
UNIT 4 FUNDAMENTALS OF PART PROGRAMMING

Structure

- 4.1 Introduction
 - Objectives
- 4.2 Types of Part Programming
 - 4.2.1 Manual Part Programming
 - 4.2.2 Computer Aided Part Programming
- 4.3 Fundamental Elements for Developing Manual Programming
 - 4.3.1 Type of Dimensioning System
 - 4.3.2 Axes Designation
 - 4.3.3 NC Words
 - 4.3.4 Standard G and M Codes
 - 4.3.5 Tape Programming Format
 - 4.3.6 Machine Tool Zero Point Setting
 - 4.3.7 Coordinate Word
 - 4.3.8 Parameter for Circular Interpolation
 - 4.3.9 Spindle Function
 - 4.3.10 Feed Function
 - 4.3.11 Tool Function
 - 4.3.12 Work Settings and Offsets
 - 4.3.13 Rapid Positioning
 - 4.3.14 Linear Interpolation
 - 4.3.15 Circular Interpolation
 - 4.3.16 Circular Interpolation
- 4.4 Symbols Used
- 4.5 Part Program for Lathe Operation
- 4.6 Part Program for Machining Centres (Milling)
- 4.7 Fixed Cycle/Canned Cycle
- 4.8 Do-Loops
- 4.9 Subroutine
- 4.10 Summary
- 4.11 Answers to SAQs

4.1 INTRODUCTION

Types of part programming, Computer aided part programming, Part programming manual, Part programme using sub routines, do loops and fixed cycles are described in this Unit.

Objectives

After studying this unit, you should be able to understand

- ,
- ,
- , and
- .

4.2 TYPES OF PART PROGRAMMING

The part program is a sequence of instructions, which describe the work, which has to be done on a part, in the form required by a computer under the control of a numerical control computer program. It is the task of preparing a program sheet from a drawing sheet. All data is fed into the numerical control system using a standardized format. Programming is where all the machining data are compiled and where the data are translated into a language which can be understood by the control system of the machine tool. The machining data is as follows :

- (a) Machining sequence classification of process, tool start up point, cutting depth, tool path, etc.
- (b) Cutting conditions, spindle speed, feed rate, coolant, etc.
- (c) Selection of cutting tools.

While preparing a part program, need to perform the following steps :

- (a) Determine the startup procedure, which includes the extraction of dimensional data from part drawings and data regarding surface quality requirements on the machined component.
- (b) Select the tool and determine the tool offset.
- (c) Set up the zero position for the workpiece.
- (d) Select the speed and rotation of the spindle.
- (e) Set up the tool motions according to the profile required.
- (f) Return the cutting tool to the reference point after completion of work.
- (g) End the program by stopping the spindle and coolant.

The part programming contains the list of coordinate values along the X, Y and Z directions of the entire tool path to finish the component. The program should also contain information, such as feed and speed. Each of the necessary instructions for a particular operation given in the part program is known as an NC word. A group of such NC words constitutes a complete NC instruction, known as block. The commonly used words are *N*, *G*, *F*, *S*, *T*, and *M*. The same is explained later on through examples.

Hence the methods of part programming can be of two types depending upon the two techniques as below :

- (a) Manual part programming, and
- (b) Computer aided part programming.

4.2.1 Manual Part Programming

The programmer first prepares the program manuscript in a standard format. Manuscripts are typed with a device known as flexo writer, which is also used to type the program instructions. After the program is typed, the punched tape is prepared on the flexo writer. Complex shaped components require tedious calculations. This type of programming is carried out for simple machining parts produced on point-to-point machine tool.

To be able to create a part program manually, need the following information :

- (a) Knowledge about various manufacturing processes and machines.
- (b) Sequence of operations to be performed for a given component.
- (c) Knowledge of the selection of cutting parameters.
- (d) Editing the part program according to the design changes.
- (e) Knowledge about the codes and functions used in part programs.

4.2.2 Computer Aided Part Programming

If the complex-shaped component requires calculations to produce the component are done by the programming software contained in the computer. The programmer communicates with this system through the system language, which is based on words. There are various programming languages developed in the recent past, such as APT (Automatically Programmed Tools), ADAPT, AUTOSPOT, COMPAT-II, 2CL, ROMANCE, SPLIT is used for writing a computer programme, which has English like statements. A translator known as compiler program is used to translate it in a form acceptable to MCU.

The programmer has to do only following things :

- (a) Define the work part geometry.
- (b) Defining the repetition work.
- (c) Specifying the operation sequence.

Over the past years, lot of effort is devoted to automate the part programme generation. With the development of the CAD (Computer Aided Design)/CAM (Computer Aided Manufacturing) system, interactive graphic system is integrated with the NC part programming. Graphic based software using menu driven technique improves the user friendliness. The part programmer can create the geometrical model in the CAM package or directly extract the geometrical model from the CAD/CAM database. Built in tool motion commands can assist the part programmer to calculate the tool paths automatically. The programmer can verify the tool paths through the graphic display using the animation function of the CAM system. It greatly enhances the speed and accuracy in tool path generation.

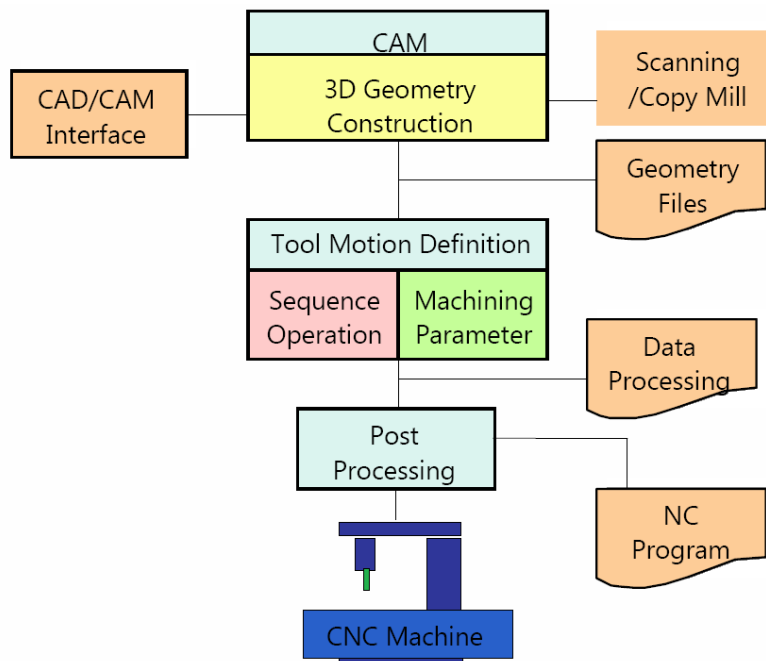


Figure 4.1 : Interactive Graphic System in Computer Aided Part Programming

4.3 FUNDAMENTAL ELEMENTS FOR DEVELOPING MANUAL PART PROGRAMME

The programmer to consider some fundamental elements before the actual programming steps of a part takes place. The elements to be considered are as follows :

4.3.1 Type of Dimensioning System

We determine what type of dimensioning system the machine uses, whether an absolute or incremental dimensional system which has been explained in Unit 2.

