The operation of the Configuration Generator Programme is to generate data which can be interpreted to control the movements of the data-plotter. For each card of input, all the data concerning that line or symbol, dimension lines and dimensioning are generated. Different pens and plotter modes must be chosen to draw these different lines. If the data were plotted in the same order it was generated, a large number of plotter mode and pen changes would be necessary. Since these are time consuming operations, the time involved would be unreasonably long. For this reason, it was decided to group the data according to different pens and plotter modes. This was accomplished by defining six arrays for storage purposes, one for each pen type and plotter mode type. After generation of the complete plot, each array is punched out. Since each array involves one pen change and/or one plotter mode change, this information must be transferred as well. This is done by automatically punching out a control card whose purpose is to choose the proper pen and plotting mode in the plotting package.

The arrays were labeled as A, B, D, E, C and IL. The main lines are stored in array A and the dimension lines in array B. In these two arrays, x, y, z coordinates defining a point are stored in the three elements of each horizontal row. Each odd-numbered row defines the starting point of a line segment and the next row the end-point of this line segment. In array D, the welding dot points are stored. Here again, the three elements of each row are the x, y, z coordinates of a point. The alphameric data and accompanying information is stored in arrays C and IL. For each line of alphameric data to be plotted, the starting point must be specified as well as the base angle, character slant and the information to be plotted. This is accomplished by making each array (C and IL) five elements wide, and incrementing them simultaneously. The first three elements of 'C' are used to store the x, y, z coordinates of the starting point. The fourth and fifth elements are used to store BSEANG (the base angle) and SLTANG (slant of the characters). The first four elements of 'IL' are set equal to FEET, AINCH, ANUM and DENOM from the input data. The fifth element in 'IL' is coded as '1' or '2' to write FEET' AINCH-ANUM/DENOM or C ELEV FEET'AINCH-ANUM/DENOM, respectively.

It is not always necessary for the alphameric information to be in the isometric plane. In this case it is desirable to use the printer head to print the data because of its greater speed. Nine storage locations (array E) are required to retain one line of data to be printed. The first three storage locations are necessary to store the three coordinates (x, y, z) of the starting point of the printing, the fourth location for ORIENT (0 for horizontal orientation and 1 for vertical orientation). The last five storage locations store up to thirty characters of alphameric data, exactly as read in.

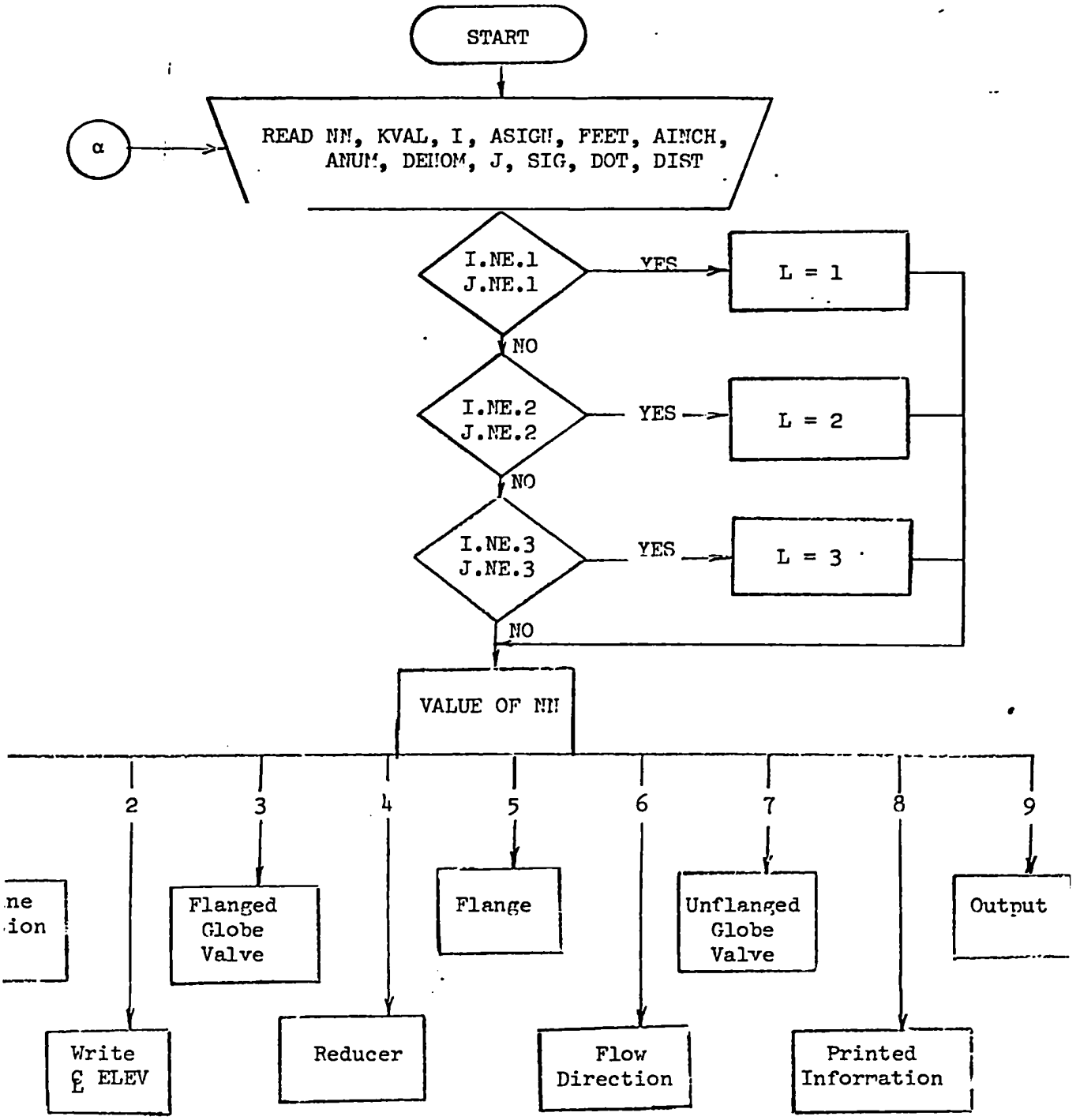


Figure 10. General Flow Chart for Configuration Generator Programme
(a) Input Phase

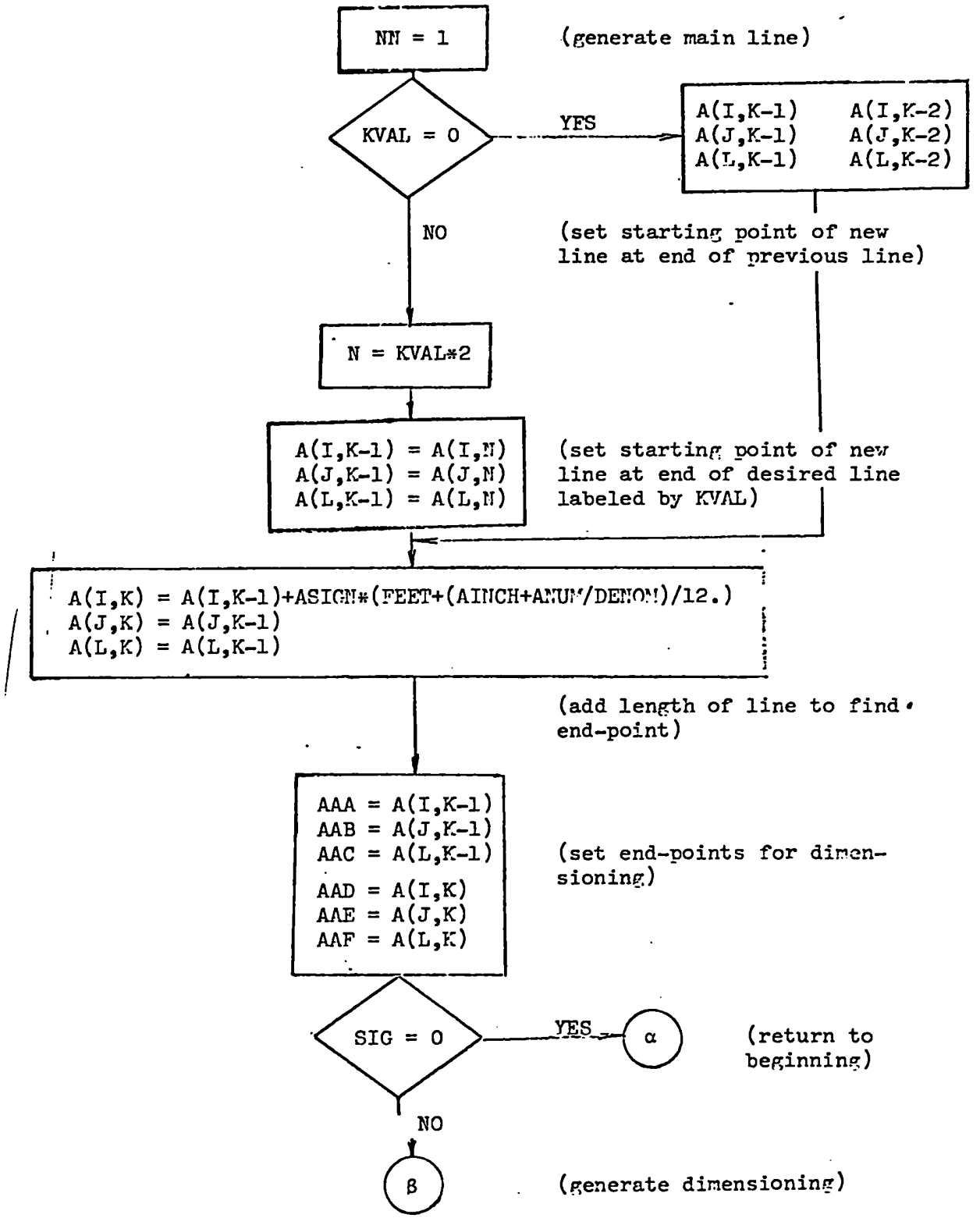


Figure 10(b) Main Line Generation

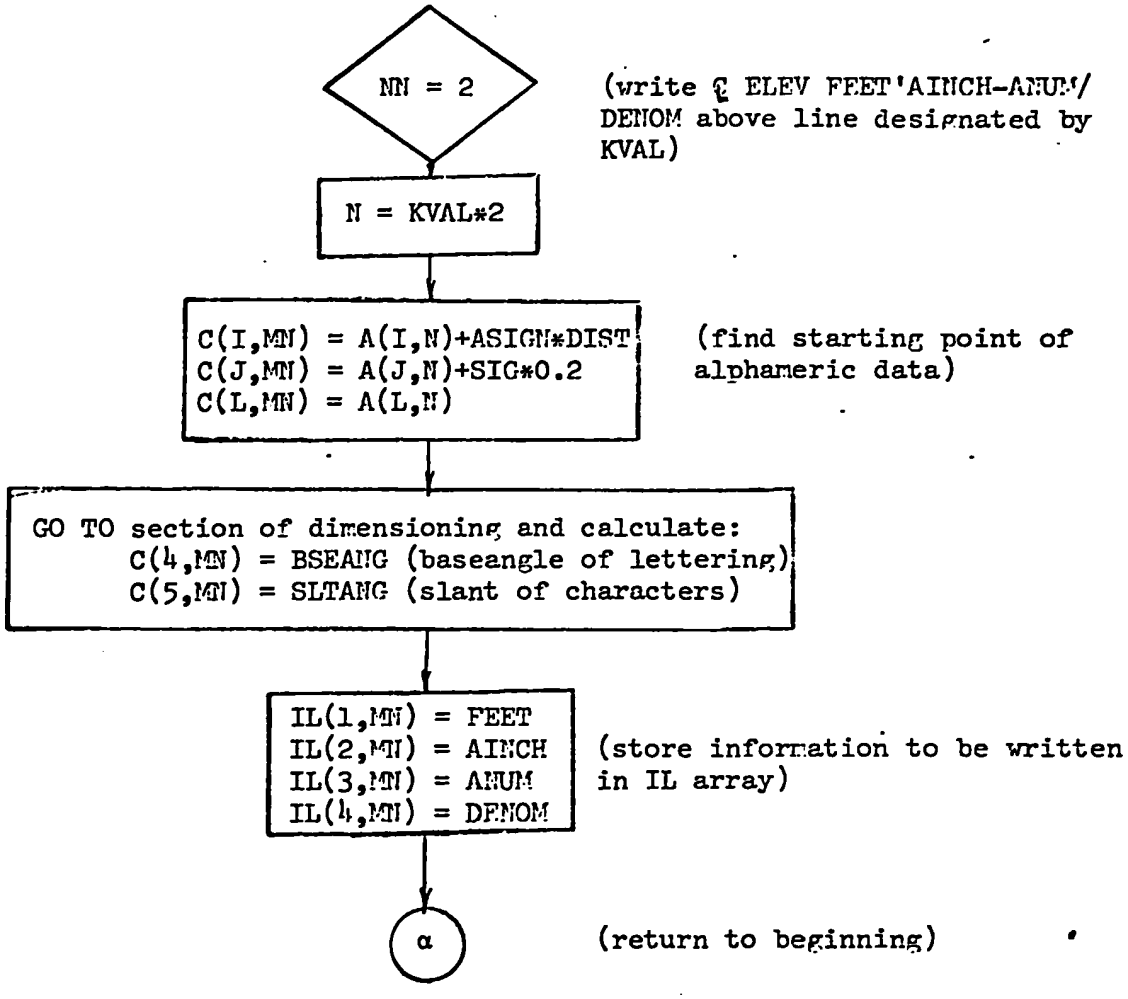


Figure 10(c) Flow Chart to Generate Instructions for Plotting Alphameric Information

