2-DIMENSIONAL DRAFTING SOFTWARES FOR MECHANICAL DRAWING

A DISSERTATION

submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
MECHANICAL ENGINEERING
(with specialization in Machine Design)

By

ALI REZA PASHMFOROOSH

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247667 (INDIA)

July, 1986

CERTIFICATE

Certified that the dissertation entitled "2-DIMENSIONAL DRAFTING SOFTWARES FOR MECHANICAL DRAWING" which is being submitted by ALI REZA PASHMFOROOSH is the partial fulfilment for the award of the Degree of Master of Engineering in MACHINE DESIGN of the UNIVERSITY OF ROORKEE, ROORKEE, is a record of student's own work carried out by him under our supervision and guidance. The matter embodied in this dissertation has not been submitted for the award of any other degree or diploma.

This is further to certify that he has worked for a period of about 13 months from June 1985 to July 1986 for preparing this dissertation for the Master of Engineering Degree at this University.

22.7.86.

(D.V.SINGH)
Professor & Head
Deptt of Mechanical &
Industrial Engineering,
University of Roorkee,
ROORKEE-247 667 INDIA

(S.C.JAIN)
Reader
Deptt of Mechanical &
Industrial Engineering,
University of Roorkee,
ROORKEE-247 667 INDIA

Roorkee:
Dated : July ,1986.

ACKNOWLEDGEMENT

I wish to express my deepest sense of gratitude to Dr. D.V.Singh, Professor and Head, Department of Mechanical & Industrial Engineering, University of Roorkee, Roorkee, for his valuable suggestions, and excellent guidance given to me at every stage of the present work.

Words do not suffice to express my feelings of appreciation and gratefulness to Dr. S.C.Jain, Reader, Department of Mechanical & Industrial Engineering for his valuable guidance, suggestions, encouragement and whole hearted co-operation throughout the entire duration of this work.

I have my humble thanks to my friends, Houshang , Cyrus, Kamal, Jamal, Sayed, Reza, Mostafa, Farhad, Masood, Ali for the constant encouragement and help. Special vote of thanks will definitely go to Mr. Jamshed Ali for typing the dissertation.

Roorkee ALI REZA PASHMFOROOSH

ABSTRACT

The drawing is the most important product of the design process in Mechanical Engineering. Drawing documentation constitutes a link between the design office and the manufacture. Traditional industrial drafters spend lot of time in the process of produce it and development including design. The implementation to CAD changes this pattern and alteration in drawings for modified design can be accomplished much more rapidly.

The work presented in this thesis develop variety of modules required in drawing in the orthographic views of an object. The programs algorithm is such that the portion to be drawn is identified and subsequently image is drawn/created, during the development of these module, portability facotr is also considered so that such a package should at the same time enable the application of plotters of various firms together with their basic software. As followed by most common graphical systems, in this thesis hierarchy approach is followed to draw the complex drawings.

Realizing the need of solid modelling in CAD/CAM activities, an initiation is taken to develop 3-D modules. In the present work simple brick modules and their assembly is demonstrated. To have infra structure for FEM grid generation and other integrated shapes of these modules.

The application of the method to some practical cases has shown that 2-D drafting can be achieved at high speed, satisfying requirements of an interactive CAD environment.

C O N T E N T S

CHAPTER			Page No.

.. ..

CHAPTER - I

INTRODUCTION

1.1 COMPUTER GRAPHICS

Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) are affecting almost every area of Engineering. CAD/CAM speeds the engineering process, stripping away the drudgery and paper work that inhibits productivity and creativity. CAD/CAM magnify man's mental power just as the machines of the industrial revolution expanded the strength of his muscles.

Computer graphics, rapidly growing in computer aided design and computer aided manufacturing, is one of the most visually spectacular branches of computer technology, producing images whose appearance and motion make them quite unlike any other form of computer output. This has been known for so many years that the human eye can absorb the information content of a displayed diagram or perfective view much faster than it can scan a table of numbers.

In the past the high cost of computer graphics technology has prevented its widespread use. Now the cost is dropping rapidly, and computer graphics is becoming available to more and more people.

Computer graphics provide a mean for creating the images, may be curves of tabular data, detail drawings showing the details of mechanical elements and displays of other objects of environments.

Computer graphic system as always done, is capable of freeing engineers from the tedious and time consuming tasks that have nothing to do with technical ingenuity.

1.2 INTERACTIVE GRAPHICS

The manipulative capability (interactive facility) of the computer graphic system, with the generated images by the user, has tremendously enhanced its uses in engineering applications.

Interactive manipulative graphics involves two way communication between computer and user. The computer, upon receiving signals from the input device, can modify the displayed picture appropriately.

To the user it appears that the picture is changing instantaneously in response to his commands. He can give a series of commands, each one generating a graphical response from the computer. In this way he maintains a conversation, or dialogue, with the computer.

1.2.a Historic Development

Several years earlier, a cathode-ray tube (CRT) has been used by the late F. Williams as an information storage device, this technique was to emerge years later, in the form of the storage CRT incorporated in many graphic display [11].

During the 1950's, computer graphics made little progress because the computers of that period were so unsuited to graphic use. Only towards the end of the decade, with the development of machines like MIT'S TX-0 and TX-2, did interactive computing become feasible, and interest in computer graphics then began to increase rapidly. By the mid 1960s large computer graphics research projects were under way at MIT, General Motors, Bell Telephone Laboratories and Lackheed Aircraft.

The single event that did most to promote interactive computer graphics as an important new field was the publication in 1962 of a brilliant thesis by Ivan E. Sutherland [8]. This thesis, entitled SKETCHPAD; A Man-Machine Graphical Communication System, proved to many readers that interactive computer graphics was a viable, useful, and exciting field of research. SKETCHPAD program which allowed the user to sketch on a cathode ray tube, opened the horizons for industrial developments by illustrating the feasibility of having a designer construct rather complex diagrams with the aid of a computer.

1.2.b A Friendly User Facility

The mathematics of design and design analysis is complex and repetitive, and so the degree to which the complexity is transparent to the user has a direct effect on the ease of use of the system. Replacing a complex geometric calculation with a simple graphical interaction on a screen is a good illustration of this practice. A simple but powerful menu structure, extensive prompting to help the user interact with the system and rapid graphical response to commands all help to reduce interruptions to the creative thought process.

Recently Reach (1986), developed, TECHNOVISION, accomplishing friendly user facility in its advanced mechanical CAD/CAM system [7].

1.3 COMPUTER AIDED DRAFTING

The drawing is the most important product of the design process in mechanical engineering. Drawing documentation constitutes a link between the design office and the manufacture. In manual design, the selection of constructional features alternates with making the drawings. In computer aided design, these two kinds of activities should be closely identified, separated and formalized. The ways of making drawings depend on the type of construction which is the object of design process. A considerable part of design process refers to the so-called variant constructions, which is a well-known concept. The designer must then work out the variant of construction corresponding to the input data of immediate interest, and at the same time make use of sketches in orthogonal projection. Most of the time spent is taken up by making of drawings. Winkler (1983), developed nodular construction of CAD drawings, in which the application of 2D automatic drawing modules are discussed [9].

Sometimes the application of variant construction proves to be insufficient and the design process is extended by the phase of working out the conception. In this phase, 3D models are often applied from which drawing in orthogonal projections are later made. The making of the 3D drawings can be computer-aided.

1.3.a Positive Aspects

Computer-aided drafting system relieve the drafter and designer from tedium. Hand made drawings are no longer required. CAD cannot, however, replace the individual. It cannot think for us. In fact, computer programmers must instruct computers with considerable detail. A CAD system should be thought of as an additional tool at user disposal. Consider it like a template which help user to draw more accurately and quickly. The computer is a tool, however, that performs at a high rate of speed. Drawings now can be revised and changed much more quickly and accurately than by hand. Thus, it is economically sound to let it prepare and revise drawings.

Reducing drafting time in a company is of prime importance. The drafting part of a project is considered to be bottleneck. Traditional industrial drafters spend approximately two-thirds of their time "laying lead", only one-third is spent for all the other job functions combined including design. The implementation of CAD changes this. Drawings and design changes can be accomplished much more rapidly. This results in quicker turn around time. Consequently, projects flow better through a company. The traditional drafting bottleneck is eliminated.

1.3.b Recognition of 3D Objects

Braid [5] and Aldefeld [1] reported two main approaches in CAD considered for defining the geometry of mechanical parts. The first of these is 3D oriented and it based on description and manipulation of elementary volumes with which, by such operations as union, difference and intersection, complex structures can be defined. The second approach is 2 D oriented and is based on the analogy with conventional engineering drawings.

1st Method :

In 3-D displays, the geometric modelling is synonymous with the creation of models by the combination of many simple primitive solids. A set of very basic solids, sometimes even just rectangular block, and cylinders, is provided as a starting point, and combinatorial operators such as addition and difference are used to build complex objects from duplicated and transformed primitives. This was the approach developed and published by Braid [5]. This approach has considerable theoretical attractions, notably the fact that with suitable rigorous definitions of the meaning of the primitives and the exact action of the combinatorial operators, a valid 3D solid is guaranteed as a result.

For example a very common feature of mechanical components is an external or internal fillet radius between adjacent plane faces. To apply such a fillet to a sharp external corner using blacks and cylinders, the corner must first be removed using the side or corner of a rectangular block, before adding a cylinder to provide the curve surface (Fig. 1.1).

2nd Method

The another possible way of specifying the geometry of mechanical parts in CAD follows a two-step procedure; input of a graphics - oriented 2 D representation corresponding of views and cuts; and interpretation of this data to give the explicit 3 D description. This topic is concerned with the problem of automating the interpretation step. The method of solution presented by Aldefelt [1] is based on viewing a part as consisting of several elementary objects and on recognising these from their specific patterns in the 2 D representation.

Fig. 1.2 gives a simplified overview of the main components of the reconstruction system, emphasizing the aspect that a number of process,

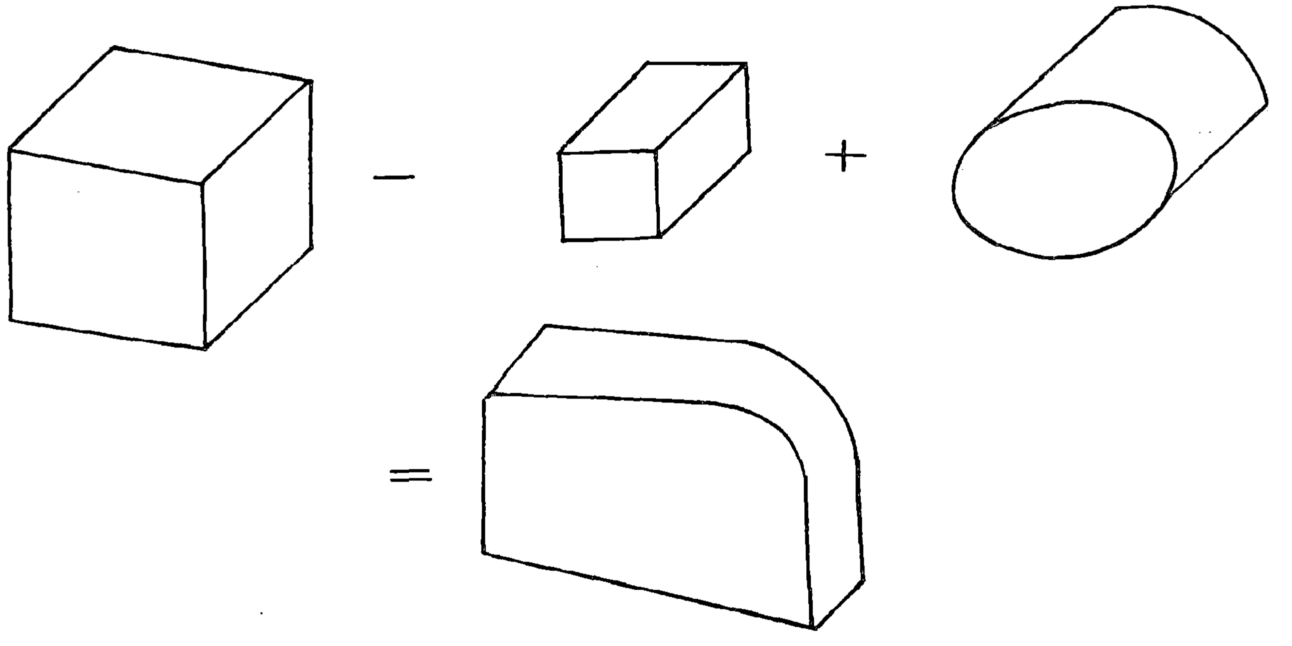


FIG.1.1. SYNTHESIS OF A FILLETED CORNER

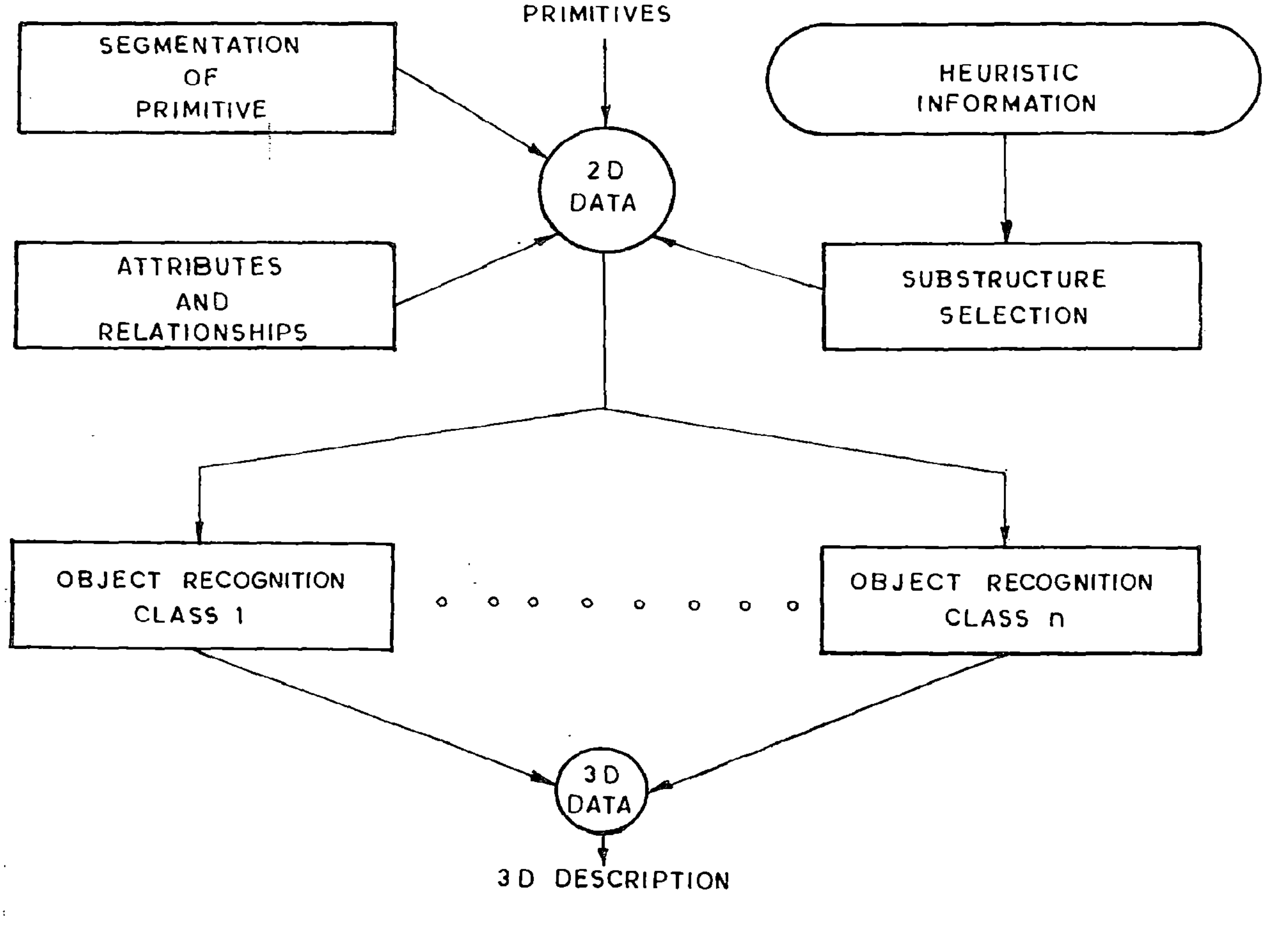


FIG.1.2. MAIN COMPONENTS OF THE SYSTEM

each having a specific function, operate on the data. Interpretation consists of invoking these processes during repetitive cycles until the 3 D structure has been completely established.

1.4 SCOPE

The drawing is the most important product of the design process in mechanical engineering. Drawing documentation constitutes a link between the design office and the manufacture.

In the manual design process a designer makes use of a certain number of drawing standards called drawing modules (primitives). Any large mechanical drawing from various field of mechanical engineering is an assembly of the modular structure of various simple modules and complex modules . Simple modules are defined as recurrent geometrical themes appearing in the lowest rank of hierarchy and linking among them points only . Complex modules on the other hand, link simple modules and/or points. This formulation treats a drawing as a whole, a supermodule.

The work presnted in this thesis, develop variety of modules required in drawing in the orthographic views of an object. The program, algorithm is such that the portion to be drawn is identified and subsequently image is drawn/created. Many modules of different levels representing common geometries are developed. In the main program they are called whenever required for preparation of drawing.

Hierarchy, adopted in this thesis related to the proposed drawing is drawn in Fig. 1.3. As followed by most common graphical systems starting from that of Sutherland's Sketchpad [8], in this thesis hierarchy approach is followed to draw the complex drawings. In the hierarchy lowest level graphic primitives are derived and called to prepare other graphic entities.

Complex geometries are defined into numerous well defined geometries, for drawing purpose. During the development of these module, portability factor is also considered so that such a package should at the same time enable the application of plotters of various firms together with their basic software. Develop drawing modules are represented in

Hierarchical Definition of Bicycle

```
Open (Spoke )
   Call Polyline (Spoke-Data) Draw Spoke
Close (Spoke)

Open (Wheel)
   For Each Spoke
     Compute  (Instance-Transform)
     Call Segment (Spoke Instance Transform)

Call Fill Area (Hub-area) Draw Hub
Call G.D.P. (Circle Centre Radius) Draw Rim

Close (Wheel)

Open (Bicycle)
   Call Segment(Wheel Front Wheel Transform)
   Call Segment(Wheel Rear Wheel Transform)
   Call Polyline (Frame Data)

Close (Bicycle)
Result 3 segments
       1 copy of spoke data
       1 copy of wheel data
       1 copy of frame data
       Dynamic Picture
       (If we re-define spoke or wheel
        all instances change automatically)
```

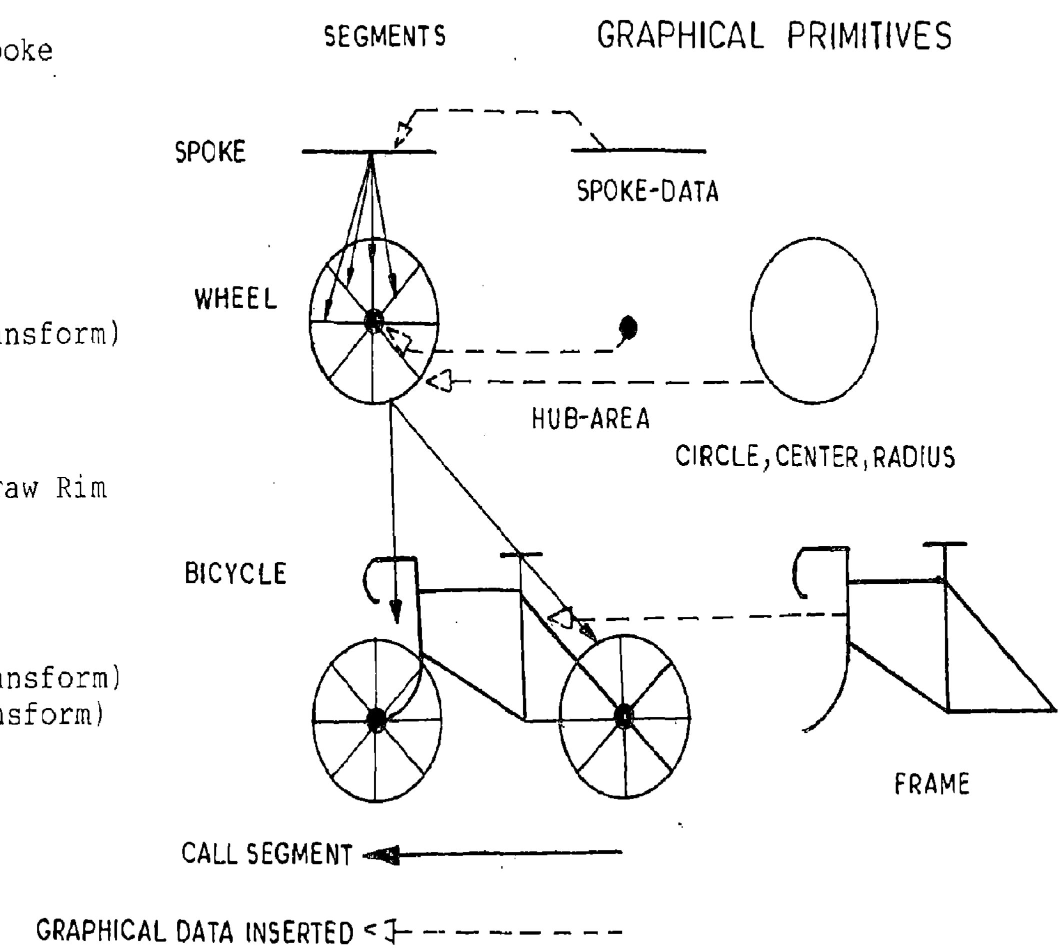


FIG. 1.3 : ILLUSTRATION OF HIERARCHY

computer memory with the use of parameters, and the values of immediate interest of parameters are stored in a set formed manually or by a computer. The above requirements have become the basis for working out a set of modules to achieve many common geometries in mechanical drawings.

Realizing the need of solid modelling in CAD/CAM activities, an initiation is taken to develop 3-D modules. In the present work simple brick modules and their assembly is demonstrated. To have infra structure for FEM grid generation and other integrated shapes of these modules.

The packages developed for plotting devices are compatible with the plotting devices Calcomp and Plot-10, DEC system-2050 in Roorkee University Regional Computer Centre.

CHAPTER - II

HIERARCHICAL DEVELOPMENT OF 2-D DRAWING PACKAGE

2.1 GRAPHIC PACKAGE

A graphic system which is a collection of hardwares and softwares is designed to make easier to use graphic inputs and outputs in computer programs. The design of graphic system is a very important aspect of computer graphics. Without such systems, graphic application programs would be extremely difficult to write. It is only by constructing graphics systems that we make it possible to exploit the potential uses of computer graphics.

The graphics package is a set of subroutines or functions used by an application program to generate pictures on a plotter or display and to handle graphical interaction.

2.1.a Characteristics of Graphic Software Design

To enhance the ability of graphic package, adequate considerations should be given on the following issues.

Simplicity

Feature that are too complex for the application program should not be used, to provide simplicity use of package.

Consistency

A consistent graphics system is one that behaves in a generally predictable manner. Function names, calling sequences, error handling, and co-ordinate systems all should follow single and consistent patterns without exceptions.

Completeness

The designer must try to design a reasonably small set of functions that can conveniently handle a wide range of applications.

Robustness

When the programmer does something seriously wrong, the system should report the error in the most helpful manner possible. Only in extreme circumstances should errors cause termination of execution, since this will generally cause the user to lose valuable results.

Performance

There is nothing much to be done here by the graphics system designer except to minimize the effect by omitting highly dynamic graphic functions that demand fast response or expensive hardware.

Economy

Graphics systems should be small and economical, so that adding graphics to an existing application program can always be considered.

2.2 GENERAL-PURPOSE GRAPHICS SOFTWARE FOR MECHANICAL DRAWINGS

The ability to operations within a standard high-level language makes programming much easier and permits the resulting programs to be run on a wide variety of different computers.

Ease of programming and portability of programs can be achieved through the use of a graphic package, a set of subroutines that provides high-level access to the graphics input-output hardware.

A good graphics package simplifies the programmer's task and makes it possile to write portable programs that can be run on different computers and with different displays. This greatly reduces the cost of writing software for graphics applications.

The need of portability [8] of computer graphics had its roots in the diversity of early graphics hardware. Some of this diversity was based on hardware, because different computer graphics systems often took quite different technological approaches to the problem of drawing pictures with computers.

In the late 1960s and early 1970s, manufacturers of computer graphics hardware usually supplied software to their customers. This software was intended to make it easier for customers to develop application software to use the vendor's hardware. These software packages from different vendors usually presented quite different interfaces to the application software level. This made utilization of new and/or different computer graphics hardware difficult and software intensive. Similar to many other user of computer graphics we intend to write our own computer graphics software interface packages in order to present a common interface to the hardware of different vendors for their applications software.

In the present study while developing the program for 2-D drafting software care has been taken.

To attain the mentioned software characteristics as much as possible, and

For achieving greater portability by using lowest rank of entities to their maximum uses in preparing the drawings.

The lowest rank of entities, discussed in the following section, are normally available to all plotting devices of computer graphic system.

2.3 DEVELOPMENT OF SOFTWARES IN ORDER OF DEGREE OF HIERARCHY

It is intended to develop the two-dimensional drafting modules capable to offer a full range of geometry creation and manipulation commands. In order to create such a facility basic interest lies to develop the commands so as points, lines, circles, arcs, ellipses, centre lines, equidistances, rectangle and polygons etc. can be generated with dimensional input from the keyboard i.e. the facility available in computer centre. Considering the portability as a basic requirement of the package for creation of geometry.

Out of two methods discussed in Chapter I, the geometry is planned to be created as group of elements which can be structured hierarchically, thus representing the true structural relationship of the mechanical parts in the drawing.

In the lowest (first) rank of hierarchy a full range of editing and manipulation commands, including translation, rotation and scaling are kept. These commands can be applied to elements, group or collections of elements contained within a user-specified window. In this rank modules for preparing required system environment i.e. routines which actually instruct the device to start and stop, line drawing and windowing function etc., are also included. Such ranked modules are usually available to any system.

Depending upon the computer drafting system, this rank software called basic software, may be supplied only for the host computer or for both the host computer and the controller. Fig. 2.1 shows the relationship between the types of software and gives the overall flow of data in an off-line graphic system. Host computer basic software consists of a set of subroutines which allow the programmer to perform elementary drawing operation such as drawing lines, scaling the plot, etc. The basic software generates the commands necessary to perform the specified operation and transmits them to the display unit or plotter if it is on-line, or writes the commands on an appropriate medium (e.g. magnetic tape) for subsequent ploting off-line. This chapter discusses the modules belonging to this level.

Combination of two or many lowest hierarchical elements are categorized in second rank of hierarchy. This rank in drawing aids includes:

A. The dimensioning and text string.

B. Simple geometrical construction i.e. development of rectangle, regular polygons, circles, ellipses and arcs etc.

C. Drawing of cross sectioning, threading, knurling etc. in which automatic cross sectioning, semi-automatic dimensioning for horizontals & vertical distances, lengths, diameters and

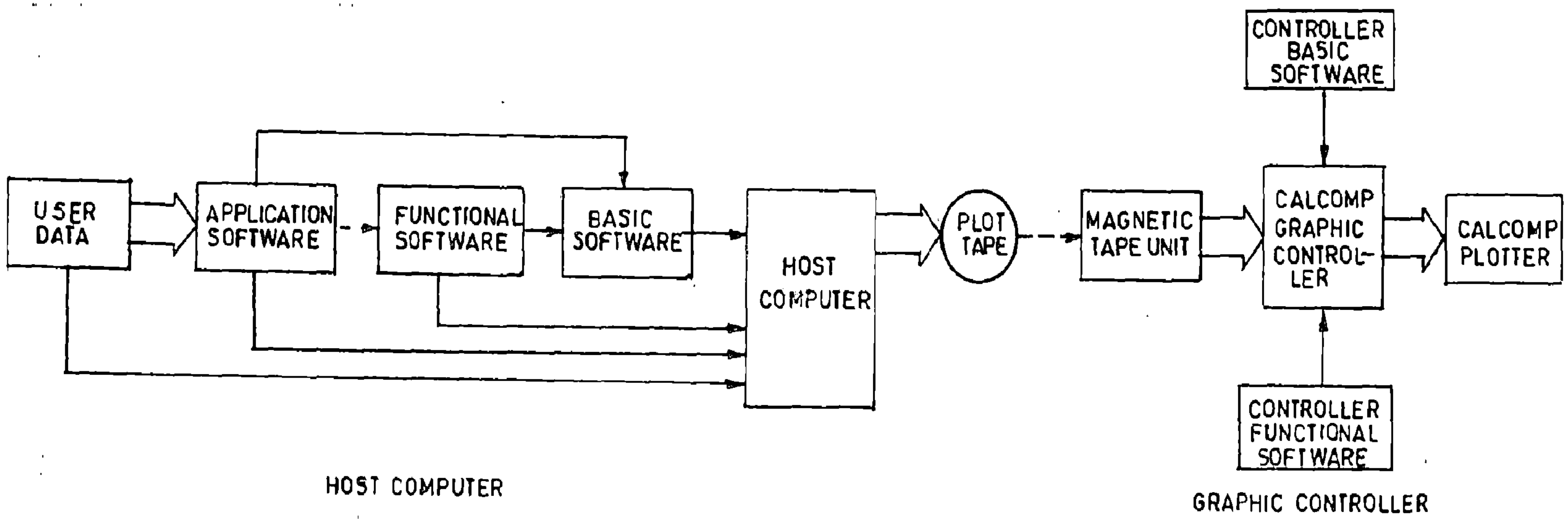


FIG.2.1.GRAPHIC SYSTEM OPERATION

radii, along with text writing are to be carried.

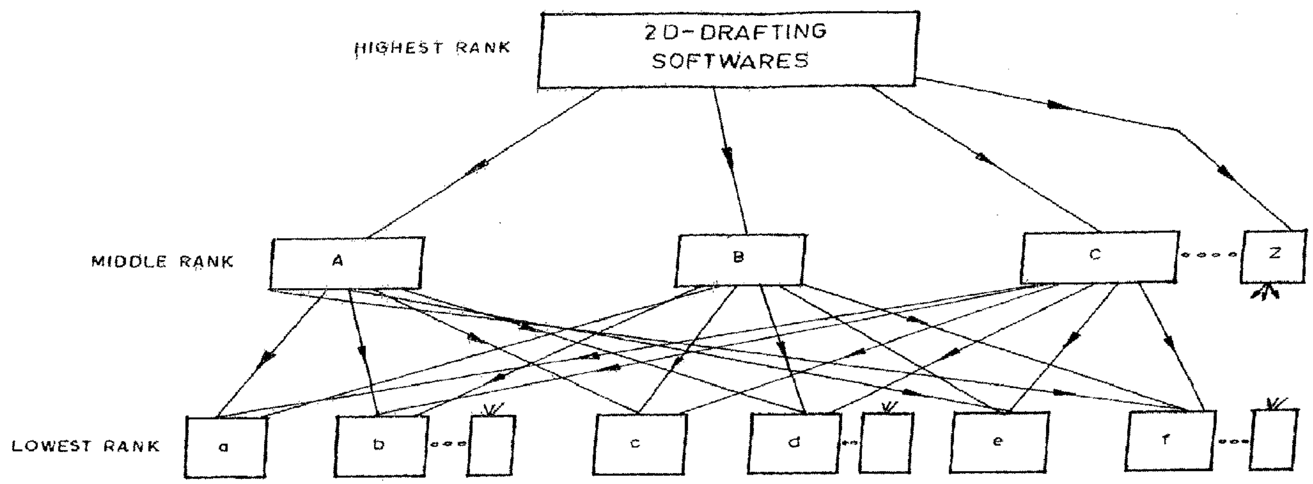
A simple but powerful menu structure, extensive prompting to help the user interact with the system and rapid graphical response to commands is deviced to provide all help for reducing interruption in the creative thought process. We shall discusses this level of modules in Chapter III.

Highest hierarchical rank of package is a complete problem solver called application software. In the present work description of softwares capable of producing detail drawings of mechanical elements are discussed in Chapter IV.

Fig. 2.2 present the schematics representation of hierarchical approach adopted for describing any 2-D drawing of mechanical object.

2.4 DRAFTING DEVICES AND LOWEST RANK ENTITIES

Modern computer graphic systems accomplish many equipments suiting to one or another applications Fig. 2.3. Roorkee University Computer Centre, though lack in recent growing facilities (equipments), for computer aided drafting but essential devices exist.