4.3.2 Axis Designation

The programmer also determines how many axes are available on machine tool. Whether machine tool has a continuous path and point-to-point control system that has been explained in Unit 2.

4.3.3 NC Words

The NC word is a unit of information, such as a dimension or feed rate and so on. A block is a collection of complete group of NC words representing a single NC instruction. An end of block symbol is used to separate the blocks. NC word is where all the machining data are compiled and where the data are translated in to a language, which can be understood, by the control system of the machine tool.

Block of Information

NC information is generally programmed in blocks of words. Each word conforms to the EIA standards and they are written on a horizontal line. If five complete words are not included in each block, the machine control unit (MCU) will not recognize the information; therefore the control unit will not be activated. It consists of a character N followed by a three digit number raising from 0 to 999.

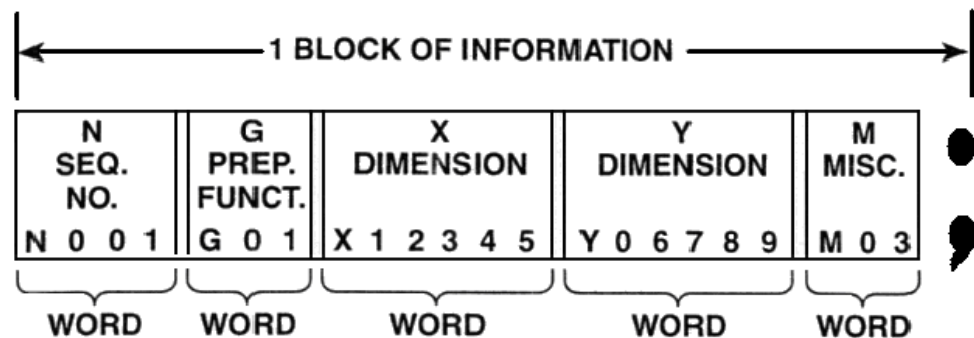


Figure 4.2 : A Block of Information

Using the example shown in Figure 4.2. The words are as follows :

N001 – represents the sequence number of the operation.

G01 – represents linear interpolation.

X12345 – will move the table in a positive direction along the X-axis.

Y06789 – will move the table along the Y-axis.

M03 – Spindle on CW and

; – End of block.

4.3.4 Standard G and M Codes

The most common codes used when programming NC machines tools are G-codes (preparatory functions), and M codes (miscellaneous functions). Other codes such as *F*, *S*, *D*, and *T* are used for machine functions such as feed, speed, cutter diameter offset, tool number, etc. G-codes are sometimes called cycle codes because they refer to some action occurring on the X, Y, and/or Z-axis of a machine tool. The G-codes are grouped into categories such as Group 01, containing codes G00, G01, G02, G03, which cause some movement of the machine table or head. Group 03 includes either absolute or incremental programming. A G00 code rapidly positions the cutting tool while it is above the workpiece from one point to another point on a job. During the rapid traverse movement, either the X or Y-axis can be moved individually or both axes can be moved at the same time. The rate of rapid travel varies from machine to machine.

The total numbers of these codes are 100, out of which some of important codes are given as under with their functions :

G-Codes (Preparatory Functions)

Code	Function
G00	Rapid positioning
G01	Linear interpolation
G02	Circular interpolation clockwise (CW)
G03	Circular interpolation counterclockwise (CCW)
G20	Inch input (in.)
G21	Metric input (mm)
G24	Radius programming
G28	Return to reference point
G29	Return from reference point
G32	Thread cutting
G40	Cutter compensation cancel
G41	Cutter compensation left
G42	Cutter compensation right
G43	Tool length compensation positive (+) direction
G44	Tool length compensation minus (-) direction
G49	Tool length compensation cancels
G 53	Zero offset or M/c reference
G54	Settable zero offset
G84	canned turn cycle
G90	Absolute programming
G91	Incremental programming

Note : On some machines and controls, some may be differ.

M-Codes (Miscellaneous Functions)

M or miscellaneous codes are used to either turn ON or OFF different functions, which control certain machine tool operations. M-codes are not grouped into categories, although several codes may control the same type of operations such as M03, M04, and M05, which control the machine tool spindle. Some of important codes are given as under with their function s:

Code	Function
M00	Program stop
M02	End of program
M03	Spindle start (forward CW)
M04	Spindle start (reverse CCW)
M05	Spindle stop
M06	Tool change
M08	Coolant on
M09	Coolant off

M10	Chuck - clamping
M11	Chuck - unclamping
M12	Tailstock spindle out
M13	Tailstock spindle in
M17	Tool post rotation normal
M18	Tool post rotation reverse
M30	End of tape and rewind or main program end
M98	Transfer to subprogram
M99	End of subprogram

Note : On some machines and controls, some may be differ.

4.3.5 Tape Programming Format

Both EIA and ISO use three types of formats for compiling of NC data into suitable blocks of information with slight difference.

Word Address Format

This type of tape format uses alphabets called address, identifying the function of numerical data followed. This format is used by most of the NC machines, also called variable block format. A typical instruction block will be as below :

N20 G00 X1.200 Y.100 F325 S1000 T03 M09 <EOB>

or

N20 G00 X1.200 Y.100 F325 S1000 T03 M09;

The MCU uses this alphabet for addressing a memory location in it.

Tab Sequential Format

Here the alphabets are replaced by a Tab code, which is inserted between two words. The MCU reads the first Tab and stores the data in the first location then the second word is recognized by reading the record Tab. A typical Tab sequential instruction block will be as below :

>20 >00 >1.200 >.100 >325 >1000 >03 >09

Fixed Block Format

In fixed block format no letter address of Tab code are used and none of words can be omitted. The main advantage of this format is that the whole instruction block can be read at the same instant, instead of reading character by character. This format can only be used for positioning work only. A typical fixed block instruction block will be as below:

20 00 1.200 .100 325 1000 03 09 <EOB>

4.3.6 Machine Tool Zero Point Setting

The machine zero point can be set by two methods by the operator, manually by a programmed absolute zero shift, or by work coordinates, to suit the holding fixture or the part to be machined.

Manual Setting

The operator can use the MCU controls to locate the spindle over the desired part zero and then set the X and Y coordinate registers on the console to zero.

Absolute Zero Shift

The absolute zero shift can change the position of the coordinate system by a command in the CNC program. The programmer first sends the machine spindle to home zero position by a command in the program. Then another command tells the MCU how far from the home zero location, the coordinate system origin is to be positioned.