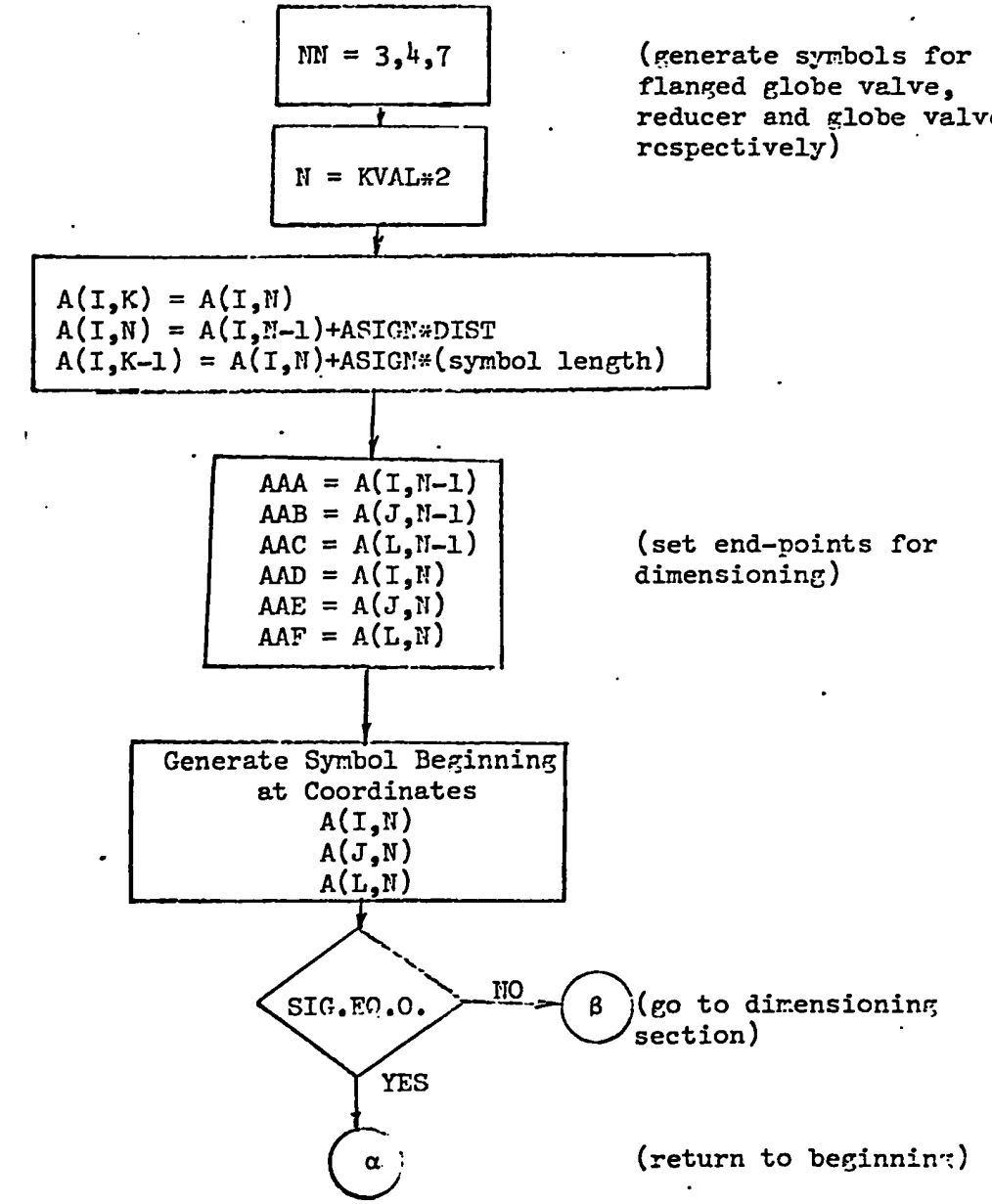


Figure 10(d) Flow Chart Showing Generation of Flanged Globe Valve, Reducer and Globe Valve

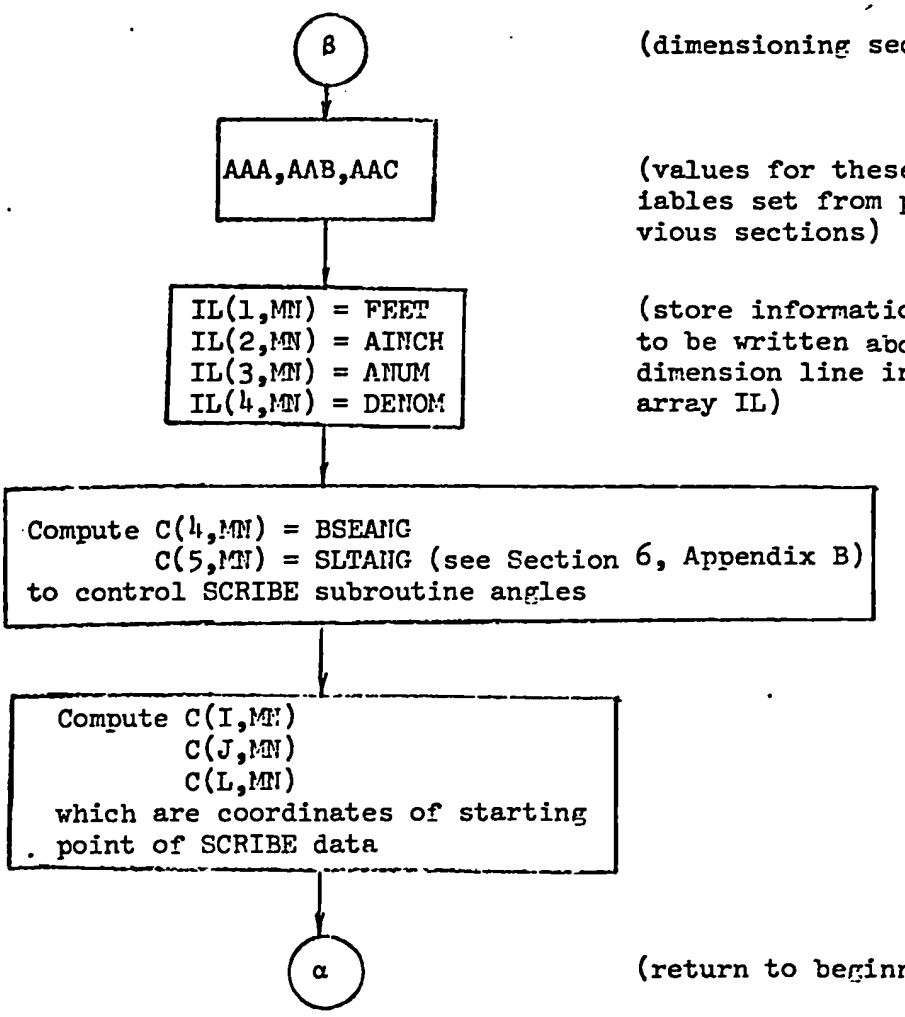


Figure 10(e) Flow Chart of Dimensioning Section

Array A

Starting points and end-points coordinates of main lines.

x_1	y_1	z_1
x_2	y_2	z_2

Array B

Same as array A for dimension lines.

Array D

Coordinates of point where welding dot is to appear.

Array C

Contains the following information for SCRIBE: coordinates of starting point of alphameric data, base angle of lettering and letter slant.

x_1	y_1	z_1	BSEANG	SLTANG

Array IL

SCRIBE information to appear in dimensioning.

		Fraction of Inch		
Feet	Inches	Numerator	Denominator	Control Character for PRINT or SCRIBE

Array E

SCRIBE information for other labeling.

x_1	y_1	z_1	ORIENT	30 characters of alphameric data

Figure 11. Storage in Configuration Generator Programme

B.6 Dimensioning

As stated in the input sheet instructions, dimension lines and dimensioning information will be generated if the variable SIG is other than blank or zero. SIG indicates on which side of the line or symbol the dimension lines are to be generated. The end-points of the line or symbol, whose length is to be dimensioned, must be stored in the computer. This is done by labeling their coordinates as AAA, AAB, AAC and AAD, AAE, AAF. After the dimension line coordinates have been generated, the starting point of the first character of the dimensioned information must be computed. This is done by computing the centre of the dimension line and adding or subtracting predetermined lengths in the 'I' and 'J' directions. The correct direction for the 'I' offset is labeled by ASIGN and determined from the diagrams in Figure 12. The logic in Figure 12 can be coded in computer language as:

```
ASIGN  =  +1.
IF ((I.EQ.1).OR.(I.EQ.3)) ASIGN = -1.
C(I,MN) = (AAA + AAD)/2. + ASIGN*1.0
```

Similarly, the 'J' direction offset is labeled by SIG and is again determined from the diagrams of Figure 12. In computer language, this becomes:

```
SIG  =  +1.
IF ((I.EQ.1).AND.(J.EQ.2)) SIG = -1.
IF ((I.EQ.2).AND.(J.EQ.1)) SIG = -1.
IF ((I.EQ.3).AND.(J.EQ.1)) SIG = -1.
C(J,MN) = B(J,M) + SIG*0.1
```

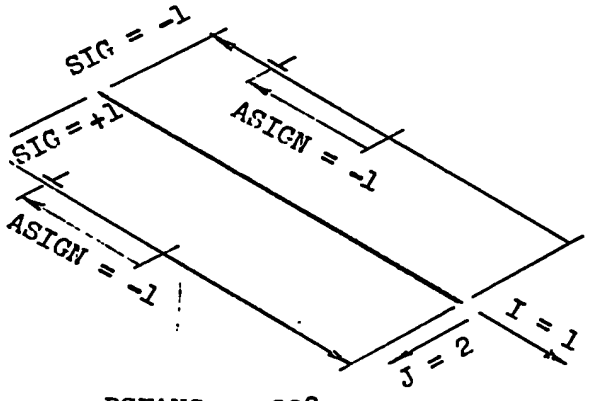
Because the writing is to be done in the plane of the drawing, it is necessary to know the base angle (BSEANG) and the slant of the characters (SLTANG) in addition to the coordinates of the starting point. As can be seen from the diagrams of Figure 12, it is possible to determine these variables from the 'I' and 'J' values. The variables BSEANG and SLTANG are stored in the C-array as C(4,MN) and C(5,MN), respectively. In computer language this logic can be written as:

```
IF (I.EQ.1) C(4,MN) = -30.
IF (I.EQ.2) C(4,MN) =  30.
IF (I.EQ.3) C(4,MN) =  90.
C(5,MN) = +30.
IF ((I.EQ.1).AND.(J.EQ.3)) C(5,MN) = -30.
IF ((I.EQ.2).AND.(J.EQ.1)) C(5,MN) = -30.
IF ((I.EQ.3).AND.(J.EQ.2)) C(5,MN) = -30.
```

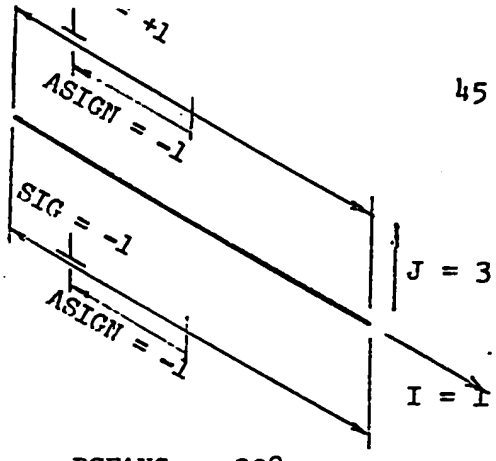
All these variables are stored and punched out one array at a time as described in Section 7 of this appendix.