WHERE
BLACK a REPRESENT THE MODULES FOR SCALING
BLOCK b WINDOWING
BLOCK c OPEN / START
BLOCK d CLOSE / STOP
BLOCK e LINE DRAWING
BLOCK f ROTATION (TRANSFORMATION)

BLOCK A MAY REPRESENT THE MODULES FOR DIMENSIONING AND TEXT STRING ETC
BLOCK B GEOMETRICAL CONSTRUCTION ETC
BLOCK C MAY REPRESENT THE MODULES FOR SECTIONING, THREADING, KNURLING ETC

FIG.2.2. HIERARCHICAL DEVELOPMENT OF 2-D DRAWING

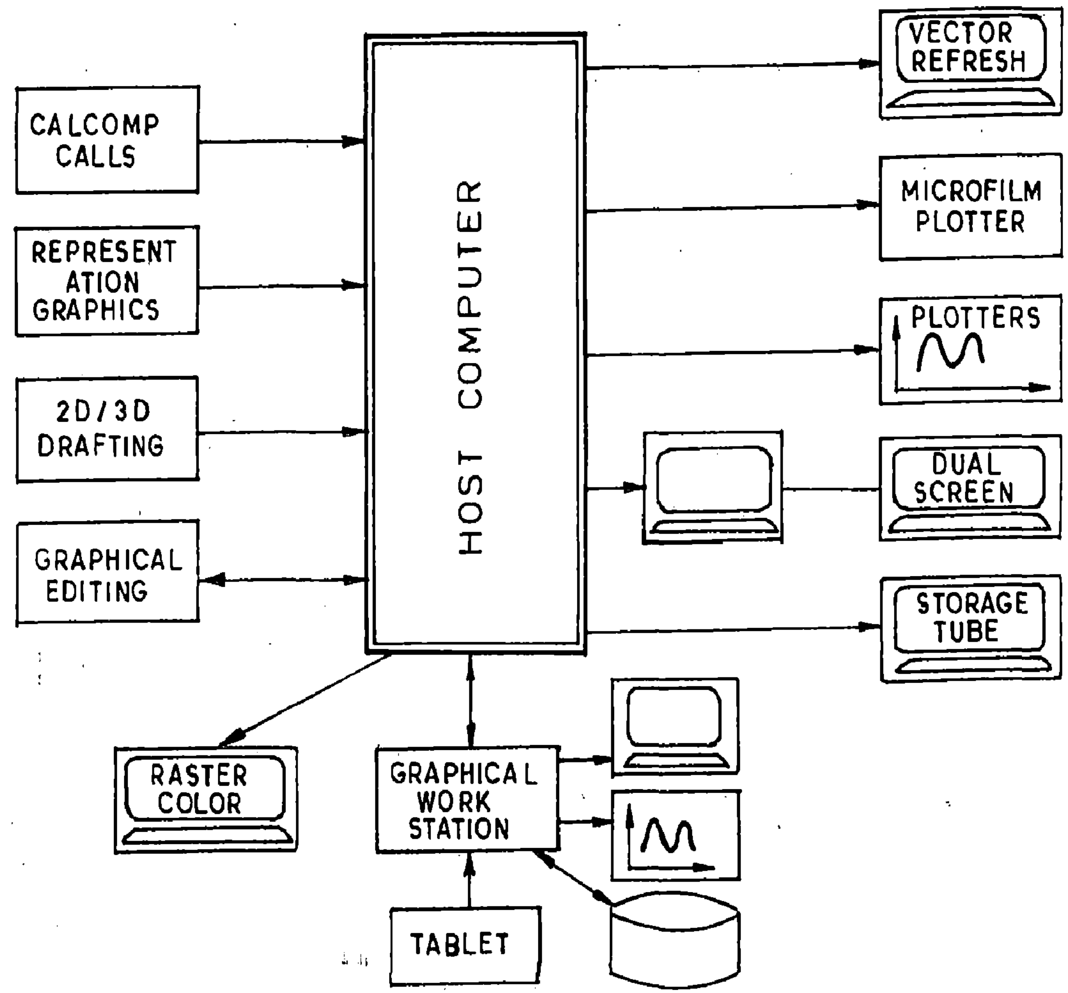


FIG.2.3. DATA FLOW IN COMPUTER DRAFTING SYSTEM

Two image creating devices which are available at the Center are :

A. The Calcomp Plotter,

B. Visual Display Unit, Tektronix Graphic Terminal (Plot - 10), Linked with DEC-2050 Host Computer System.

A. The Calcomp Plotter

This is an automatic electromechanical plotter of drum type. The other type available with many other systems is the flat-bed type, but dram plotters are more compact and less expensive than flat-bed plotters, while flat-beds offer greater accuracy and resolution plus ability to plot on a variety of materials. In drum plotters, the paper moves over a series of rollers in X-direction and the pen moves over a slider in Y-direction. The plotter is controlled by a set of instruction, specifying pen position (up and down on paper) and X and Y movements of the paper and pen. These instructions are obtained from subroutine stored in the library of the computer when executed with data from a plotting program.

A drawing is started by initializating the plotter (calling subroutine PLOTS with arguments 0, 0 and 5,5 being the code of device), setting an origin with respect to the at rest position of pen and then specifying the co-ordinates of the point to be drawn [13].

B. Visual Display Unit, The Tektronix Graphic Terminal (Plot 10)

Tektronix Plot 10 terminal is the unit attached directly with the computer through computer display terminals. The 4010C01 plot 10 Interactive Graphics Library (IGL) is a software package supporting the existing hardwares. IGL source code is written in MORTRAN, an "extended FORTRAN" structured programming language [12].

2.5 LOWEST LEVEL SOFTWARES

2.5.a The Calcomp Plotter

The library of subroutines in which all calling routines for the calcomp plotting device, are stored, is named as CALCOM. The following is the list of subroutines with their arguments, and their function [13].

CALL PLOTS (0,0,LDEV)

CALL PLOT (XPAGE, YPAGE, ± IPEN)

CALL FACTOR (FACT)

LDEV=5, is the logical number of the device

XPAGE, YPAGE are the x,y coordinates, in cms, from the current origin, location of a pen movements terminal point.

IPEN = 2, pen is down during the movement

= 3, pen is up during the movement

=-2, down } new coordinates x,y will be defined as

=-3, up } the new origin.

= 999, device to stop after reaching point (x,y).

FACT is the scale factor that determines the enlargement or reduction of the entire plot.

The other subroutines are -

CALL SYMBOL(XPAGE, YPAGE, HEIGHT, IBCD, ANGLE, ± NCHAR)

CALL SYMBOL (XPAGE, YPAGE, HEIGHT, INTEQ, ANGLE, -ICODE)

CALL NUMBER (XPAGE, YPAGE, HEIGHT, FPN, ANGLE, +NDEC)

XPAGE, YPAGE define the relative origin of the character string (usually the lower left corner of the first character position).

HEIGHT is the height (and width), in cms, of a character position.

IBCD is the array of the text that is to be written.

ANGLE	is the angle at which the character string is to be plotted.
NCHAR	is the number of characters in IBCD array.
INTEQ	is the integer equivalent of a special centered plotting symbol.
-ICODE =-1,	pen is up during the move, after which a single symbol is plotted.
=-2,	pen is down during the move, after which a single symbol is produced.
FPN	is the floating point number that is to be converted and plotted.
NDEC	specifies the number of decimal places to be plotted.
= 0,	only the number's integer portion with a decimal point are plotted, after rounding.
=-1,	only the number's integer position is plotted, after rounding.

Value of NDEC should not exceed 9.

2.5.b The Tektronix Graphic Terminal (Plot 10).

The Tektronix 4010C01 PLOT 10 Interactive Graphics Library (IGL) is a Host-independent library of routines providing a versatile and easily accessible means of graphic and text interaction. Programs calling IGL library routines may be written in any language that can call a FORTRAN library subroutine.

The IGL source code is written in IGL MORTRAN and is compiled into FORTRAN when IGL is installed. The entire library subroutines can be divided into five major categories.

(i) Routines Preparing Required Systems Environment

This contains subroutines like GRSTRAT, GRSTOP, BAUDRT etc. which actually instruct the device to start, stop and set baudrate etc.

(ii) Graphic Environemnt

This contains subroutines like DEGRER, RADIAN, INCHES, MILLIM, ROTATE, SCALE, WINDOW etc. which set angle either in degrees or in radians, dimensions either in inches or millimeters, scale of the figure, window to be viewed etc., as is obvious from the names of the subroutines.

(iii) Graphic Action

This contains all subroutines responsible for the actual graphic action. Subroutines like BELL, MOVE, DRAW, NEWPAG etc. fall under this category. Different routines serve the purpose as may be clear by their names itself as BELL rings bell whenever called. DRAW draws a straight line between any two points, MOVE moves the pen to the specified point with no line drawn in between, NEWPAG washes the screen and starts with the fresh page etc.

(iv) Text Environment

Routines under this again are responsible for preparing the environment for the text writing on the figure. Subroutine TXAI species that text is in AI format.

(v) Text Action

This contains the subroutines instructing the action to write the text. Subroutine TEXT display the integer, real numbers and string of alphanumeric text, at the screen.

All subroutines have their respective list of arguments which can be seen in Ref.No.[12].

2.5.1 Similarities with Routines Available on Both Devices

There are only a few subroutines which have the similarity on the basis of their function on the two devices. The library CALCOM contains very few subroutines which are available at this computer of Roorkee University, where as in IGL some subroutines are not supported by hardware. The pen up movement is defined in IGL by subroutine MOVE and pen down movement by subroutine DRAW while in CALCOM, one subroutine PLOT with a counter IPEN does both the things.

For starting and stopping of device in IGL we have subroutines GRSTRT and GRSTOP respectively while in CALCOM we have subroutine PLOTS and PLOT (with IPEN = 999).

..

CHAPTER - III

HIERARCHICAL MIDDLE RANK MODULES

3.1 BASIC CONCEPT AND MODULES FACILITIES

This chapter describe the hierarchical middle rank modules developed for mechanical drawings. The modules of this rank are divided into three major categories.

I- Modules for simple geometrical constructions and arc drawings etc.

II- Modules for dimensioning and text string etc.

III- Modules for sectioning, threading and knurling etc.

Middle rank modules or functional software is an intermediate level of software which relieves the user from programming of many commonly used graphic functions. Functional software is further subdivided into host-computer resident and controller - resident type software. Host-computer resident functional software is usually written in FORTRAN and grouped into packages by general application.

Each package consists of a set of subroutines to perform particular operations.

A controller - resident functional software product is a program written in assembly language for a Calcomp programmable controller, which permits the user to obtain graphical output more easily and efficiently.

Middle rank of entities are planned to create geometries as single element or as group which are structured hierarchically. These entities form the pillar to represent the true structural relationship of the mechanical parts in the drawing. These can be redefined if they are not appropriate.

A particularly productive feature of the system is that graphical display commands can be executed at any stage in the drawing process.

Individual users can often multiply their productivity by tailoring the system to their special application needs. This requirement is catered for by the provision of a FORTRAN interface. The FORTRAN interface gives the user access to almost all the drawing facilities and allows him to capture the drafting logic of his special application in a very efficient manner.

The system is menu-driven through a simple user interface, and although the commands are structured, the user does not have to climb a command tree to exit from a function as most of the commands (Fig. 3.1) can be selected at any stage in the drawing process. The commands contained in the menu are easy to understand for the movic user.

It can be seen from the above description that the two-dimensional middle rank modules for mechanical drawings contains many features which can improve productivity.

Thus the identification of middle rank modules, which link to themselves give as large number of drawings as possible, is of great importance.

3.2 GENERAL MODULES DESCRIPTION

The graphic modules in the language FORTRAN IV, consists of main operations of calculating required coordinates, and also formation of connectivites of them. In the curve drawing, modules has the facility of calculating the total number of points that must be calculated to get a smooth boundary curve if all these calculated points are joined by a straight line.

I	Modules for geometrical constructions and arcs drawing	-->	RECTAN	POLYG	CONIC	ARC	NEWTR	-	-	-	-	-
II	Modules for dimensioning and text string	-->	SYMBOL	ARHI	ARHE	ARVI	ARVE	INAF	INAH	-	-	-
III	Modules for sectioning, threading and knurling	-->	CONIC1	CONIC2	CONIC3	CONIC4	CONIC5	CONIC6	HARETA	HAPOYG	THRD	KNURL

FIG. 3.1 ; MENU OF MIDDLE RANK MODULES

Modules for sectioning reads the parameters defining the boundary of geometry and section line. Then the equation of boundary will be identified and intersecting points will be obtained by solving the equation of section line and boundary curves. Parameters in section line will be adjusted to get required intersecting points all over the region of interest. Subsequently module identify the portion which suppose to be hatched and subsequent connectivity will be done.

Geometrical constructions of simple geometries consists of main operations of calculating the end coordinates of sides and connectivities between them.

Separate subroutines supplementing the two devices, Tektronix Graphic Terminal (Plot 10) and Calcomp Plotter are written.

A brief description of each subroutine is given below.

3.3 MODULES FOR SIMPLE GEOMETRICAL CONSTRUCTIONS AND ARCS DRAWINGS

This section deals with simple problems of geometrical constructions based on straight lines and arcs which are essential in the preparation of mechanical drawings i.e. rectangle, triangles, square, regular polygons (pentagon, hexagon etc.), circles, ellipses and arc drawings.

Three modules are developed to achieve common geometries (Fig. 3.2) in mechanical drawings and two for arcs drawings.

3.3.1 Subroutine RECTAN

Basic Feature: To draw any number of rectangles

Data : Number of rectangles to be drawn, NREC

: Coordinate of origin (0_i), XO(I), YO(I)

: Length of sides, LX(I),LY(I)

: Angle which rectangle makes with horizontal axis, THETA(I)

Lower entities used : (MOVE, DRAW)/PLOT

New entity name : RECTAN(XO(I), YO(I), LX(I), LY(I) ,THETA(I))

Tailored connection : Mathematical connection between variables.

The flow chart of subroutine is given in Fig. 3.3.

TWO-DIMENSIONAL DRAFTING MODULES

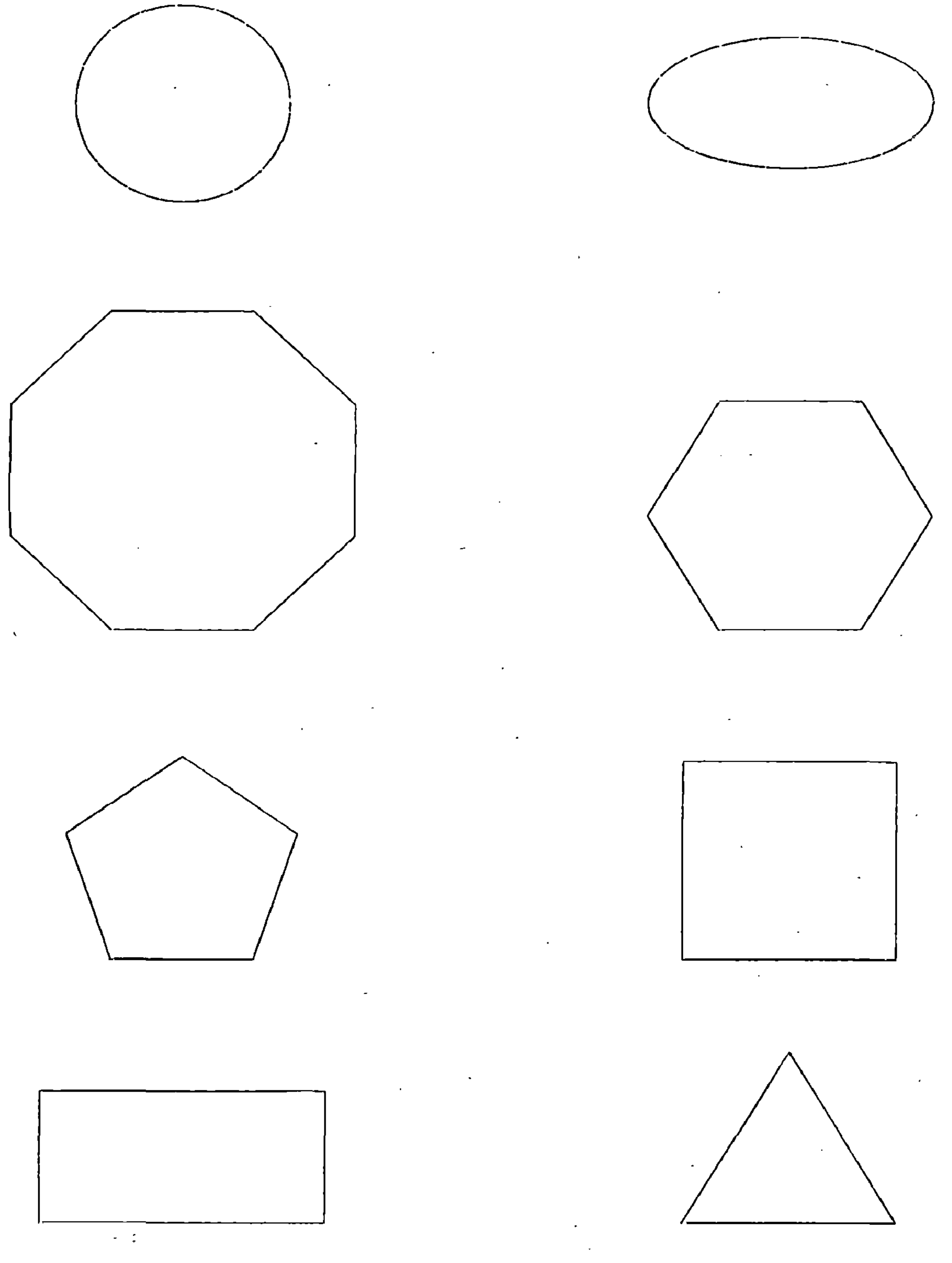
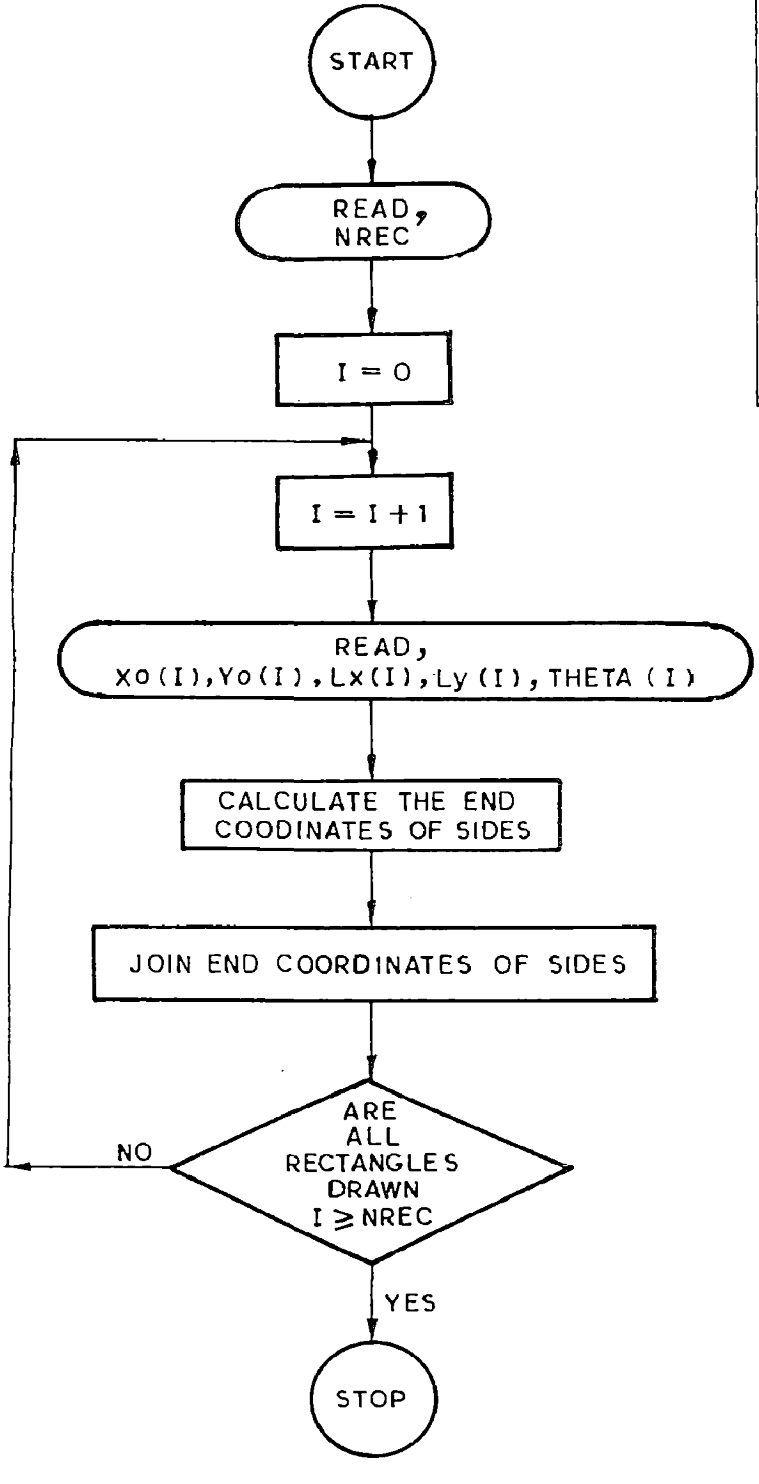


FIGURE No.3.2. GEOMETRICAL CONSTRUCTIONS MODULES



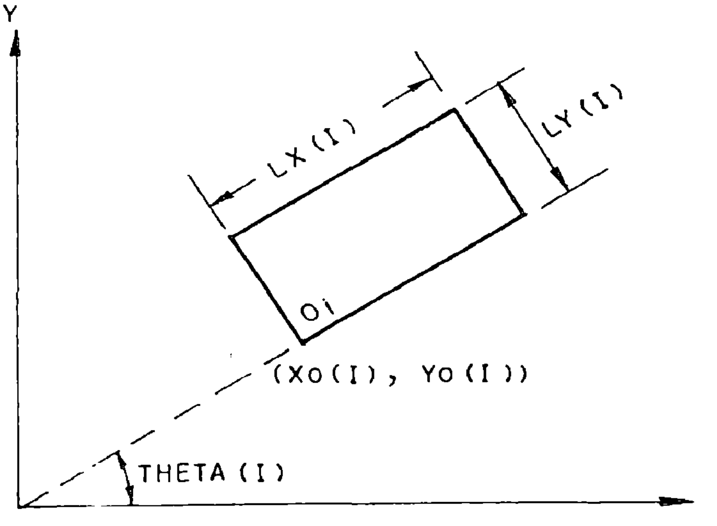


FIG.3.3. FLOW CHART FOR SUBROUTINE RECTAN

3.3.2 Subroutine POLYG

Basic Feature: Since in the mechanical drawings, mostly we required to draw regular polygons, therefore this subroutine has the facility to draw any number of regular geometries of equal sides i.e. equilateral triangle, square, pentagon, hexagon etc.

Data : Coordinate of origin (0_i), X(I), Y(I)
: Length of side, A(I)
: Number of sides, NL(I)
: Number of regular polygons to be drawn ,NPOLY

Lower entities used : (MOVE,DRAW)/PLOT

New entity Name ; POLYG(X(I),Y(I),A(I),NL(I))

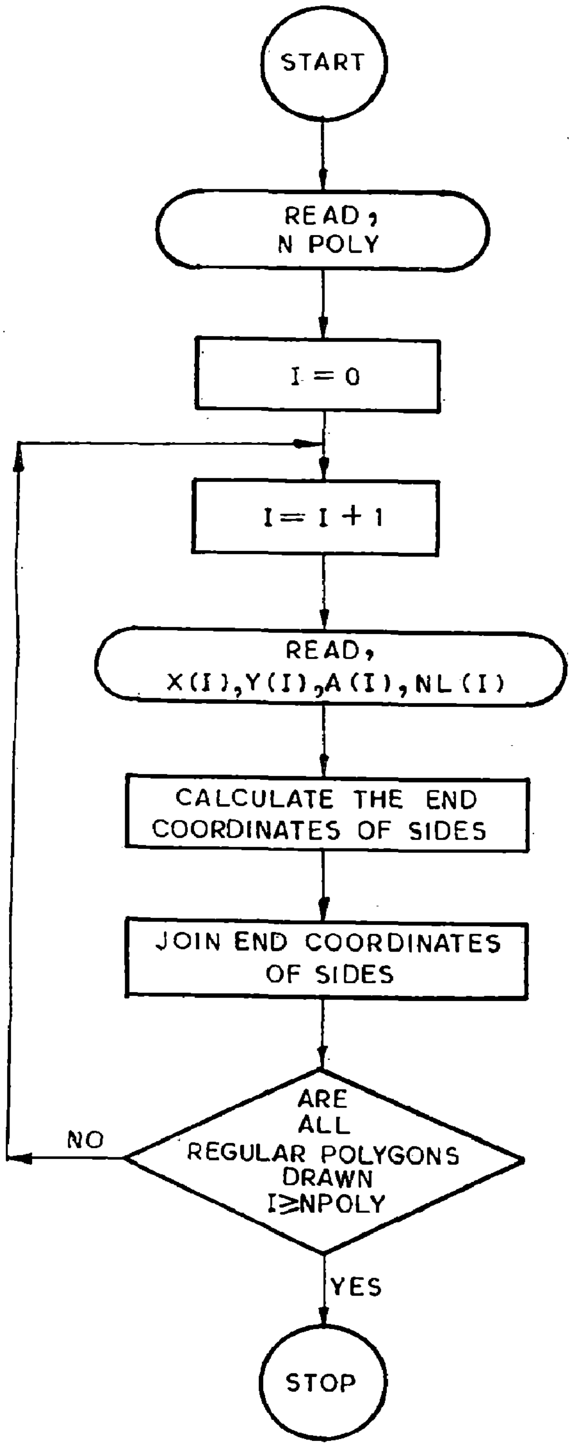
Tailored connection : Mathematical connection between variables.

The flow chart of subroutine is given in Fig. 3.4.

3.3.3 Subroutine CONIC

Basic Feature: To draw any number of conic section geometries i.e. circular or elliptical geometries.

Data : Coordinate of center, XC(I), YC(I)
: Vector of conic section, C(I)



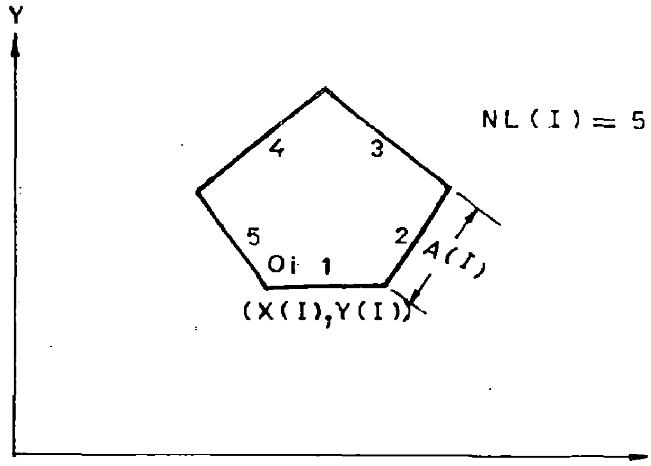


FIG.3.4. FLOW CHART FOR SUBROUTINE POLYG

: Constants of conic section equation, A(I),B(I) (In case of circle A(I)=B(I) and in case of ellips A(I)≠B(I))

: Number of conics to be drawn, NARC

Lower entities used : (MOVE, DRAW)/PLOT

New entity name : CONIC (A(I),B(I),C(I),XC(I),YC(I))

Tailored connection : Mathematical connection between coordinates find out from equation

$$A(X-XC)^2 + B(Y-YC)^2 = C^2$$

The flow chart of subroutine is given in Fig. 3.5.

3.3.4 Subroutine ARC

Basic Feature: To draw any number of circular or elliptical arcs.

Data : Coordinate of center, XC(I), YC(I)

: Vector of arc, C(I)

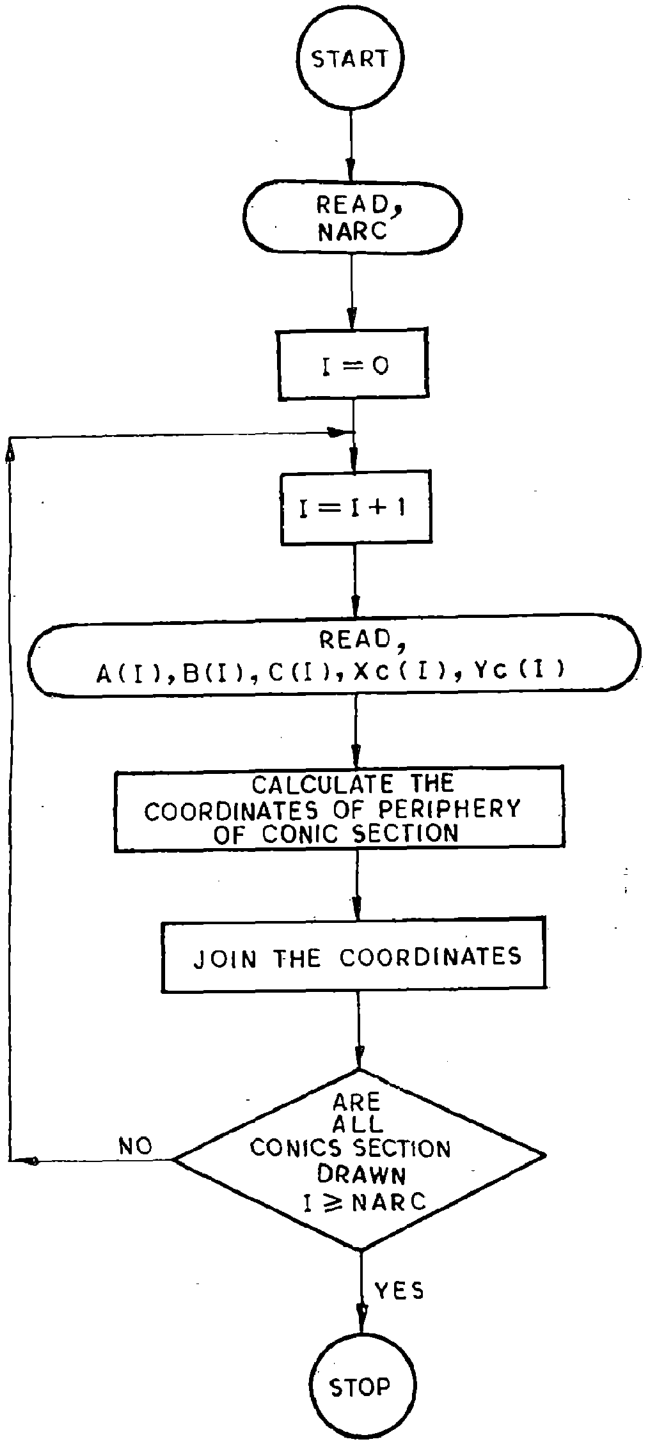
: Constants of conic section equation, A(I),B(I)

: Number of arcs to be drawn, NARC

: Initial and final coordinates of arc, XOD(I), XOD2(I)

Lower entities used : (MOVE, DRAW)/PLOT

New entity name : ARC(A(I), B(I), C(I), XC(I), YC(I), XOD(I), XOD2(I))



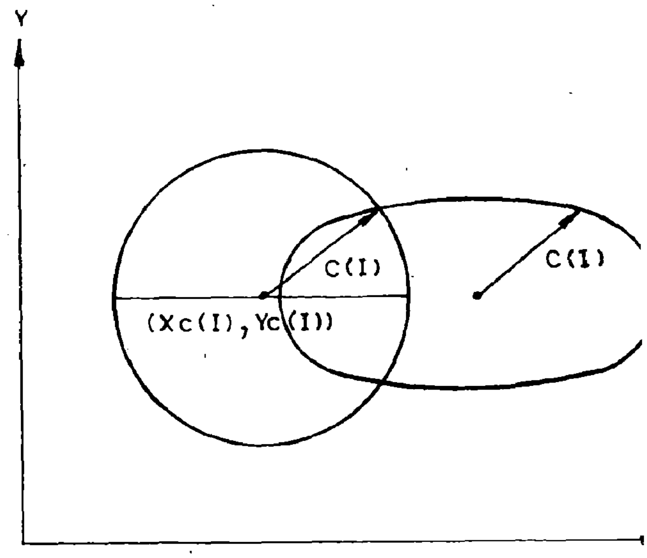


FIG.3.5. FLOW CHART FOR SUBROUTINE CONIC

Tailored connection : Mathematical connection between coordinates find out from equation

$$A(X-XC)^2 + B(Y-YC)^2 = C^2$$

The flow chart of subroutine is given in Fig. 3.6.

3.3.5 Subroutine NEWTR

Basic Feature : The subroutine is used to find out the intersecting or tangentical coordinates of any two conic curves. The utility of subroutine is that when we have the curves with different curvatures, it can find out the intersecting or tangential coordinates, then with the help of subroutine ARC we can draw the complex curves. In fact it solve a set of non linear, multivariable simultaneous equations with the help of Newton-Raphson Method.

Here an iterative approach is used to solve the polynomial equations. Initially approximate values of the roots are assumed and after each iterative cycle, more correct values are calculated till the desired accuracy is achieved. Let the two simultaneous equations be $\phi(x,y) = 0$ and $\psi(x,y) = 0$.