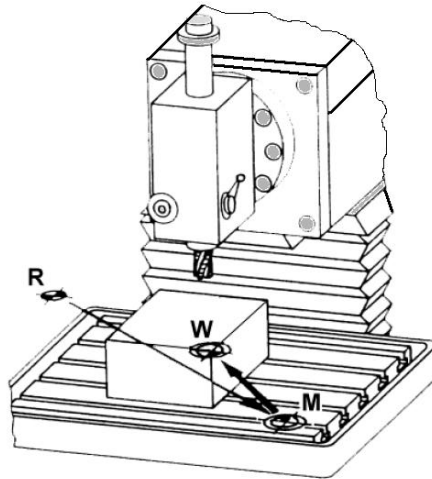


Figure 4.3 : Machine Tool Zero Point Setting

R = Reference point (maximum travel of machine)



W = Part zero point workpiece coordinate system



M = Machine zero point (X0, Y0, Z0) of machine coordinate system



The sample commands may be as follows :

N1 G28 X0 Y0 Z0 (sends spindle to home zero position or Return to reference point).

N2 G92 X3.000 Y4.000 Z5.000 (the position the machine will reference as part zero or Programmed zero shift).

4.3.7 Coordinate Word

A coordinate word specifies the target point of the tool movement or the distance to be moved. The word is composed of the address of the axis to be moved and the value and direction of the movement.

Example

X150 Y-250 represents the movement to (150, -250). Whether the dimensions are absolute or incremental will have to be defined previously using G-Codes.

4.3.8 Parameter for Circular Interpolation

These parameters specify the distance measured from the start point of the arc to the center. Numerals following *I*, *J* and *K* are the *X*, *Y* and *Z* components of the distance respectively.

4.3.9 Spindle Function

The spindle speed is commanded under an *S* address and is always in revolution per minute. It can be calculated by the following formula :

$$\text{Spindle Speed} = \frac{\text{Surface cutting speed in m/min} \times 1000}{\pi \times \text{Cutter Diameter in mm}}$$

Example

S1000 represents a spindle speed of 1000 rpm.

4.3.10 Feed Function

The feed is programmed under an *F* address except for rapid traverse. The unit may be in mm per minute or in mm per revolution. The unit of the federate has to be defined at the beginning of the programme. The feed rate can be calculated by the following formula :

$$\text{Feed Rate} = \frac{\text{Chip Load}}{\text{Tooth} \times \text{No. of tooth} \times \text{Spindle speed}}$$

Example

F100 represents a feed rate of 100 mm/min.

4.3.11 Tool Function

The selection of tool is commanded under a *T* address. T04 represents tool number 4.

4.3.12 Work Settings and Offsets

All NC machine tools require some form of work setting, tool setting, and offsets to place the cutter and work in the proper relationship. Compensation allows the programmer to make adjustments for unexpected tooling and setup conditions. A retraction point in the *Z*-axis to which the end of the cutter retracts above the work surface to allow safe table movement in the *X-Y* axes. It is often called the rapid-traverse distance, retract or work plane. Some manufacturers build a workpiece height distance into the MCU (machine control unit) and whenever the feed motion in the *Z*-axis will automatically be added to the depth programmed.

When setting up cutting tools, the operator generally places a tool on top of the highest surface of the work piece. Each tool is lowered until it just touches the workpiece surface and then its length is recorded on the tool list. Once the work piece has been set, it is not generally necessary to add any future depth dimensions since most MCU do this automatically.

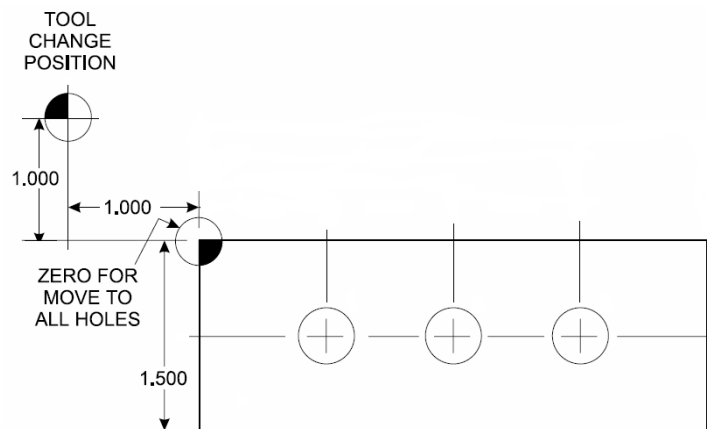


Figure 4.4 : Work Settings

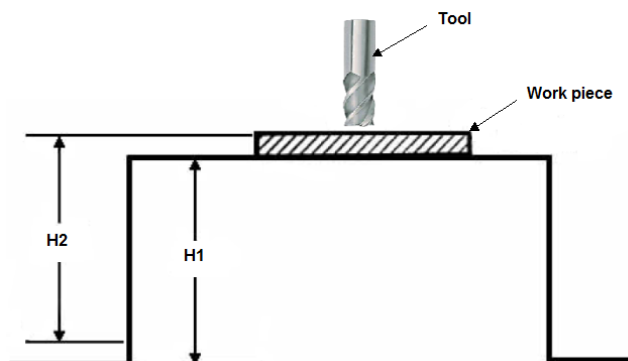


Figure 4.5 : Offsets

4.3.13 Rapid Positioning

This is to command the cutter to move from the existing point to the target point at the fastest speed of the machine.

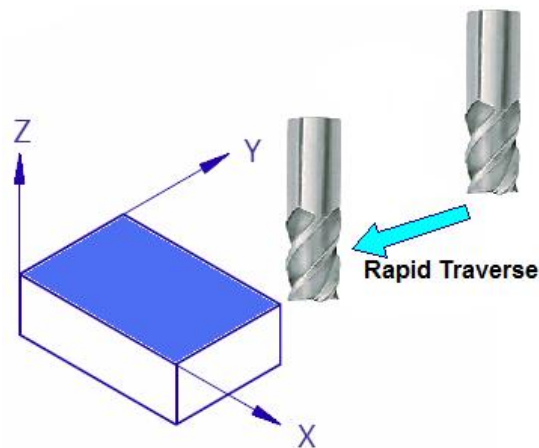


Figure 4.6 : Rapid Positioning

4.3.14 Linear Interpolation

This is to command the cutter to move from the existing point to the target point along a straight line at the speed designated by the F address.