B.7 Output

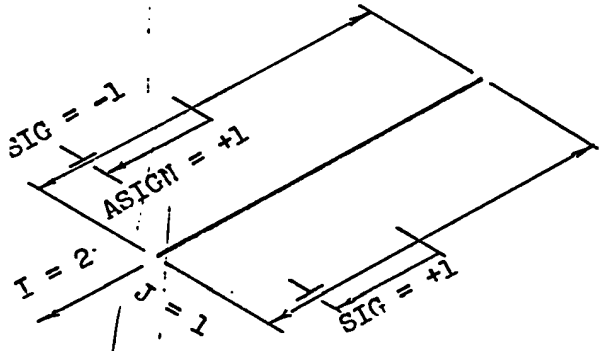
This phase of the Configuration Generator Programme is triggered by the last input card where 'NN' takes on the value '9'. In this section, the stored data along with the appropriate control cards to control the plotting package are punched out. First the main line data is punched out. The control card is written according to FORMAT (6H99999.,7X,2H2., 18X,2H1.). The first group of characters (99999.) tells DRAFT that this card is a control card. The second group (2.) tells the plotting package which pen to choose, in this case the medium width (pen #2). The third



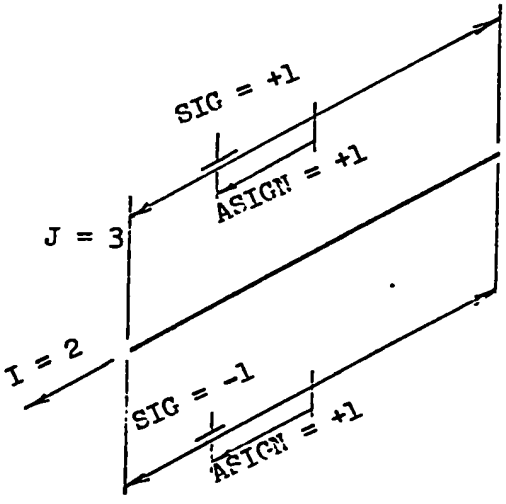
BSEANG = -30°
SLTANG = +30°



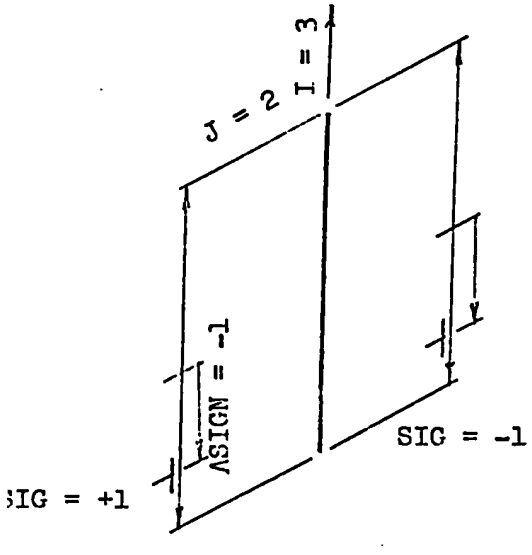
BSEANG = -30°
SLTANG = -30°



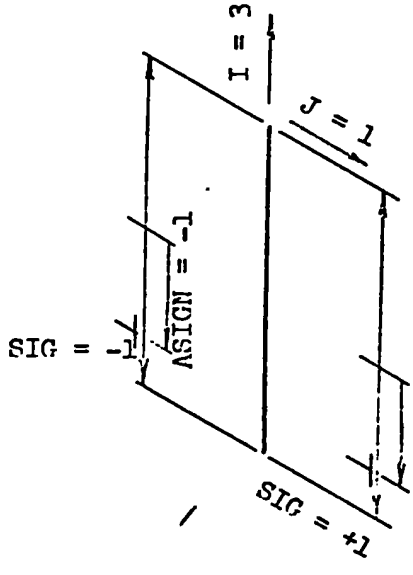
BSEANG = +30°
SLTANG = -30°



BSEANG = +30°
SLTANG = +30°



BSEANG = 90°
SLTANG = -30°



BSEANG = 90°
SLTANG = +30°

Figure 12. Dimensioning Variables

group (1.) tells the plotting package which plotting mode to enter, in this case the line mode. Then the A-array is punched out, such that two rows appear on each card. In this way each card contains x_1, y_1, z_1 and x_2, y_2, z_2, defining the two end-points of a line segment.

After the last card of this array has been punched, another control card is punched. This control card is punched according to FORMAT (6H99999.,7X,2H3.,18X,2H1.). Again 99999. identifies this to be a control card. The second group (3.) tells the plotting package to pick pen #3, the thin width. The third group (1.) again instructs the plotting package to enter the line mode. Then the B-array, which contains the coordinates used in drawing dimension lines, is punched out.

Upon completion of output from the B-array, another control card is punched to set the correct parameters in the plotting package for entering the SCRIBE subroutine. The format for this control card is (6H99999.,7X,2H3.,18X,2H4.). Again pen #3 (thin width) is called, and the numeral 4 appearing in the third group is a signal to DRAFT to read the succeeding data as SCRIBE information and to call the SCRIBE subroutine from the plotting package. The C- and IL-arrays are then punched simultaneously. The first three elements of the C-array contain x, y, z coordinates of the starting points of the alphameric information. The next two elements of the C-array contain the baseangle and the character slant of the alphameric information. The first four elements of the IL-array contain the length in feet, inches and fractions of an inch. The fifth element of the IL-array contains the signal 1 or 2 to indicate the length is to be written as FEET'AINCH-ANUM/DENOM and C ELEV FEET'AINCH-ANUM/DENOM, respectively. In order to avoid writing zeroes in the dimen-

sioned length, the elements of the IL-array are checked for zeroes and corresponding formats used in the output.

Then the welding dot D-array is punched out. The control card is punched according to format (6H99999.,7X,2H1.,18X,2H2.). This will signal the plotting package to select pen #1 (very wide) and to enter mode #2, which is the point mode. The data from the D-array is punched two rows per card such that each card contains x_1, y_1, z_1 and x_2, y_2, z_2, the coordinates of two points to be plotted.

The final array to be punched out is the E-array. This array contains the data to be printed by the printer head. The control card is punched according to format (6H99999.,7X,2H1.,18X,2H3.). For the printed output, a pen is not required, but to simplify programming of DRAFT we merely select the same pen as used in the previous array. The last group (3) is a signal to DRAFT to enter the printing mode. The first three elements of the E-array are the x, y, z coordinates of the starting point of the printed information. The fourth element is either 0 or 1, which chooses the horizontal or vertical orientation, respectively, for the printed information. The last five elements of the E-array contain up to thirty characters of information to be printed exactly as they are read in.

The final card is punched according to format (6H99999.), indicating it to be a control card and that no more data cards are to be read in.

During the output, if any of the arrays A, B, D, E, C or IL contain no entries, the programme will automatically cause its control card and output of that array to be omitted.

B.8 Definition of Variables and Fortran Names

The description of variables read in as input data has been given in Section 1 of this appendix. Other variables whose meaning might be useful to future users of this programme are listed below, along with a short description:

L The coordinate which remains constant during the generation of any line, piping symbol or dimensioning.

A Array of main line and piping symbol coordinates. Each line contains three coordinates (x, y, z) defining one point. Two rows of storage are required for each line segment. Data in this array is plotted using the medium width pen.

K Index of A-array

B Array of dimension line coordinates. Two rows of storage are required for each line segment. Data in this array is plotted using the thin width pen.

M Index of B-array

D Array of welding symbol coordinates. One row of storage defines each welding dot.

LN Index of D-array

E Array of data to be plotted by SCRIBE routine. Storage locations contain coordinates of starting point, orientation of scribed information and thirty characters of information to be scribed.

II Index of E-array

C Array of starting coordinates, base angle and character slant of information to be plotted as dimensioning.

IL Array of length to be plotted as labeling of dimensioning. Con-

tains feet, inches, numerator and denominator.

MN Index of C-array and IL-array. Both arrays are indexed simultaneously.

N Index of beginning of branch line or beginning of piping symbol from A-array. The value of this index is set to twice the value of KVAL.

SKIP Length of offset of dimension lines perpendicular to main line or piping symbol.

DD Length of dimension line perpendicular to main line or piping symbol.

V Length of valve to which its dimensions are proportioned.

AL Length of arrow used to indicate direction of flow of fluid in pipe.

B.9 Input Data Instructions

The sample input is shown in Table 2, and a description of input data is given in Section 1 of this appendix. The data cards must be punched according to the Fortran format given below:

Format(3I5,5F5.0,I5,3F5.0,I5)

To signal the end of data and cause the computer to enter the output phase of this programme, a value of 9 must appear in the first field of the data card.