Let approximate values of roots be x_1 and y_1 and let the correct values of roots be $x_1 + s$ and $y_1 + t$, then we have

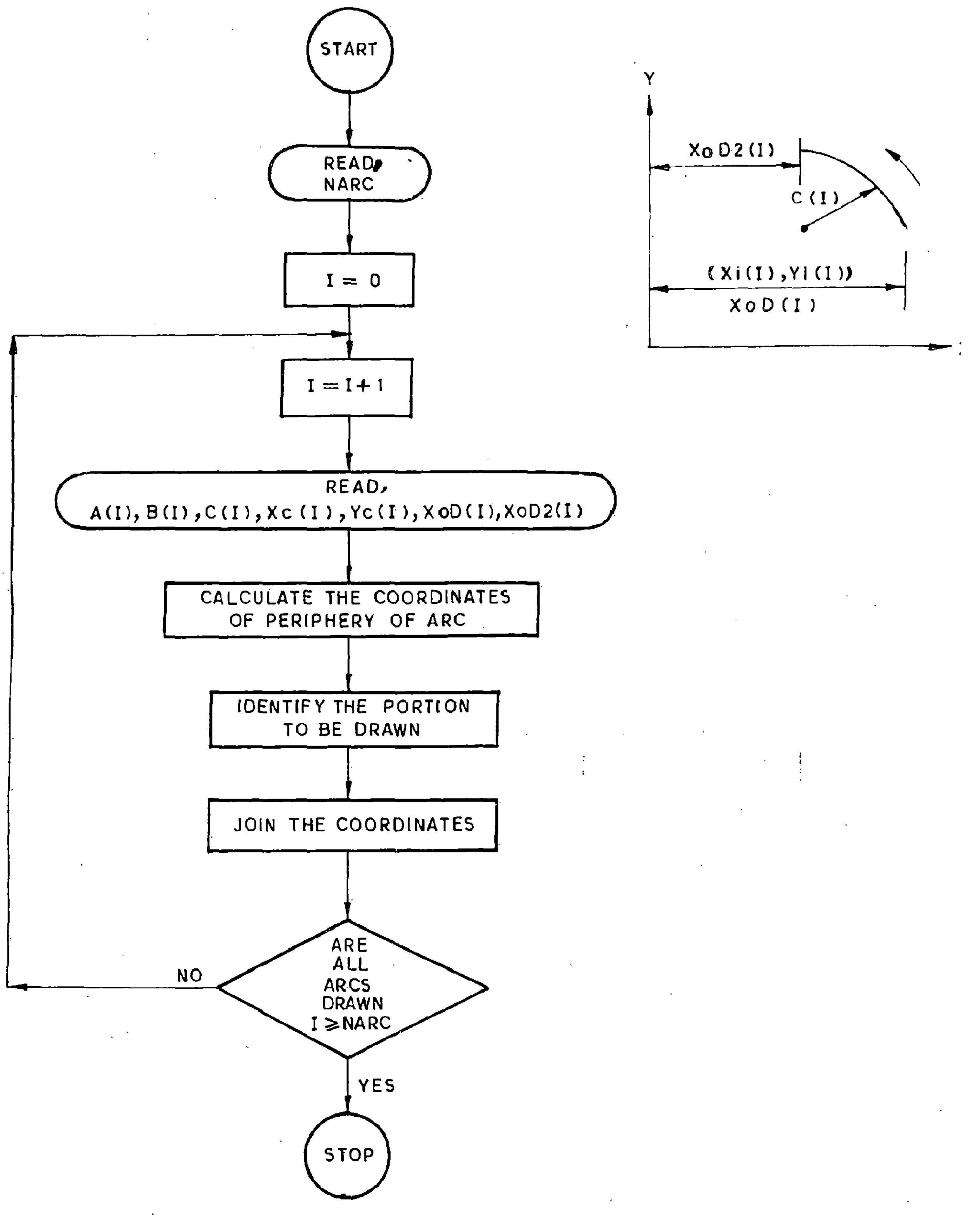


FIG.3.6. FLOW CHART FOR SUBROUTINE ARC

$\emptyset(x_1+s,\ y_1+t) = 0$ and $\psi(x_1+s,\ y_1+t) = 0$

These may be expanded in Taylor's series form (higher order terms contains powers of s and t , which are small quantities and hence may be neglected). Then we have

$$\emptyset(x_1+s,\ y_1+t) = \emptyset(x_1,y_1)+s\left[\frac{\partial\emptyset}{\partial x}\right]_{(x_1,y_1)} + t\left[\frac{\partial\emptyset}{\partial y}\right]_{(x_1,y_1)}$$

and

$$\psi(x_1+s,\ y_1+t) = \psi(x_1,y_1)+s\left[\frac{\partial\psi}{\partial x}\right]_{(x_1,y_1)} + t\left[\frac{\partial\psi}{\partial y}\right]_{(x_1,y_1)}$$

Let $\left[\frac{\partial\emptyset}{\partial x}\right]_{(x_1,y_1)} = p_1$, $\left[\frac{\partial\psi}{\partial x}\right]_{(x_1,y_1)} = p_2$ and

$$\left[\frac{\partial\emptyset}{\partial y}\right]_{(x_1,y_1)} = q_1, \qquad \left[\frac{\partial\psi}{\partial y}\right]_{(x_1,y_1)} = q_2$$

$\emptyset(x_1,y_1) = r_1$ and $\psi(x_1,y_1) = r_2$ then above equations may be rewritten as $p_1s + q_1t + r_1 = 0$ and $p_2s + q_2t + r_2 = 0$, where s and t are the unknown. Then by method of determinates :

$$s = \frac{\begin{vmatrix} q_1 & r_1 \\ q_2 & r_2 \end{vmatrix}}{\begin{vmatrix} p_1 & q_1 \\ p_2 & q_2 \end{vmatrix}} = \frac{(q_1r_2 - q_2r_1)}{(p_1q_2 - p_2q_1)}$$

$$t = \left| \frac{\begin{matrix} p_1 & r_1 \\ p_2 & r_2 \end{matrix}}{\begin{matrix} p_1 & q_1 \\ p_2 & q_2 \end{matrix}} \right| = - \frac{(p_1 r_2 - p_2 r_1)}{(p_1 q_2 - p_2 q_1)}$$

Thus we obtain more accurate values of roots :

$x_2 = x_1 + s$ and $y_2 = y_1 + t$. Now the procedure is repeated by assuming x_2 and y_2 to be the approximate roots and s and t are recalculated. This procedure may be repeated till the desired accuracy achieved.

For our requirement the two simultaneous equations of curves are :

$$R_1 = A1(X)^2 + B1(Y)^2 - 2A1(XC1)(x) - 2B1(YC1)(Y) + A1(XC1)^2 + B1(YC1)^2 - C1^2 \quad .. \quad (I)$$

$$R_2 = A2(X)^2 + B2(Y)^2 - 2A2(XC2)^2(x) - 2B2(YC2)(Y) + A2(XC2)^2 + B2(YC2)^2 - C2^2 \quad .. \quad (II)$$

and partial derivatives are

$P_1 = 2\ A1(X) - 2A1(XC1)$

$Q_1 = 2\ B1(Y) - 2B1(YC1)$

$P_2 = 2\ A2(X) - 2\ A2(XC2)$

$Q_2 = 2\ B2(Y) - 2\ B2(YC2)$

The subroutine is meant for N sets of roots of two equtions. Input consists of values of N and N sets of values of X and Y the approximate roots and also parameter defining the types of curves,output comprises

N sets of values of X and Y the correct roots.

The flow chart of subroutine is given in Fig.3.7.

3.4 MODULES FOR TEXT STRING AND DIMENSIONING

This section covers the dimensioning and text strings. One module related to text string has been developed, and four modules for dimensioning of horizontal and vertical distance and length, also two modules for dimensioning of diameters and radii. The modules are capable to draw projection lines, arrow head and dimension line (Fig. 3.8).

3.4.1 Subroutine SYMBOL

Basic Feature : This provides the user with the facility of writing any text in the figure, anywhere on the figure at any slant. Any number of strings can be plotted of different sizes of letters and also at different angles.

Data : Total number of strings to be plotted, NT
: Their text, TITLE
: Number of characters in each text, NC
: Coordinates of extreme left low corner of the string, XT, YT
: The angle of string, ANG
: The size of characters, SIZE

New entity name : SYMBOL (XT, YT,SIZE,TITLE,ANG, NC)

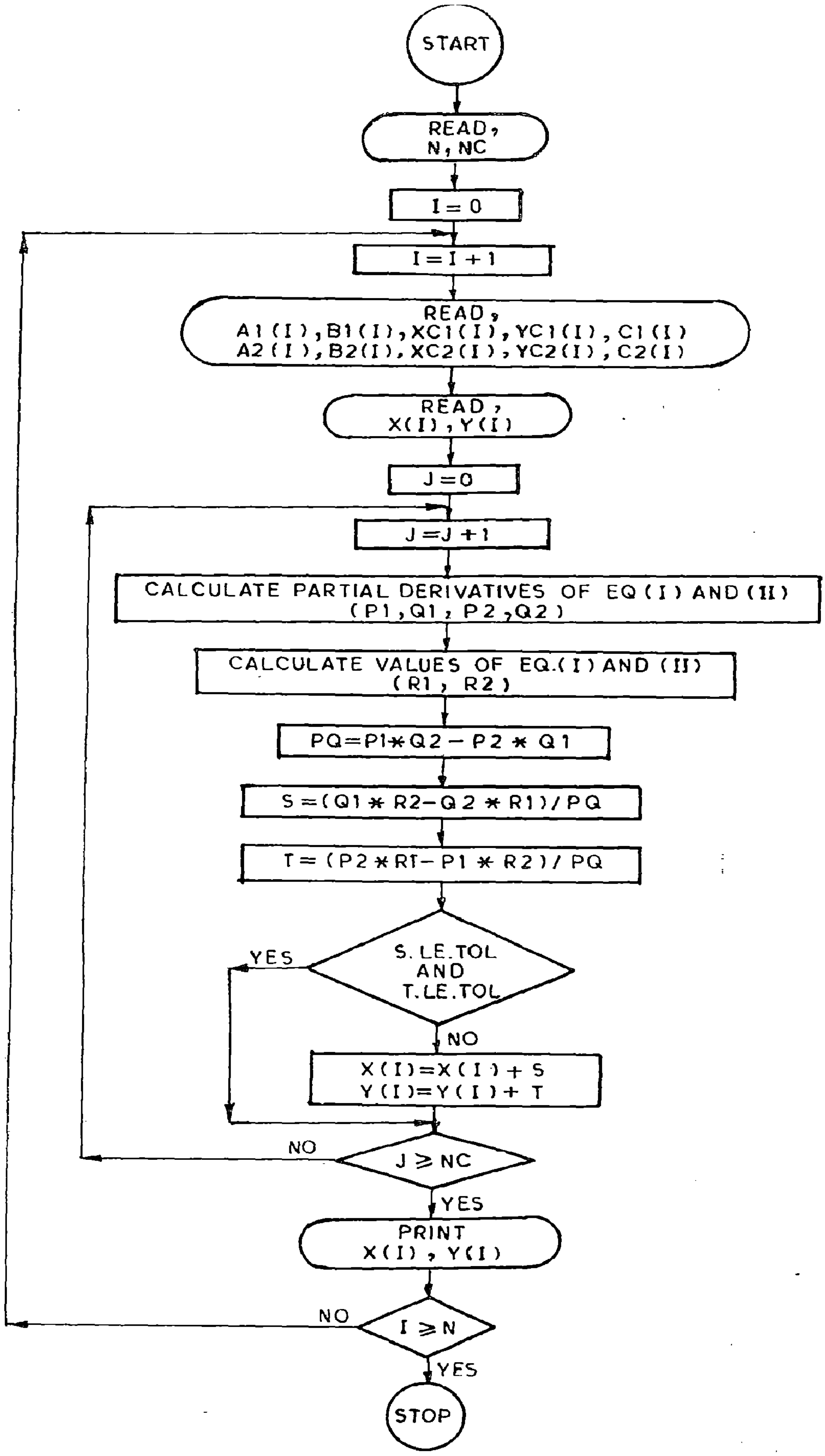


FIG.3.7. FLOW CHART FOR SUBROUTINE NEWTR

TWO-DIMENSIONAL DRAFTING MODULES

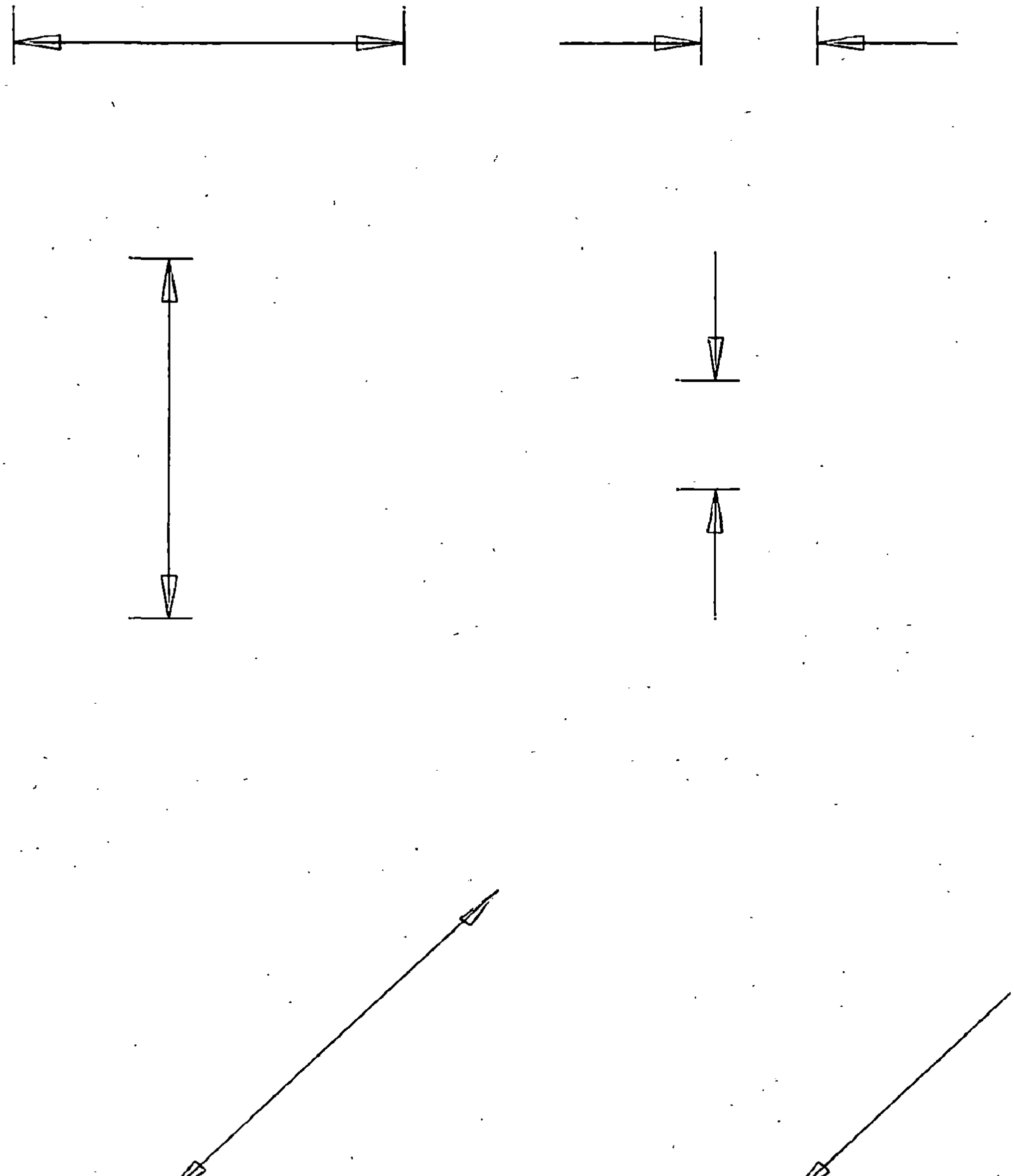


FIGURE NO 3.8 DIMENSIONIG MODULES

3.4.2 Subroutines for Dimensioning of Horizontal and Vertical Distance and Length

Basic Feature : The modules has the facility to calculate, all the coordinates of projection line, arrow head and dimensioning line relative to origin O_i (Fig. 3.9).

Subroutine ARHI : To draw dimensioning of horizontal distance and length (Fig. 3.9.A).

Subroutine ARHE : To draw dimensioning of horizontal distance and length (Fig. 3.9.B).

Subroutine ARVI ; To draw dimensioning of vertical distance and length (Fig. 3.9 .C).

Subroutine ARVE : To draw dimensioning of vertical distance and length (Fig. 3.9.D).

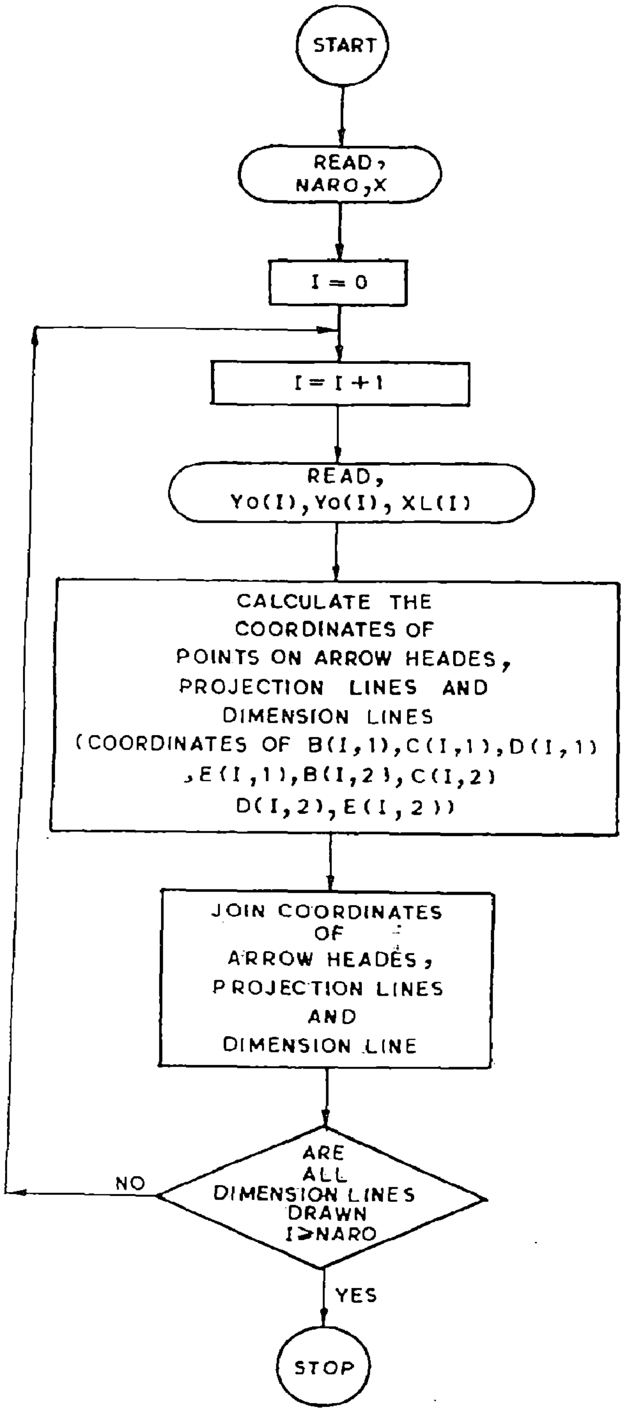
Data : Coordinate of origin (O_i), XO(I), YO(I)
: Depth of arrow head, X
: Number of dimensioning lines, NARO
: Length of dimension line, XL(I)

Lower entities used : (MOVE, DRAW)/PLOT

New entity name : ARHI(XO(I), YO(I), XL(I),X)
: ARHE (XO(I), YO(I), XL(I), X)
: ARVI(XO(I),YO(I),XL(I),X)
: ARVE(XO(I),YO(I),XL(I),X)

Tailored connection : Mathematical connection between variables.

The flow chart of subroutines is given in Fig. 3.10.



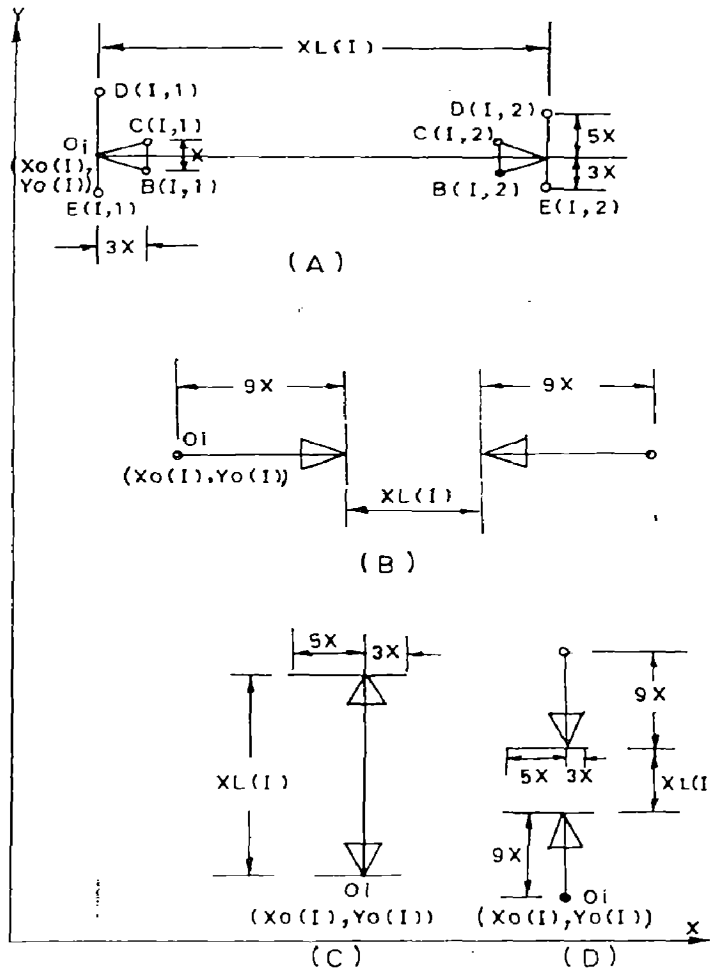


FIG 3.9

FIG.3.10. FLOW CHART FOR SUBROUTINES ARHI, ARHE, ARVI, ARVE

3.4.3 Subroutines for Dimensioning of Diameters and Radii

Basic Feature : The modules has the facility to calculate all the coordinates of arrow head and dimensioning line relative to origin 0_i (Fig. 3.11).

Subroutine INAF : To draw dimensioning of diameters (Fig. 3.11.A)

Subroutine INAH : to draw dimensioning of radii (Fig. 3.11.B)

Data : Coordinate of origin (0_i), XO(I), YO(I)

: Depth of arrow head, X

: Number of dimensioning lines, NARO

: Length of dimension line, XL(I)

: The angle of dimension line, ALFA(I)

Lower entities used : (MOVE, DRAW)/PLOT

New entity name : INAF (XO(I), YO(I), XL(I), X, ALFA(I))

: INAH (XO(I), YO(I), XL(I), X, ALFA(I))

Tailored connection : Mathematical connection between variables.

The flow chart of subroutines is given in Fig. 3.12.

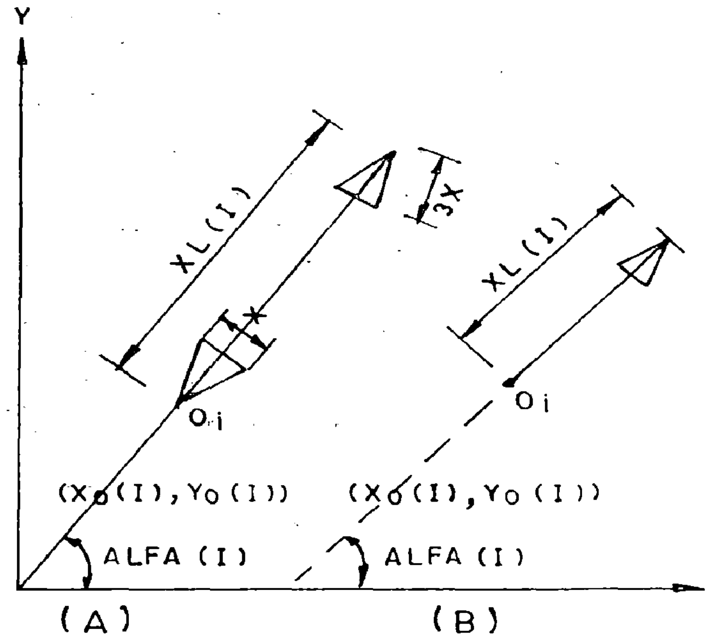


FIG. 3.11

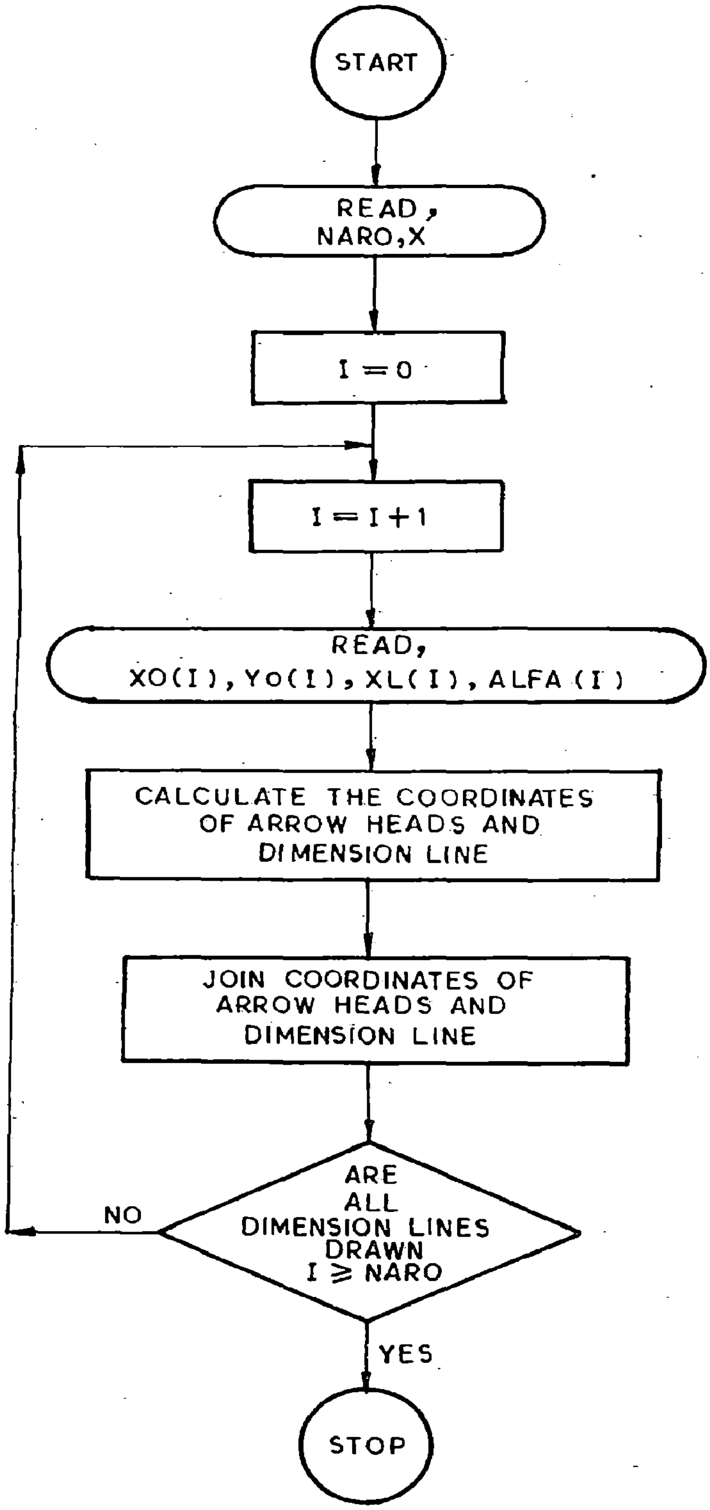


FIG.3.12. FLOW CHART FOR SUBROUTINE INAF, INAH

3.5 MODULES FOR SECTIONING, THREADING AND KNURLING

This section deals with problems of sectioning, threading and knurling based on straight lines and arcs which are essential in the preparation of mechanical drawings i.e. sectioning of rectangle, triangles, circle, ellipses, hollow conic section etc.

Eight modules are developed to achieve sectioning of common geometries in mechanical drawings and one module for threading and one for knurling.

3.5.1 Subroutine HARETA

Basic Feature : To hatch any number of rectangles.

Data : Number of rectangles to be hatches, NRECT

: Slope of section line, M

: Parameters defining the rectangle boundary, P(K), Q(K), R(K), S(K).

: Parameter A (defining the distance between section lines)

: Parameter B (defining the region of hatching).

Lower entities used : (MOVE, DRAW)/PLOT

New entity name : HARETA(P(K), Q(K), R(K),S(K),M)

Tailored connection : Mathematical connection between intersecting points obtained by solving intersection equation of Y = Mx+D and boundary of rectangle.

The flow chart of subroutine is given in Fig. 3.13 and output in Fig. 3.14.

3.5.2 Subroutine HAPOYG

Basic Feature: To hatch a set of triangles.

Data : Slope of section line, M

: Number of sides, NL

: Initial and final coordinates of sides, XA(I), XF(I), YA(I), YF(I)

: Parameter A (defining the distance between section lines)

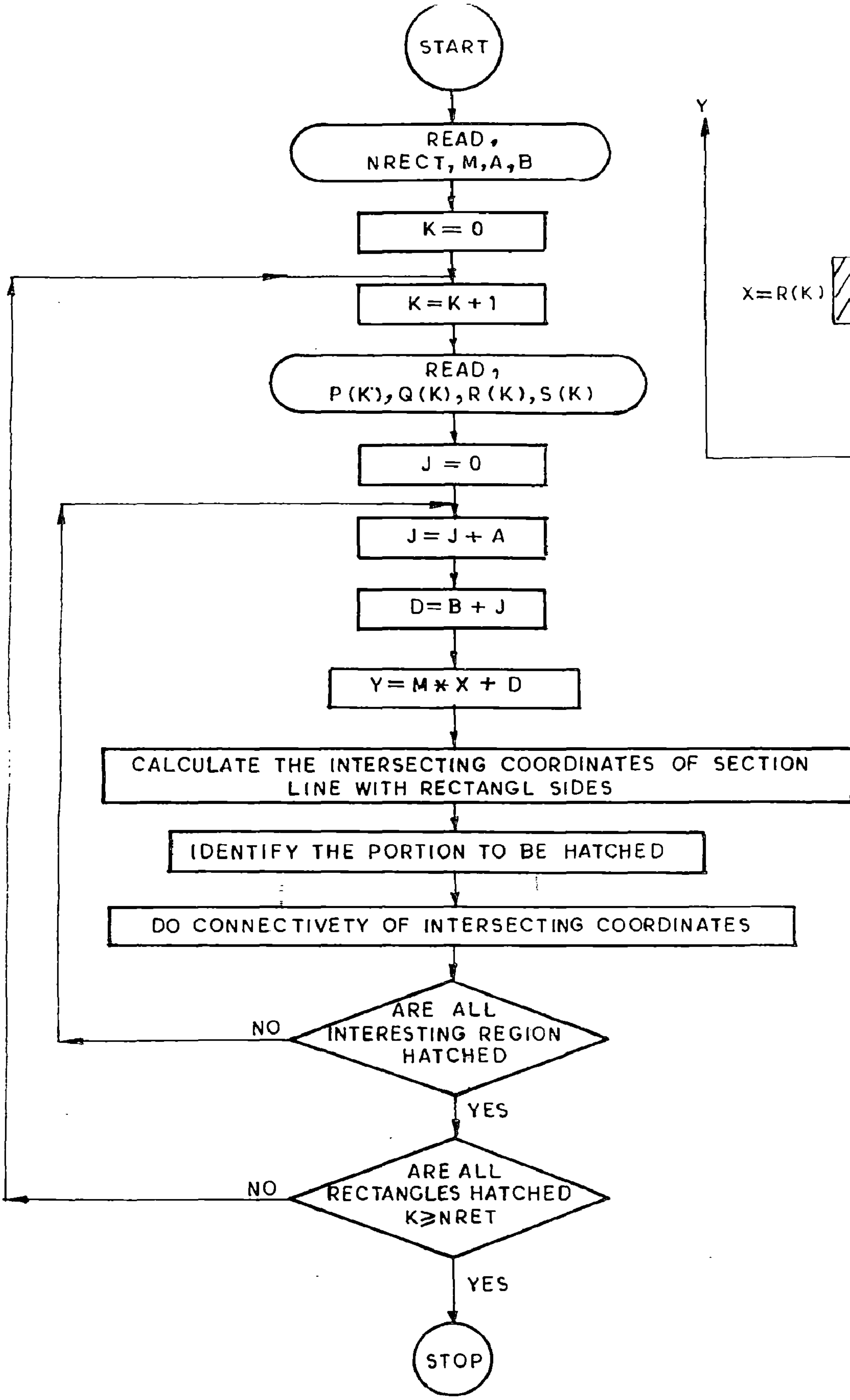
: Parameter B (defining the region of hatching).

Lower entities used : (Move, DRAW)/PLOT

New entity name : HAPOYG (XA(I),XF(I), YA(I) ,YF(I), NL,M)

Tailored connection : Mathematical connection between intersecting points obtained by solving intersection equation of Y=MX+D and boundary of triangles.

The flow chart of subroutine is given in Fig. 3.15 and output in Fig. 3.14.