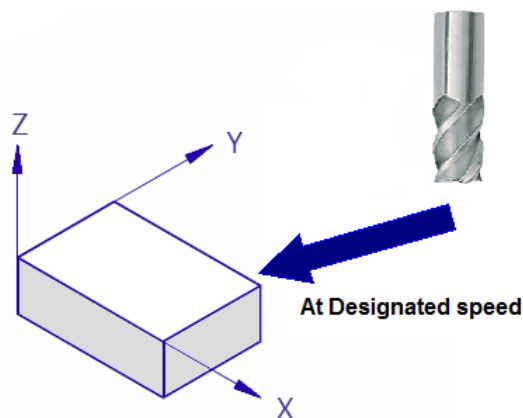


Figure 4.6 : Linear Interpolation

4.3.15 Circular Interpolation

This is to command the cutter to move from the existing point to the target point along a circular arc in clockwise direction or counter clockwise direction. The parameters of the center of the circular arc is designated by I , J and K addresses. I is the distance along the X -axis, J along the Y , and K along the Z . This parameter is defined as the vector from the starting point to the center of the arc.

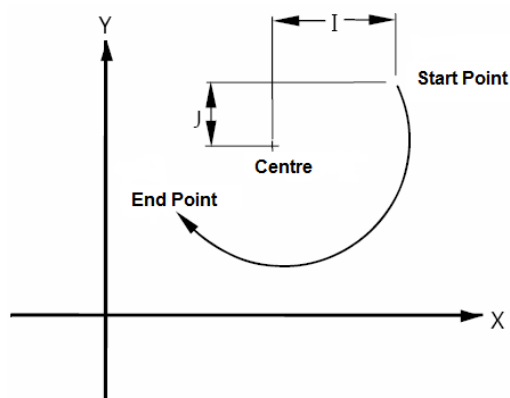


Figure 4.7 : Clockwise Circular Interpolation

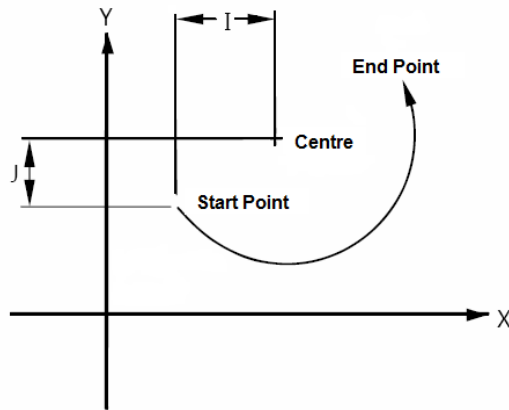


Figure 4.8 : Counter Clockwise Circular Interpolation

4.3.16 Circular Interpolation

In NC machining, if the cutter axis is moving along the programmed path, the dimension of the workpiece obtained will be incorrect since the diameter of the cutter has not been taken into account. What the system requires are the programmed path, the cutter diameter and the position of the cutter with reference to the contour. The cutter diameter is not included in the programme. It has to be input to the NC system in the tool setting process.

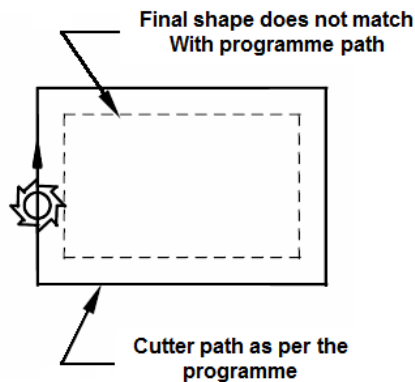


Figure 4.9 : Tool Path without Cutter Compensation

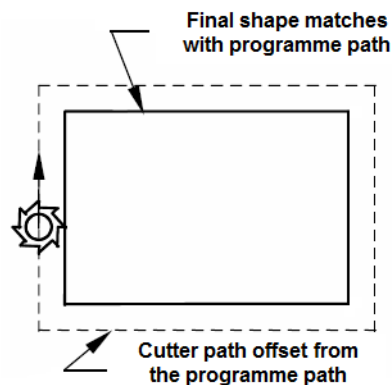


Figure 4.10 : Tool Path with Cutter Compensation

4.4 SYMBOLS USED

- % – Main Programme (1 to 9999)
- L – Sub program (1 to 999)/Home position
- N – Sequence of block number.
- Lf – Block end (EOB) means “; or *”
- T – Tool number or Tool station number

D – Tool offset
 S – Spindle speed
 F – Feed
 M – Switching function
 G – Transverse commands
 R – Parameters
 I, J, K – Circle parameters
 B/U/R – Radius
 X/Y/Z – Axis coordinates
 P – Passes.

4.5 PART PROGRAM FOR LATHE OPERATION

The CNC lathe operation such as simple facing, turning, taper turning, thread, boring, parting off etc. The X-axis and Z-axis are taken as the direction of transverse motion of the tool post and the axis of the spindle respectively. To prepare part programs using G-codes and M-codes. The following examples illustrated the part program for different components.

Example

01 (All dimensions are in mm).

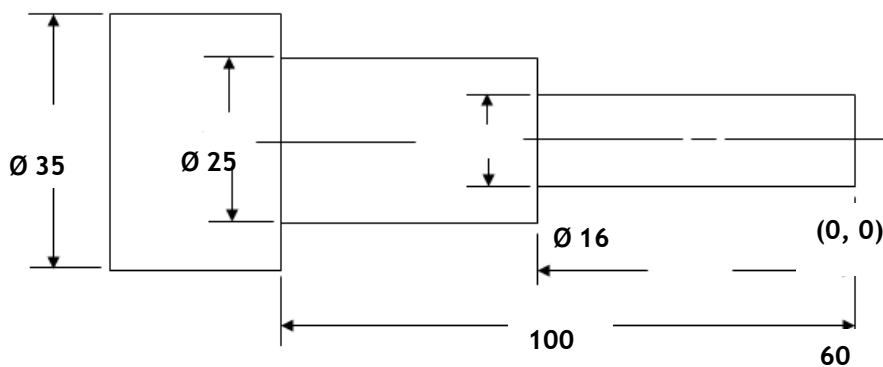


Figure 4.11 : Turning Operation

% 1000;	(Main programme)
N01 G54 G90 G71 G94 M03 S800;	(Parameters Setting)
N05 G01 X-12.5 Z0 F2;	(Facing the job)
N10 G00 Z1;	(Retrieval of tool)
N15 G00 X00;	(Tool clearance)
N20 G01 Z-100;	(Starting cut)
N25 G00 X1 Z1;	(Clearance position)
N30 G00 X-2;	(Position of cut)
N35 G01 Z-60;	(Cutting length)
N40 G00 X-1 Z1;	(Retrieval of tool)
N45 G00 X-3;	(Position of cut)
N50 G01 Z-60;	(Cutting length)
N55 G00 X-2 Z1;	(Retrieval of tool)
N60 G00 X-4;	(Position of cut)
N65 G01 Z-60;	(Cutting length)

N70 G00 X-3 Z1;	(Retrieval of tool)
N75 G00 X-4.5;	(Position of cut)
N80 G01 Z-60;	(Cutting length)
N85 G00 X5 Z5;	(Final position of tool)
N90 M02;	(End of programme)

Example

02 (All dimensions are in mm).