B.10 Computer Listing of the Configuration Generator Programme

```
$B   WATFOR  003711 PAULS            060   010   030
$JOB         NODECK                                          51
$FTC CONFIG
      DIMENSION A(3,  500), B(3,500),  C(5,100), IL(5, 100), D(3,500),
     1E(9,50) .
      DO 10 I = 1, 3
      DO 10 J = 1, 500
      A(I,J) = 0.0
      B(I,J) = 0.0
   10 D(I,J) = 0.0                          (set all the array locations
                                             equal to zero)
      DO 17 I = 1,5
      DO 17 J = 1, 100
      C(I,J) = 0.0
   17 IL(I,J) = 0
      DO 18 I = 1,9
      DO 18 J = 1,50
   18 E(I,J) = 0.0
      K = 2
      M = 2
      MN = 1
      LN = 1
      II = 1
   20 READ (5,30) NN,KVAL,I,ASIGN,FEET,AINCH,ANUM,DENOM,J,SIG,DOT,DIST
     1 , MTRI                               (read in one line of input data)
   30 FORMAT (3I5,5F5.0, I5,2F5.0,I5)
      IF (I.NE.1.AND.J.NE.1) L = 1
      IF (I.NE.2.AND.J.NE.2) L = 2          (compute value of constant coordinate)
      IF (I.NE.3.AND.J.NE.3) L = 3
    CONTROL IS TRANSFERRED TO THE CORRECT SECTION OF THE CATALOGUE
    DEPENDING ON THE SYMBOL DESIRED.
```

```
      GO TO (40,100,300,400,500,600,700,800,900), MN
      GENERATE COORDINATES OF MAIN LINES
   40 IF (KVAL.LT.1) GO TO 80
      N = KVAL * 2                          (set index of beginning of
      IF (N.GE.K) GO TO 80                   branch line)
      A(1,K-1) = A(1,N)
      A(2,K-1) = A(2,N)
      A(3,K-1) = A(3,N)
      GO TO 90
   80 IF (K.EQ.2) GO TO 90
      A(1,K-1)= A(1,K-2)
      A(2,K-1)= A(2,K-2)
      A(3,K-1)= A(3,K-2)
   90 IF (DENOM.EQ.0.) DENOM = 1.           (add length of line to proper coordinate)
      A (I,K) = A(I,K-1) + ASIGN*(FEET +(AINCH + ANUM/DENOM)/12.)
      IF (ABS(A(I,K)-A(I,K-1)).GT.18.)     A(I,K) = A(I,K-1)+ASIGN*18.
      IF (DIST.NE.0.) A(I,K) = A(I,K-1) + ASIGN * DIST
      A(J,K) = A(J,K-1)
      A(L,K) = A(L,K-1)
      IF (DOT.EQ.0.) GO TO 87
      IF (DOT.EQ.(+1.)) GO TO 86
      D(I,LN) = A(I,K-1) + ASIGN*0.4
                                            (generate symbol for welded joint)
      D(J,LN) = A(J,K-1)
      D(L,LN) = A(L,K-1)
      LN = LN + 1
      IF (DOT.EQ.(-1.)) GO TO 87
   86 D(I,LN) = A(I,K) - ASIGN*0.4
      D(J,LN) = A(J,K)
      D(L,LN) = A(L,K)
```

```
      LN = LN + 1
   87 IF (MTRI.EQ.0) GO TO 92
      B(I,M-1) = A(I,K-1)
      B(J,M-1) = A(J,K-1)                    (triangulation for non-coordinate
      B(L,M-1) = A(L,K-1)                     direction lines)
      B(I,M) = A(I,K)
      B(J,M) = A(J,K)
      B(L,M) = A(L,K)
      M = M + 2
   92 AAA = A(I,K-1)
      AAB = A(J,K-1)
      AAC = A(L,K-1)                         (set starting coordinates for
      AAD = A(I,K)                            generating dimensioning)
      AAE = A(J,K)
      AAF = A(L,K)
      IF (MTRI.LE.1) GO TO 97
      K = K - 2
      A(1,K-1) = A(1,K-2)
      A(2,K-1) = A(2,K-2)
      A(3,K-1) = A(3,K-2)
      A(1,K) = A(1,K+2)
      A(2,K) = A(2,K+2)
      A(3,K) = A(3,K+2)
   97 K = K + 2
      IF (SIG.EQ.0.) GO TO 20                (if line is not to be dimensioned,
                                              return to statement 20)
      DD=.2.6
      SKIP = 0.2
      GO TO 1500
  100 N = KVAL * 2
```

```
      WRITE 'C ELEV FEET' AINCH-ANUM/DENOM' ABOVE THE LINE
      DD= 0.1
      C(I,MN) = A(I,N-1) + (ASIGN*DIST)
      IF (I.EQ.3) GO TO 20
      IF ((I.EQ.1).AND.(J.EQ.2)) DD= -DD        (compute starting point coordinates
      IF ((I.EQ.2).AND.(J.EQ.1)) DD= -DD         of labeling)
      C(J,MN) = A(J,N-1) + DD
      C(L,MN) = A(L,N-1)
      GO TO 1512
 120  IL(5,MN) = 2
      MN = MN + 1
      GO TO 20
 300  V = 1.                                    (length of valve)
      GENERATE SYMBOL FOR FLANGED GLOBE VALVE
      MN = 1
      N = KVAL * 2
      A(J,K-1) = A(J,N)
      A(J,K) = A(J,K-1)
      A(L,K-1) = A(L,N)
      A(L,K) = A(L,K-1)
      A(I,K) = A(I,N)
      A(I,N) = A(I,N-1) + ASIGN*DIST
      A(I,K-1) = A(I,N) +ASIGN*V*1.25
      AAA = A(I,N-1)
      AAB = A(J,N-1)
      AAC = A(L,N-1)                            (set starting coordinates for
      AAD = A(I,N)                               generation of dimensioning)
      AAE = A(J,N)
      AAF = A(L,N)
```

```
      A(I,K+1) = A(I,N)
      A(J,K+1) = A(J,N) - V/4.
      A(L,K+1) = A(L,N)
      IF (DOT.EQ.0.) GO TO 310
      D(I,LN) = A(I,N)
      D(J,LN) = A(J,N)
      D(L,LN) = A(L,N)
      LN = LN + 1
      D(I,LN) = A(I,K-1)
      D(J,LN) = A(J,K-1)
      D(L,LN) = A(L,K-1)
      LN = LN + 1
  310 K = K + 2
      A(I,K) = A(I,K-1)
      A(J,K) = A(J,K-1) + V/2.
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K) + ASIGN *V/8.
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1) + ASIGN * V
      A(J,K) = A(J,K-1) - V/2.
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K)
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1)
      A(J,K) = A(J,K-1) + V/2.
```

```
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K)
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1) - ASIGN * V
      A(J,K) = A(J,K-1) - V/2.
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K)
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1)
      A(J,K) = A(J,K-1) + V/2.
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K) + ASIGN*V*1.125
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1)
      A(J,K) = A(J,K-1) - V/2.
      A(L,K) = A(L,K-1)
      K = K + 2
      IF (SIG.EQ.2.) GO TO 20
      DD = 1.8
      SKIP = 0.4
      GO TO 1500
  400 DIA1 = 0.9
      GENERATE SYMBOL FOR REDUCER
```

```
      DIA2 = 0.6
      IF (KVAL.GT.0) GO TO 405
      DIA1 = 0.6
      DIA2 = 0.9
405   N = ABS(KVAL)*2
      AAA = A(I,N-1)
      AAB = A(J,N-1)                    (beginning coordinates of
      AAC = A(L,N-1)                     dimensioning)
      A(J,K-1) = A(J,N)
      A(J,K) = A(J,K-1)
      A(L,K-1) = A(L,N)
      A(L,K) = A(L,K-1)
      A(I,K) = A(I,N)
      A(I,N) = A(I,N-1) + ASIGN*DIST
      A(I,K-1) = A(I,N) +ASIGN*0.72
      A(I,K+1) = A(I,N)
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      IF (DOT.EQ.0.) GO TO 410
      D(I,LN) = A(I,N)
      D(J,LN) = A(J,N)
      D(L,LN) = A(L,N)
      LN = LN + 1
      D(I,LN) = A(I,K-1)
      D(J,LN) = A(J,K-1)
      D(L,LN) = A(L,K-1)
      LN = LN + 1
410   K = K + 2
      AAD = A(I,K-1)
```

```
AAE = A(J,K-1)
AAF = A(L,K-1)
A(I,K) = A(I,K-1)
A(J,K) = A(J,K-1) + DIA1/2.
A(L,K) = A(L,K-1)
A(I,K+1) = A(I,K)
A(J,K+1) = A(J,K)
A(L,K+1) = A(L,K)
K = K + 2
A(I,K) = A(I,K-1) + ASIGN*0.72
A(J,K) = A(J,K-1) - DIA1/2. + DIA2/2.
A(L,K) = A(L,K-1)
A(I,K+1) = A(I,K)
A(J,K+1) = A(J,K)
A(L,K+1) = A(L,K)
K = K + 2
A(I,K) = A(I,K-1)
A(J,K) = A(J,K-1) - DIA2
A(L,K) = A(L,K-1)
A(I,K+1) = A(I,K)
A(J,K+1) = A(J,K)
A(L,K+1) = A(L,K)
K = K + 2
A(I,K) = A(I,K-1) - ASIGN*0.72
A(J,K) = A(J,K-1) + DIA2/2. - DIA1/2.
A(L,K) = A(L,K-1)
A(J,K+1) = A(J,K)
A(I,K+1) = A(I,K)
A(L,K+1) = A(L,K)
```

```
      K = K + 2
      A(I,K) = A(I,K-1)
      A(J,K) = A(J,K-1) + DIA1
      A(L,K) = A(L,K-1)
      K = K + 2
      IF (SIG.EQ.0.) GO TO 20
      DD = 1.8
      SKIP = 0.4
      GO TO 1500
      GENERATE SYMBOL FOR FLANGE
500   N = KVAL * 2
      A(I,K-1) = A(I,N)
      A(J,K-1) = A(J,N) - 0.25
      A(L,K-1) = A(L,N)
      A (I,K) = A(I,K-1)
      A (J,K) = A(J,K-1) +0.5
      A (L,K) = A(L,K-1)
      K = K + 2
      GO TO 20
600   N = KVAL * 2
      GENERATE ARROW IN CENTER OF LINE TO INDICATE FLOW DIRECTION
      AL = 0.8
      A(I,K-1) = (A(I,N) + A(I,N-1))/2.
      A(J,K-1) =  A(J,N) + AL/6.
      A(L,K-1) =  A(L,N)
      A(I,K) = A(I,K-1) + ASIGN*AL
      A(J,K) = A(J,N)
      A(L,K) = A(L,K-1)
      K = K + 2
```

```
      A(I,K-1) = A(I,K-2)
      A(J,K-1) = A(J,K-2)
      A(L,K-1) = A(L,K-2)
      A(I,K) = A(I,K-3)
      A(J,K) = A(J,N) - AL/6.
      A(L,K) = A(L,K-1)
      K = K + 2
      GO TO 20
  700 V = 1.5
      GENERATE SYMBOL FOR GLOBE VALVE
      N = KVAL * 2
      AAA = A(I,N-1)
      AAB = A(J,N-1)
      AAC = A(L,N-1)
      A(J,K-1) = A(J,N)
      A(J,K) = A(J,K-1)
      A(L,K-1) = A(L,N)
      A(L,K) = A(L,K-1)
      A(I,K) = A(I,N)
      A(I,N) = A(I,N-1) + ASIGN*DIST
      A(I,K-1) = A(I,N) +ASIGN*V
      A(I,K+1) = A(I,N)
      A(J,K+1) = A(J,N-1) - V/4.
      A(L,K+1) = A(L,N-1)
      AAD = A(I,N)
      AAE = A(J,N)
      AAF = A(L,N)
      K = K + 2
      A(I,K) = A(I,K-1)
```

```
      A(J,K) = A(J,K-1) + V/2.
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K)
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1) + ASIGN * V
      A(J,K) = A(J,K-1) - V/2.
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K)
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1)
      A(J,K) = A(J,K-1) + V/2.
      A(L,K) = A(L,K-1)
      A(I,K+1) = A(I,K)
      A(J,K+1) = A(J,K)
      A(L,K+1) = A(L,K)
      K = K + 2
      A(I,K) = A(I,K-1) - ASIGN*V
      A(J,K) = A(J,K-1) - V/2.
      A(L,K) = A(L,K-1)
      K = K + 2
      IF (SIG.EQ.0.) GO TO 20
      DD = 1.8
      SKIP = 0.4
      GO TO 1500                          (read line of information for labeling
  800 READ (5,810) KVAL,I,ASIGN,DISTI,J,SIG,DISTJ,ORIENT,V,W,X,Y,Z
```

```
810 FORMAT (2I5,2F5.0,I5,3F5.0,5A6)
    IF (KVAL.GE.900) GO TO 20
    N = KVAL*2
    IF (I.NE.1.AND.J.NE.1) L = 1
    IF (I.NE.2.AND.J.NE.2) L = 2
    IF (I.NE.3.AND.J.NE.3) L = 3
    E(I,II) = A(I,N) + ASIGN*DISTI
    E(J,II) = A(J,N) + SIG*DISTJ
    E(L,II) = A(L,N)
    E(4,II) = ORIENT
    E(5,II) = V
    E(6,II) = W
    E(7,II) = X
    E(8,II) = Y
    E(9,II) = Z
    II = II + 1
    GO TO 800
500 AL = 0.5
    GENERATE DIMENSIONING LINES AND DIMENSIONING INFORMATION
    B(I,M-1) = AAA
    B(J,M-1) = AAB + SIG*SKIP
    B(L,M-1) = AAC
    B(I,M) = B(I,M-1)
    B(J,M) = B(J,M-1) + SIG*DD
    B(L,M) = B(L,M-1)
    M = M + 2
    B(I,M-1) = AAD
    B(J,M-1) = AAE + SIG*SKIP
    B(L,M-1) = AAF
```

```
      B(I,M) = B(I,M-1)
      B(J,M) = B(J,M-1) + SIG*DD
      B(L,M) = B(L,M-1)
      M = M + 2
      B(I,M-1) = B(I,M-2)
      B(J,M-1) = B(J,M-2) - SIG*SKIP
      B(L,M-1) = B(L,M-2)
      B(I,M) = B(I,M-4)
      B(J,M) = B(J,M-4) - SIG*SKIP
      B(L,M) = B(L,M-4)
  112 IL(1,MN) = FEET
      IL(2,MN) = AINCH
      IL(3,MN) = ANUM
      IL(4,MN) = DENOM
      IF (I.EQ.1) C(4,MN) = -30.
      IF (I.EQ.2) C(4,MN) =  30.
      IF (I.EQ.3) C(4,MN) =  90.                    (set base angle and slant of
                                                     lettering)
      C(5,MN) = +30.
      IF ((I.EQ.1).AND.(J.EQ.3)) C(5,MN) = -30.
      IF ((I.EQ.2).AND.(J.EQ.1)) C(5,MN) = -30.
      IF ((I.EQ.3).AND.(J.EQ.2)) C(5,MN) = -30.
  114 IF (NN.EQ.2) GO TO 120
      ASIGN = +1.
      IF ((I.EQ.1).OR.(I.EQ.3)) ASIGN = -1.
      C(I,MN) = (AAA + AAD)/2. + ASIGN*1.0
      SIG = +1.                                     (set starting coordinates of
                                                     dimensioning label)
      IF ((I.EQ.1).AND.(J.EQ.2)) SIG = -1.
      IF ((I.EQ.2).AND.(J.EQ.1)) SIG = -1.
      IF ((I.EQ.3).AND.(J.EQ.1)) SIG = -1.
```

```
      C(J,MN) = B(J,M) + SIG*0.1
      C(L,MN) = AAC
      IL (5,MN) = 1
      MN = MN + 1
      M = M + 2
      MM = 1
      KK = 1
      AA = B(I,M-2)
      AB = B(J,M-2)
      AC = B(L,M-2)
  520 B(I,M-1) = AA - AL
      IF (ABS(B(I,M-1)-B(I,M-3)).LT.ABS(AA - B(I,M-3))) GO TO 1540
      KK = KK + 1
      IF (KK.GT.3) GO TO 1540
      AL = -AL
      GO TO 1520
  540 B(J,M-1) = AB + AL/6.
      B(L,M-1) = AC
      B(I,M) = AA
      B(J,M) = AB
      B(L,M) = AC
      M = M + 2
      B(I,M-1) = B(I,M-2)
      B(J,M-1) = B(J,M-2)
      B(L,M-1) = B(L,M-2)
      B(I,M) = B(I,M-1) - AL
      B(J,M) = B(J,M-1) - AL/6.
      B(L,M) = B(L,M-1)
      M = M + 2
```

```
      IF (MM.GE.2) GO TO 20
      AA = B(I,M-7)
      AB = B(J,M-7)
      AC = B(L,M-7)
      MM = MM + 1
      KK = 1
      AL = -AL
      GO TO 1520
   00 IF (K.EQ.2) GO TO 950
      PUNCH OUT ALL THE STORED INFORMATION DESCRIBING THE DRAWING
      WRITE (7,910)
  910 FORMAT (6H99999., 7X,2H2.,18X,2H1.)
      K = K-2
      DO 945 I = 2,K,2
      WRITE (7,940)A(1,I-1),A(2,I-1),A(3,I-1),A(1,I),A(2,I),A(3,I)
  940 FORMAT (6F10.5)
  945 CONTINUE
  950 IF (M.EQ.2) GO TO 980
      WRITE (7,960)
  960 FORMAT (6H99999., 7X,2H3.,18X,2H1.)
      M = M - 2
      DO 970 I = 2,M,2
      WRITE (7,940)B(1,I-1),B(2,I-1),B(3,I-1),B(1,I),B(2,I),B(3,I)
  970 CONTINUE
  980 IF (MN.EQ.1) GO TO 1105
      WRITE (7,985)
  985 FORMAT (6H99999., 7X,2H3.,18X,2H4.)
      MN = MN - 1
      DO 1100 I = 1,MN
```

```
      N = IL(5,I)
      GO TO (1000,1060,1100), N
 1000 IF (IL(1,I).EQ.0) GO TO 1030
      IF (IL(3,I).EQ.0) GO TO 1020
      IF (IL(4,I).LT.10) GO TO 1010
      WRITE (7,990) C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(1,I),IL(2,I),
     1IL(3,I),IL(4,I)
  990 FORMAT (3F10.5,    2F10.3,I2,2H'-, I2,1H-,I2,1H/,I2)
      GO TO 1100
 1010 WRITE (7,1015)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(1,I),IL(2,I),
     1IL(3,I),IL(4,I)
 1015 FORMAT (3F10.5,    2F10.3,I2,2H'-, I2,1H-,I1,1H/,I1)
      GO TO 1100
 1020 WRITE (7,1025)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(1,I),IL(2,I)
 1025 FORMAT (3F10.5,    2F10.3,I2,2H'-, I2)
      GO TO 1100
 1030 IF (IL(3,I).EQ.0) GO TO 1040
      IF (IL(4,I).LT.10) GO TO 1050
      WRITE (7,1035)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(2,I),IL(3,I),
     1 IL(4,I)
 1035 FORMAT (3F10.5,2F10.3,I2,1H-,I2,1H/,I2)
      GO TO 1100
 1040 WRITE (7,1045)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(2,I)
 1045 FORMAT (3F10.5,2F10.3,I2)
      GO TO 1100
 1050 WRITE (7,1055)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(2,I),IL(3,I),
     1 IL(4,I)
 1055 FORMAT (3F10.5,2F10.3,I2,1H-,I1,1H/,I1)
      GO TO 1100
```

```
 160 IF (IL(3,I).EQ.0) GO TO 1070
     IF (IL(4,I).LT.10) GO TO 1080
     WRITE (7,1065)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(1,I),IL(2,I),
    1IL(3,I),IL(4,I)
 165 FORMAT (3F10.5,    2F10.3,5HC EL.,I3,2H'-, I2,1H-,I2,1H/,I2)
     GO TO 1100
 170 WRITE (7,1075)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(1,I),IL(2,I)
 175 FORMAT (3F10.5,    2F10.3,5HC EL.,I3,2H'-,I2)
     GO TO 1100
 180 WRITE (7,1085)C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),IL(1,I),IL(2,I),
    1IL(3,I),IL(4,I)
 185 FORMAT (3F10.5,    2F10.3,5HC EL.,I3,2H'-, I2,1H-,I1,1H/,I1)
 100 CONTINUE
 105 IF (LN.EQ.1) GO TO 1130
     WRITE (7,1110)
 110 FORMAT (6H99999., 7X,2H1.,18X,2H2.)
     DO 1120 I = 2,LN,2
 120 WRITE (7,1040)D(1,I-1),D(2,I-1),D(3,I-1),D(1,I),D(2,I),D(3,I)
 130 IF (II.EQ.1) GO TO 1170
     WRITE (7,1140)
 140 FORMAT(6H99999.,7X,2H1.,18X,2H3.)
     II = II - 1
     DO 1150 I = 1,II
 150 WRITE (7,1160)E(1,I),E(2,I),E(3,I),E(4,I),E(5,I),E(6,I),F(7,I),
    1E(8,I),E(9,I)
 160 FORMAT (3F10.5,F10.3,10X,5A6)
 170 WRITE (7,1180)
 180 FORMAT (6H99999.)
     STOP
```

```
    END
NTRY
   1          1 +1.  2.                              3
   1          3 +1.  15.  6.                         1 +1.  +1.  7.
   1          2 +1.  42.  2.    1.    2.             3 +1.  2.   8.
   1          1 -1.  6.   5.    1.    2.             3 -1.  2.
   1          3 +1.  23.  7.    3.    4.             2 +1.  -1.  8.
   1          3 +1.  1.         3.    4.             2 -1.        2.
   1          3 +1.        15.                       2 -1.  +1.  2.
   1          2 +1.         12.   1.    2.           3 +1.  -1.  2.
   1     1    3 -1.         7.    1.    2.           1 +1.  +1.  2.
   7     1    1 +1.                                  3
   5     8    2 +1.                                  3 +1.
   5     9    3 -1.                                  1 +1.
   8
   2     3 +1.  1.        1 +1.  0.5   0.   E-1
   9     3 -1.  3.        1 +1.  0.5   0.   CONT. ON 1-8-C/14001A
9999
   9
B.SYS
```