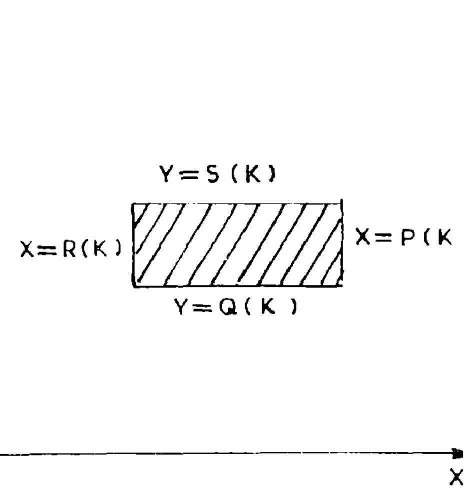


FIG.3.13. FLOW CHART FOR SUBROUTINE HARETA

TWO-DIMENSIONAL DRAFTING MODULES

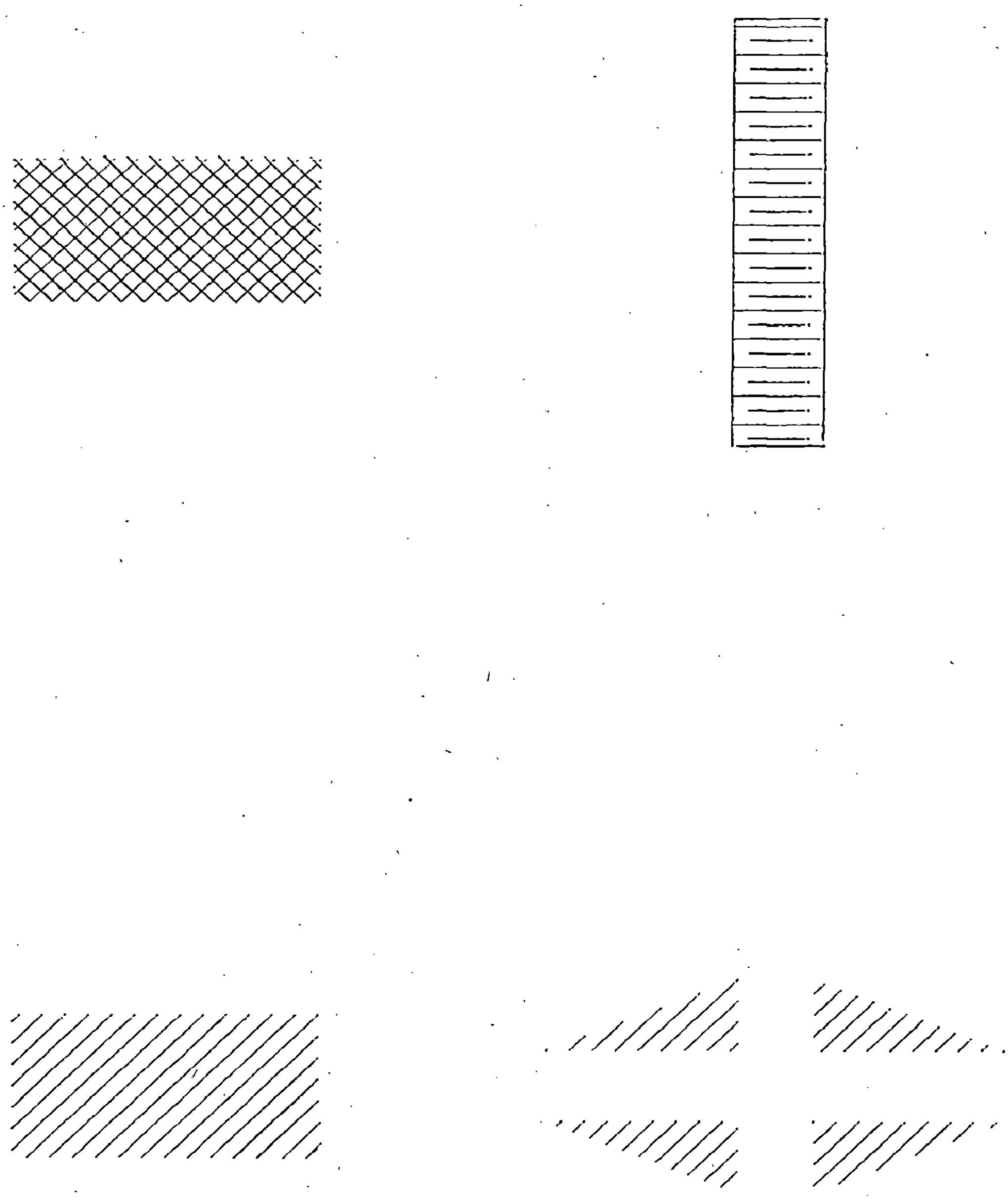


FIGURE No.3.14 SECTIONING, THREADING AND KNURIRING MODULES

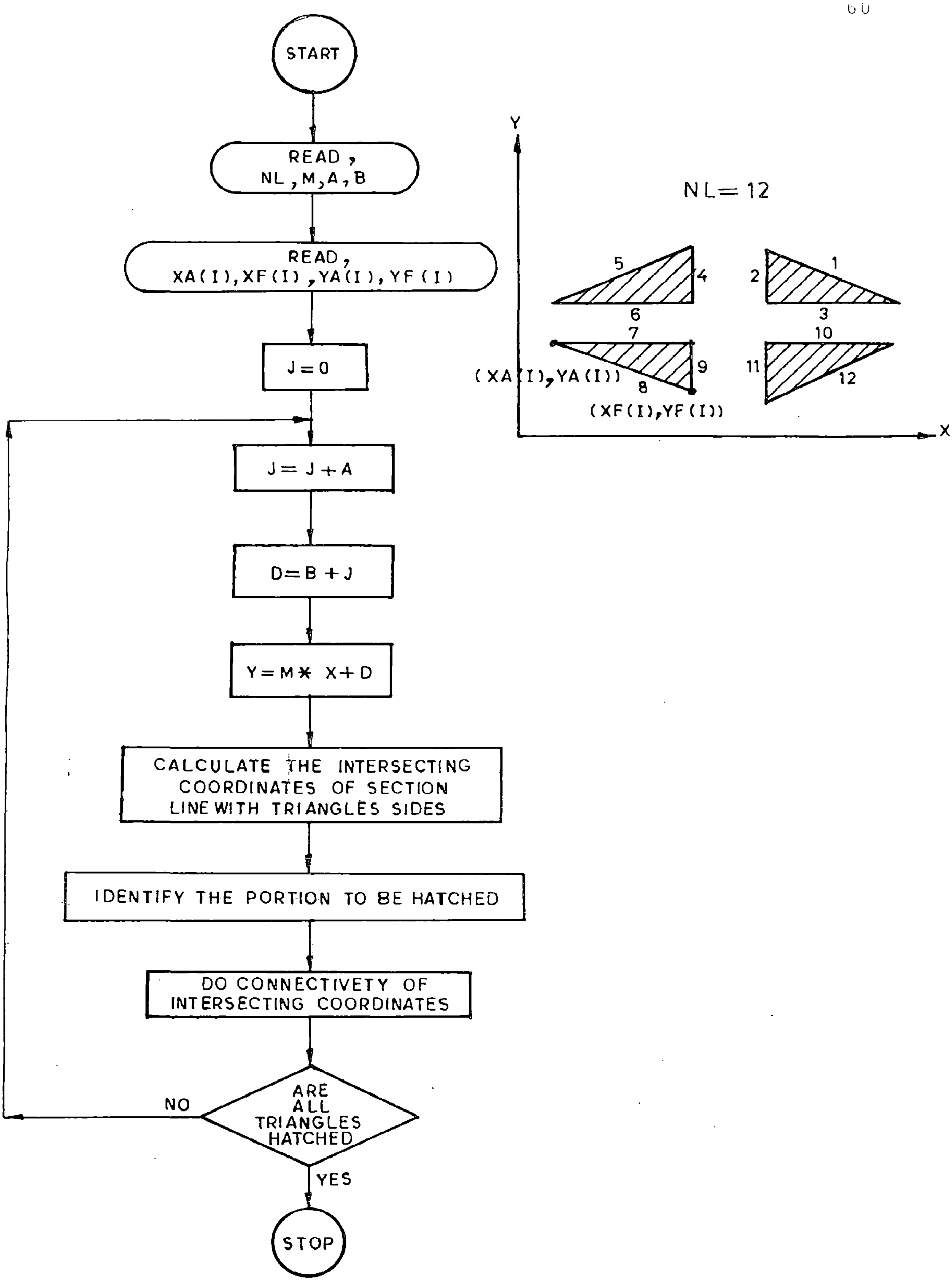


FIG.3.15. FLOW CHART FOR SUBROUTINE HAPOYG

3.5.3 Subroutine KNURL

Basic Feature : Knurling of rectangle

Data : Slope of section lines, M(K)

: Parameters defining the rectangle boundary, P(K), Q(K), R(K), S(K)

: Parameter A (defining the distance between section lines)

: Parameter B (defining the region to be knurled)

Lower entities used : (MOVE, DRAW)/PLOT, HARETA

New entity name : KNURL (P(K), Q(K), R(K), S(K),M(K))

Tailored connection : Mathematical connection between intersecting points obtained by solving intersection equation of Y=M(K)*X+D and boundary of rectangle . (here boundary of rectangles remain same but slope of section line will get change)

The flow chart of subroutine is given in Fig. 3.16 and output in Fig. 3.14.

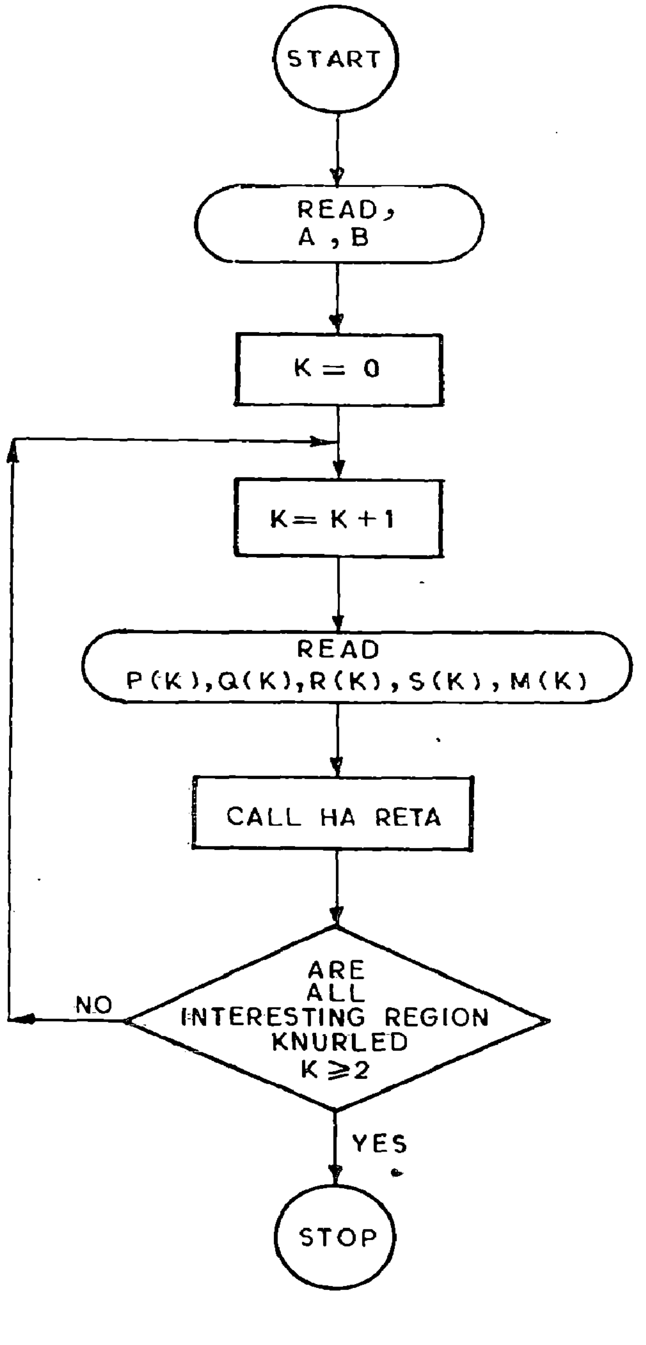


FIG.3.16. FLOW CHART FOR SUBROUTINE KNURL

3.5.4 Subroutine THRD

Basic Feature : To thread a rectangle section

Data : Slope of section lines, M(K)

: Parameters defining the rectangles boundary, P(K), Q(K), R(K), S(K)

: Parameter A(defining the distance between section lines)

: Parameter B (defining the region of threading)

Lower entities used : (MOVE, DRAW)/PLOT, HARETA

New entity name : THRD (P(K), Q(K), R(K), S(K), M(K))

Tailored connection : Mathematical connection between intersecting points obtained by solving intersection equation of Y=M(K)*X+D and boundary of rectangles (here boundary of rectangles will get change but slope of section line remain same).

The flow chart of subroutine is given in Fig. 3.17 and output in Fig. 3.14.

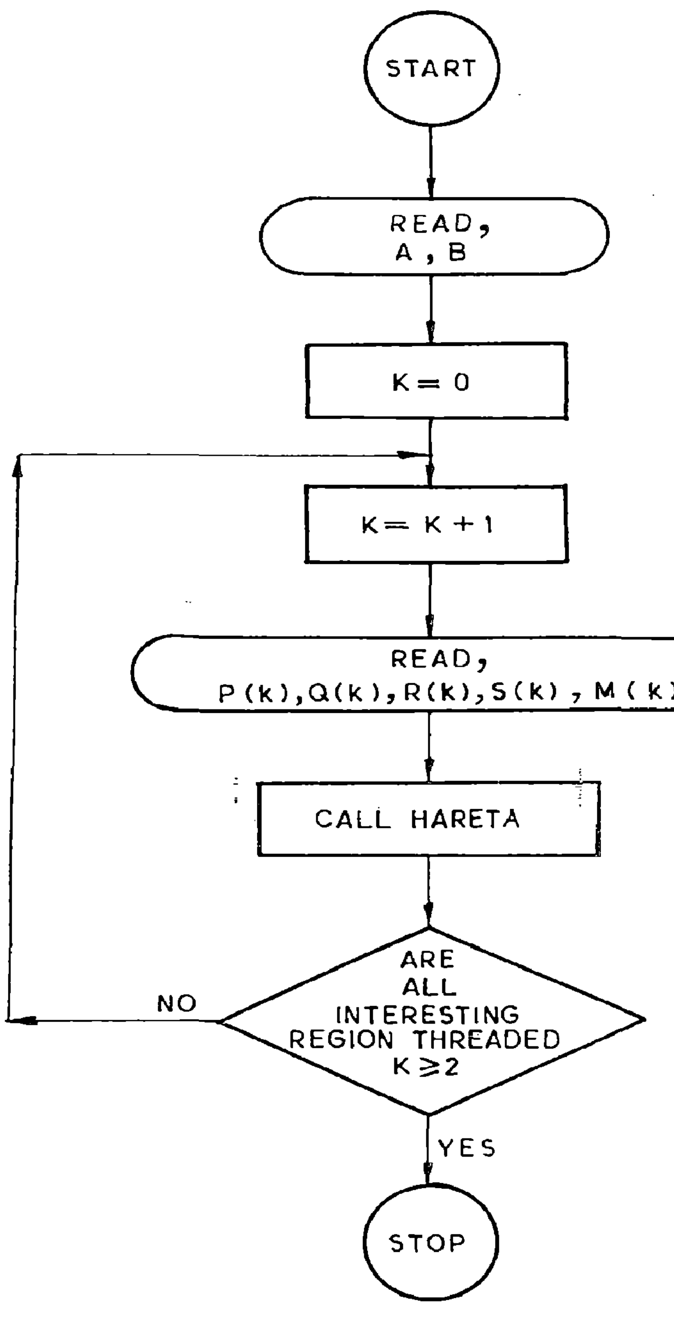


FIG. 3.17. FLOW CHART FOR SUBROUTINE THRD

3.5.5 Subroutine CONIC1

Basic Feature : To hatch any number of conics section

Data : Number of conics section to be hatched, NARC

: Slope of section line, M

: Parameters defining the conic section equation, A(I), B(I), XC(I), YC(I)

: Vector of conic section , C(I)

: Parameter AD(defining the distance between section lines)

: Parameter BD(defining the region of hatching)

Lower entities used : (MOVE, DRAW)/PLOT, CONIC

New entity name : CONIC1(A(I), B(I), C(I), XC(I), YC(I), M)

Tailored connection : Mathematical connection between intersecting points, obtained by solving intersection equation of Y=MX+D and boundary equation of $A(X-XC)^2+B(Y-YC)^2=C^2$

The flow chart of subroutine is given in Fig. 3.18 and output in Fig. 3.19.A.

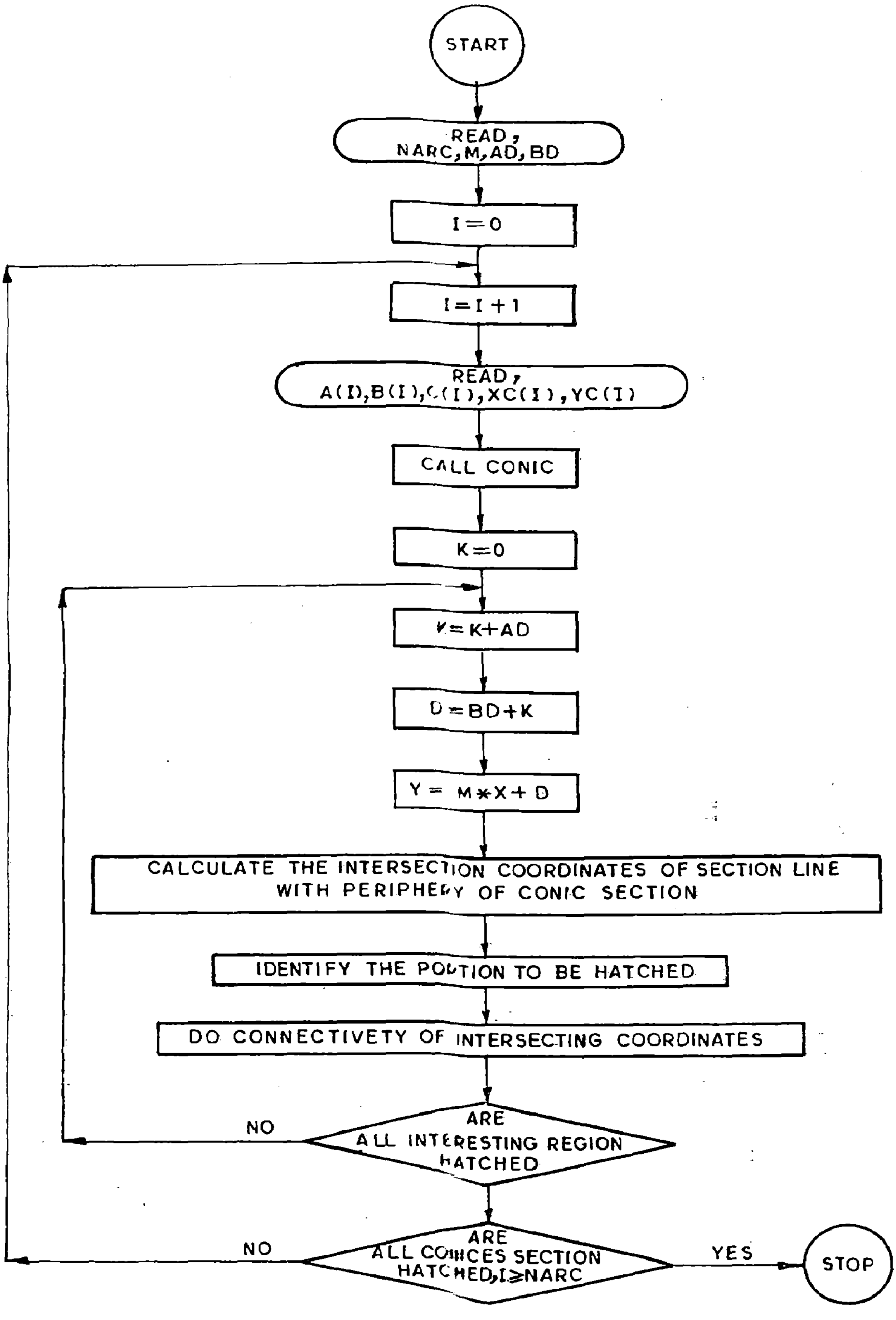


FIG.3.18. FLOW CHART FOR SUBROUTINE CONIC1

TWO-DIMENSIONAL DRAFTING MODULES

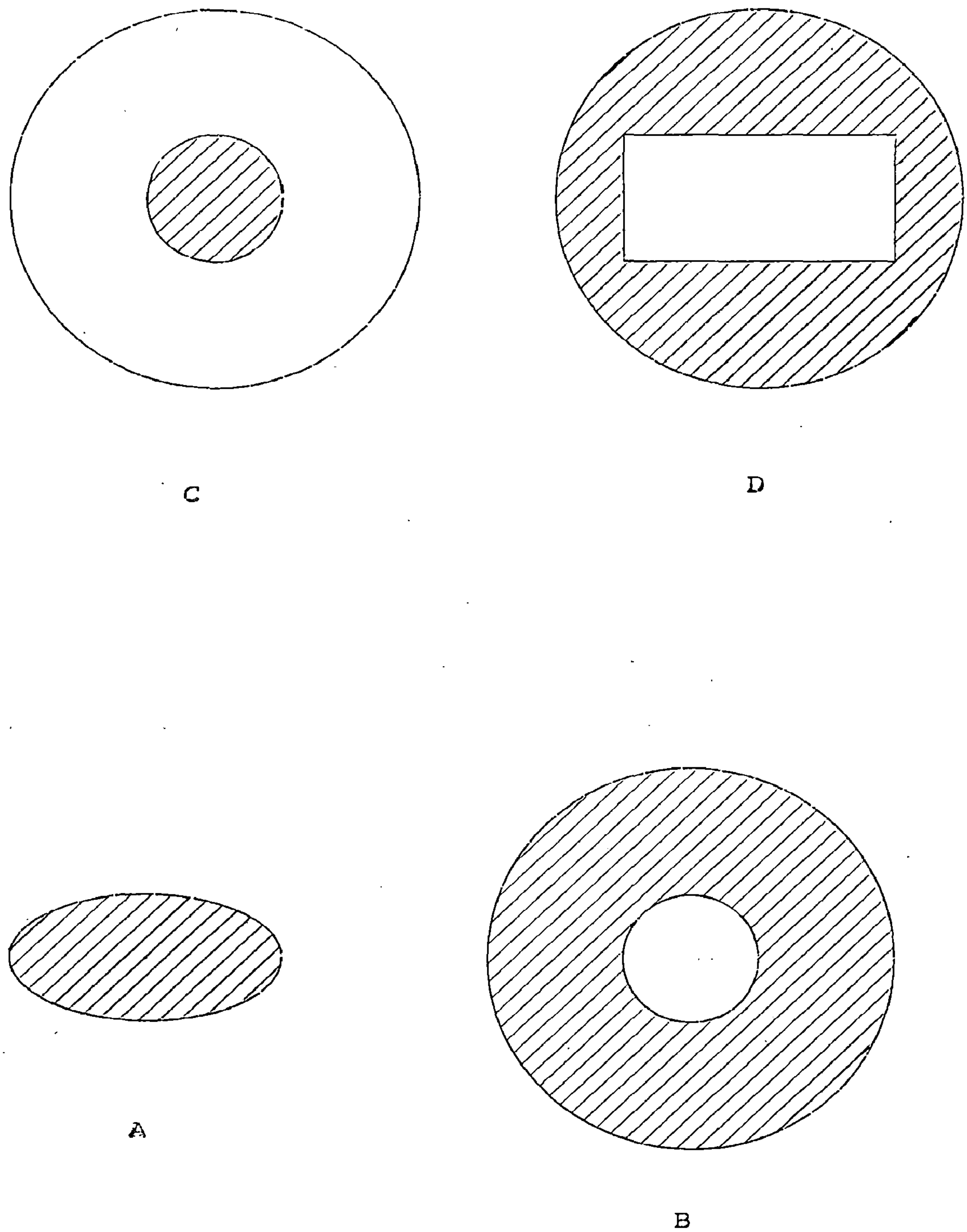


FIGURE NO 3.15 SECTIONING MODULES

3.5.5 Subroutine CONIC2 and CONIC3

Basic Feature : To hatch a hollow conic section

Data : Number of conics section, NCONIC=2

: Slope of section line, M

: Parameters defining the conics section equation, A(K), B(K)

: Vector of conics section, C(K)

:Parameter AD(defining the distance between section lines.

:Parameter BD(defining the region of hatching)

Lower entities used : (MOVE, DRAW)/PLOT, CONIC

New entity names : CONIC2 (A(K), B(K), C(K), M)

: CONIC3 (A(K), B(K), C(K), M)

Tailored connection : Mathematical connection between intersecting points obtained by solving intersection equation of Y=MX+D and boundary equations of $A(K)[X]^2+B(K)[Y]^2 = C(K)^2$

The flow chart of subroutines is given in Fig. 3.20 and output of CONIC2 in Fig. 3.19.B and CONIC3 in Fig. 3.19.C.

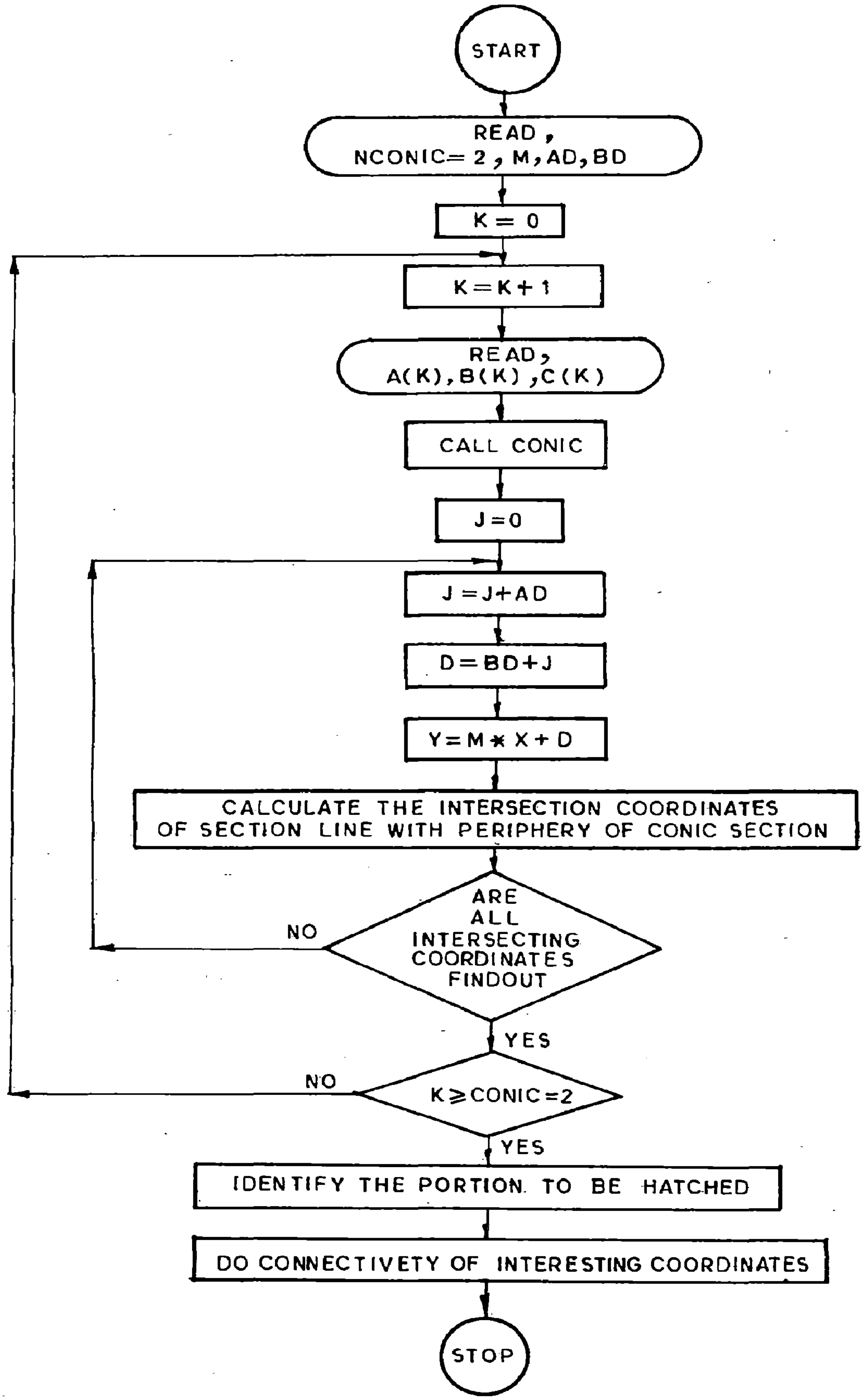


FIG.3.20. FLOW CHART FOR SUBROUTINES CONIC 2 AND CONIC 3

3.5.7 Subrutine CONIC4 and CONIC5

Basic Feature : To hatch a hollow conic section by a polygon.

Data : Slope of section line, M

: Parameters defining the conic section equation, A, B, C

: Number of sides of polygon, NL

: Initial and final coordinates of sides of polygon, XA(I), XF(I), YA(I), YF(I)

: Parameter AD(defining the distance between section lines)

: Parameter BD(defining the region of hatching)

Lower entities used : (MOVE, DRAW)/PLOT, CONIC

New entity names : CONIC4 (A,B,C,M,NL,XA(I), XF(I), YA(I), YF(I))

: CONIC5 (A,B,C,M,NL,XA(I), XF(I), YA(I), YF(I))

Tailored connection : Mathematical connection between intersecting points obtained by solving intersection equation of Y=M*X+D, boundary equation of $AX^2+BY^2 = C^2$ and boundary of polygon.

The flow chart of subroutines is given in Fig. 3.21 and output of CONIC4 in Fig. 3.19.D and CONIC 5 in Fig. 3.22.A.

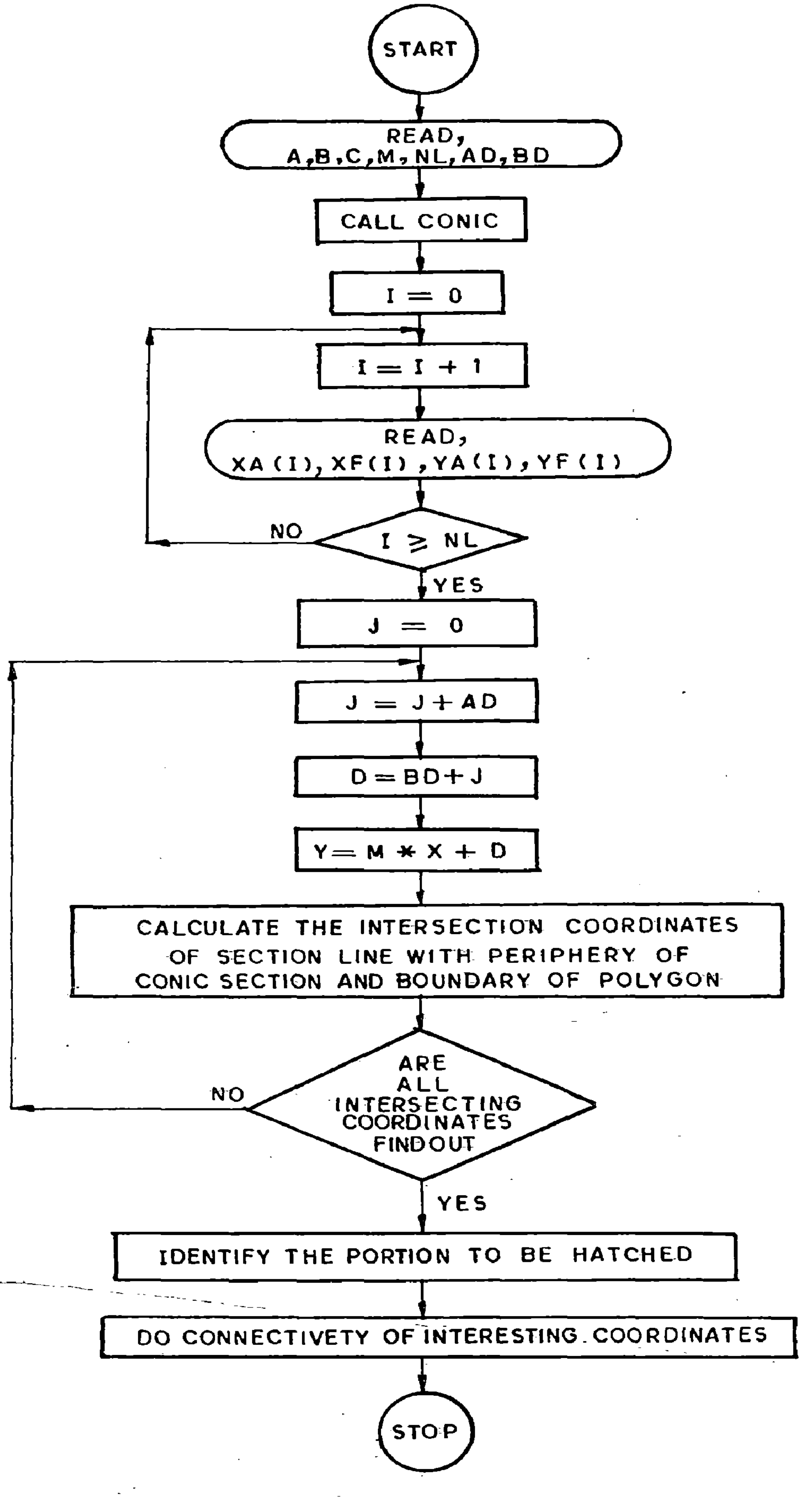
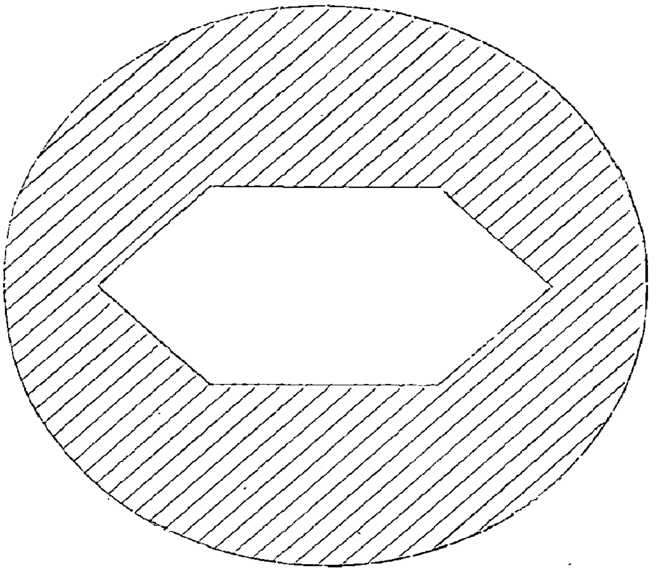
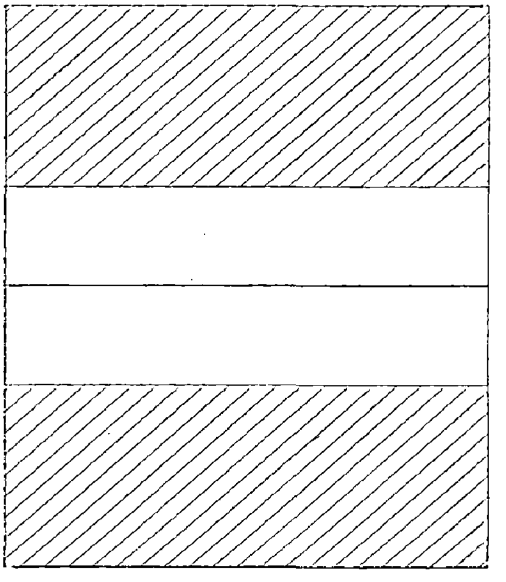


FIG.3.21. FLOW CHART FOR SUBROUTINES CONIC 4 AND CONIC 5

TWO-DIMENSIONAL DRAFTING MODULES



A



B

FIGURE NO 3.22 SECTIONING MODULES

3.5.8 Subroutine CONIC6

Basic Feature	:	To hatch hollow conic section by polygon in longitudinal direction.
Data	:	Slope of section line, M
	:	Number of rectangles to be hatched, NRECT
	:	Number of sides, NLIN
	:	Parameters defining the rectangle boundary, P(K), Q(K), R(K), S(K)
	:	Parameter A (defining the distance between section lines)
	:	Parameter B (defining the region of hatching)
	:	Initial and final coordinates of sides, XO(I), YO(I), XF(I),YF(I)
Lower entities used	:	(MOVE,DRAW)/PLOT,HARETA
New entity name	:	CONIC6 (M,NRECT,NLIN,P(K),Q(K), R(K),S(K),XO(I),YO(I),XF(I),YF(I))
Tailored connection	:	Mathematical connection between intersecting points obtained by solving intersection equation of Y=MX+D and boundary of rectangles.

The flow chart of subroutine is given in Fig.3.23 and output in Fig. 3.22.B.