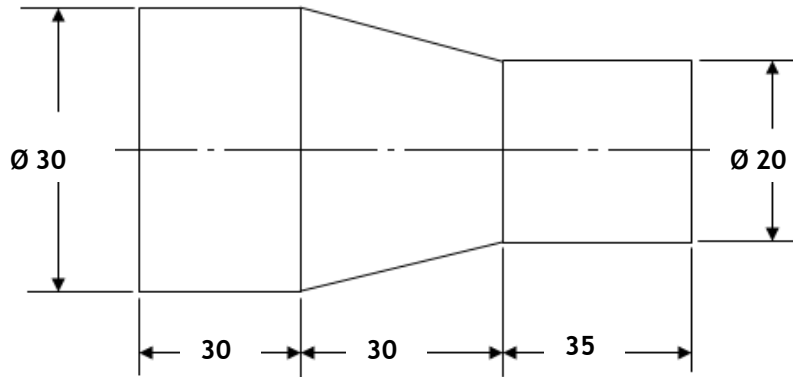


Figure 4.12 : Taper Turning

% 2000;	(Main programme)
N01 G54 G91 G71 G94 M03 S800;	(Parameters Setting)
N05 G01 X-15 Z0 F2;	(Facing the job)
N10 G00 Z1;	(Tool clearance)
N15 G00 X10;	(Tool clearance from the centre)
N20 G01 Z-36;	(Turning operation)
N25 G01 X5 -Z30;	(Taper turning operation)
N30 G00 X1 Z66;	(Final position of tool)
N35 M02;	(End of programme)

Example

03 (All dimensions are in mm).

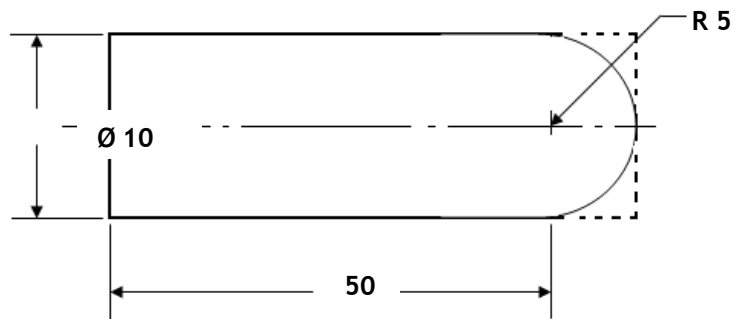


Figure 4.13 : Circular Interpolation

% 2000;	(Main programme)
N01 G91 G71 G94 M03 S800;	(Parameters Setting)
N05 G01 X-5 Z0 F1;	(Facing the job)
N10 G02 X5 Z-5 I0 K5;	(Circular Interpolation)
N15 G00 X6 Z6;	(Final position of tool)
N20 M02;	(End of programme).

4.6 PART PROGRAM FOR MACHINING CENTRES (MILLING)

The CNC milling machine, the motion is possible in three axes, X-axis, Y-axis and Z-axis. The movement of Z-axis is taken as positive when tool moves away from the job or vice versa.

Example

01 (All dimensions are in mm).

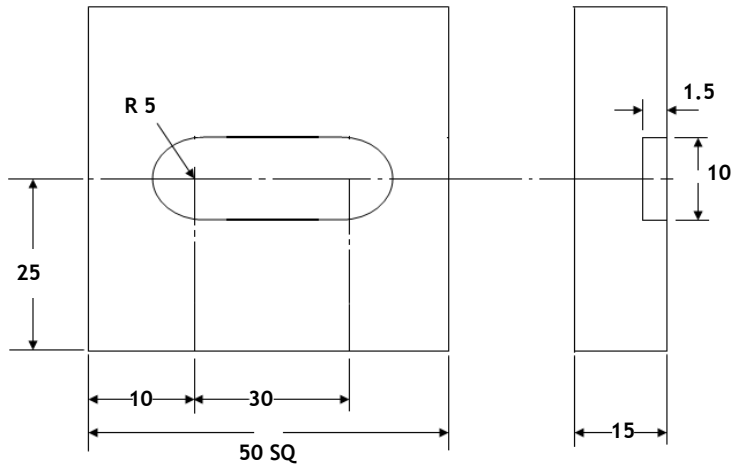


Figure 4.14 : Straight Line

% 100;	(Main programme)
N5 G17 G71 G90 G94 G54;	(Parameters Setting)
N10 T2 L90;	(Home position)
N15 G00 D2 Z50 M3 S700 X10 Y-25;	(Position of tool)
N20 G01 Z-1.5;	(Position of cut)
N25 G01 X4 F100 M8;	(Cutting slat)
N30 G00 Z100 M9;	(Final position of tool)
N35 M30;	(Main programme end)

Example

02 (All dimensions are in mm).

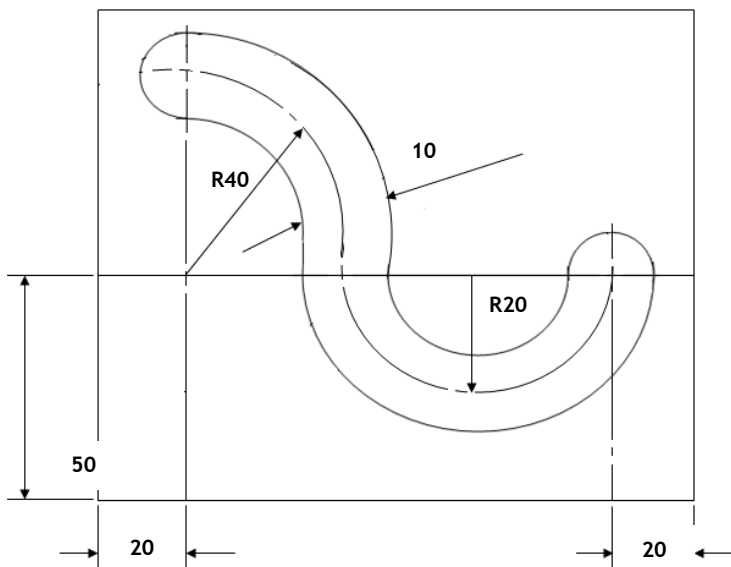


Figure 4.15 : Circular Interpolation

% 101;	(Main programme)
N2 G17 G71 G90 G94 G54;	(Parameters Setting)
N4 T1 L90;	(Home position)
N6 G00 Z5 D5 M3 S500 X20 Y90;	(Position of tool)
N8 G01 Z-2 F50;	(Position of cut)
N10 G02 X60 Y50 I0 J-40;	(Circular interpolation clockwise-CW)
N12 G03 X80 Y50 I20 J0;	(Circular interpolation clockwise-CCW)
N14 G00 Z100;	(Final position of tool)
N16 M02;	(End of programme)