APPENDIX C

Digital Computer Programme DRAFT

C.1 Operation of the Programme DRAFT

The three-dimensional (X, Y, Z) coordinates of all the end-points are read in. Mathematical operations are performed on each point and transformed to the new coordinates (X*, Y*, Z*). If these operations are performed on all the points, the whole model will be transformed as desired. The operations can be defined easily in a matheratical form.

(1) To translate the object by an amount ΔX, ΔY and ΔZ in the respective directions X, Y, Z

$$X^* = X + \Delta X$$

$$Y^* = Y + \Delta Y$$

$$Z^* = Z + \Delta Z$$

(2) To rotate the object by an angle θ about the X-axis

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

A rotation about an axis is positive if a right-handed screw, when pointed in the positive direction of the axis, would advance when given the rotation.

(3) To rotate the object by an angle θ about the Y-axis

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

(4) To rotate the object by an angle θ about the Z-axis

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

After performing the required mathematical operations on all the points, the model will be in the desired orientation and position, but still in three-dimensional (X, Y, Z) coordinates. The problem now remains of transforming these points into two-dimensional form, so the object will appear in perspective. As seen in Figure 13, the mathematical formulation of the problem is to determine the image point of each object point on the picture plane. In Figure 13 we have the object point P^o, the picture plane PP, and the observer's eye E. Assuming light travels in a straight line, the image of P^o is at P_{pp}. For all calculations we shall use a Cartesian system of coordinates with the origin on the line from E perpendicular to PP; the Z-axis will be along this same line and the coordinate of PP will be larger than the Z-coordinates of E. Let X' and Y' be the coordinates on the plotting paper. We note that by similar triangles

$$\frac{X'_{pp}}{X_o} = \frac{DP}{DO + Z_o}$$

and

$$\frac{Y'_{pp}}{Y_o} = \frac{DP}{DO + Z_o}$$

Solving for the needed quantities we get:

$$X'_{pp} = \frac{DP}{DO + Z_o} X_o$$

$$Y'_{pp} = \frac{DP}{DO + Z_o} Y_o$$

These are the coordinates (X', Y') of the image point P_{pp} in the desired picture plane. If this mathematical operation is performed on each end-point of the straight line segments, a two-dimensional image will be formed of the three-dimensional object. The same calculations are performed on the starting point of alphameric information to keep the starting point at the same point relative to the drawing.

As can be seen in Figure 13, the magnification of the reproduction will be DP/DO. A perspective drawing can be achieved by making DP and DO small. The drawing will become an orthographic projection if both DP and DO are made very large.

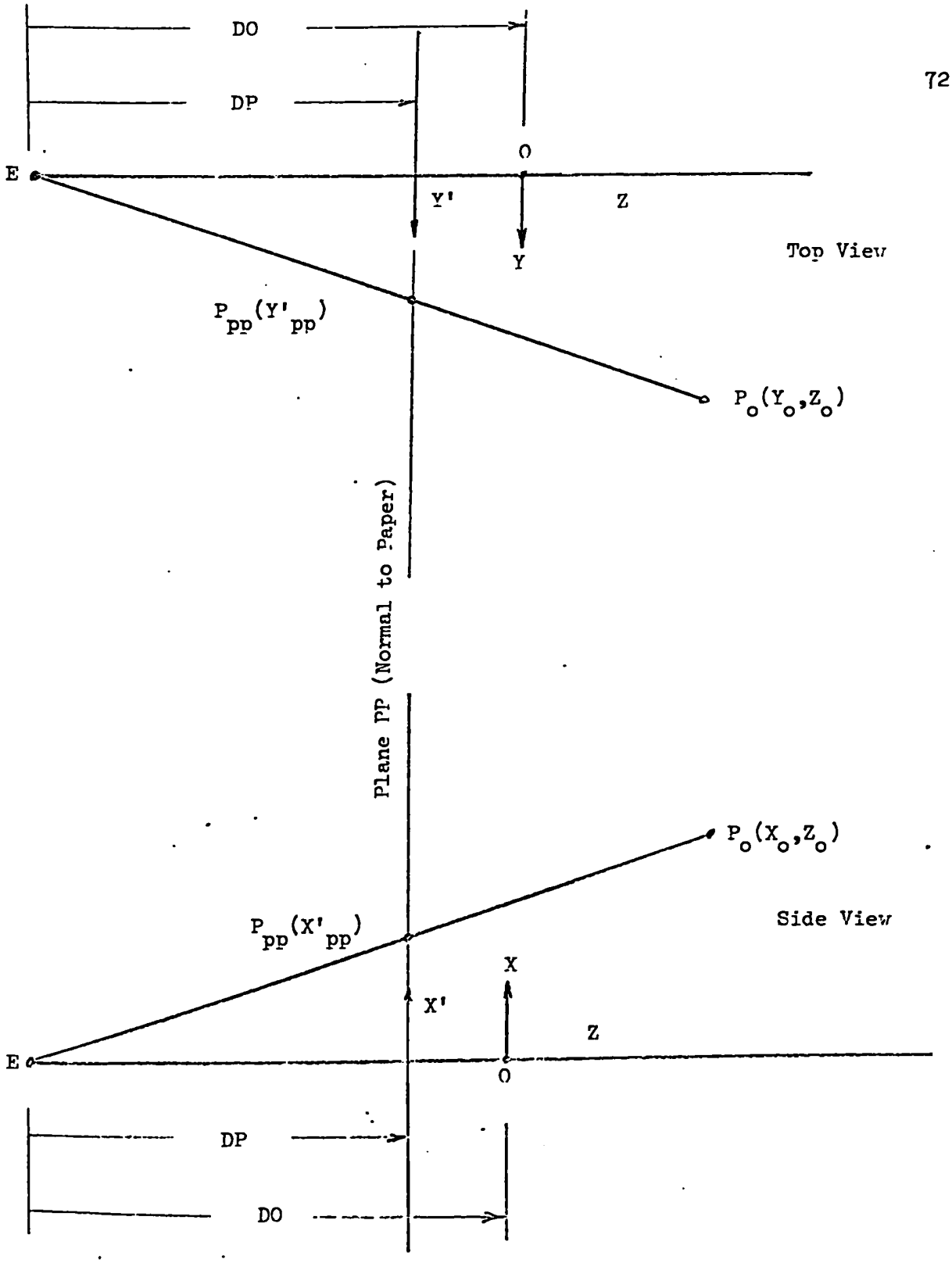


Figure 13. Perspective Geometry

C.2 Modifications to the Programme DRAFT

The programme DRAFT had to be modified to make it suitable for plotting isometric piping drawings, and to make it compatible with the EAI 3500 dataplotter. The principal subroutine of the EAI 3500 plotting package is PLOT. The subroutines for alphameric output are SCRIBE and PRINT, where SCRIBE uses the pen for output and PRINT the 48-character printer head.