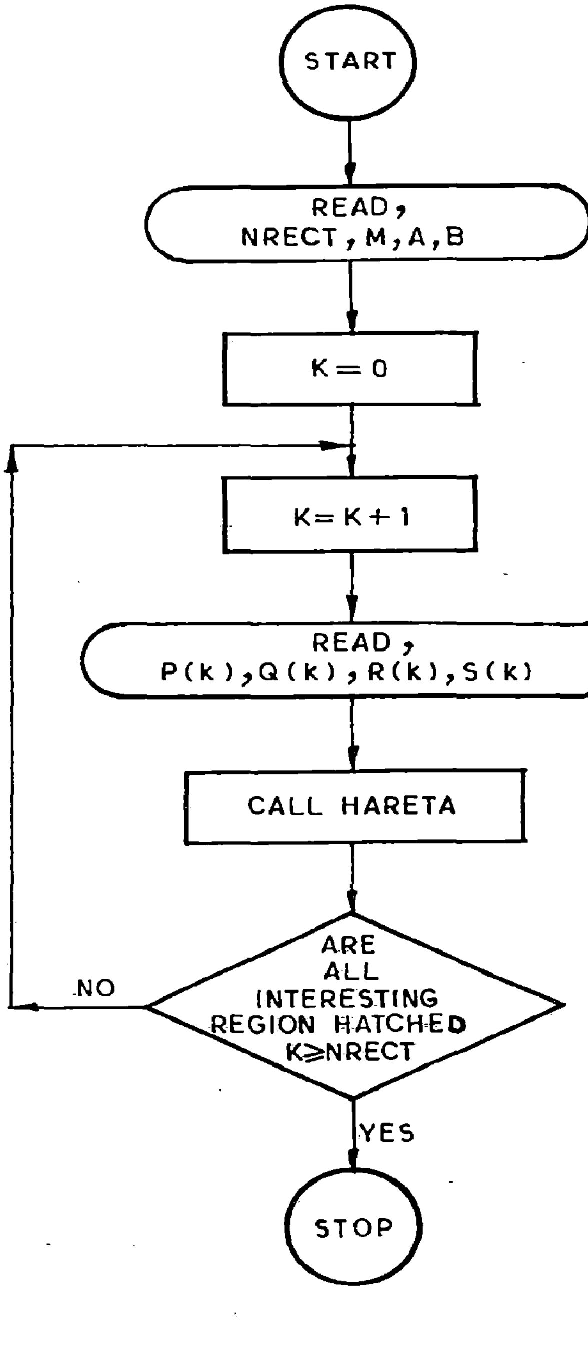


FIG. 3.23. FLOW CHART FOR SUBROUTINE CONIC 6

3.6 HATCHING OF COMPLEX GEOMETRIES

As mentioned before, individual users can often multiply their productivity by tailoring the modules to their special application needs. For instance, we have been able to hatch the complex geometry as shown in Fig. 3.24, by writing a tailored set of functions which prompt for a minimal number of parameters before automatically generating complete hatching.

Here the geometry has been discretized in to three rectangles and two triangles (Fig. 3.25.A). The flow chart coupling the various modules in the required program is given in Fig. 3.25.B.

TWO-DIMENSIONAL DRAFTING MODULES

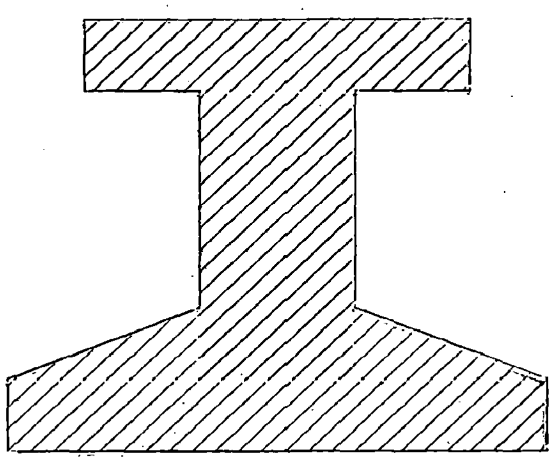


FIGURE NO 3.24 SECTIONING MODULES

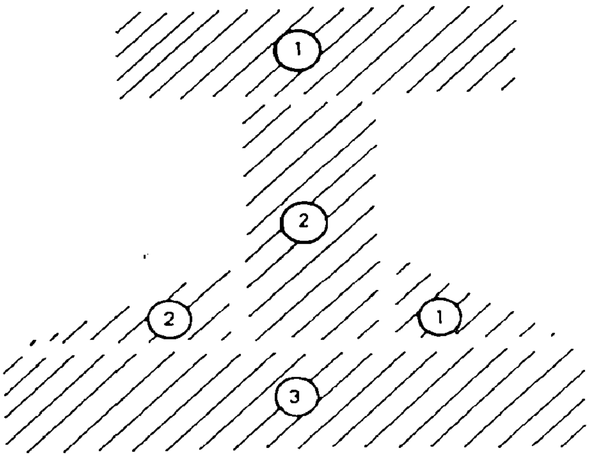


FIG. 3.25. A

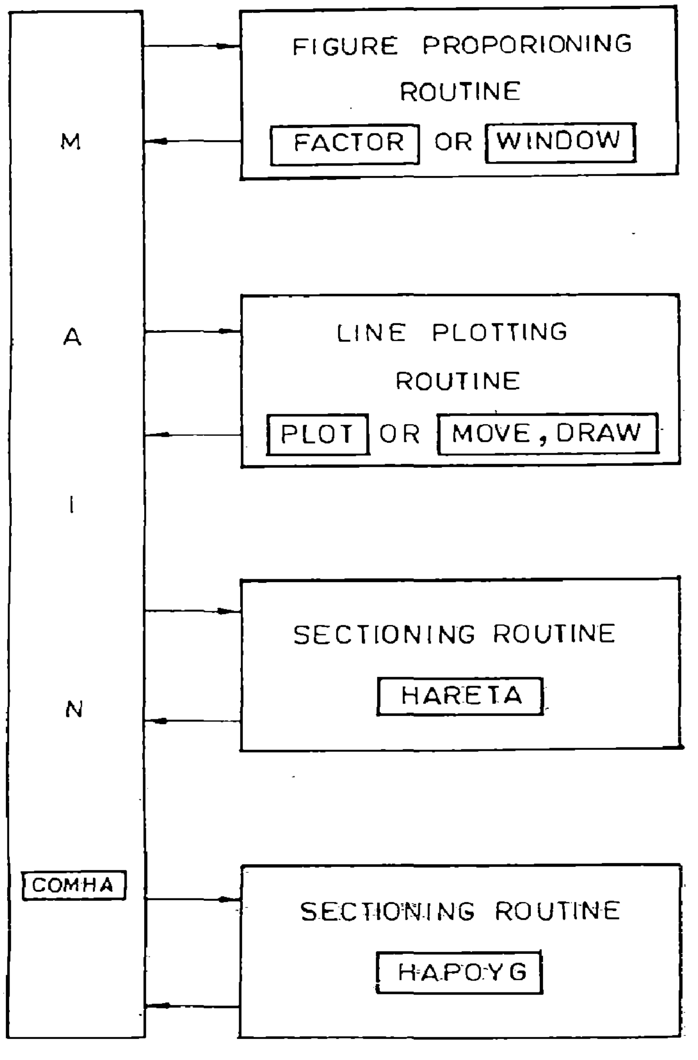


FIG.3.25.B. FLOW CHART FOR HATCHING OF COMPLEX GEOMETRY

CHAPTER - IV

HIERARCHICAL HIGHEST RANK MODULES

4.1 BASIC CONCEPT

The highest rank of graphic software is the application software. An application program is a complete problem solver. A user need only supply data and select among program options to obtain the desired graphical output, no programming is required on the user's part. Application program are written in a higher level language such as FORTRAN IV.

4.2 LINK BETWEEN SOFTWARES

The matter how all these modules can be linked together into a coherent system which will provide a quick and cost effective means of converting lowest and middle rank modules into an application program, is discussed below.

There are many problems to be faced in terms of compatibility of different computer systems and hardware peripherals, compatibility of computer languages, compatibility of software system in terms of their input requirements, structure and logic .

Thereafter it is mainly a question of ensuring that the various software modules are readily portable, and that the style and form of input and output are clearly defined so as to facilitate the writing of application software interfaces. This will enable the modules to be linked together, but will also allow them each to be used in a stand-alone manner when desired. This offer the prospects of reduction in time.

In the present chapter, lower level and middle level softwares tailoring, capable of producing detail drawing of mechanical elements, is discussed.

Description of some program capable of converting data bases to graphic images is given in the following sections. Section 4.3 is devoted for creating title block and sheet layout as per I.S.I. code. Section 4.4 deals with simple problems of drawing. Section 4.5, 4.6 and 4.7 describe the concatenated structure of the program, built up using different level softwares.

4.3 PROGRAM TO DRAW A1SHEET LAYOUT AND TITLE BLOCK

The program contain two subroutines A1SLA and TITBL.

4.3.1 Subroutine A1SLA

Basic Feature : To draw A1 sheet layout

Data : Initial and final coordinates of sides, X1(I), Y1(I), X2(I), Y2(I)

: Number of sides, NLI

Lowest entities used : (MOVE, DRAW)/PLOT,SYMBOL, FACTOR/ WINDOW

New entity name: A1SLA

4.3.2 Subroutine TITBL

Basic Feature : To draw title block

Data : Initial and final coordinates of sides, X1(I), Y1(I), X2(I), Y2(I)

: Number of sides, NLI

Lowest entities used : (MOVE,DRAW)/PLOT,SYMBOL,CONIC, FACTOR/WINDOW

New entity name: TITBL

The flow chart for plotting of A1 sheet layout and title block is given in Fig. 4.1.

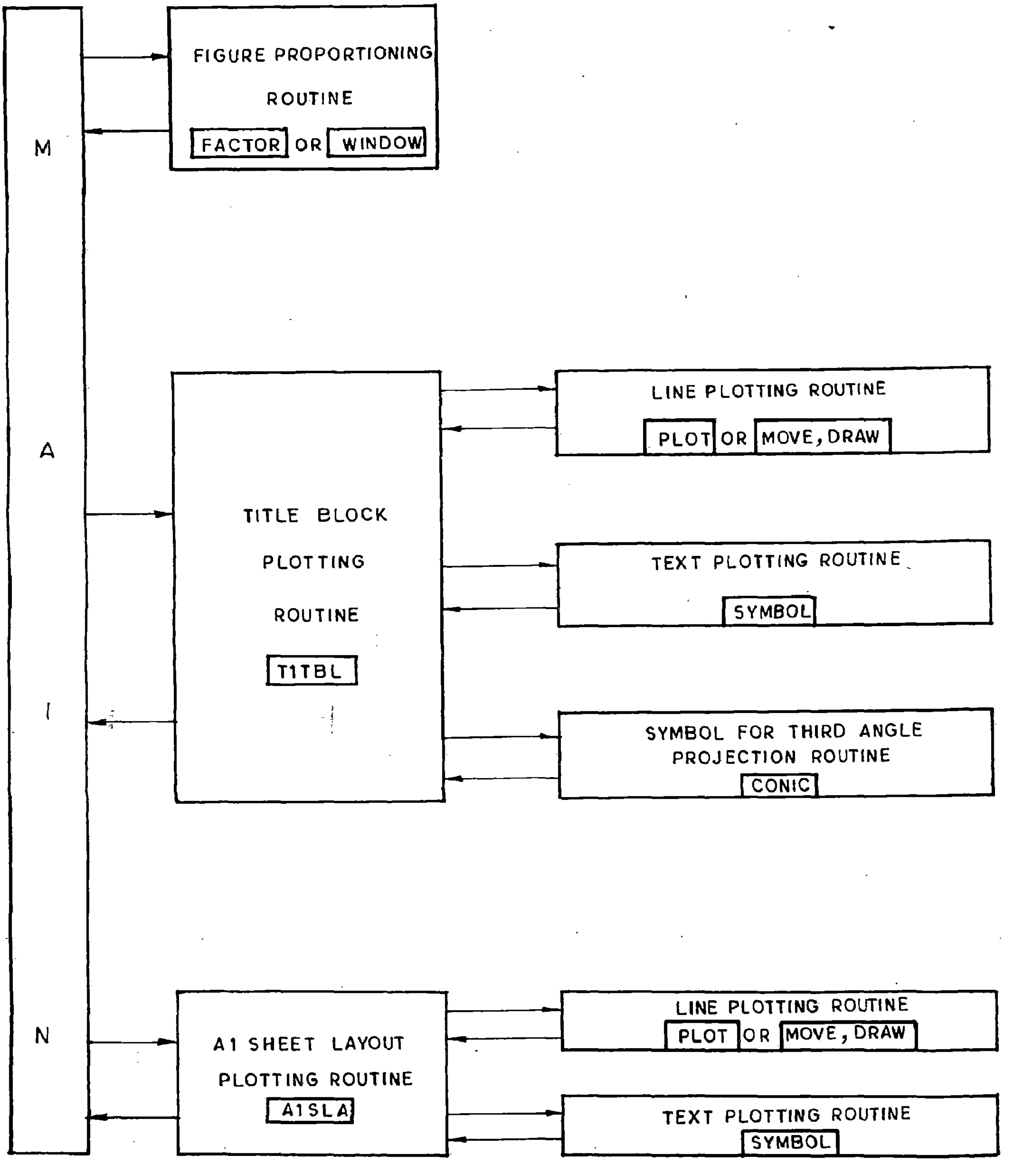


FIG.4.1. FLOW CHART TO PLOT A1 SHEET LAYOUT AND TITLE BLOCK

4.4 PROGRAM SNOOC AND SNOOT

This program is capable to plot the incomplete orthographic views of simple object (missing lines).

Here we have executed the program for a set of eight objects, the output of orthographic views is shown in sheet 4.1 and the flow chart of program is shown in Fig. 4.2. The program SNOOC is applicable to Calcomp plotter and SNOOT TO Tektronix graphic terminal.

4.5 PROGRAM SNO1C AND SNO1T

This program is capable to plot the details drawing of the caster. It contains five part, namely, support arm (body), spacer, rod, wheel, pin.

Fig. 4.3 shows the flow chart of details drawing of caster, Fig. 4.4 shows the flow chart of plotting of support arm (PARNO1), Fig. 4.5 shows the flow chart of plotting of spacer (PARNO2), Fig. 4.6 shows the flow chart of plotting of rod (PARNO3), Fig. 4.7 shows the flow chart of plotting of wheel (PARNO4), Fig 4.8 shows the flow chart of plotting of pin (PARNO5).

The program SNO1C is applicable to Calcomp plotter and program SNO1T is applicable to Tektronix graphic terminal. The output of program is shown in Sheet 4.2.

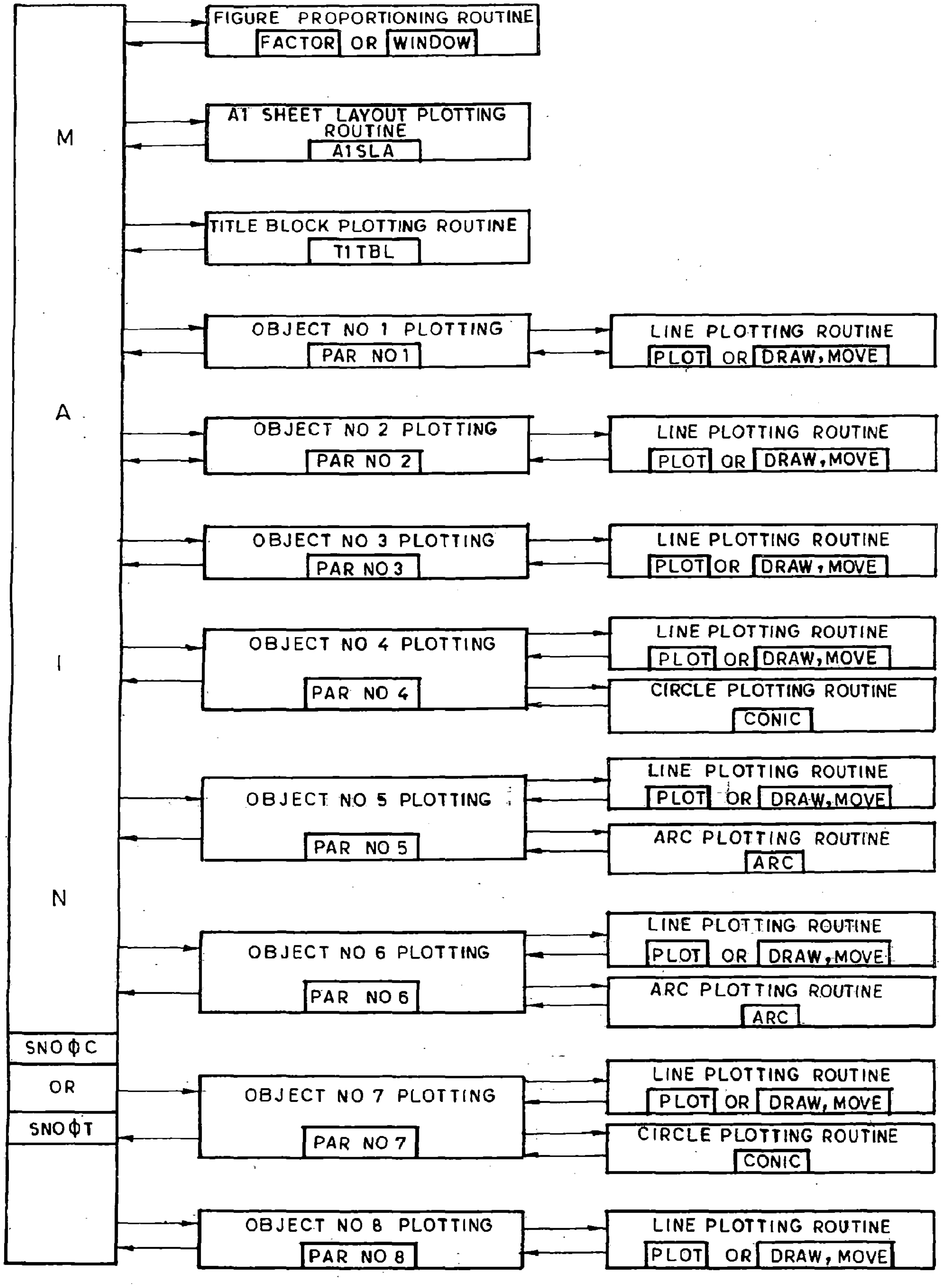
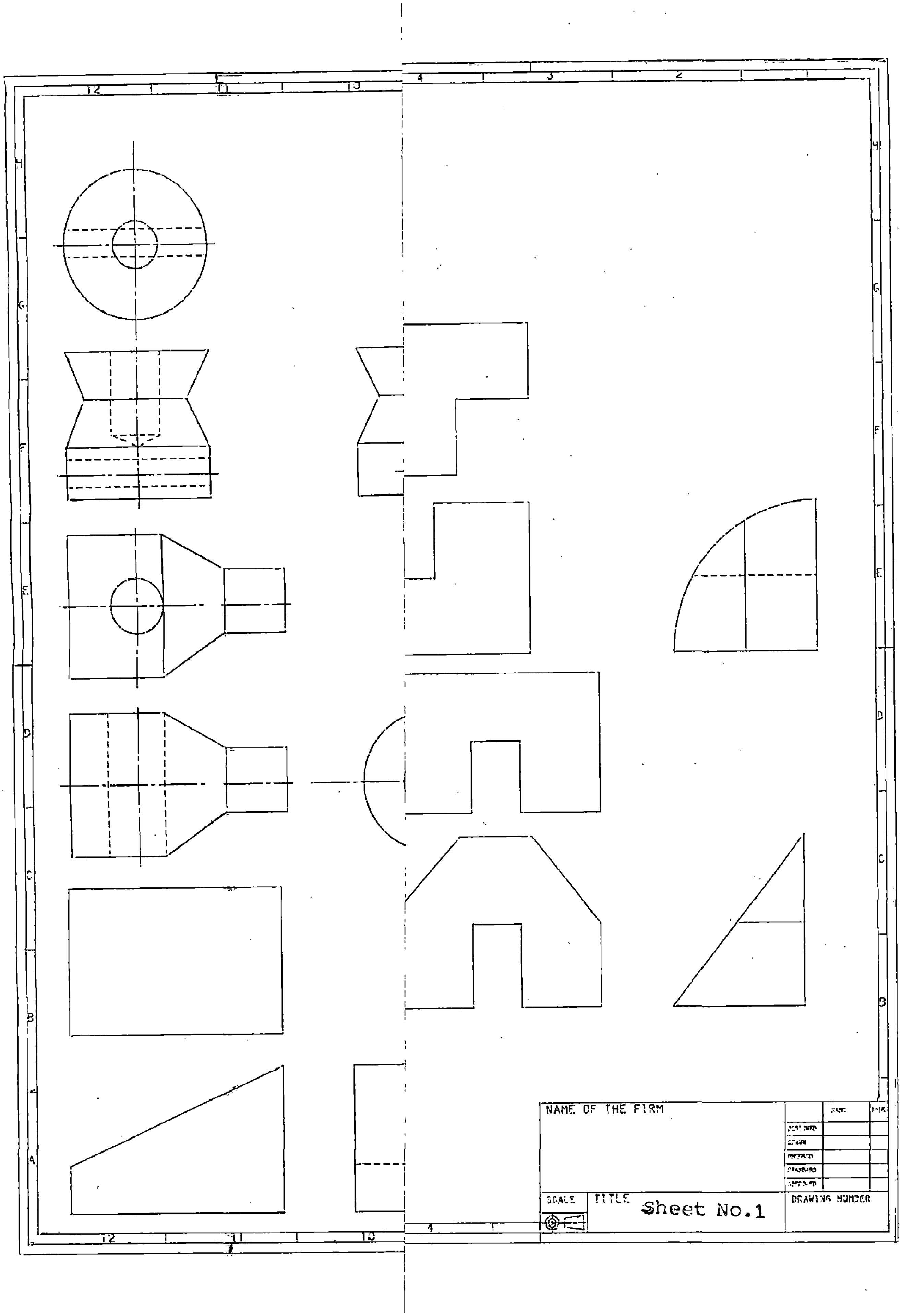


FIG.4.2.FLOW CHART FOR PROGRAM SNOΦC AND SNOΦT


NAME OF THE FIRM
SCALE
TITLE
Sheet No.1
DRAWING NUMBER

4.5.1 Data Sequencing for Supporting Arm (PAR NO 1)

PLOT	NLI	-	-	-	-	-	-	Refer sub-routine PLOT	Appendix (I.1)
	X1(2)	Y1(I)	X2(I)	Y2(I)	-	-	-		
ARHI	NARO	X	-	-	-	-	-	Refer Fig.3.9	"
	XO(I)	YO(I)	XL(I)	-	-	-	-		
ARHE	NARO	X	-	-	-	-	-	Refer Fig.3.9	"
	XO(I)	YO(I)	XL(I)	-	-	-	-		
ARVI	NARO	X	-	-	-	-	-	" "	"
	XO(I)	YO(I)	XL(I)	-	-	-	-		
ARVE	NARO	X	-	-	-	-	-	" "	"
	XO(I)	YO(I)	XL(I)	-	-	-	-		
CONIC	NARC	-	-	-	-	-	-	Refer Fig.3.5	"
	A(I)	B(I)	C(I)	XC(I)	YC(I)	-	-		
NEWTR	N	NC	-	-	-	-	-	Refer Fig.3.7	"
	A1(I)	B1(I)	XC1(I)	YC1(I)	C1(I)	-	-		
	A2(I)	B2(I)	XC2(I)	YC2(I)	C2(I)	-	-		
	X(I)	Y(I)	-	-	-	-	-		
ARC	NARC	-	-	-	-	-	-	Refer Fig.3.6	"
	A(I)	B(I)	C(I)	XC(I)	YC(I)	XOD(I)	XOD2(I)		
INAF	NARO	X	-	-	-	-	-	Refer Fig.3.11	"
	KO(I)	YO(I)	XL(I)	ALFA(I)	-	-	-		
INAH	NARO	X	-	-	-	-	-	Refer Fig.3.11	"
	XO(I)	YO(I)	XL(I)	ALFA(I)	-	-	-		
SYMBOL	NT	-	-	-	-	-	-	Refer subroutine SYMBOL	"
	TITLE	-	-	-	-	-	-		
	XT	YT	SIZE	ANG	NC	-	-		

4.5.2 Data Sequencing for Wheel (PARNO 4)

PLOT	NLI	-	-	-	-	Refer subroutine PLOT	Appendix (I.2)
	X1(I)	Y1(I)	X2(I)	Y2(I)	-		
ARHI	NARO	X	-	-	-	Refer Fig.3.9	Appendix (I.2)
	XO(I)	YO(I)	XL(I)	-	-		
ARVI	NARO	X	-	-	-	"	"
	XO(I)	YO(I)	XL(I)	-	-		
INAF	NARO	X	-	-	-	Refer Fig.3.11	"
	XO(I)	YO(I)	XL(I)	ALFA(I)	-		
INAH	NARO	X	-	-	-	Refer Fig.3.11	"
	XO(I)	YO(I)	XL(I)	ALFA(I)	-		
CONIC	NARC	-	-	-	-	Refer Fig.3.5	"
	A(I)	B(I)	C(I)	XC(I)	YC(I)		
HARETA	M	NRECT	-	-	-	Refer Fig.3.13	"
	P(K)	Q(K)	R(K)	S(K)	-		
HAPOY	M	NL	-	-	-	Refer Fig.3.15	"
	XA(I)	XF(I)	YA(I)	YF(F)	-		
SYMBOL	NT	-	-	-	-	Refer subroutine SYMBOL	"
	TITLE	-	-	-	-		
	XT	YT	SIZE	ANG	NC		