4.7 FIXED CYCLE/CANNED CYCLE

Machining holes is probably the most common operation, mainly done on CNC milling machines and machining centers. Even in the industries traditionally known for their complex parts, such as aircraft and aerospace components manufacturing, electronics, instrumentation, optical or mold making industries, machining holes is a vital part of the manufacturing process. Machining on simple hole may require only one tool but a precise and complex hole may require several tools to be completed. Number of holes required for a given job is important for selection of proper programming approach. In the majority of programming applications, hole operations offer a great number of similarities from one job to another. Hole machining is a reasonably predictable is an ideal subject to be handled very efficiently by a computer. Several advance technique are used such that a sequence can be programmed just once and given an identity so that it can be called back into the main programme as and when required. These sequences are referred to in a number of ways like cycle, subroutines and loops, etc.

A fixed cycle is a combination of machine moves resulting in a particular machining function such as drilling, milling, boring and tapping. By programming one cycle code number, as many as distinct movements may occur. These movements would take blocks of programme made without using Fixed or Canned cycles. The corresponding instructions of a fixed cycle are already stored in the system memory. The advantages of writing a part programme with these structures are :

- (a) Reduced lengths of part programme.
- (b) Less time required developing the programme.
- (c) Easy to locate the fault in the part programme.
- (d) No need to write the same instructions again and again in the programme.
- (e) Less memory required in the control unit.

The following examples are some basic and fixed cycle codes available with a number of machines, assigned by EIA.

Example

- 01 (G81 Drilling Cycle) (All dimensions are in mm).
- R00 – Dwell time at the starting point for chip removal.
- R02 – Reference plane absolute with sign.
- R03 – Final depth of hole absolute with sign.
- R04 – Dwell time at the bottom of drilled hole for chip breaking.
- R10 – Retract plane without sign.
- R11 – Drilling axis number 1 to 3.
- % 400;

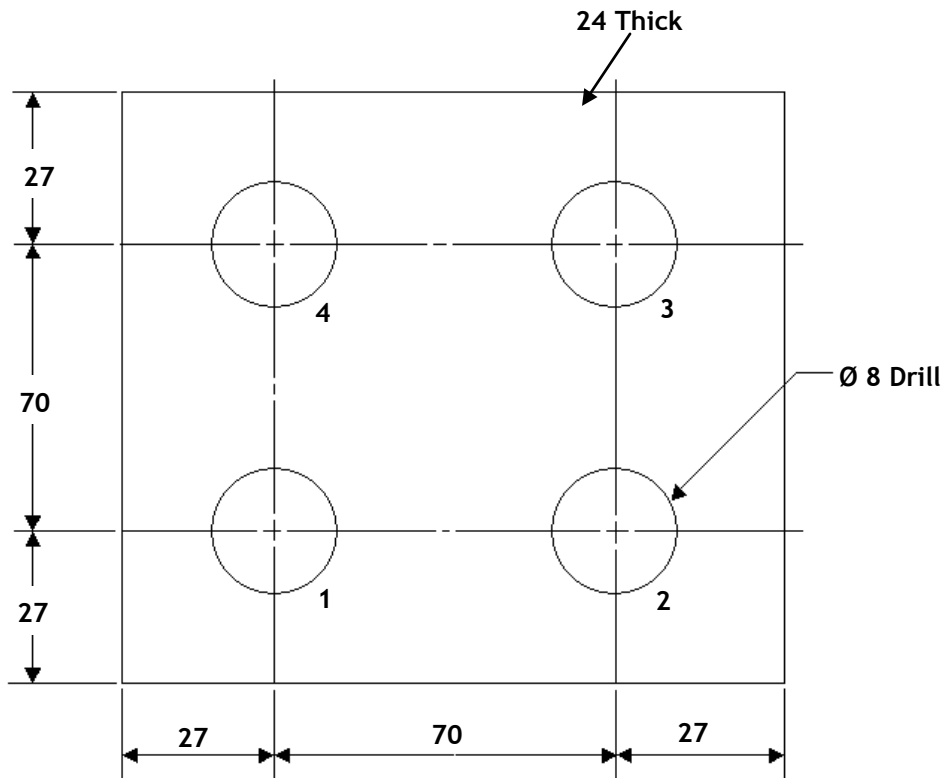


Figure 4.16 : Drilling Cycle

```

N5 G17 G71 G90 G94 G55;
N10 T1 L90;
N15 G00 D5 Z5 M3 S600 X27 Y27;
N20 G81 R02=5, R03=-33, R11=3, F50 M7;
N25 X97;
N30 Y97;
N35 X27;
N40 G00 G80 Z100 M9;
N45 M02;
    
```

Example

02 (G83 Deep Drilling Cycle) (All dimensions are in mm).

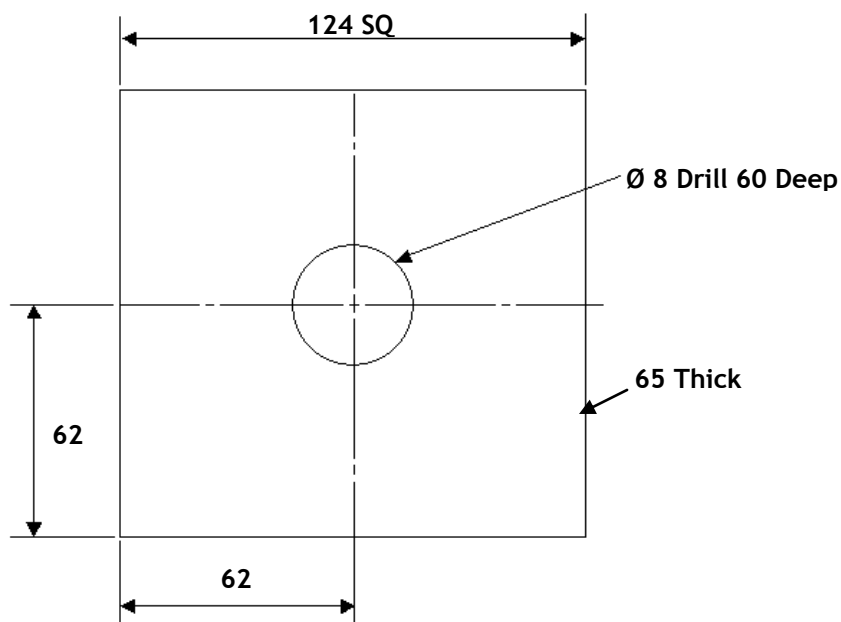


Figure 4.17 : Deep Drilling Cycle

R00 – Dwell time at the starting point for chip removal.
 R01 – First drilling depth (incremental) without sign.
 R02 – Reference plane absolute with sign.
 R03 – Final depth of hole absolute with sign.
 R04 – Dwell time at the bottom of drilled hole for chip breaking.
 R05 – Amount of digression is without sign.
 R10 – Retract plane without sign.
 R11 – Drilling axis number 1 to 3.
 % 401;
 N5 G17 G71 G90 G94 G55;
 N10 T1 L90;
 N15 G00 D5 Z5 M3 S600 X62 Y62;
 N20 G83 R00=30, R01=15, R02=5, R03=-60, R04=1, R05=15, R10=80, R11=3,
 F50 M7;
 N25 G00 G80 Z100 M9;
 N30 M02;

Example

03 (G84 Tapping Cycle) (All dimensions are in mm).