The calling sequence for the subroutine PLOT takes the following form:

```
CALL PLOT (OPCODE, X, Y)
```

where OPCODE is an integer representing a PLOT function and X and Y are parameters whose interpretation depends on OPCODE.

The operation of DRAFT will be described in the sequence in which operations occur.

```
CALL PLOT (34, XCOR, YCOR)
```

By this statement, the lower left-hand corner of the permissible plotting surface is set to the coordinates XCOR, YCOR in inches relative to the lower left-hand corner of the plotting board.

```
CALL PLOT (31, 4)
```

This sets the error procedure to procedure 4, where an error message will be printed and control returned to the user's programme, if for instance the pen will be driven beyond the limits of the plotting surface.

```
CALL PLOT (32, 1)
```

This sets the units to inches, so that the coordinates will be interpreted to be in inches.

CALL PLOT (35, 30, 22.5)

The dimensions of the permissible plotting area are set to 30 by 22.5 inches in the X and Y directions, respectively.

At this point in the programme, the basic PLOT parameters have been set and the data may be read in. The coordinates are read in with the FORMAT (6F10.5) and stored in the A-array. Alphameric data is signalled by the proper control card and read in with FORMAT (3F10.5,5A6). The first three fields on the data input card contain the coordinates of the starting point and the next two fields, the base angle and character slant in degrees. The five words in the A-field designation allow a maximum of thirty characters to be read and stored in a separate array C. The end of the data is signalled by another control card.

Following the input section of the programme, the three dimensional coordinates of all end-points must be rotated into isometric form (see Figure 7). To obtain the isometric form, the following rotations were found necessary:

Rotate about X-axis by 180°

Rotate about Z-axis by -45°

Rotate about X-axis by 54.7356°

These rotations were performed on the coordinates of all the points, including the starting points of the alphameric data. These three-dimensional points then have to be transformed into two-dimensional points so they can be plotted. This transformation is accomplished by

the following mathematics:

$$X^* = \frac{X}{Z + DO} DP$$

$$Y^* = \frac{Y}{Z + DO} DP$$

By making DP (distance from the eye to the picture plane) and DO (distance from the eye to the object origin) both large, the resulting coordinates will be in projection. As these coordinates (X*, Y*) are computed, they are scanned for the largest and smallest values. The scaling factors are then computed from these limits so the drawing will fit in a 16" by 12" format.

```
XSCALE = (BIGX - SMAX)/16.
YSCALE = (BIGY - SMAY)/12.
IF (XSCALE.GT.YSCALE) YSCALE = XSCALE
IF (YSCALE.GT.XSCALE) XSCALE = YSCALE
```

The scaling factors are set equal to the largest value. This is necessary to prevent distortion from the isometric form.

Using these computed scaling factors and the limits in the coordinate directions, it is now possible to compute the data origin to centre the drawing in the 16" by 12" format. The programming for this can be written as

```
XORIG = 15.0  - ((BIGX+SMAX)/12.)/XSCALE
YORIG = 11.35 - ((BIGY+SMAY)/12.)/YSCALE
```

These parameters are then set in the subroutine PLOT by the statements

```
CALL PLOT (36, XORIG, YORIG)
CALL PLOT (37, XSCALE, YSCALE)
```

where XORIG and YORIG are the X and Y coordinates of the data origin relative to the paper corner and XSCALE and YSCALE are the scaling factors in user's units per inch of board surface.

Now that these parameters have been set, the two-dimensional coordinates are again computed from the three-dimensional coordinates and placed on magnetic tape along with plotter instructions. For each plotting mode, different parameters have to be set and these are called by specific values appearing on the control cards. A control card is signalled by the first variable being '99999.'. The variable KTYPE is set equal to A(2,I) which appears in the second field of the control card. NSIG is set equal to A(1,I+1) which appears in the fourth field. The value of NSIG determines which plotting mode to enter.

NSIG = 1

Line mode is entered and either the main lines or dimension lines will be plotted depending on the value of the variable KTYPE.

```
CALL PLOT (19, KTYPE)
```

This statement causes PLOT to choose the correct pen from the pen turret. If the control card signals main line plotting, KTYPE will have the value '2', indicating a medium width pen. If the control card preceeds dimension line data, KTYPE will have the value '3' and a narrow width pen will be chosen.

The PLOT subroutine is entered, and various other line mode parameters are set as indicated below:

CALL PLOT (15, 2)

The plotter is instructed to enter line mode (may draw lines).

CALL PLOT (16, 1)

The long line mode is chosen (lines segmented into long lengths).

CALL PLOT (33, 1.0)

All lines will be divided into one inch lengths to insure straight lines between coordinates.

CALL PLOT (2, X1, Y1)

X1, Y1 are the coordinates of the starting point of a line and they are calculated by the mathematics described above. The plotter is instructed to lift the pen and move to this position using the calculated scaling factor.

CALL PLOT (3, X2, Y2)

X2, Y2 are the coordinates of the end-point of the line. The plotter is instructed to lower the pen and draw a straight line to this point using the selected scaling factor.

The last two commands are repeated for all the pairs of points until another control card is read.

NSIG = 2

If the control card sets NSIG equal to 2, the plotting package is instructed to plot the succeeding points as welding dots. These instructions are described below:

CALL PLOT (19, KTYPE)

In this case KTYPE will have the value of '1', which is the instruction

to pick the wide pen for plotting.

```
CALL PLOT (15, 1, 1)
```

By this statement, the plotter is instructed to enter the point mode and plot points with the most recently selected pen.

```
CALL PLOT (1, X1, Y1)
```

The pen moves to the point X1, Y1, using the scaling factor and plots a point at this position.

```
CALL PLOT (1, X2, Y2)
```

The procedure is repeated for the second point. These two commands are repeated until another control card is encountered.

NSIG = 3

If it is desired to use the printer head to print some alphameric information, A(1,I+1) of the control card will take on the value '3'. Then each succeeding card will contain the X, Y, Z coordinates of the starting point of the printed information, the orientation of the printing, and the information to be printed. The orientation can be either '0' for horizontal or '1' for vertical. From this the X-incrementation and Y-incrementation between letters is computed by the following statements:

```
      LLL  =  A(1,I+1)  (orientation)
      IF (LLL.EQ.0) GO TO 595
      XINC  =  0.0
      YINC  =  0.1
      GO TO 596
  595 XINC = 0.1
      YINC = 0.0
```

A line of data will be printed by the following statement:

CALL PRINT (C(1,MM),30,LLL,X1,Y1,XINC,YINC)

Here C(1,MM) is the first word of the alphameric information and 30 indicates the next 30 characters of storage are to be printed. LLL indicates the orientation, and X1, Y1 are the starting point coordinates in inches calculated from the X, Y, Z coordinates by DRAFT.

NSIG = 4

The value of NSIG = 4 read from the control card will cause the programme to enter the SCRIBE subroutine and plot the alphameric information according to the succeeding card instructions. Each card contains the X, Y, Z coordinates of the starting point, the baseangle, the slant and the alphameric information to be plotted. The starting point is computed by DRAFT and then scaled to the same units as the plotted data. The SCRIBE subroutine is called from the plotting package by

CALL SCRIBE (C(1,MM),30,X1,Y1,0.10,A(1,I+1),A(2,I+1))

Here C(1,MM) is again the first word of the alphameric information, 30 indicates there are 30 characters to be plotted, X1, Y1 are the coordinates of the starting point, 0.10 is the height of the characters in inches, A(1,I+1) is the base angle, and A(2,I+1) the slant of the characters.

NSIG = 0

A control card with NSIG = 0 signals the end of data to the programme.

C.3 Computer Listing of the Modified Programme DRAFT

For a detailed description of the programme DRAFT as written by the Grumman Aircraft Corporation see reference [3]. The modifications have been described in Section C.2. In the listing that follows, some of the modifications made have been briefly described.

```
$JOB            003711 PAULS            060   010   030
$IBJOB                                                          81
$IBFTC DRAFT
      XCOR = 0.0
      YCOR = 22.5
    5 CALL PLOT (34,XCOR, YCOR)
      DIMENSION A(3,2000),B(3,2000),C(5,50),DATA(1005)
      DATA MVR,NVR/3,2000/                                      DRAF102
      CALL PLOT (31, 4)
      CALL PLOT (32, 1)
      CALL PLOT (35,30.,22.5)
      DO 10 I = 1,MVR                                           DRAF106
      DO 10 J = 1,NVR                                           DRAF107
      A(I,J) = 0.0                                              DRAF108
   10 B(I,J) = 0.0                                              DRAF109
      DO 15 I = 1,5
      DO 15 J = 1,50
   15 C(I,J) = 0.0
      NN = 1
      M = 1
      NSIG = 0                                                  DRAF110
      MM = 1
      BIGX = -1000.
      BIGY = -1000.
      SMAX = +1000.
      SMAY = +1000.
      D1 = 0.0                                                  DRAF112
      D2 = 0.0                                                  DRAF113
      D3 = 0.0                                                  DRAF114
      D4 = 0.0                                                  DRAF115
```