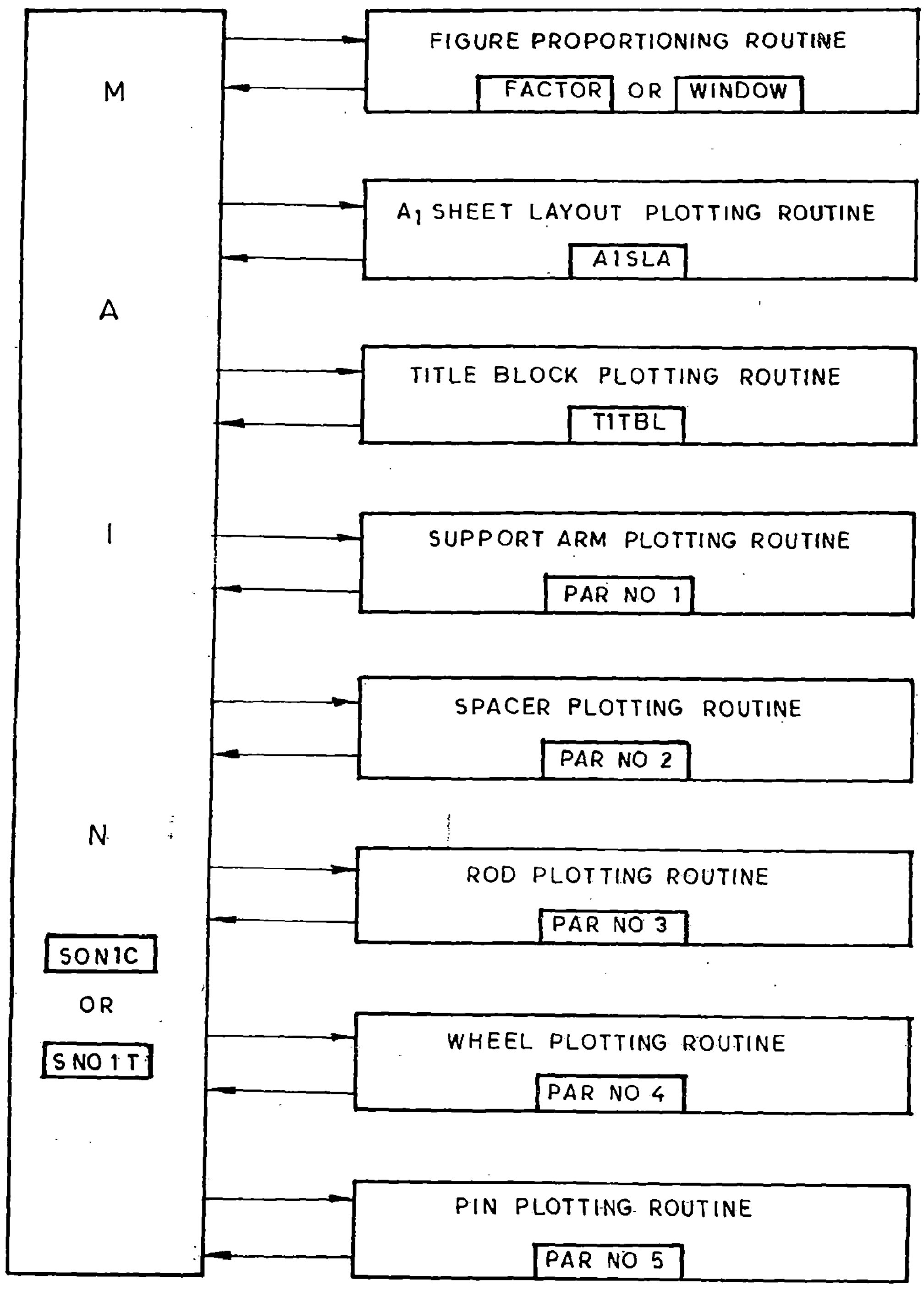


FIG.4.3. FLOW CHART TO PLOT DETAILS DRAWING OF CASTER ON A1 SHEET LAYOUT

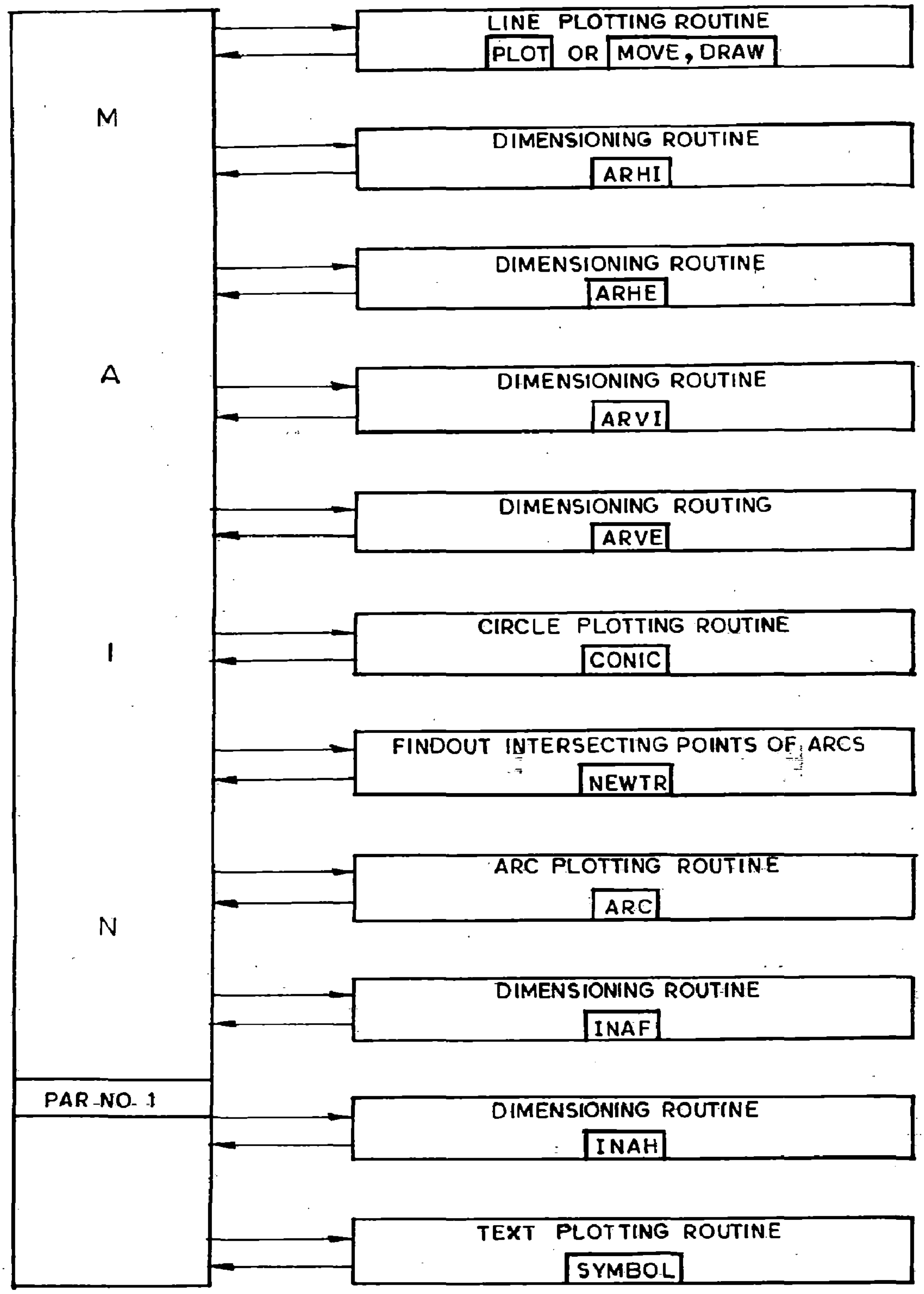


FIG.4.4. FLOW CHART TO PLOT SUPPORT ARM OF CASTER

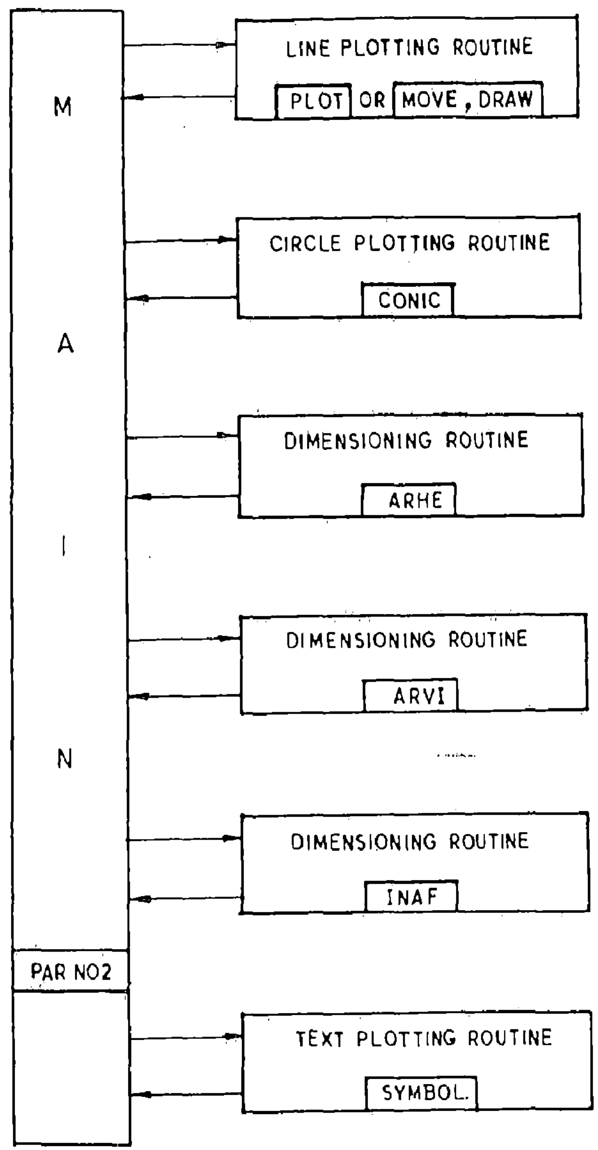


FIG.4.5. FLOW CHART TO PLOT SPACER

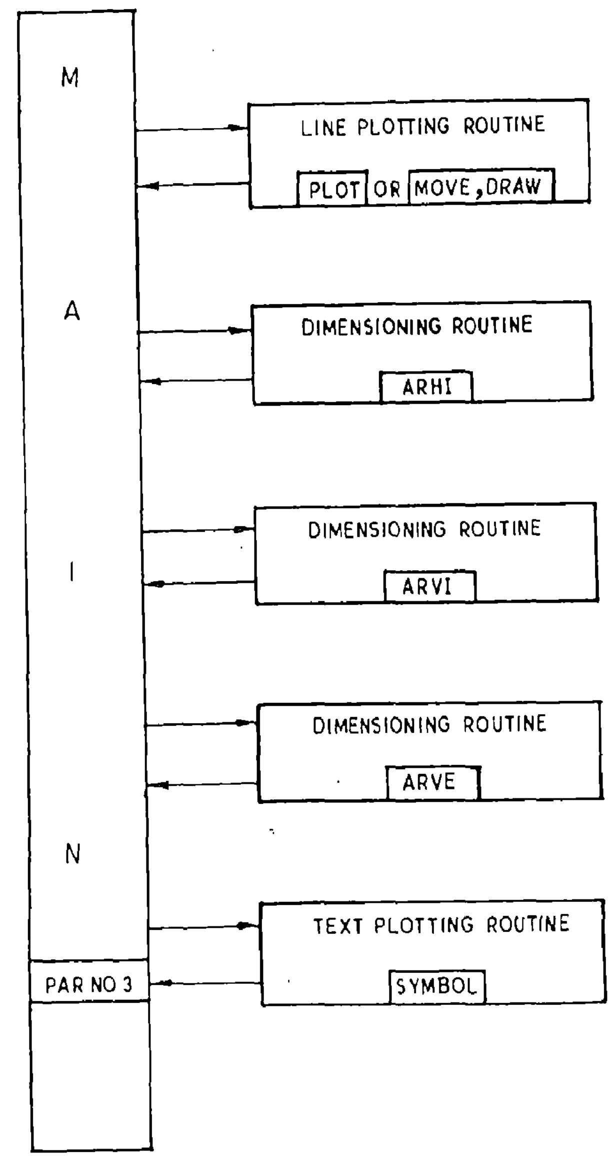


FIG.4.6. FLOW CHAR TO PLOT ROD

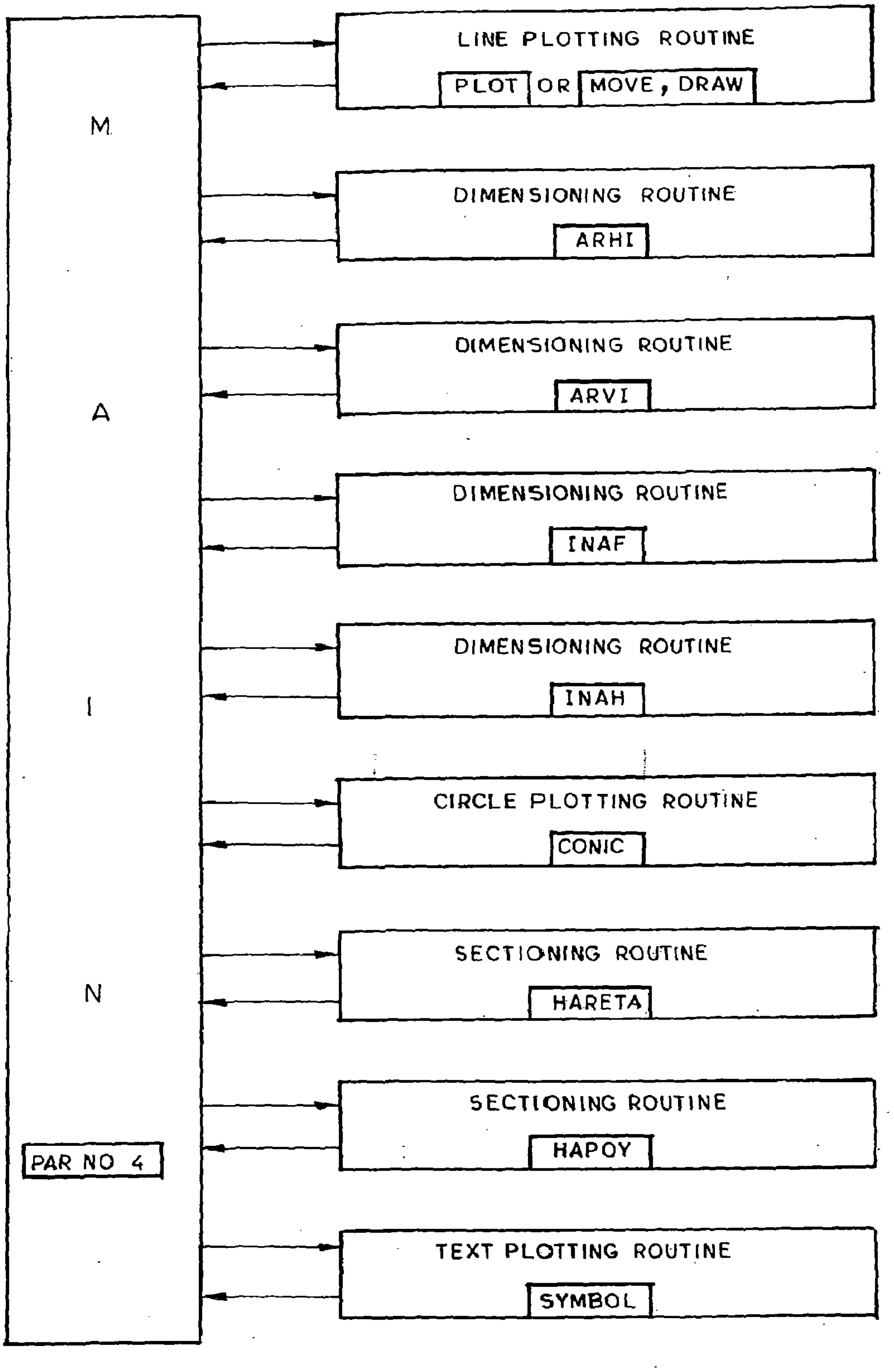


FIG. 4.7. FLOW CHART TO PLOT WHEEL

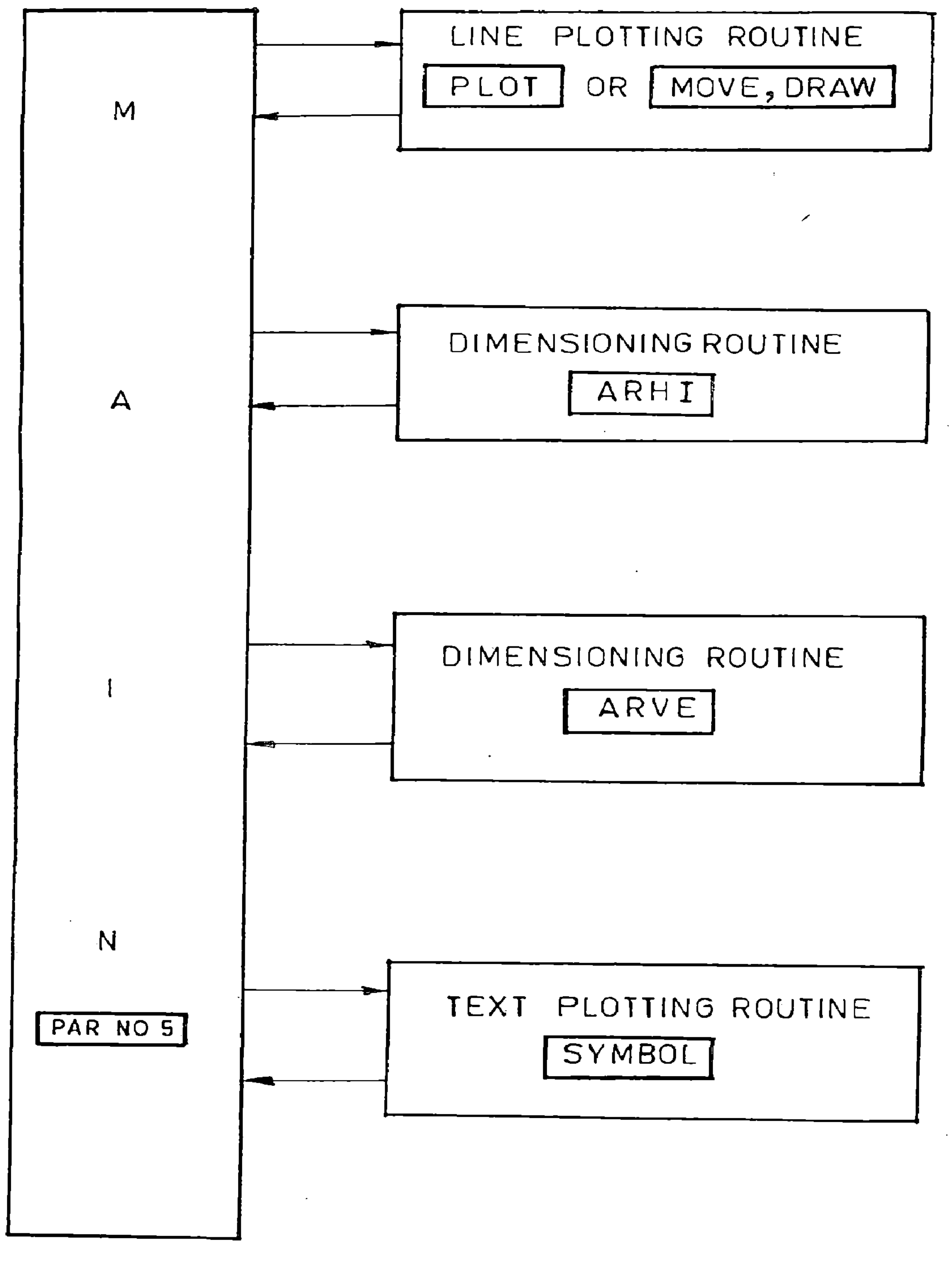
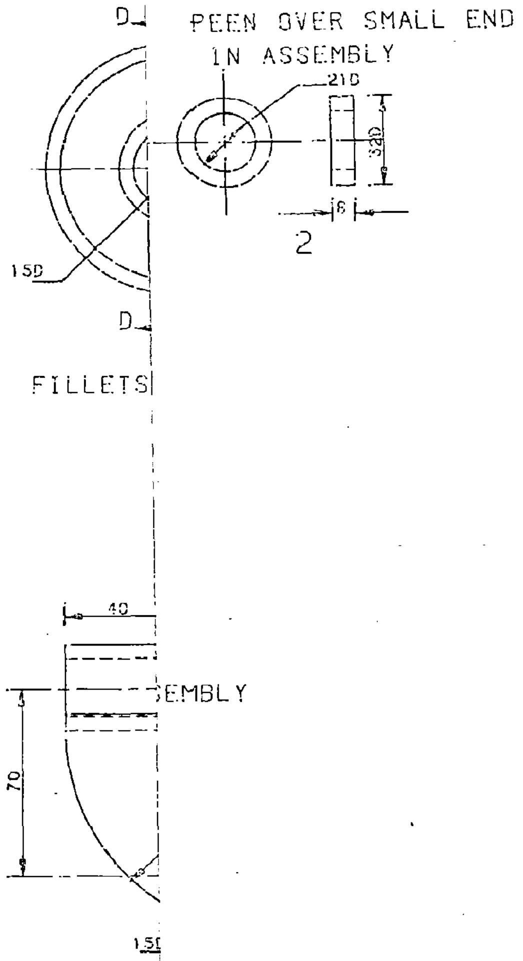


FIG. 4.8. FLOW CHART TO PLOT PIN



NAME OF THE FIRM

SCALE

TITLE Sheet No.2

DRAWING NUMBER

4.6 PROGRAM SNO2C AND SNO2T

This program is capable to plot the details drawing of tool maker's vice. It contains orthographic views of fined jaw (1 off), stop plate (1 off), orthographic views of moving jaw (1 off), spindle (1 off), M6 set screw (1 off) and M8CSK cap screw (2 off).

Fig. 4.9 shows the flow chart of details drawing of tool maker's vice, Fig. 4.10 shows the flow chart of M6 set screw (1 off) and M8CSK cap screw (2 off) (PARNO 1), Fig.4.11 shows the flow chart of spindle (1 off) (PARNO 2), Fig. 4.12 shows the flow chart of orthographic views of moving jaw (1 off) (PARNO 3), Fig. 4.13 shows the flow chart of stop plate (1 off) (PARNO 5), Fig. 4.14 shows the flow chart of orthographic views of fixed jaw (1 off) (PARNO 6).

The program SNO2C is applicable to Calcomp plotter and program SNO2T is applicable to Tektronix graphic terminal. The output of program is shown in sheet4.3.

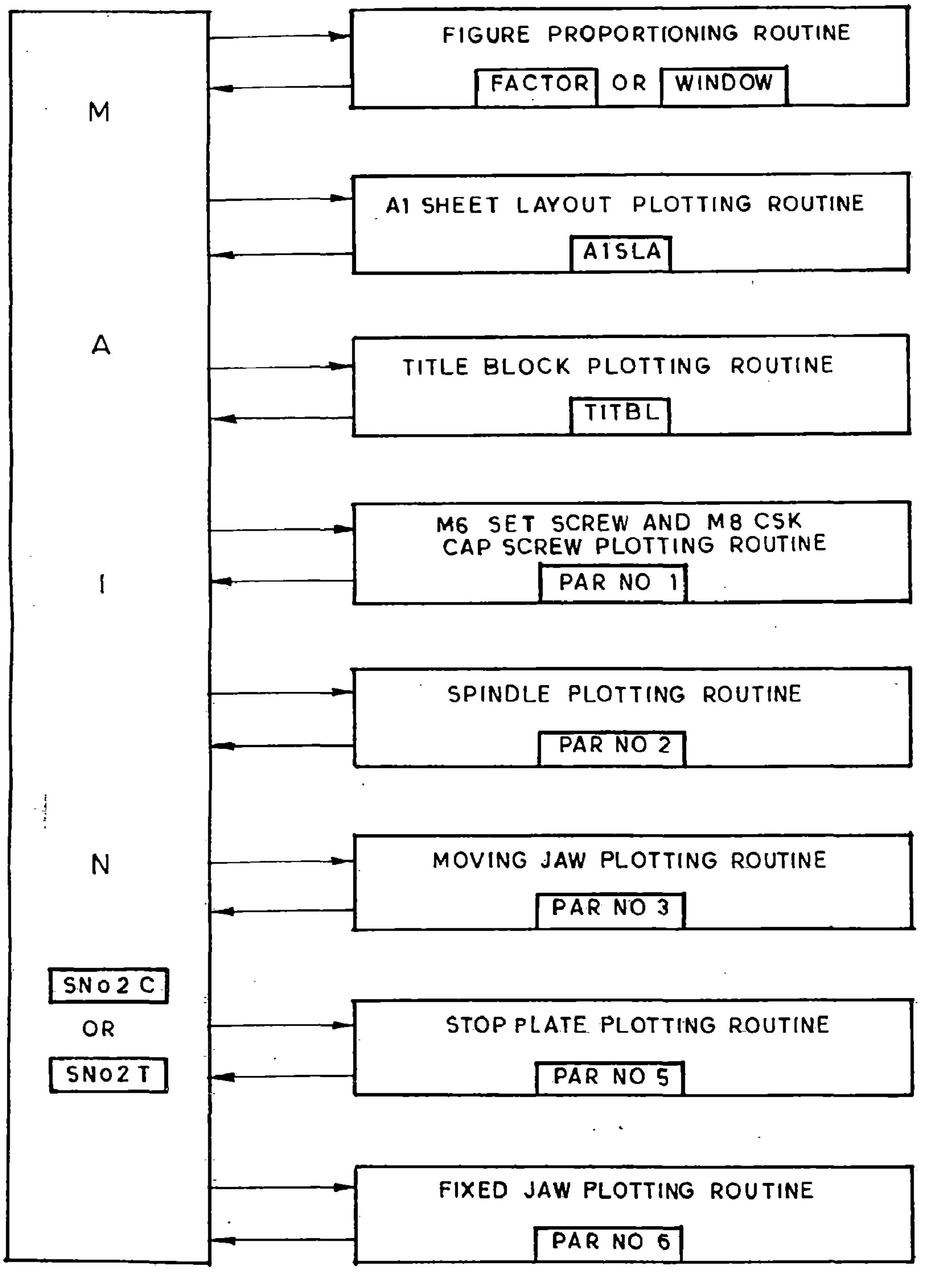


FIG.4.9. FLOW CHART TO PLOT DETAILS DRAWING OF TOOL MAKER'S VICE, ON A1 SHEET LAYOUT

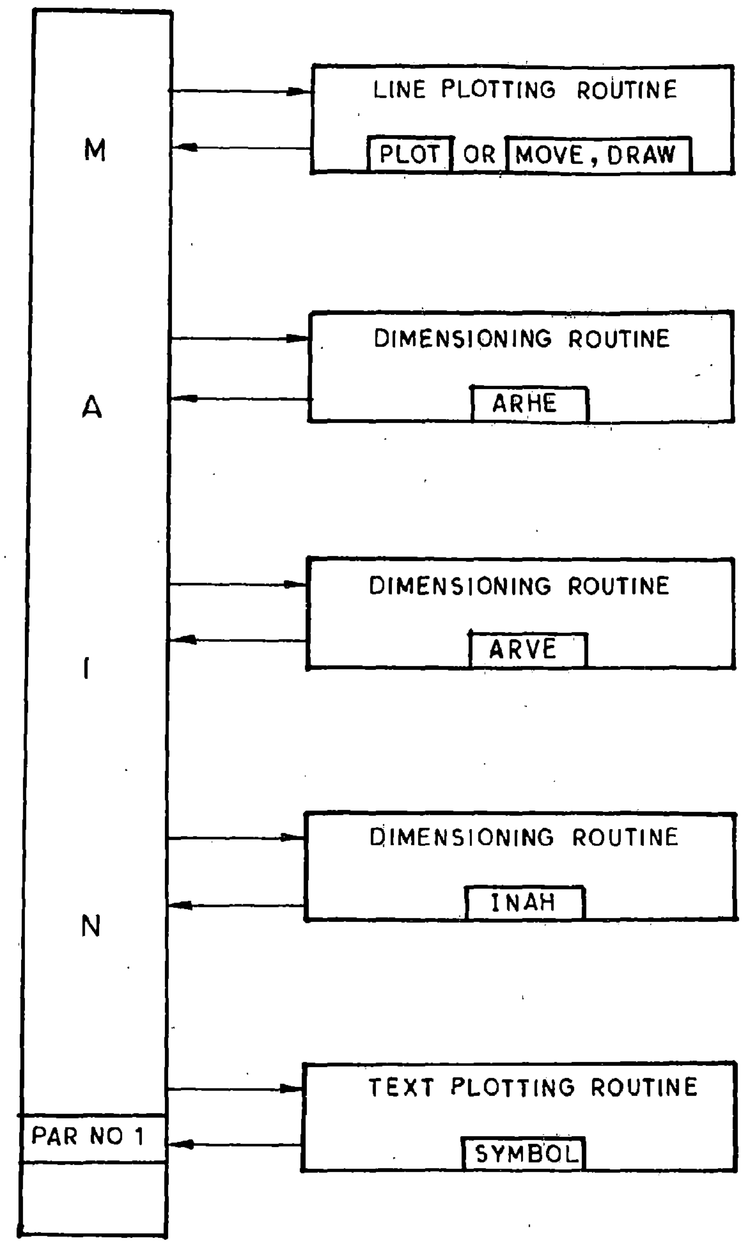


FIG.4.10. FLOW CHART TO PLOT M6 SET SCREW (1OFF) & M8 CSK CAP SCREW (2 OFF)

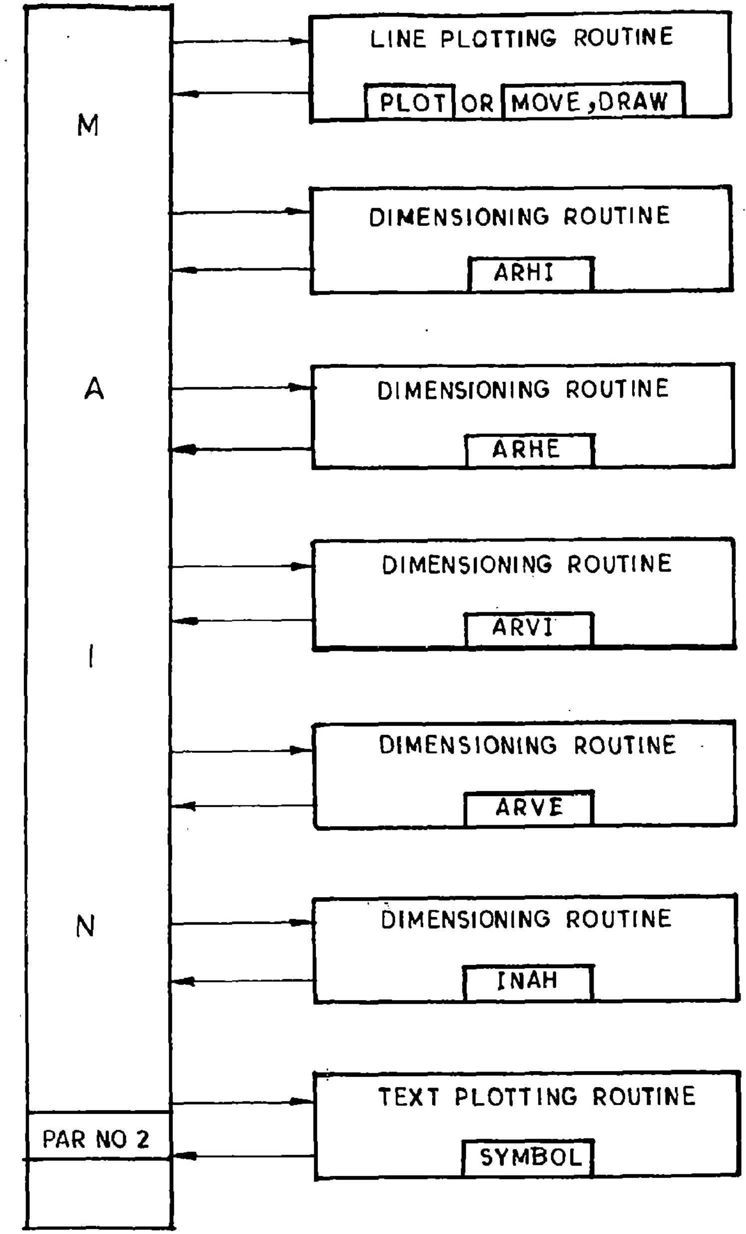


FIG.4.11. FLOW CHART TO PLOT SPINDLE (1 OFF)

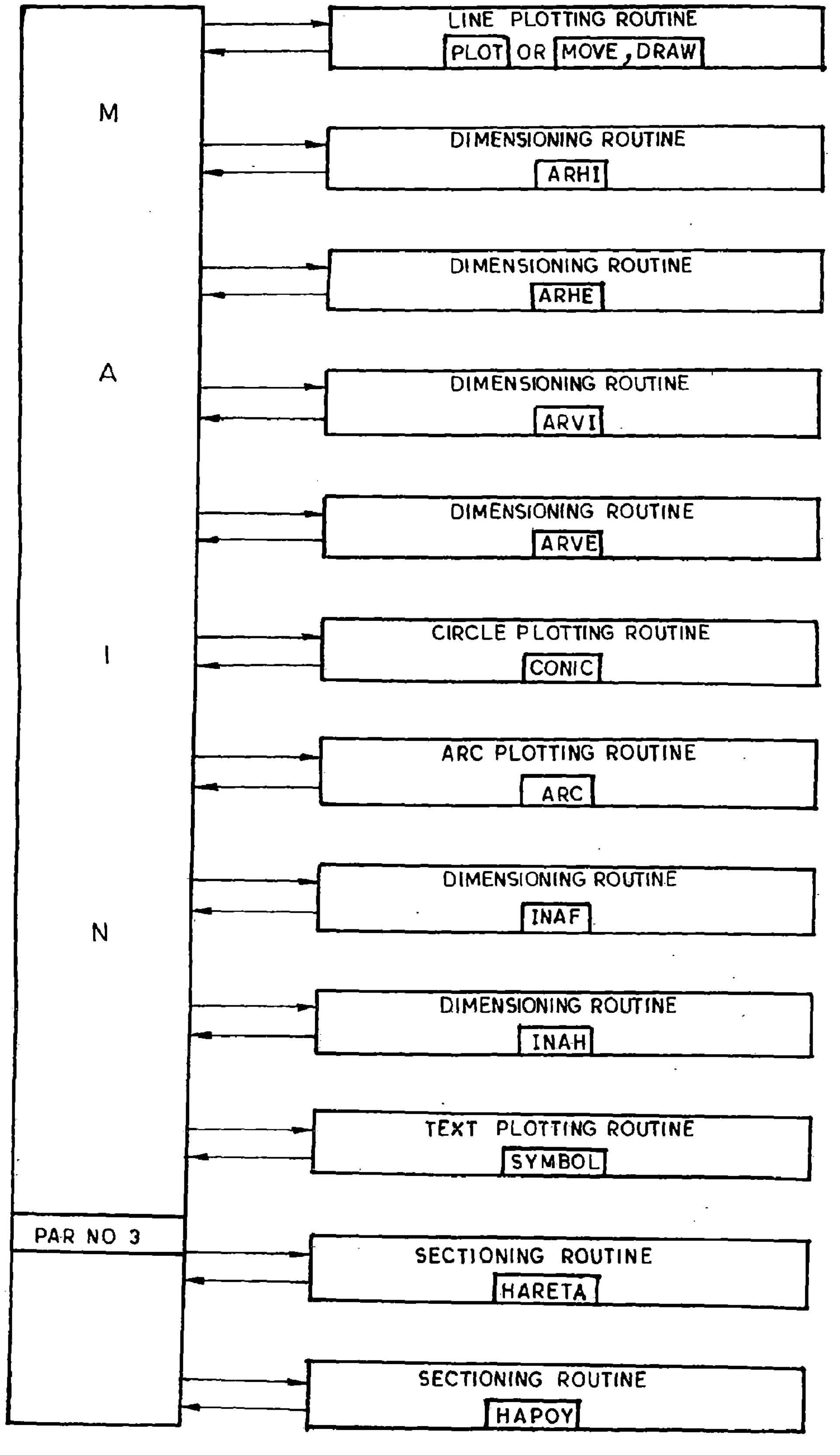


FIG.4.12. FLOW CHART TO PLOT ORTHOGRAPHIC VIEWS OF MOVING JAW (1 OFF

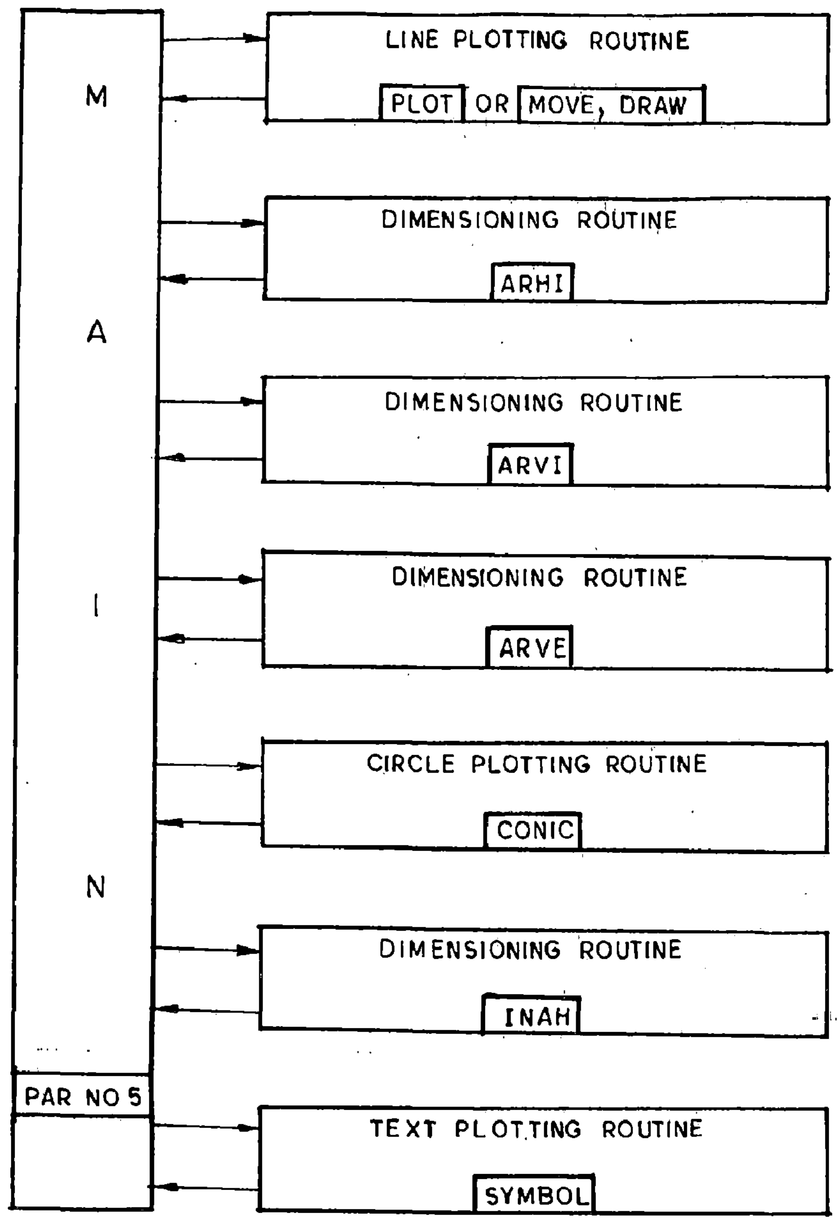


FIG.4.13. FLOW CHART TO PLOT STOP PLATE(1 OFF)

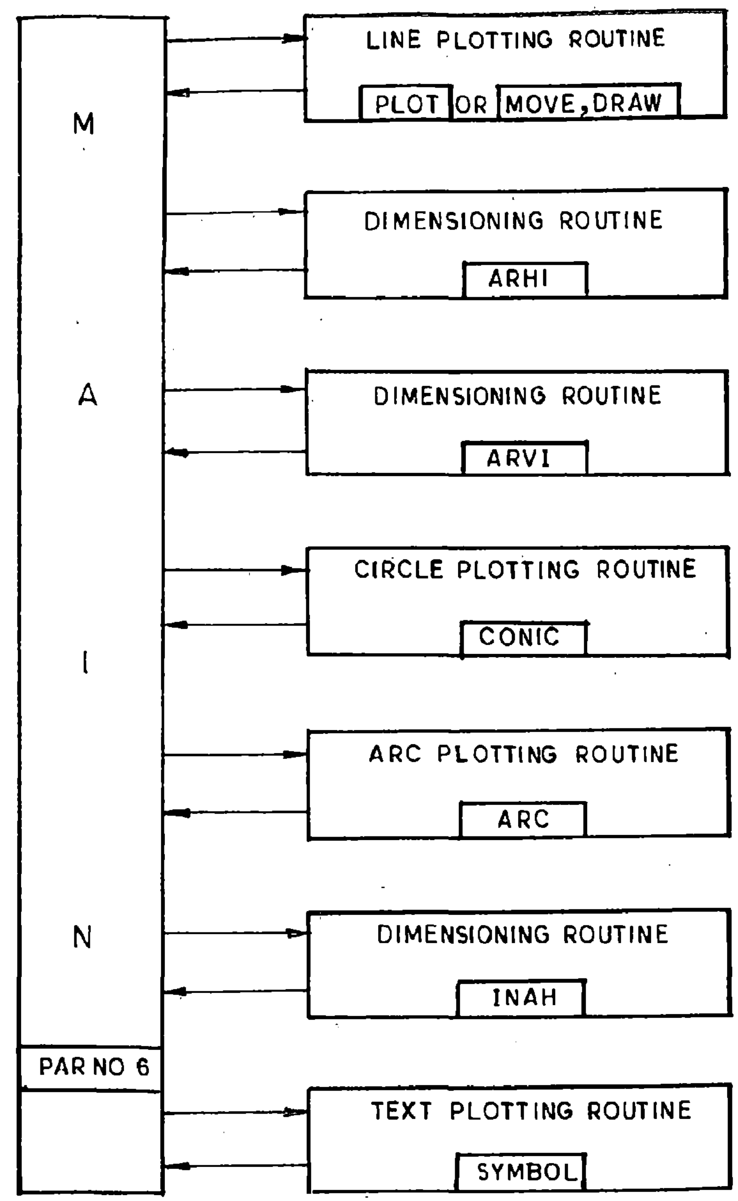
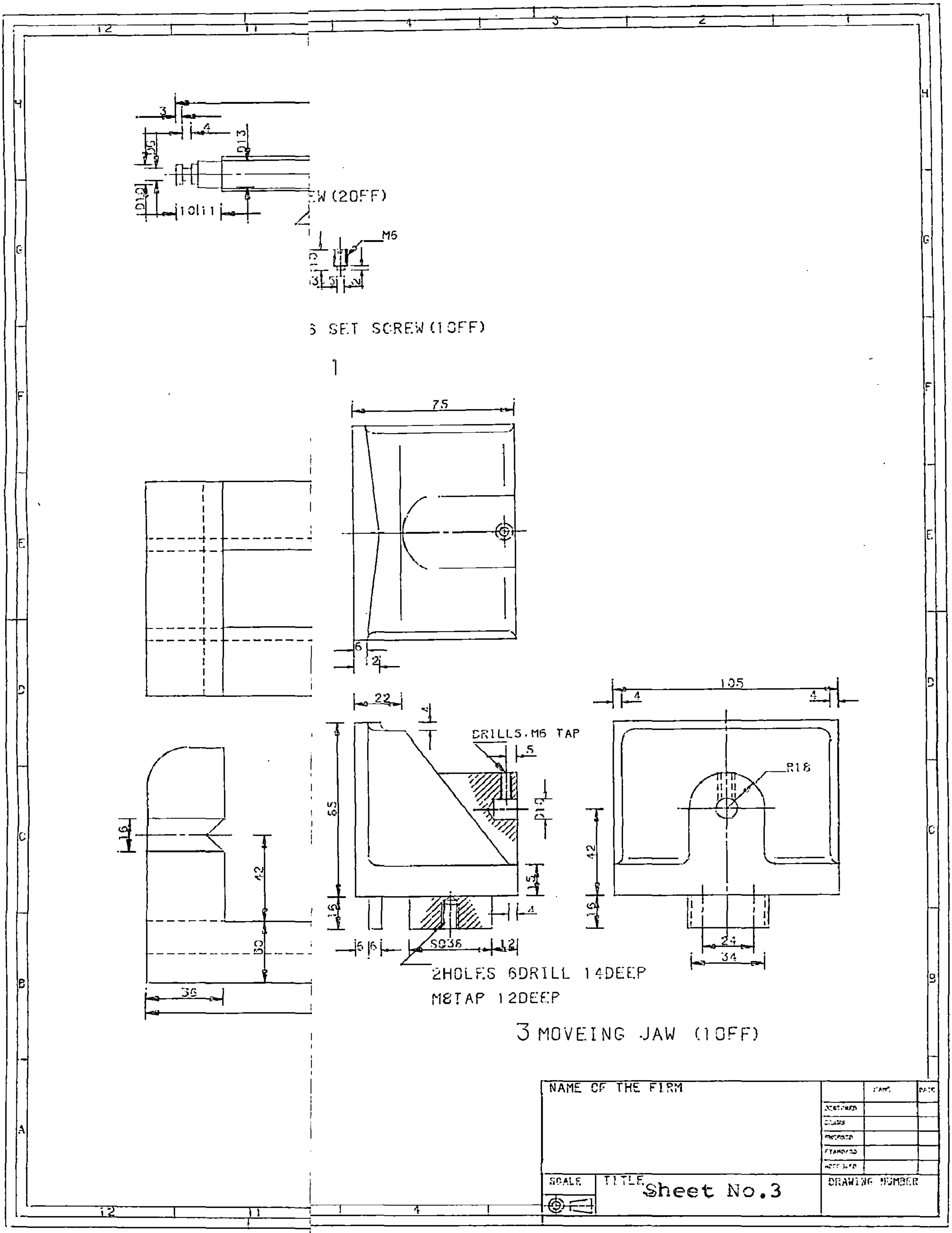


FIG.4.14. FLOW CHART TO PLOT ORTHOGRAPHIC VIEWS OF FIXED JAW (1 OFF)


EW (2OFF)
M6
6 SET SCREW (1OFF)
DRILL5.M6 TAP
R18
2HOLES 6DRILL 14DEEP
M8TAP 12DEEP
3 MOVEING JAW (1OFF)
NAME OF THE FIRM
SCALE
TITLE Sheet No.3
DRAWING NUMBER

4.7 PROGRAM SNO3C AND SNO3T

This program is capable to plot the details drawing of screw jack. It contains body and half plan of body, screw, cup, washer, set screw and tommay bar.