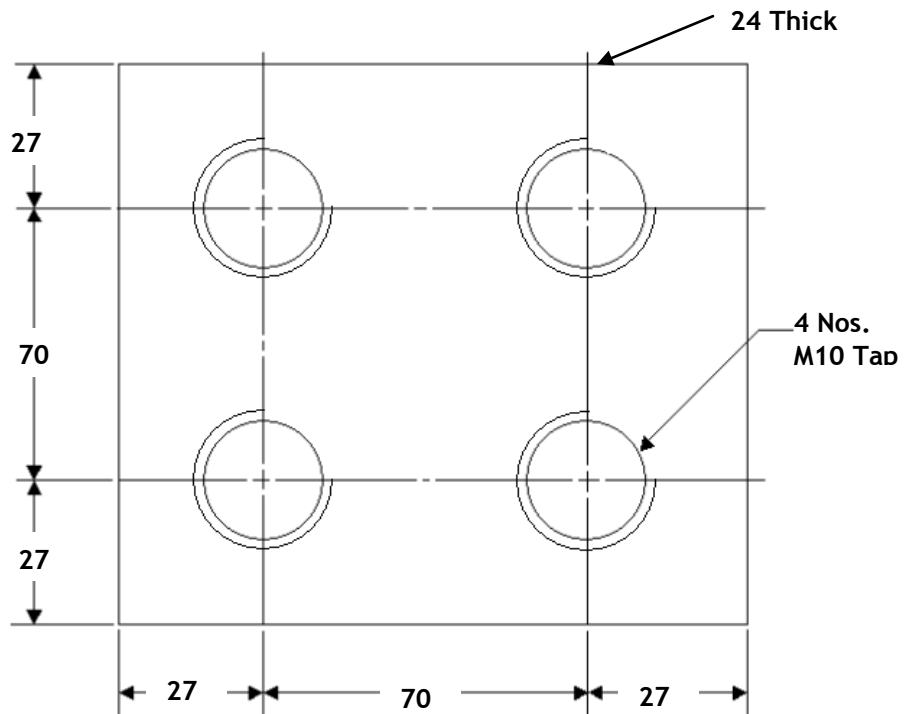


Figure 4.18 : Tapping Cycle

R02 – Reference plane absolute with sign.
 R03 – Final depth of hole absolute with sign.
 R04 – Dwell time at the bottom of drilled hole for chip breaking.
 R06 – Reverse direction of spindle rotation.
 R07 – Return to the original direction of spindle rotation.
 R08 – Machine data setting.
 R09 – Thread pitch.


```

R11 – Drilling axis number 1 to 3.
% 403;
N5 G17 G71 G90 G94 G55;
N10 T1 L90;
N15 G00 D5 Z5 M3 S600 X27 Y27;
N20 G81 R02=5, R03=-32, R11=3, F50 M7;
N25 X97;
N30 Y97;
N35 X27;
N40 G00 G80 Z100 M9;
N45 T2 L90;
N43 G00 D10 Z5 M3 S60 X27 Y27;
N50 G84 R02=5, R03=-29, R04=1, R06=4, R07=3, R08=0, R09=1, R11=3
F60 M7;
N55 X97;
N60 Y97;
N65 X27;
N70 G00 G80 Z100 M9;
N75 M02;

```

Example

04 (G86 Boring cycle) (All dimensions are in mm).

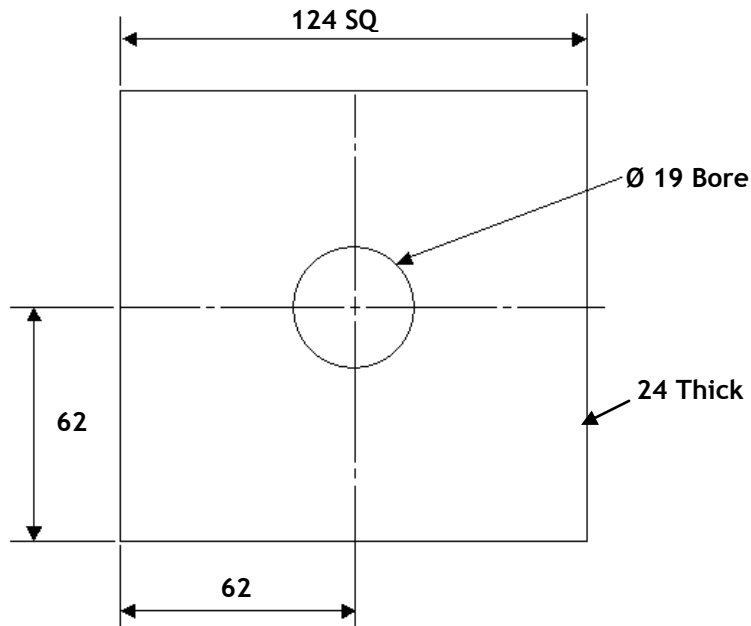


Figure 4.19 : Boring Cycle

R02 – Reference plane.
R03 – Final depth of hole.
R04 – Dwell time at the bottom of drilled hole for chip breaking.
R07 – Spindle on after M05.
R10 – Retract plane.
R11 – Drilling axis.

```

R12 – X distance.
R13 – Y distance.
% 404;
N5 G17 G71 G90 G94 G55;
N10 T1 L90;
N15 G00 D5 Z5 M3 S600 X62 Y62;
N20 G81 R02=5, R03=-27, R11=3, F50;
N25 G00 G80 Z100 M9;
N30 T2 L90;
N35 G00 D10 Z5 M3 S600 X62 Y62;
N40 G81 R02=5, R03=-30, R11=3, F50 M7;
N45 G00 G80 Z100 M9;
N50 T3 L90;
N55 G00 D15 Z5 M3 S600 X62 Y62;
N60 G81 R02=5, R03=-33, R11=3, F50 M7;
N65 G00 G80 Z100 M9;
N70 T4 L90;
N75 G00 D20 Z5 M3 S800 X62 Y62;
N80 G86 R02=5, R03=-33, R04=1, R07=3, R10=60, R11=3, R12=0.1, R13=0.1,
F50 M7;
N85 G00 G80 Z100 M9;
N90 M02;

```

4.8 DO-LOOPS

In a few jobs some portion of the programme needs to be repeated, which do not fit into standardized category. Some of the non-standardized cycles are Do-loops and Subroutines. Do-loop is a number of operations repeated over a number of equal steps for a previously fixed number of times.