(set all the array locations equal to zero)

```
      E = 0.0                                                          DRAF1160
      F = 0.0                                                          DRAF1170
   20 READ (5,30)N                                                     DRAF1180
   30 FORMAT(I3)                                                       DRAF1190
      GO TO(40,120,120,120,120,120,120,220,220,690,690,740,750),N      DRAF1200
   40 K=2                                                              DRAF1210
   50 READ (5,130)A(1,K-1),A(2,K-1),A(3,K-1),A(1,K),A(2,K),A(3,K)      DRAF1220
      IF (A(1,K-1).LT.99999.) GO TO 100                                DRAF1230
   60 IF (A(1,K).LT.3.) GO TO 90
   70 K = K + 2                                                        DRAF1250
      IF (K.GT.NVR) GO TO 110                                          DRAF1260
      READ (5,80) A(1,K-1),A(2,K-1),A(3,K-1),A(1,K),A(2,K),C(1,M),C(2,M)
     1,C(3,M),C(4,M),C(5,M)
   80 FORMAT (3F10.5,2F10.3,5A6)
      M = M + 1
      IF (A(1,K-1)-99999.) 70,60,60
   90 IF (A(2,K-1).EQ.0.) GO TO 110
  100 K = K + 2                                                        DRAF1300
      IF (K.GT.NVR) GO TO 110                                          DRAF1310
      GO TO 50                                                         DRAF1320
  110 K = K - 2                                                        DRAF1330
      GO TO 20                                                         DRAF1340
  120 READ(5,130) THETA                                                DRAF1350
  130 FORMAT(6F10.5)                                                   DRAF1360
      THETA = THETA / 57.295795                                        DRAF1370
      GO TO (40,140,150,160,140,150,160),N                             DRAF1380
  140 D1=1.                                                            DRAF1390
      D2=0.                                                            DRAF1400
      D3=0.                                                            DRAF1410
```

```
      D4=0.                                                          DRAF1430
      D5=COS(THETA)                                                  DRAF1440
      D6=-SIN(THETA)                                                 DRAF1450
      D7=0.                                                          DRAF1460
      D8=-D6                                                         DRAF1470
      D9=D5                                                          DRAF1480
      GO TO 170                                                      DRAF1490
  150 D1=COS(THETA)                                                  DRAF1500
      D2=0.                                                          DRAF1510
      D3=SIN(THETA)                                                  DRAF1520
      D4=0.                                                          DRAF1530
      D5=1.                                                          DRAF1540
      D6=0.                                                          DRAF1550
      D7=-D3                                                         DRAF1560
      D8=0.                                                          DRAF1570
      D9=D1                                                          DRAF1580
      GO TO 170                                                      DRAF1590
  160 D1=COS(THETA)                                                  DRAF1600
      D2=-SIN(THETA)                                                 DRAF1610
      D3=0.                                                          DRAF1620
      D4=-D2                                                         DRAF1630
      D5=D1                                                          DRAF1640
      D6=0.                                                          DRAF1650
      D7=0.                                                          DRAF1660
      D8=0.                                                          DRAF1670
      D9=1.                                                          DRAF1680
  170 IF(N-4) 180,180,200                                            DRAF1690
  180 DO 190 I=1,K                                                   DRAF1700
      B(1,I)=D1*A(1,I)+D2*A(2,I)+D3*A(3,I)                           DRAF1710
```

```
      B(2,I)=D4*A(1,I)+D5*A(2,I)+D6*A(3,I)                          DRAF1720
  190 B(3,I)=D7*A(1,I)+D8*A(2,I)+D9*A(3,I)                          DRAF1730
      GO TO 20                                                      DRAF1740
  200 DO 210 I=1,K                                                  DRAF1750
      E      =D1*B(1,I)+D2*B(2,I)+D3*B(3,I)                         DRAF1760
      F      =D4*B(1,I)+D5*B(2,I)+D6*B(3,I)                         DRAF1770
      G      =D7*B(1,I)+D8*B(2,I)+D9*B(3,I)                         DRAF1780
      B(1,I)=E                                                      DRAF1790
      B(2,I)=F                                                      DRAF1800
  210 B(3,I)=G                                                      DRAF1810
      GO TO 20                                                      DRAF1820
  220 READ (5, 130) DP, DO                                          DRAF1830
      IF(N-8) 250,250,240                                           DRAF1840
  240 N=2                                                           DRAF1850
      GO TO 260                                                     DRAF1860
  250 N=1                                                           DRAF1870
  260 DO 680 I=1,K,2                                                DRAF1880
      IF (A(1,I).LT.99999.) GO TO 300
      KTYPE=A(2,I)                          (choose proper pen)     DRAF1940
      NSIG = A(1,I+1)
      IF (NN.EQ.1) GO TO 400
      CALL PLOT (19, KTYPE)
      GO TO (280,290,680,680),NSIG
  280 CALL PLOT (15,2)
      CALL PLOT (16,1)
      CALL PLOT (33, 1.0)
      GO TO 680
  290 CALL PLOT (15,1,1)
      CALL PLOT (19, KTYPE)
```

```
      GO TO 630                                                    DRAF2010
  300 GO TO(310,320),N                                             DRAF2020
  310 D1=A(1,I)                                                    DRAF2030
      D2=A(1,I+1)                                                  DRAF2040
      D3=A(2,I)                                                    DRAF2050
      D4=A(2,I+1)                                                  DRAF2060
      D5=A(3,I)                                                    DRAF2070
      D6=A(3,I+1)                                                  DRAF2080
      GO TO 330                                                    DRAF2090
  320 D1=B(1,I)                                                    DRAF2100
      D2=B(1,I+1)                                                  DRAF2110
      D3=B(2,I)                                                    DRAF2120
      D4=B(2,I+1)                                                  DRAF2130
      D5=B(3,I)                                                    DRAF2140
      D6=B(3,I+1)
  330 IF (NSIG.LT.3) GO TO 350                                     DRAF2150
      D2=D1                                                        DRAF2160
      D4=D3                                                        DRAF2170
      D6=D5                                                        DRAF2180
  350 E=(D5+DC)/DP                                                 DRAF2190
      F=(D6+DC)/DP                                                 DRAF2200
      X1 = D1/E
      Y1 = D3/E                                                    DRAF2500
      X2 = D2/F                                                    DRAF2510
      Y2 = D4/F                                                    DRAF2520
      IF (NN.GT.1) GO TO 560
      IF (X1.GT.BIGX) BIGX = X1
      IF (X2.GT.BIGX) BIGX = X2
      IF (X1.LT.SMAX) SMAX = X1
```

```
      IF (X2.LT.SMAX) SMAX = X2
      IF (Y1.GT.BIGY) BIGY = Y1
      IF (Y2.GT.BIGY) BIGY = Y2
      IF (Y1.LT.SMAY) SMAY = Y1
      IF (Y2.LT.SMAY) SMAY = Y2
  400 IF (I.LT.K-1) GO TO 680
      XSCALE = (BIGX - SMAX)/16.
      YSCALE = (BIGY - SMAY)/12.
      IF (XSCALE.GT.YSCALE) YSCALE = XSCALE
      IF (YSCALE.GT.XSCALE) XSCALE = YSCALE
      XORIG = 15.0 - ((BIGX + SMAX)/2.)/XSCALE
      YORIG = 11.25- ((BIGY + SMAY)/2.)/YSCALE
      WRITE (6,800) XSCALE
  800 FORMAT (1H0,16HSCALE FACTOR IS ,E16.6)
      WRITE (6,810) XORIG, YORIG
  810 FORMAT (1H ,11HX-ORIGIN = ,E16.6,10X,11HY-ORIGIN = ,E16.6)
      CALL PLOT (36,XORIG,YORIG)
      CALL PLOT (37,XSCALE,YSCALE)
      GO TO 680
  560 GO TO (570,580,590,600),MSIG
  570 CALL PLOT (2, X1, Y1)
      CALL PLOT (3, X2, Y2)
      GO TO 680
  580 CALL PLOT (4,X1,Y1)
      CALL PLOT (4,X1,Y1)
      CALL PLOT (4,X1,Y1)
      CALL PLOT (4,X2,Y2)
      CALL PLOT (4,X2,Y2)
      CALL PLOT (4,X2,Y2)
```

(find maximum and minimum coordinates in the x- and y-directions)

(compute scaling factor)

(compute origin)

(plot straight line between points (x_1, y_1) and (x_2, y_2))

(plot symbol for welded joint at (x_1, y_1) and at (x_2, y_2))

```
      GO TO 680
  590 X1 = X1/XSCALE + XORIG
      Y1 = Y1/YSCALE + YORIG
      LLL = A(1,I+1)
      IF (LLL.EQ.0) GO TO 595
      XINC = 0.0
      YINC = 0.1
      GO TO 596                                  (plot labels using PRINT routine)
  595 XINC = 0.1
      YINC = 0.0
  596 CALL PRINT (C(1,MM),30,LLL,X1,Y1,XINC,YINC)
      MM = MM + 1
      GO TO 680
  600 X1 = X1/XSCALE + XORIG
      Y1 = Y1/YSCALE + YORIG                     (plot dimensioning labels using
                                                  SCRIBE routine)
      CALL SCRIBE (C(1,MM),30,X1,Y1,0.10,A(1,I+1),A(2,I+1))
      MM = MM + 1
  680 CONTINUE                                                    DRAFC880
      NN = NN + 1
      IF (NN.EQ.2) GO TO 260
      GO TO 20                                                    DRAFC8[illegible]
  690 READ(5,130) DX,DY,DZ                                        DRAFC[illegible]
      IF(N-10) 700,700,720                                        DRAFC[illegible]
  700 DO 710 I=1,K                                                DRAFC[illegible]
      B(1,I)=A(1,I)+DX                                            DRAFC[illegible]
      B(2,I)=A(2,I)+DY                                            DRAFC[illegible]
  710 B(3,I)=A(3,I)+DZ                                            DRAFC[illegible]
      GO TO 20                                                    DRAFC[illegible]
  720 DO 730 I=1,K                                                DRAFC[illegible]
```

```
      B(1,I)=B(1,I)+DX                                                  DRAF2930
      B(2,I)=B(2,I)+DY                                                  DRAF2940
  730 B(3,I)=B(3,I)+DZ                                                  DRAF2950
      GO TO 2                                                           DRAF2960
  740 READ (5,130) XCOR, YCOR          (set up starting point
      GO TO 5                           of new plot)
  750 CONTINUE
      STOP                                                              DRAF3000
      END                                                               DRAF3010
```

APPENDIX D

Material List (Recommendations)

In incorporating the production of the material list in the Configuration Generator Programme, the first step will be drawing up specifications for each fitting, concerning size, material and type of face end, as well as associated hardware such as gaskets, bolts and nuts. Then counters will be coded to each of these fittings and the various pieces of hardware. These counters have to all be set to zero at the beginning of the programme. Within each subsection of the programme, where the symbols representing the various fittings are generated, additional programming is necessary to increment the appropriate counters when a particular fitting is called.

If, for example, the code representing a flanged globe valve were KFGVAL, then the following statement would be inserted in the subsection NN = 3:

```
KFGVAL  =  KFGVAL + 1
```

If this fitting were to be installed with two four inch gaskets and the symbol for four inch gaskets were KGAS4, then the following statement would store this in the computer's memory:

```
KGAS4  =  KGAS4 + 2
```

This type of logic could be programmed in each subsection with overrides provided from the input data.

Upon completion of the input data, the material list could be punched out, with an appropriate control card preceeding.

Carrying through the example of the flanged globe valve, its output could be of the form:

```
      WRITE (7, 1200) KFGVAL
1200 FORMAT (30X, I3, 2X, 21H FLANGED GLOBE VALVES)
      WRITE (7, 1300) KGAS4
1300 FORMAT (30X, I3, 2X, 15H 4 INCH GASKETS)
```

The programme DRAFT would then be instructed to read this data by an A-format and store it as alphameric data in the array C. (DRAFT allows a maximum of 30 characters per line of alphameric data.) Upon entering the output stage, the starting coordinates of the first line can be set to X = 12 and Y = 12. The succeeding lines can be incremented down the page by the statement

```
Y = Y + 0.5
```

The plotter can then be instructed to plot this data by the statement

```
CALL PRINT (C(1,I), 30, 0, X, Y, 0.1, 0.0)
```

This will then be repeated until all the lines of the material list have been printed out.

APPENDIX E

Rotational Subroutine (Recommendations)

At the present stage of development of the plotting system, the Configuration Generator Programme may generate symbols, dimension lines and labeling (SCRIBE) in coordinate directions only. Occasionally situations are encountered where sections of piping occur in other than the coordinate directions. This special case can be taken care of by incorporating in the programme the rotational section of DRAFT. This has been called the subroutine ROTATE. The following information is required by this subroutine to rotate a particular segment of the drawing stored in the A, B, D and C arrays: THETA, NAXIS, KSTART, K, MSTART, M, LNSTAR, LN, MNSTAR, MN. Here NAXIS refers to the axis about which the rotation is to take place (1, 2 or 3); THETA is the angle in degrees of the desired rotation. These two parameters will have to be added to the input sheet to bring them under the user's control. The variables KSTART, MSTART, LNSTAR and MNSTAR will have to be defined by the Configuration Generator Programme. These variables could be redefined at the start of generation of each line or symbol, thus always available if that section is to be rotated. The variables K, M, LN and MN are the current indices of the arrays A, B, D and C and thus become the upper limits of the sections to be rotated. After all the sections of the above arrays have been generated, the subroutine may be called from the main programme by the following statement:

```
IF (NAXIS.NE.0) CALL ROTATE
```

The subroutine will rotate all the desired sections of storage and store each entry in its same location.

Subroutine ROTATE

This subroutine is taken directly from the rotational section of the programme DRAFT with only the input and output sections altered for this special application. The variables described above are transferred to the subroutine by a COMMON statement. Depending upon the axis about which the rotation is to occur (NAXIS) and the angle of rotation (THETA), the variables D1, D2, D3, D4, D5, D6, D7, D8, D9 are set. These variables are the coefficients used in the rotational matrix (see Appendix C). The desired rotation will in the general case not be about the origin (0, 0, 0), but rather about some point P_1 (x_1, y_1, z_1) (see Figure 14). The way the mathematics is written, the rotation must occur about the origin; thus the origin has to be shifted to the point P_1. This is done by subtracting the coordinates (x_1, y_1, z_1) of the point P_1 from the coordinates of each point to be rotated. Then the rotation can be performed and the new coordinates temporarily stored as (U, V, W). The final rotated coordinates are computed by adding the coordinates (x_1, y_1, z_1) of the point P_1 to the coordinates (U, V, W). This procedure is repeated for all the points to be rotated of the A, B, D and C arrays. Control is then returned to the Configuration Generator Programme.