Fig. 4.15 shows the flow chart of details drawing of screw jack, Fig. 4.16 shows the flow chart of body and half plan of body (PARNO 1), Fig. 4.17 shows the flow chart of screw, (PARNO 3), Fig. 4.18 shows the flow chart of cup (PANO 4), Fig. 4.19 shows the flow chart of washer (PANO 5), Fig. 4.20 shows the flow chart of set screw (PARNO 6), Fig. 4.21 shows the flow chart of tommay bar(PARNO 7).

The program SNO3C is applicable to Calcomp plotter and program SNO3T is applicable to Tektronix graphic terminal. The output of program is shown in sheet 4.4.

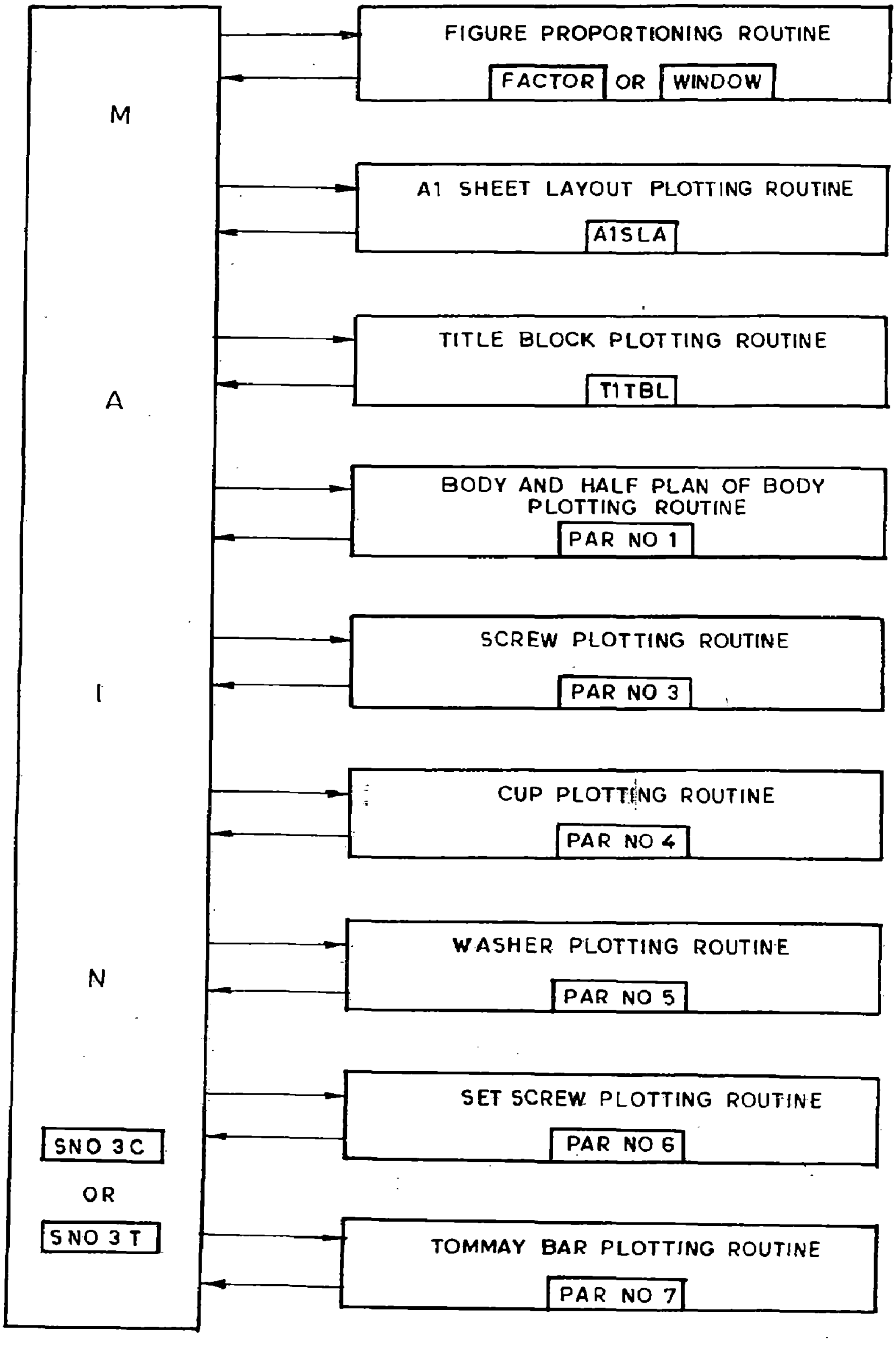


FIG. 4.15. FLOW CHART TO PLOT DETAILS DRAWING OF SCREW JACK ON A1 SHEET LAYOUT

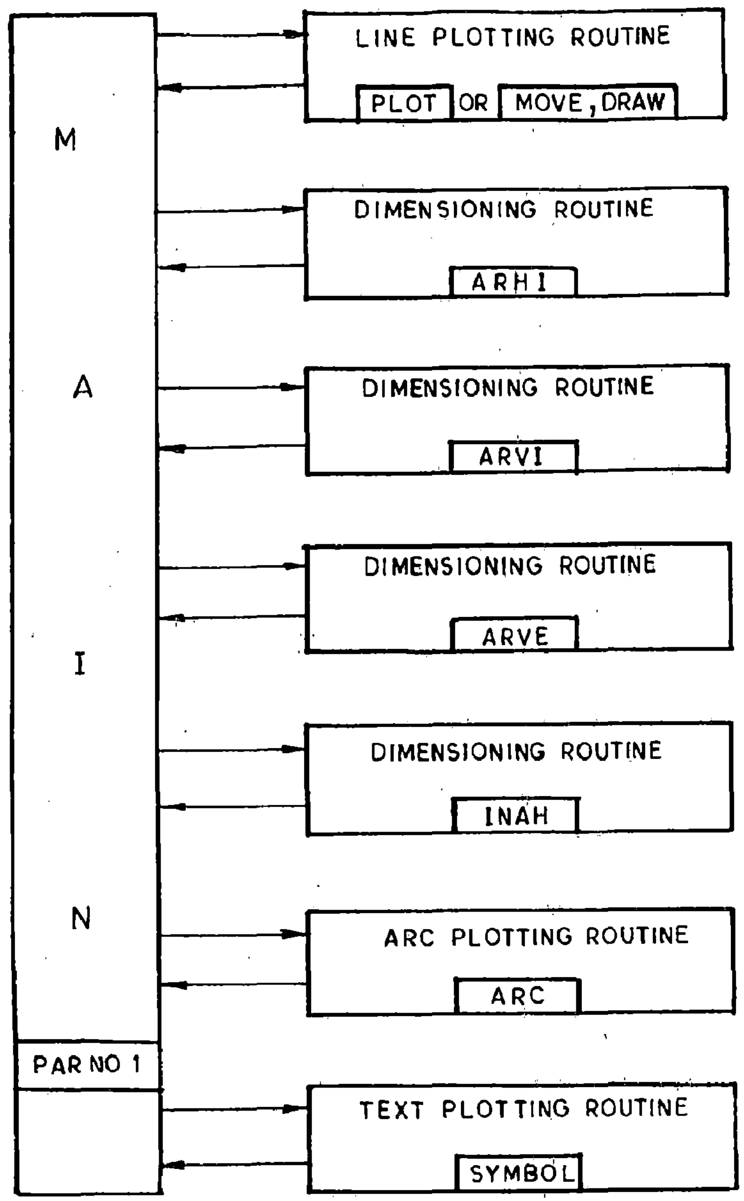


FIG.4.16. FLOW CHART TO PLOT BODY AND HALF PLAN OF BODY

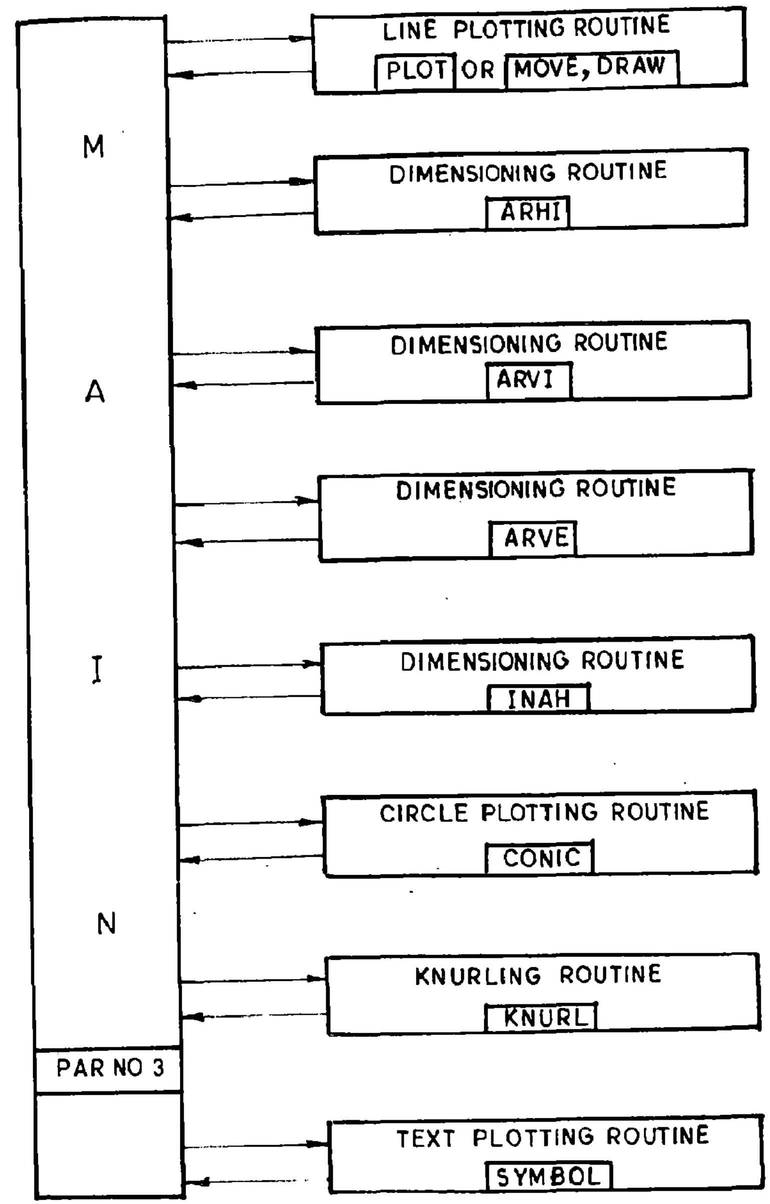


FIG.4.17. FLOW CHART TO PLOT SCREW

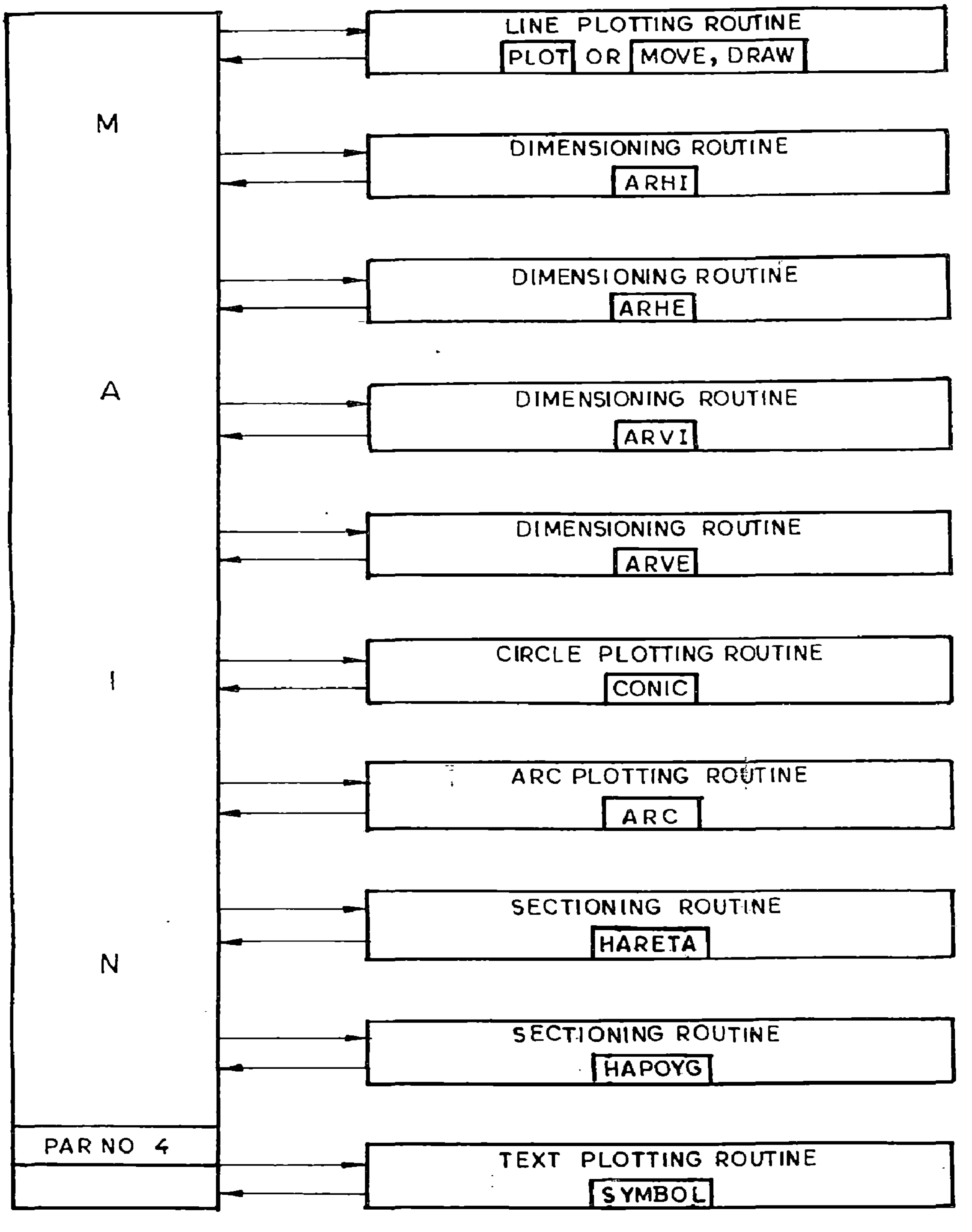


FIG.4.18. FLOW CHART TO PLOT CUP

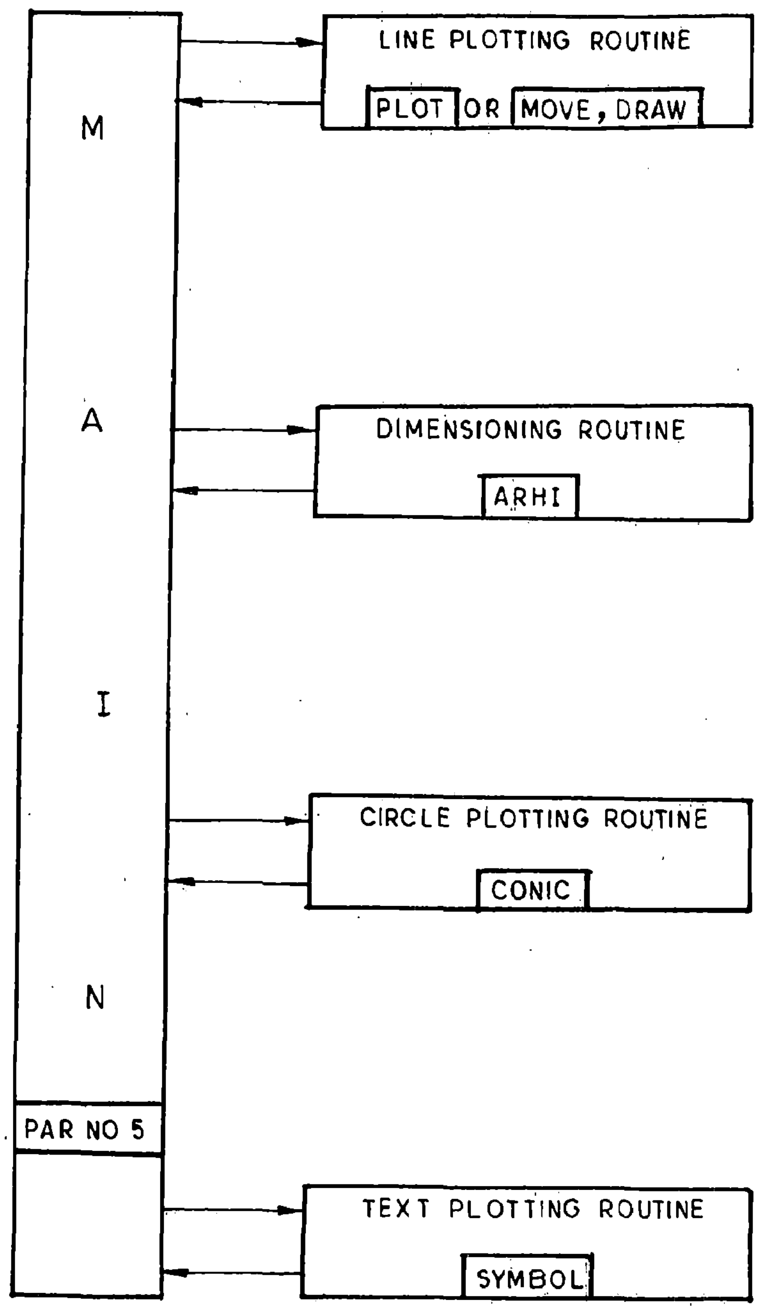


FIG.4.19. FLOW CHART TO PLOT WASHER

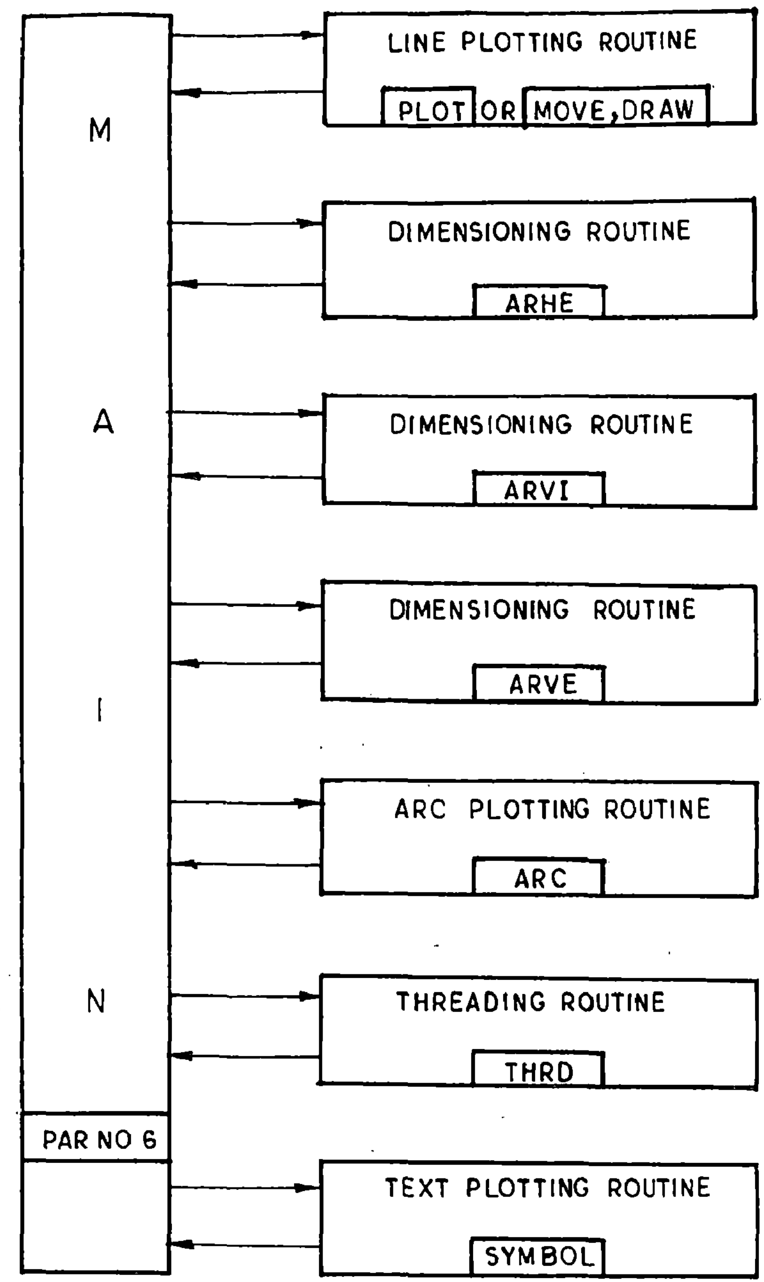


FIG.4.20. FLOW CHART TO PLOT SET SCREW

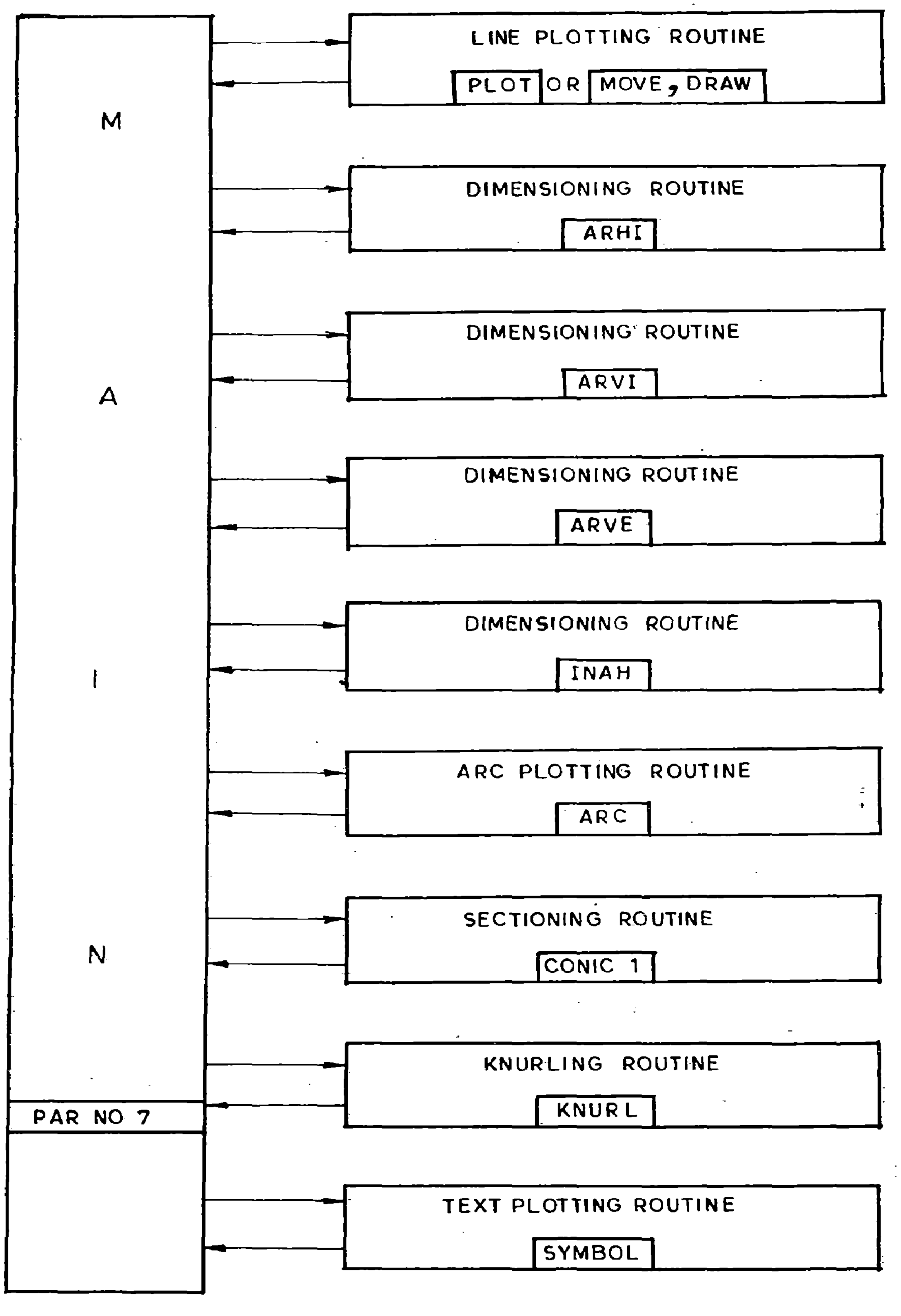
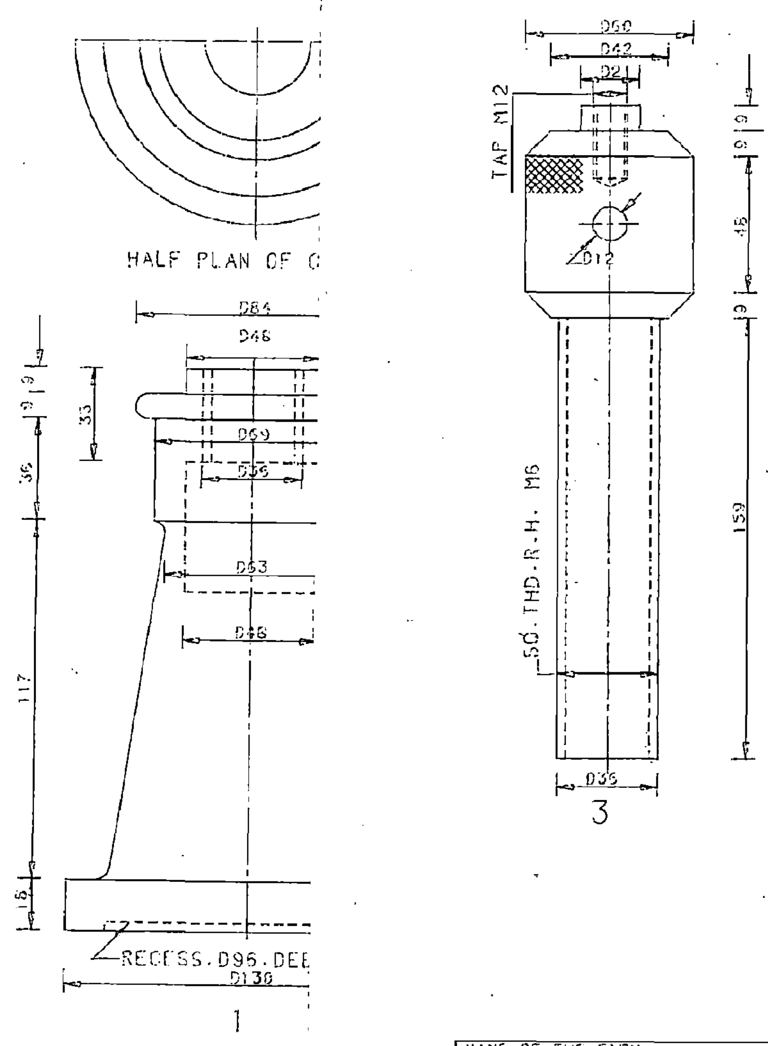


FIG. 4.21. FLOW CHART TO PLOT TOMMAY BAR



NAME OF THE FIRM

SCALE | TITLE Sheet No.4 | DRAWING NUMBER

CHAPTER - V

THREE-DIMENSIONAL DISPLAY

5.1 GENERAL

The designer constructs a geometric model on the terminal screen to describe the shape of a object to the computer. The computer then converts this pictorial representation into a mathematical model which it stores in a data base for later use. The model may be recalled and refined by the engineer at any point in the design process. And it may be used as an input for virtually all other CAD/CAM functions.

Because so many functions depend on the model, geometric modelling is considered by many experts to be the most important feature of CAD/CAM For example, the geometric model may be used to create a finite element model for stress analysis. It may serve as an input for computer assisted drafting to produce engineering drawing, or it can be used as a basis for producing NC tapes for fabricating the part.

The aim of this chapter is to create parameterized shapes. The idea of parameterized shapes is to supply the user with a procedure for generating frequently used classes of shapes. A particular shape within the class can be specified by a few parameters [2] from which the procedure computes a larger, sufficient set of shape information. An example might be a rectangular box (brick), parameterized by width, hight and length. Other parameterized shapes may be supplied by the system to provide a set of primitives from which many other shapes can be constructed. The developed software offers a range of standard parameterized shapes i.e. cube, rectangular box, cylindrical and conical geometries.

The another aim of this chapter is describe a finite element grid generation, using the assembly of brick module of regular rectangular box body. The user calls the geometric model of the part and create a finite element model quickly and easily using automatic node - generation and element generation routines.

A brief description of each program is given below.

5.2 PROGRAM 3DIMN

Basic Feature : To draw any number of cube, or rectangular block (Fig. 5.1) and create computer-oriented mesh generation which can serve as preprocessors to finite element method.

THREE-DIMENSIONAL DISPLAY MODULES

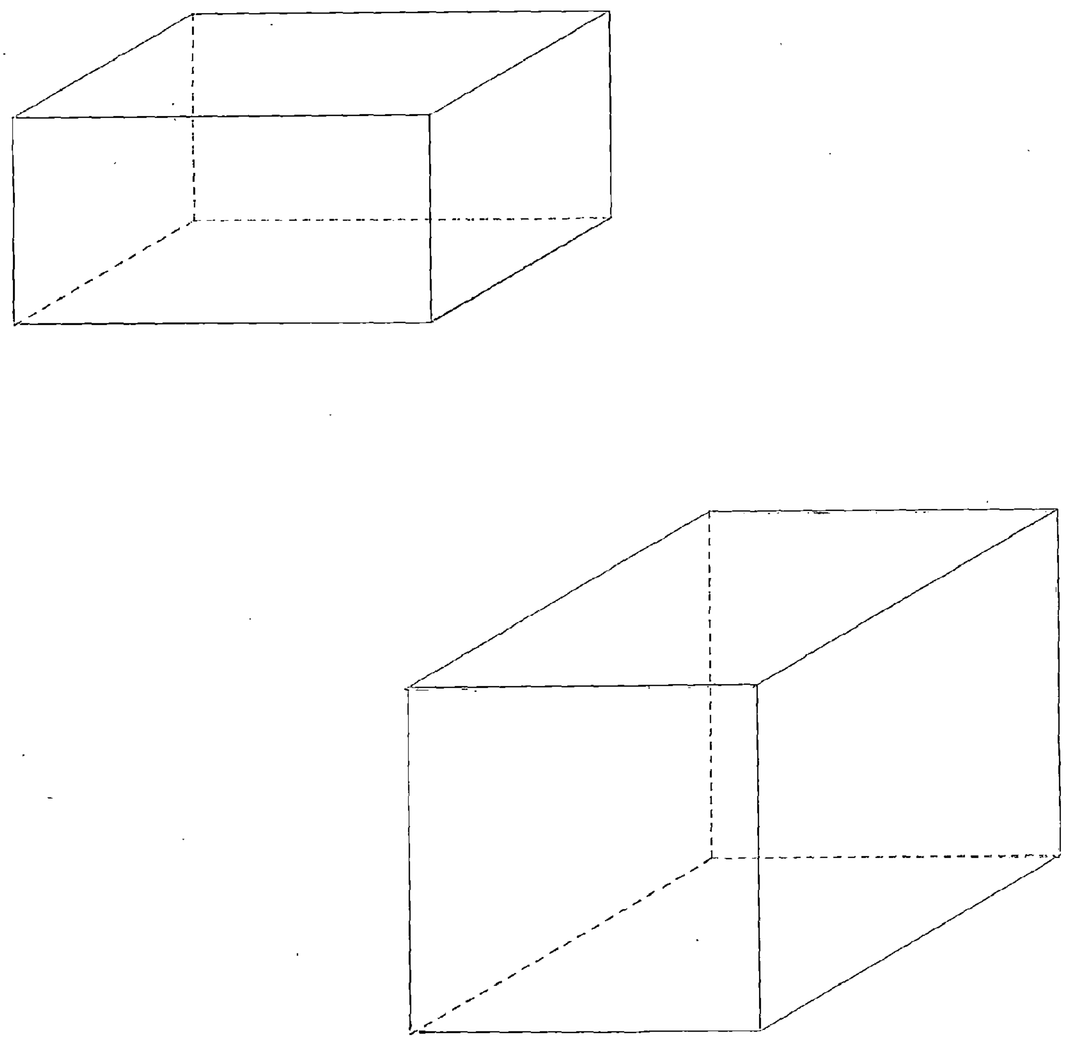


FIGURE NO 5.1 CUBIC AND RECTANGULAR BLOCK MODULES

Using the present program of computer graphics the coordinates and connectivities of nodes and elements for stiffness analysis can be prepared. The program is capable of generating two and three dimensional mesh and also plotting for various plotting devices.