Do-loops always are implemented on incremental mode because each previous position becomes reference for next iteration. Do-loop is actually jumping back to an already written initial portion of the program for the number of times a loop count.

Example

01 (Do-loop) (All dimensions are in mm).

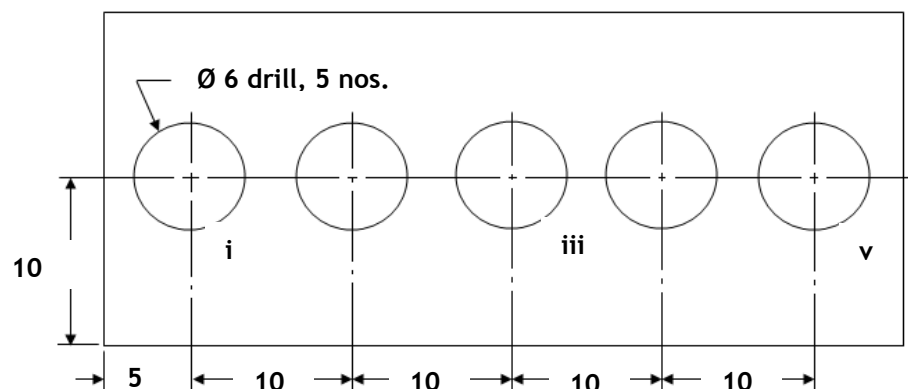


Figure 4.20 : Do-loop

```

% 500;
N2 G71 G90 G94;
N4 G92 X0 Y0 Z0 ;
N6 T1 M06;
N8 G81 G99 X5 Y10 Z-8 R.2 F100 S500 M03 M08;   (Canned Drill cycle)
N10 G51 P4;                                       (Start loop 4 times)
N12 G91 X 10 ;
N14 G50 ;
N16 G80 G90 M09;                                  (Cancel cycle)
N18 T2 M06 ;
N20 G81 G99 X5 Y10 Z-8 R.2 F100 S500 M03 M08;   (Canned Drill cycle)
N22 G51 P4 ;                                       (Start loop 4 times)
N24 G91 X 10 ;
N26 G50;
N28 G80 G90 M09;                                  (Cancel cycle)
N30 M30;

```

4.9 SUBROUTINE

A subroutine is a portion of a programme, complete in itself, which is stored in computer after programming once. It is called with required data when required again in a programme.

Example

01 (Subroutine) (All dimensions are in mm).

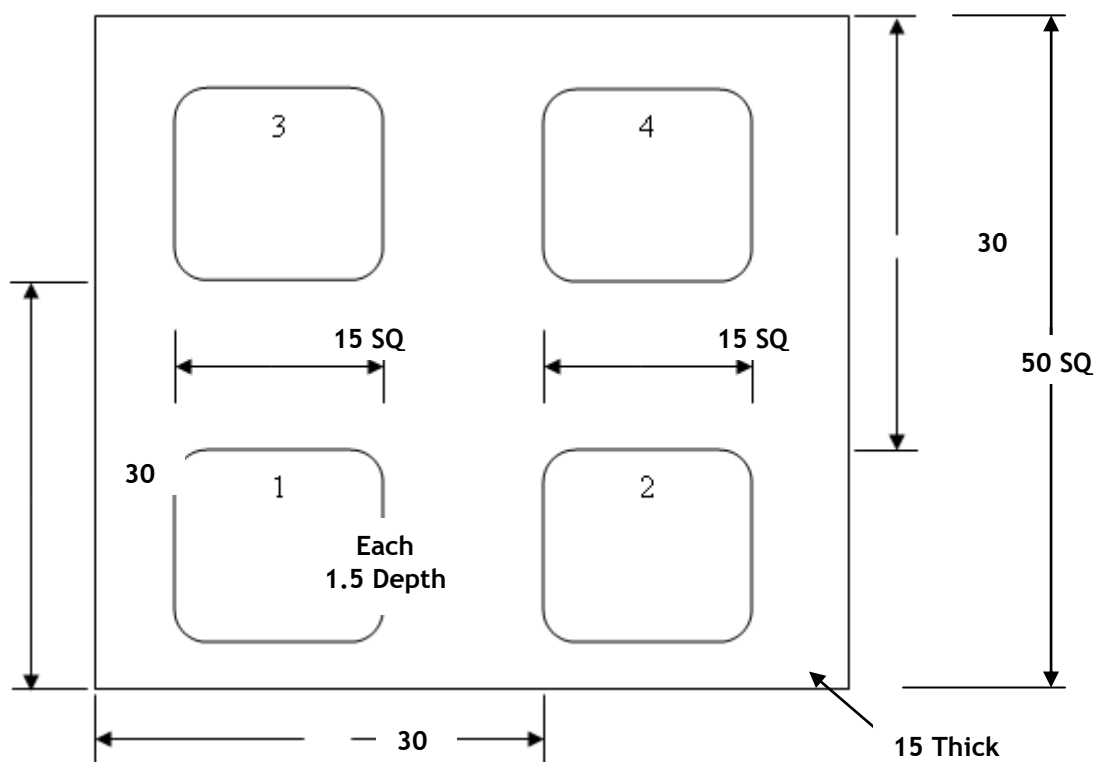


Figure 4.21 : Subroutine

```
%1001;  
N2 G17 G71 G94 G90 G54;  
N4 T1 L90;  
N6 G00 D5 Z5 M3 S500 X9 Y16;  
N8 G01 Z0 F500;  
N10 L100 P1; (Call the subroutine)  
N12 G00 X34 Y16;  
N14 G01 Z0 F500;  
N16 L100 P1; (Call the subroutine)  
N18 G00 X9 Y41;  
N20 G01 Z0 F500;  
N22 L100 P1; (Call the subroutine)  
N24 G00 X34 Y41;  
N26 G01 Z0 F500;  
N28 L100 P1; (Call the subroutine)  
N30 G00 Z100;  
N32 M02;
```

Subroutine Programme is below :

```
L100;  
N2 G01 G91 Z-1.5 F100 M7;  
N4 X7;  
N6 Y-7;  
N8 X-7;  
N10 Y7;  
N12 G00 G90 Z5 M9;  
N14 M17;
```

This has been called as a subroutine in the main programme as above.

4.10 SUMMARY

4.11 ANSWERS TO SAQs