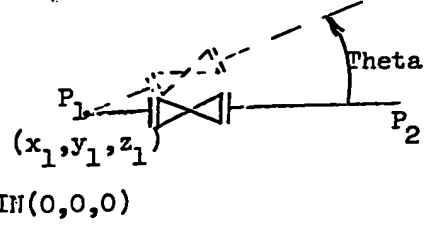


Figure 14. Desired Rotation of Section P_1 P_2, by Angle THETA, about Point P_1

The subroutine ROTATE described above should increase the versatility of the Configuration Generator Programme by allowing data to be plotted at angles other than coordinate directions. The problem of plotting alphameric data at angles other than isometric will, however, be encountered and it may be necessary to use the PRINT subroutine as opposed to using SCRIBE. The reason for this is that, as described in Appendix B,6, the programme will automatically SCRIBE in isometric directions but cannot handle non-isometric directions. It would be possible to use the PRINT subroutine and print the data either horizontally or vertically.

```
      SUBROUTINE ROTATE
      COMMON A,B,C,D,THETA,MAXIS,KSTART,K,MSTART,M,LNSTART,LN,MNSTART
      THETA = THETA / 57.295795
      GO TO (140,150,160),MAXIS
  140 D1=1.
      D2=0.
      D3=0.
      D4=0.
      D5=COS(THETA)
      D6=-SIN(THETA)
      D7=0.
      D8=-D6
      D9=D5
      GO TO 170
  150 D1=COS(THETA)
      D2=0.
      D3=SIN(THETA)
      D4=0.
      D5=1.
      D6=0.
      D7=-D3
      D8=0.
      D9=D1
      GO TO 170
  160 D1=COS(THETA)
      D2=-SIN(THETA)
      D3=0.
      D4=-D2
      D5=D1
      D6=0.
```

```
      D7=0.
      D8=0.
      D9=1.
  170 DO 180 I = KSTART,K
      A(1,I) = A(1,I) - A(1,KSTART)
      A(2,I) = A(2,I) - A(2,KSTART)
      A(3,I) = A(3,I) - A(3,KSTART)
      U      =D1*A(1,I)+D2*A(2,I)+D3*A(3,I)
      V      =D4*A(1,I)+D5*A(2,I)+D6*A(3,I)
      W      =D7*A(1,I)+D8*A(2,I)+D9*A(3,I)
      A(1,I) = U + A(1,KSTART)
      A(2,I) = V + A(2,KSTART)
  180 A(3,I) = W + A(3,KSTART)
      DO 190 I = KSTART,L
      B(1,I) = B(1,I) - A(1,KSTART)
      B(2,I) = B(2,I) - A(2,KSTART)
      B(3,I) = B(3,I) - A(3,KSTART)
      U      =D1*B(1,I)+D2*B(2,I)+D3*B(3,I)
      V      =D4*B(1,I)+D5*B(2,I)+D6*B(3,I)
      W      =D7*B(1,I)+D8*B(2,I)+D9*B(3,I)
      B(1,I) = U + A(1,KSTART)
      B(2,I) = V + A(2,KSTART)
  190 B(3,I) = W + A(3,KSTART)
      DO 200 I = LNSTART,LN
      D(1,I) = D(1,I) - A(1,KSTART)
      D(2,I) = D(2,I) - A(2,KSTART)
      D(3,I) = D(3,I) - A(3,KSTART)
      U      =D1*D(1,I)+D2*D(2,I)+D3*D(3,I)
      V      =D4*D(1,I)+D5*D(2,I)+D6*D(3,I)
```

```
      W      =D7*D(1,I)+D8*D(2,I)+D9*D(3,I)
      D(1,I) = U + A(1,KSTART)
      D(2,I) = V + A(2,KSTART)
  200 D(3,I) = W + A(3,KSTART)
      DO 210 I = MKSTART,IM1
      C(1,I) = C(1,I) - A(1,KSTART)
      C(2,I) = C(2,I) - A(2,KSTART)
      C(3,I) = C(3,I) - A(3,KSTART)
      U      =D1*C(1,I)+D2*C(2,I)+D3*C(3,I)
      V      =D4*C(1,I)+D5*C(2,I)+D6*C(3,I)
      W      =D7*C(1,I)+D8*C(2,I)+D9*C(3,I)
      C(1,I) = U + A(1,KSTART)
      C(2,I) = V + A(2,KSTART)
  210 C(3,I) = W + A(3,KSTART)
      RETURN
      END
```

APPENDIX F

Subprogramme to Generate Arc between Two Perpendicular Lines

In order to produce computer generated drawings more closely resembling manually produced drawings, it was decided to add a subprogramme to generate arcs depicting 90° bends between pairs of perpendicular lines. The information required for this subprogramme was found to be: location in storage of the coordinates (3-dimensional) defining the end-points of the two lines, the directions of these two lines, the radius of the bend and the number of points desired around the bend. After reading in these variables, the first step is to see in which direction these lines run and add or subtract the length of the radius 'R' from one end of each line to leave a space for the arc. These coordinates also define the centre of curvature of the arc (see Figure 15).

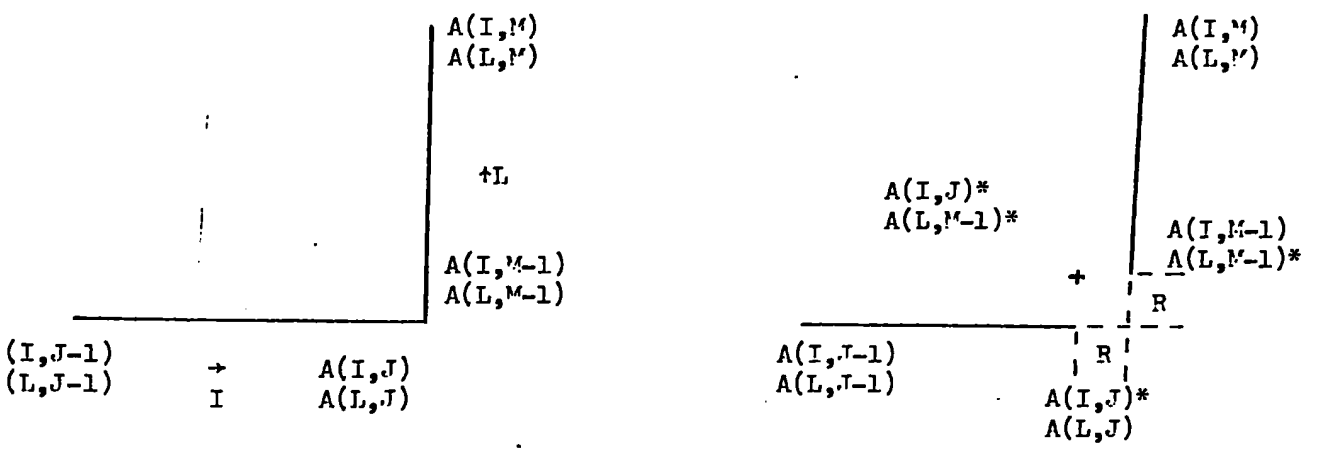
As the Configuration Generator Programme is set up, the lines can only be in coordinate directions, and since the subprogramme has computed the centre of curvature, the 90° arc can be generated in four different positions (see Figure 15). Beginning with $(\theta+\Delta\theta)^\circ$ and $(\theta+\Delta\theta)^\circ$, coordinates of the first point are computed by the statements:

$$\begin{aligned} A(I,K) &= A(I,J) + R*\cos(\theta+\Delta\theta) \\ A(L,K) &= A(L,M-1) + R*\sin(\theta+\Delta\theta) \end{aligned} \qquad (1)$$

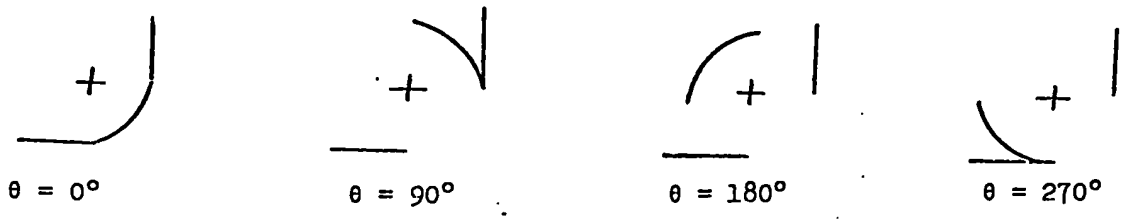
These coordinates are compared with the altered end-point coordinates of both the lines. If the comparison proves this point of the arc to coincide (within R/10), with either line end-point, then the coordinates of the points around the curve are computed. If the first point of the arc does not coincide with either line end-point, θ is increased by

90° and the comparison repeated. Once the beginning of the arc is found to coincide with a line end-point, the rest of the arc is generated by incrementing $\Delta\theta$ from 0 to 90° (Eq. (1)). The end point of the arc, when $\Delta\theta = 90°$, is compared with both end-points of the lines. If the arc end-point coincides with one of the line end-points, we have the correct arc. If not, θ is again incremented by 90° and control returned to the beginning of the loop.

Since the arc can only be generated in four ways with one of these arcs being the correct one, the computations are quite fast. It was found that ten points on the arc produced a reasonably smooth curve. The disadvantage, however, was that ten additional storage locations were required for each arc. It was decided that the additional cost in computer and plotter time were not justified in the slight improvement in the plot. For this reason this section was deleted.



(a) Modifications required to a Pair of Lines to leave Room for a 90° Arc



(b) Possible Ways to Generate Arc, Given Centre of Curvature and Plane of Pair of Lines

Figure 15. Subroutine to Generate Arc between Two Perpendicular Lines

```
C     SUBPROGRAM TO GENERATE ARC BETWEEN TWO PERPENDICULAR LINES
  100 READ (5,105) I,J,L,Y,R,N                                          100
  105 FORMAT (4I5,F10.0,I5)
C     'I' REFERS TO DIRECTION OF FIRST LINE, 'J' REFERS TO STORAGE
C     LOCATION OF FIRST LINE, 'L' AND 'Y' REFER TO DIRECTION AND
C     LOCATION OF SECOND LINE, R = RADIUS, N = NO. OF POINTS.
      THETA = 0.0
      IF (I.EQ.L) GO TO 20
      B = R*.1
      IF (I.EQ.ITEMP.AND.J.EQ.JTEMP) GO TO 130
C     COMPUTE NEW END POINTS OF LINES LEAVING SPACE FOR ARC BETWEEN
C     THE TWO LINES.
      IF (A(I,J)-A(I,J-1)) 120,130,110
  110 A(I,J) = A(I,J) - B
      GO TO 130
  120 A(I,J) = A(I,J) + B
  130 IF (A(L,Y)-A(L,Y-1)) 150,160,140
  140 A(L,Y-1) = A(L,Y-1) + B
      GO TO 160
  150 A(L,Y-1) = A(L,Y-1) - B
  160 CONTINUE
      DO 250 II = 1,N
      CII = FLOAT (II)-1.0
      CN = FLOAT (N)-1.0
      D = THETA + (90.*CII/CN)/57.295795
C     COMPUTE POINTS ON ARC.
      A(I,K) = A(I,J) + R*COS(D)
      A(L,K) = A(L,Y-1) + R*SIN(D)
      IF (I.NE.1.AND.L.NE.1) A(1,K) = A(1,J)
      IF (I.NE.2.AND.L.NE.2) A(2,K) = A(2,J)
```

```
      IF (I.NE.3.AND.L.NE.3) A(3,K) = A(3,J)
      A (1,K+1) = A(1,K)
      A (2,K+1) = A(2,K)
      A (3,K+1) = A(3,K)
      IF (CII.GT.0.) GO TO 210
C     SEE IF FIRST POINT OF ARC CORRESPONDS WITH EITHER END POINT
C     OF THE TWO LINES
      IF ((ABS(A(I,K)-A(I,J))).GT.B.OR.(ABS(A(L,K)-A(L,J))).GT.B) GO
     1TO 180
      GO TO 200
  180 IF((ABS(A(I,K)-A(I,K-1))).GT.B.OR.(ABS(A(L,K)-A(L,K-1))).GT.B) GO
     1TO 190
  200 A(1,K-1) = A(1,K)
      A(2,K-1) = A(2,K)
      A(3,K-1) = A(3,K)
      GO TO 250
C     IF NOT, INCREMENT THETA AND START OVER
  190 THETA = THETA + 90./57.265795
      IF (THETA.GT.6.)  GO TO 20
      GO TO 160
  210 IF (II.NE.N) GO TO 250
C     SEE IF LAST POINT OF ARC CORRESPONDS WITH EITHER END POINT OF LINES
      IF((ABS(A(I,K)-A(I,K-1))).GT.B.OR.(ABS(A(L,K)-A(L,K-1))).GT.B) GO
     1 TO 220
      GO TO 250
  220 IF ((ABS(A(I,K)-A(I,J))).GT.B.OR.(ABS(A(L,K)-A(L,J))).GT.B) GO
     1 TO 230
      GO TO 250
C     IF BOTH ABOVE CONDITIONS ARE SATISFIED WE HAVE THE CORRECT ARC,
```

```
C     IF NOT, INCREMENT THETA BY 90 DEGREES AND START AGAIN.
  230 THETA = THETA + 90./57.265795
      IF (THETA.GT.6.)  GO TO 20
      K = K - 2*(N-1)
      GO TO 160
  250 K = K + 2
      GO TO 20
```

References

1. Fetter, W.A.

Computer Graphics in Communication

McGraw-Hill 1965

2. Aicher, J.R.

Piping Isometric Drawings Via Computer Plotter

Research & Engineering, Sun Oil Co. October 1966

3. Stoodley, G.R.

An IBM 7094 Program for Making Perspective Drawings

Grumman Research Dept. CR 64-2 September 1964

4. Instruction Manual for EAI 3500 Dataplotter

Electronics Associates Incorporated 1966

5. Nuss, W.

Personal Communication with Mr. Nuss of Ontario Dept. of Highways

6. McCracken, D.D.

A Guide to Fortran IV Programming

John Wiley & Sons 1966

7. McKenzie, A.

Personal Communication with Mr. McKenzie of Beamer & Lathrop

1966