The facility for irregular rectangular block pattern is also incorporated in the program i.e. the length, width and height need not continue in the same manner for all the elements. The interior elements of the rectangular object can also be eliminated in the plot (Fig. 5.2.A) or can be shown by dotted line (Fig. 5.2.B).

The given rectangular object is discretized as the assembly of number of rectangular block in such a manner that the geometric configuration of each of the rectangular block is identical in all the three direction, then for each rectangular block, mesh is generated to give node and element assembly of each block and hence to total rectangular object. A few subroutines supplement the main program in order to achieve 3D and 2D plotting of rectangular object. A brief description of the developed program is given below.

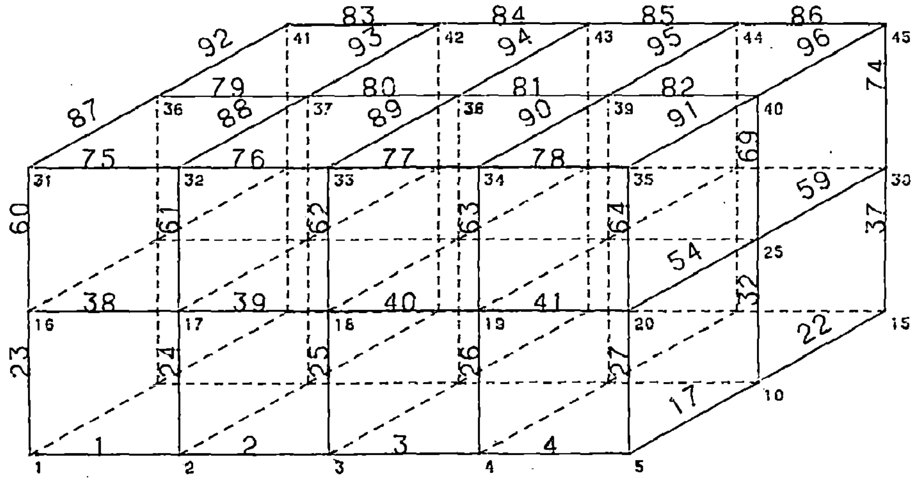


FIG 5.2.A 3D REPRESENTATION OF BRICK ELEMENT WITH INTERIOR ELEMENTS

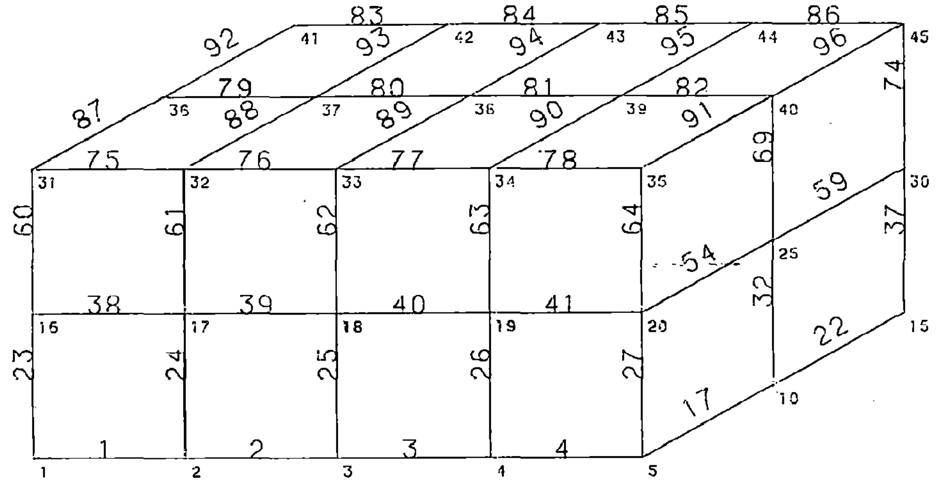


FIG 5.2.B 3D REPRESENTATION OF BRICK ELEMENT WITHOUT INTERIOR ELEMENTS

5.2.1 Main Program

Main program reads about the number of independent problems to be plotted and the data defining the geometry of the rectangular blocks i.e. the length, width and height and also number of rectangular blocks in x, y and z direction. This also reads all the control cards and calls for the desired subroutines.

Data : Number of independent problems, NPRB

: Number of rectangular block in x,y and z direction, NBX, NBY, NBZ

: If all the rectangular block in x,y and z direction are equal in length, width & height then IBX = 0 , IBY = 0 and IBZ = 0 otherwise if not so then IBX $\neq$ 0, IBY $\neq$ 0 and IBZ $\neq$ 0

: Nodal number of the initial node NP1

: Member number of the first element, NE1

: NCORD = 2, for 2-dimensional problem

= 3, for 3-dimensional problem

: X,Y,Z coordinates of initial node, COX, COY, COZ

: If IBX $\neq$ 0, length of rectangular blocks in X direction, BX(I) (I=1,NBX)

: If IBX=0 length of rectangular block in X direction, BX(1)

: If IBY ≠ 0, width of rectangular blocks in Y direction, BY(I), (I=1,NBY)

: If IBY=0 width of rectangular block in Y direction, BY(I)

: If IBZ ≠ 0 , height of rectangular blocks in Z direction, BZ(I), (I=1,NBZ)

: If IBZ=0 height of rectangular block in Z direction, BZ(1)

: Angle at which isometric view to be plotted, ANG (Fig. 5.3)

: If NDEV=1, program to be executed on graphic terminal and if NDEV=2, program to be executed on Calcomp plotter.

: Factor by which the whole figure to be enlarged or reduced, FACT

: If NOD = 0, neither element nor node numbers to be plotted.

= 1, only node numbers to be plotted

= 2, only element numbers to be plotted

= 3, both node and element numbers to be plotted.

: Angle at which whole the isometric view to be rotated, AXIS

: If interior elements to be plotted then, INTRIR $\neq$ 0

: If text string to be plotted then, NTEXT $\neq$ 0

5.2.2 Subroutine TRANS

The routine is called if the counter NCORD= 3. It does not need any extra input data, and provides the facility of isometric plotting. In isometric plotting coordinates are transformed according to the direction of viewing. To plot an isometric view of 3-D figure the special coordinates (X,Y,Z) must be transformed to the two dimensional coordinates (X,Y) such that when plotted an isometric view emerges.

5.2.3 Subroutine DDRAW

In the routine an algorithm, to generate the elements which will be visible is described for 3-D case,on the other hand in 2-D case all the elements created in a DOLOOP will be visible and hence plotted.

5.2.4 Subroutine DRRAW

The subroutine is called after each element is generated in the later routine (DDRAW). This is the main subroutine which carries out the process of plotting. For each element, both the connectivities of which have already been obtained, plotting function starts with the instruction of moving to the first connectivity of the same and then drawing the line to the other connectivity. The process is repeated for all the elements.

5.2.5 Subroutine DASH

The subroutine is called in the main program if the counter INTRIR is specified any non-zero value. It provides the facility of plotting the interior elements by hidden lines and thus giving a true and total picture of object.

For interior line plotting of the elements, the concept involved is as follows, for each element depending upon the coordinates of both connectivities, certain number of points are obtained and the line is drawn by considering alternate points.

5.2.6 Subroutine NUMB

The subroutine is used when counter NOD $\neq 0$, it calls the inbuilt subroutine NUMBER from Calcomp library for plotting the numerics.

The routine provides the flexibilities for-

(I) any node number to be plotted

(II) any element number to be plotted

(III) both element and nodes to be plotted.

Fig. 5.4 shows the complete flow chart of the program.

5.3 PROGRAM CYLINC AND CYLINT

Basic Feature : To draw any number of cylindrical and conical geometries (Fig. 5.5).

Data : Number of cylindr or conic to be drawn, NCYL

: Constants of conics section equation, A(I), B(I)

: Vectors of conic section, C(I)

: Coordinates of center of conics section, XC(I), YC(I)

The program CYLINC is applicable to Calcomp plotter and program CYLINT to Tektronix graphic terminal. The flow chart of program is given in Fig. 5.6.

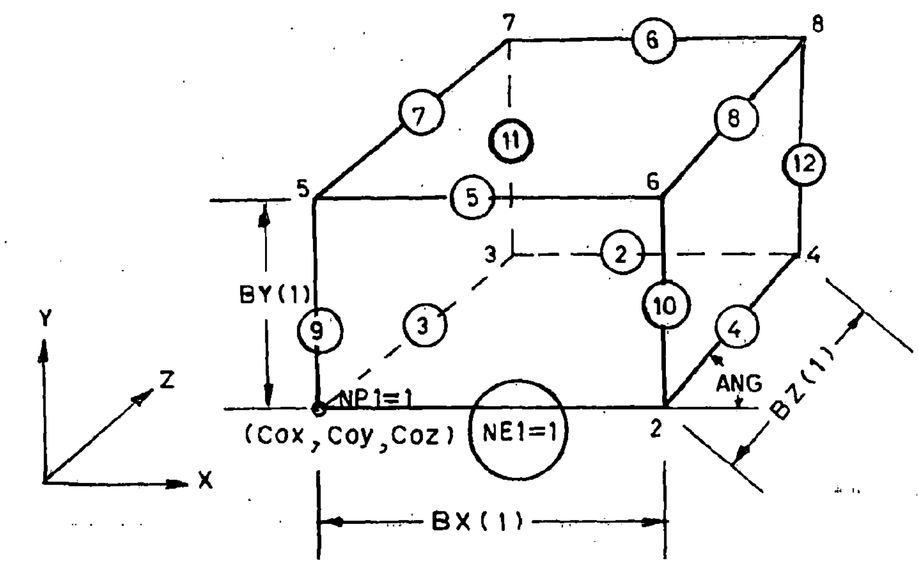


FIG.5.3

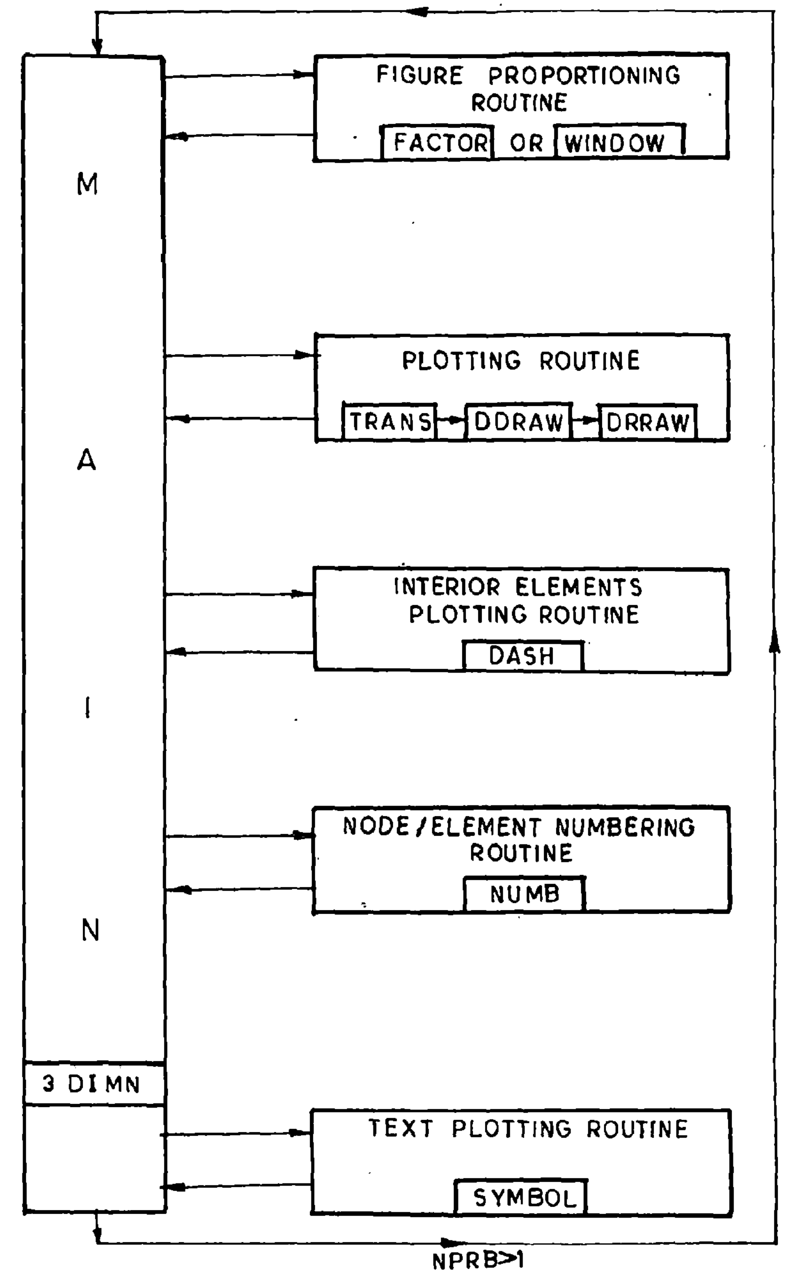


FIG.5.4. FLOW CHART FOR PROGRAM 3 DIMN

THREE-DIMENSIONAL DISPLAY MODULES

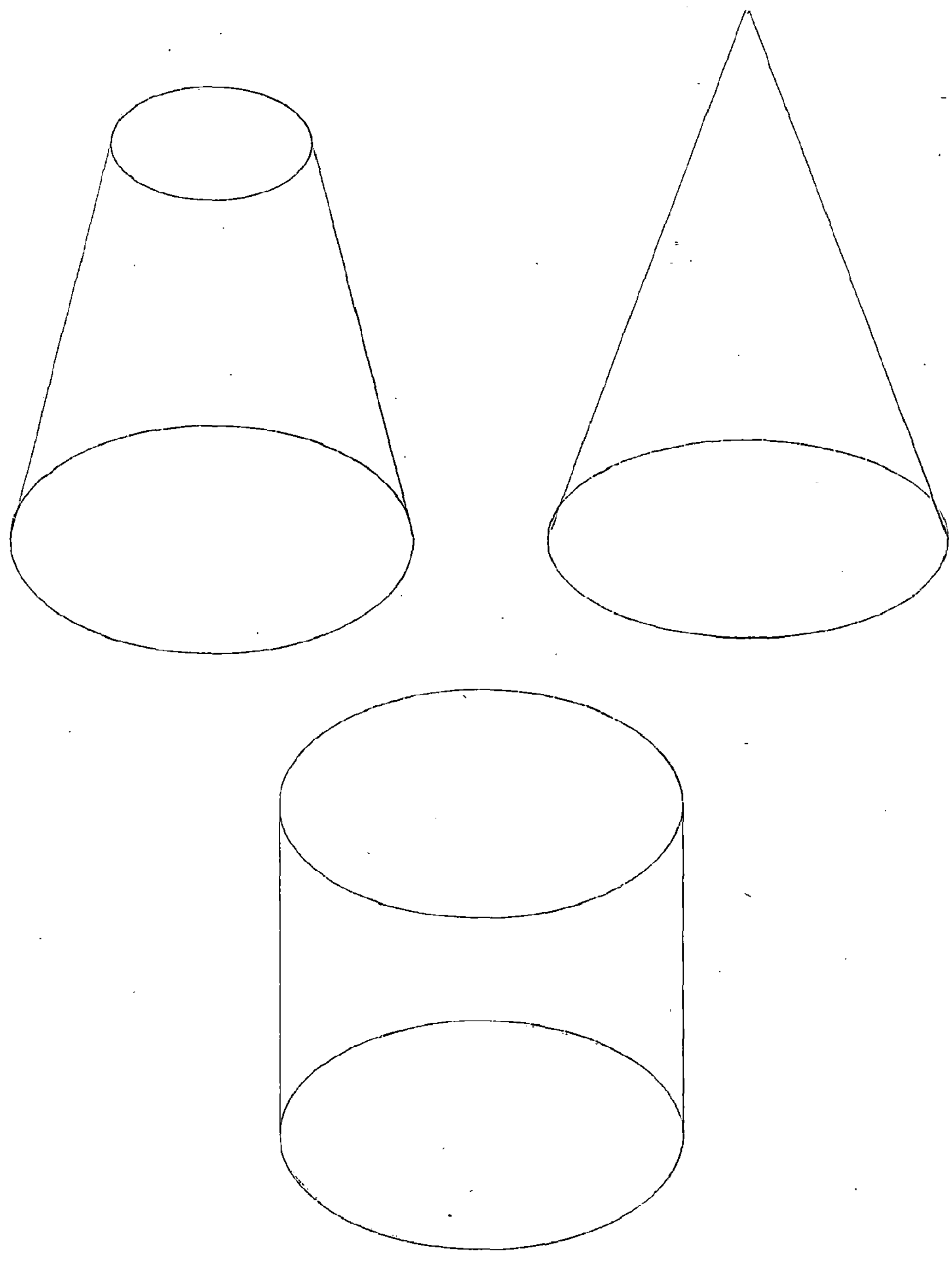
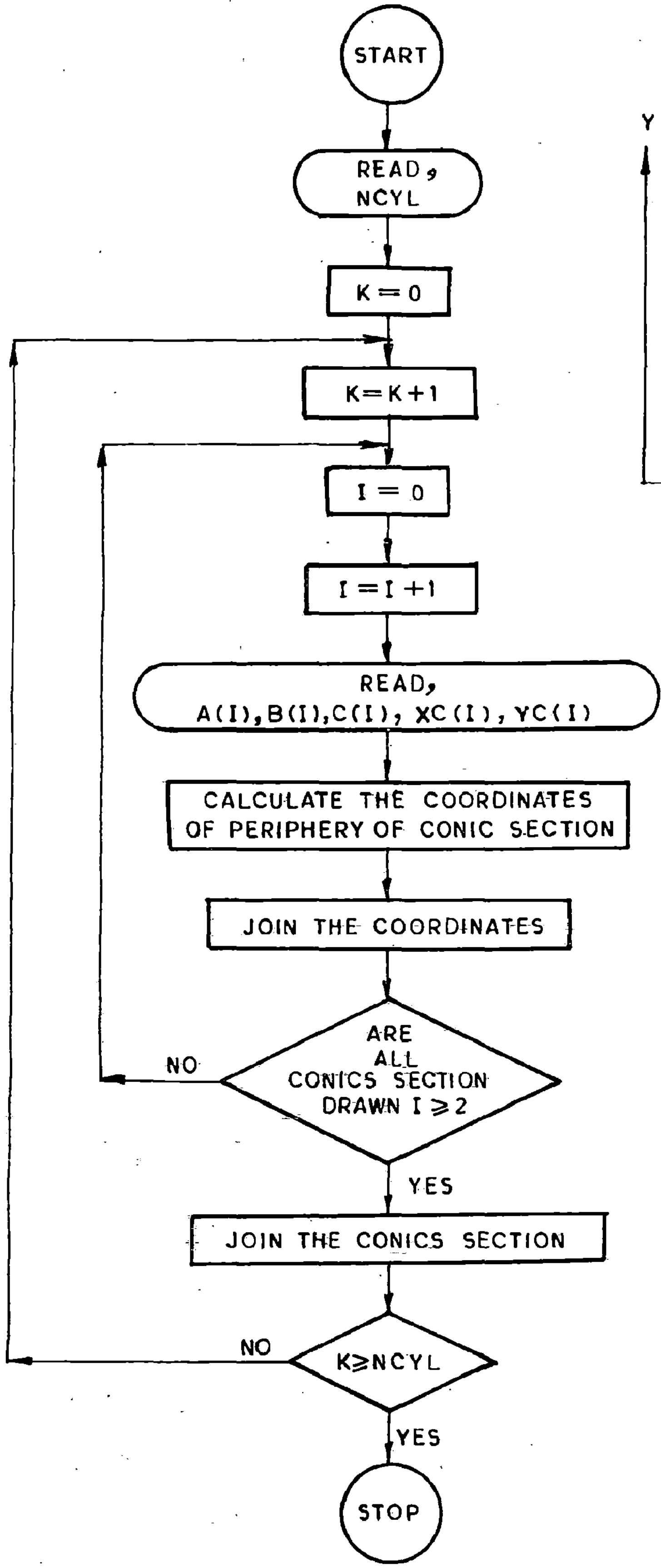


FIGURE NO5.5 CYLINDRICAL AND CONICAL MODULES



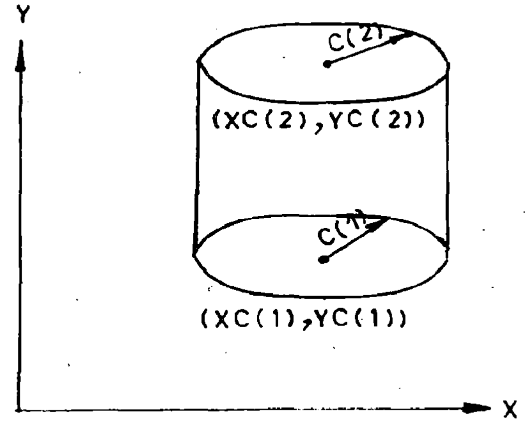


FIG.5.6. FLOW CHART TO PLOT CYLINDRICAL AND CONICAL GEOMETRIES

As a starting point, to utilize the developed primitives for basic solids i.e. rectangular block and cylinders, can operate for combinatorial operators such as addition and difference to build complex objects from duplicated and transformed primitives. Therefore with suitable rigorous definition of the meaning of the primitives and the exact action of the combinatorial operators a valid 3D solid is drawn (Fig. 5.7).

5.4 CONCLUSIOS AND POSSIBLE EXTENSIONS

The application of the method to some practical cases has shown that 2-dimensional drafting can be achieved at high speed, satisfying requirements of an interactive CAD environemnt. However, in the light of the limited system capabilities and other practical constraint, the present work is an initiation in the direction of proper documentation and reproduction of engineering drawings use in the computer graphics.

An obvious shortcoming so far is that the full semantics of the 2D representation is not entirely recovered. A possible extension of the method would be to develope more modules, so that a broader

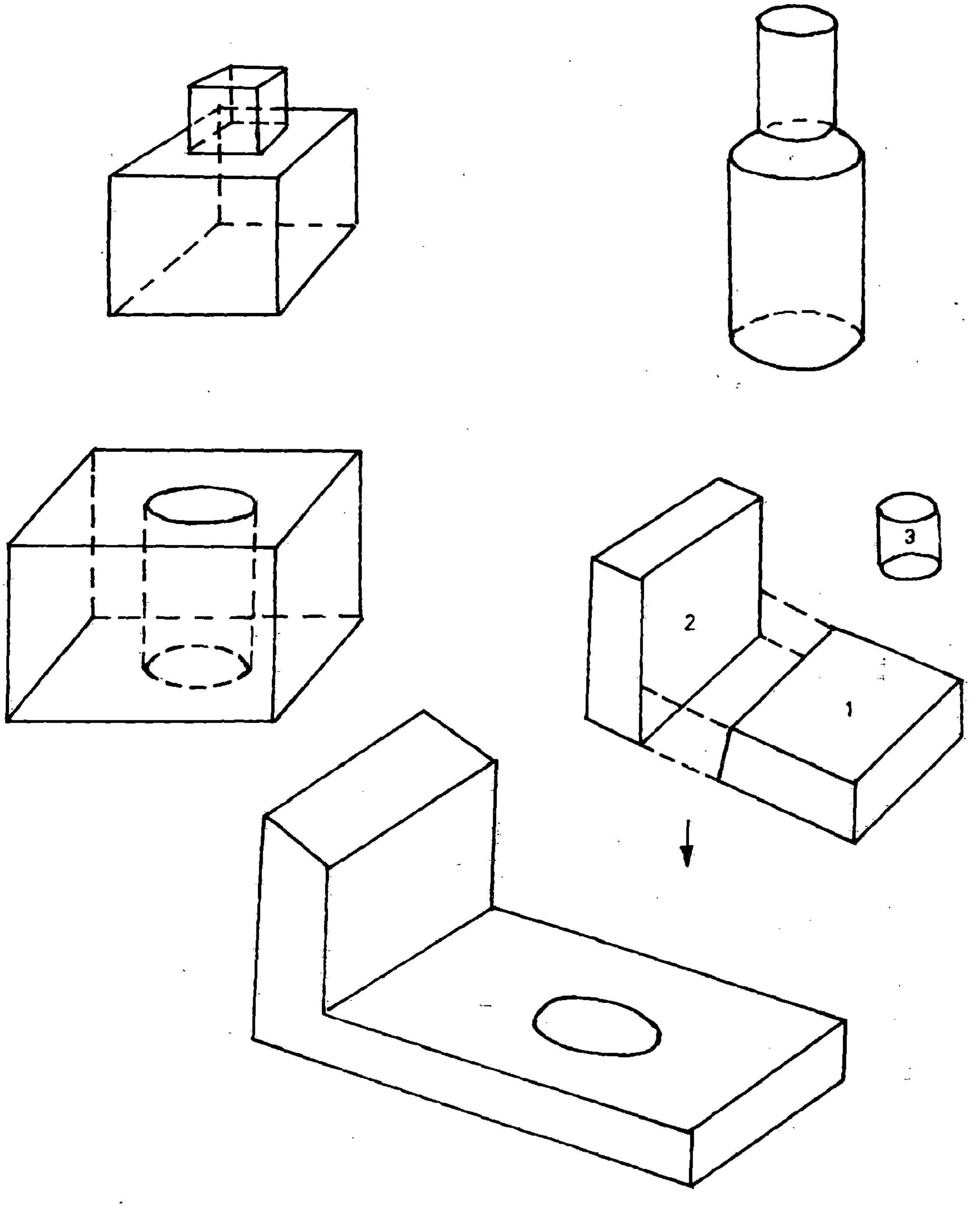


FIG.5.7

range of geometries can be covered. Further, in general, any class whose 2D pattern can be described by some kind of formal rules, its geometries can be described using only a small number of lowest rank modules, the philosophy of having a special module for each class remains reasonable for a wide range of practical applications.

Another possible extension of the work would be on automatic recognition of 3D structures from 2D representations. It follows two step procedure, input of a graphic - oriented 2 D representation corresponding to views and cuts, and interpretation of this data to give the explicit 3D description.

Lastly extension of the work would be to develope more solid modules. Notably the fact that with suitable rigorous definitions of the meaning of solid modules and the exact action of the combinatorial operators, a valid 3D solid can be drawn.

REFERENCES

1. Aldefed, B., 'Computer-aided dėsign', Phillips Research Laboratory, Vol 15, No. 2 (1983), pp 59-64.

2. Bear, B., Eastman, C., Hension, M., 'Computer-aided design', Institute of Physical Planning, Vol 11, No. 5 (1979), pp 253-272.

3. Bron, G.P., Chetaikin, A.V., 'Soviet Engineering Research', Stanki I Instrument, Vol 2, No. 7 (1982), pp 63-64.

4. Bron, G.P., 'Soviet Engineering Research' Stanki I Instrument, Vol 4, No. 5 (1984), pp 47-48.

5. Grayer, A.R., 'Computer-aided design', Shape Data Limited, Vol 12, No. 4 (1980), pp 189-192.

6. Marshall, P., 'Computer-aided Engineering', Automation and Control Department, October (1985), pp 167-172.

7. Reach, P.W., 'Computer-aided Engineering', Norsk Data Ltd., February (1986), pp 9-16.

8. Straayer, D.H., 'Computer and Graphics', System Engineering Department, Vol 9, No. 1 (1985), pp 27-32.

9. Winkler, T., 'Computer-aided design', Imstytut Mechaniki I Podstan Konstrukeji Maszyn, Vol 15, No. 1 (1983), pp 15-18.

10. Krouse, Jhn K., What every engineer should know about computer-aided design and computer-aided manufacturing, Marcel Dekker, Inc., New York and Basel 1982.

11. Newman, W., and Sproull, R., Principles of Interactive Computer Graphics, McGraw Hill, International Book Co., Tokyo, 1982.

12. Plot 10, 4010C01, Interactive Graphics Library, User's Manual, Tektronix, Inc., 1978.

13. Programming Calcomp Electromechanical Plotters, CALCOMP, California, 1977.

14. Zienkiewicz, O.C., The Finite Element Methods, 3rd Edn., McGraw Hill, New York (1977).

APPENDIX - I

I.1 Data Sequencing for Supporting Arm (Body) (PARNO 1)

Input data for routine PLOT OR (MOVE, DRAW)

where NLI = Number of lines

X1(I), Y1(I) = initial coordinates of lines

X2(I), Y2(I) = final coordinates of lines

NLI

11

X1(I)	Y1(I)	X2(I)	Y2(I)
40.	119.	40.	150.
40.	150.	80.	150.
80.	150.	80.	119.
40.	125.	80.	125.
170.	125.	170.	46.
170.	46.	178.	46.
178.	46.	178.	116.
181.	119.	234.	119.
237.	116.	237.	46.
237.	46.	245.	46.
245.	46.	245.	125.

Input Data for routine ARHI (Refer Fig. 3.9)

NARO	X	
XO(I)	YO(I)	XL(I)
40.	160.	40.
80.	160.	20.
170.	25.	76.
208.	160.	30.

Input Data for routine ARHE (Refer Fig. 3.9)

NARO	X	
2.	2.	
XO(I)	YO(I)	XL(I)
152.	35.	8.
220.	35.	8.

Input Data for rutine CONIC (Refer Fig. 3.5)

NARC

2.

A(I)	B(I)	C(I)	XC(I)	YC(I)
1.	1.	56.25	100.	67.
1.	1.	110.25	208.	134.

Input Data for routine NEWTR (Refer Fig. 3.7)

N	NC
2.	5.

A1(I)	B1(I)	XC1(I)	YC1(I)	C1(I)
1.	1.	246.	150.	25.
1.	1.	208.	134.	16.
A2(I)	B2(I)	XC2(I)	YC2(I)	C2(I)
1.	1.	170.	150.	25.
1.	1.	208.	134.	16.

X(I)	Y(I)
220.	135.
190.	135.

Input data for routine ARC (Refer Fig. 3.6)

NARC

8

A(I)	B(I)	C(I)	XC(I)	YC(I)	XOD(I)	XOD2(I)
1.	1.	5625.	115.	119.	40.	115.
1.	1.	1225.	115.	119.	80.	115.
1.	1.	400.	100.	67.	100.	100.
1.	1.	9.	235.	116.	238.	235.
1.	1.	9.	181.	116.	181.	178.
1.	1.	484.	170.	150.	170.	*
1.	1.	484.	246.	150.	*	246.
1.	1.	256.	208.	134.	*	*

*This values will be find out from subroutine NEWTR.

Input data for routine INAF (Refer Fig. 3.11)

NARO	X		
XO(I)	YO(I)	XL(I)	ALFA(I)
95.	61.	15.	45.
127.	201.	21.	45.

Input data for routine INAH (Refer Fig. 3.11)

NARO	X		
4.	2.		
XO(I)	YO(I)	XL(I)	ALFA(I)
62.	65.	75.	45.
82.	106.	35.	20.
186.	131.	25.	135.
114.	52.	20.	135.

Input data for routine SYMBOL (Refer subroutine SYMBOL)

```
NT
17
TITLE
40
XT,YT,SIZE,ANG,NC
55.,161.,4.,0.0,2
20
87.,161.,4.,0.0,2
76
205.,26.,4.,0.0,2
30
222.,161.,4.,0.0,2
8
172.,35.,4.,0.0,1
8
240.,35.,4.,0.0,1
15D
63.,41.,4.,0.0,3
20R
136.,36.,4.,0.0,3
22R
176.,150.,4.,0.0,3
21D
231.,150.,4.,0.0,3
```

contd...

I.2 Data Sequencing for wheel (PARNO 4)

Input data for routine PLOT OR (MOVE, DRAW)

NLI

22

X1(I)	Y1(I)	X2(I)	Y2(I)
242.5	145.	242.5	115.5
242.5	140.	227.5	140.
227.5	140.	227.5	120.
227.5	120.	245.	115.
245.	115.	245.	85.
245.	85.	227.5	80.
227.5	80.	227.5	60.
227.5	60.	242.5	60.
242.5	84.5	242.5	55.
242.5	55.	192.5	55.
192.5	55.	192.5	84.5
192.5	60.	207.5	60.
207.5	60.	207.5	80.
207.5	80.	190.	85.
190.	85.	190.	115.
190.	92.5	245.	92.5
190.	107.5	245.	107.5

contd...

192.5	115.5	192.5	145.
190.	115.	207.5	120.
207.5	120.	207.5	140.
207.5	140.	192.5	140.
192.5	145.	242.5	145.

Input data for routine ARHI (Refer Fig. 3.9)

NARO	X	
4.	2.	
XO(I)	YO(I)	XL(I)
192.5	45.	15.
207.5	45.	20.
192.5	35.	50.
190.	25.	55.

Input data for routine INAH (Refer Fig. 3.11)

NARO	X		
2.	2.		
XO(I)	YO(I)	XL(I)	ALFA(I)
65.	43.	15.	0.0.
65.	152.	15.	0..0

Input data for routine CONIC (Refer Fig. 3.5)

NARC

5.

A(I)	B(I)	C(I)	XC(I)	YC(I)
1.	1.	56.25	80.	100.
1.	1.	125.	80.	100.
1.	1.	400.	80.	100.
1.	1.	1600.	80.	100.
1.	1.	2025.	80.	100.

Input data for routine HAPOY (Refer Fig. 3.15)

M	NL		
1.	12.		
XA(I)	XF(I)	YA(I)	YF(I)
245.	227.5	115.	120.
227.5	227.5	120.	115.
227.5	245.	115.	115.
207.5	207.5	115.	120.
207.5	190.	120.	115.
190.	207.5	115.	115.
207.5	190.	85.	85.
190.	207.5	85.	80.
207.5	207.5	80.	85.
245.	227.5	85.	85.
227.5	227.5	85.	85.
227.5	245.	80.	85.

Input data for routine SYMBOL (Refer subroutine SMBOL)

```
NT
14
TITLE
15
XT, YT, SIZE, ANG, NC
197., 46.,4., 0., 2
20
215., 46., 4., 0.,2
50
215., 36., 4., 0.0, 2
55
215., 26., 4., 0., 2
30D
179., 98., 4., 90., 3
40D
169., 98., 4., 90., 3
80D
159., 98., 4., 90., 3
90D
149., 98., 4., 90., 3
15D
23., 62., 4., 0.0, 3
```

contd...