



Total Commitment to Education and Training WorldWide.

G and M Programming for CNC Milling Lathes.

Denford Limited reserves the right to alter any specifications and documentation without prior notice. No part of this manual or its accompanying documents may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Denford Limited.

All brands and products are trademarks or registered trademarks of their respective companies.

Copyright Denford Limited - Version 1.05.01. All rights reserved.

Contact Details

Address:	Denford Limited, Birds Royd, Brighouse, West Yorkshire, HD6 1NB, UK.
Telephone:	General Enquiries +44 (0) 1484 712264 Customer Services +44 (0) 1484 722733
Fax:	+44 (0) 1484 722160
e-mail:	for customer services and technical support contact, customer_services@denford.co.uk

About this Manual

Use:	This manual provides information relating to G and M programming on a Denford CNC lathe.
Disclaimer:	We take great pride in the accuracy of information given in this manual, but due to nature of continuous development, be aware that software, hardware and operational specifications and features of this product can change without notice. No liability can be accepted by Denford Limited for loss, damage or injury caused by any errors in, or omissions from, the information supplied in this manual. Please note that not all the G and M codes listed in this guide are supported by every Denford CNC machine and/or control software.
Screenshots:	Please note that any screenshots and diagrams are used for explanation purposes only.
Language:	This manual is written using European English.
Contact:	Any comments regarding this manual should be referred to the following e-mail address: customer_services@denford.co.uk
Updates:	Any updates to this manual will be posted on the Denford website: http://www.denford.com

Contents

Preface.

Contact Details	2
About this Manual	2
Contents	3

Section 1: Introduction.

Introduction to G & M Code Programming	6
What is CNC?	7
Example CNC Manufacturing Process	8

Section 2: Basic Programming.

Composition of a Part Program	9
Main Program Structure	10
Sub Program Structure	12
Sub Program Repeat Command	14
Billet Definition	15
Program Numbering and Saving	16
Program Block Numbering	17
Block Configuration	18
G Codes (Preparatory Functions)	18
Tool Movement	19
Thread Cutting	21
Feed Function	22
M Codes (Miscellaneous Functions)	22
Spindle Speed Function (Cutting Speed)	23
Tool Function	23
Tool Compensation (Tool Offset)	24
Absolute and Incremental Co-ordinates	25
Optional Block Skip	26
Tutorials and Comments	26

Contents

Section 3: G Codes (Preparatory Functions).

G Codes (Preparatory Functions) - Introduction	27
List of G Codes supported by Denford CNC Controls	29
G00 (Rapid Positioning/Traverse)	30
G01 (Linear Interpolation)	34
G02 G03 (Circular Interpolation).....	37
G04 (Dwell)	41
G20 G21 (Inch/Metric Data Input)	42
G28 (Reference Point Return)	43
G40 G41 G42 (Tool Nose Radius Compensation)	46
G50 (Multiple Command Functions - 1. Clamping Max. Spindle Speed)	55
G50 (Multiple Command Functions - 2. Co-ordinate System Setting)	56
G50 (Multiple Command Functions - 3. Co-ordinate System Shift).	59
G70 (Finishing Cycle)	60
G71 (Stock Removal in X Axis)	61
G72 (Stock Removal in Facing)	68
G73 (Pattern Repeating)	72
G74 (End Face Peck Drilling Cycle)	74
G75 (Outer Diameter/Internal Diameter Drilling and Grooving Cycle).....	78
G76 (Multiple Thread Cutting Cycle)	80
G81 (Deep Hole Drilling Cycle)	84
G90 (Outer Diameter/Internal Diameter Cutting Cycle)	85
G92 (Thread Cutting Cycle)	89
G94 (End Face Turning Cycle)	91
G96 (Constant Surface Speed Control)	94
G97 (Spindle Speed in Rev/Minute)	95
G98 (Per Minute Feed)	96
G99 (Per Revolution Feed)	98

Contents

Section 4: M Codes (Miscellaneous Functions).

M Codes (Miscellaneous Functions) - Introduction	99
List of M Codes supported by Denford CNC Controls	100
M00 (Program Stop)	101
M01 (Optional Stop)	101
M02 (End of Program)	101
M03 (Spindle Forward)	101
M04 (Spindle Reverse)	101
M05 (Spindle Stop)	102
M06 (Automatic Tool Change)	102
M08 (Coolant On)	102
M09 (Coolant Off)	102
M10 (Chuck Open)	102
M11 (Chuck Close)	102
M13 (Spindle Forward and Coolant On)	103
M14 (Spindle Reverse and Coolant On)	103
M25 (Tailstock Quill Extend)	103
M26 (Tailstock Quill Retract)	103
M30 (Program Stop and Reset)	104
M40 (Parts Catcher Extend)	104
M41 (Parts Catcher Retract)	104
M62, M63, M64, M65, M66, M76, M77 (Auxiliary Output Functions)	104
M98 (Sub Program Call)	105
M99 (Sub Program End and Return)	105

1: Introduction to G & M Code Programming

The Denford CNC (Computer Numerical Control) unit fitted to Denfords range of machine tools is a FANUC compatible system which uses ISO code format.

This manual covers the stages involved in producing the coded instructions, used by the CNC unit to make the component. These coded instructions are called the part program. Each part program contains a number of different codes, the most important being the collection of G and M codes. Essentially, these form the basic language used to describe how a component will be manufactured, the order in which to carry out machining tasks, when to change tools, how far to cut into the material etc.

Section 2 of this manual cover the basics of part programming, including guidelines for general layout and commands. Each topic builds progressively, using plain, easy to follow text, to cover the most common aspects of programming. At the end of section 2, the operator should be confident enough to tackle basic part programming.

Section 3 numerically lists the full range of G Codes available on Denford CNC milling machines, whilst section 4 numerically lists the full range of M Codes available. G and M Codes contain information which are more specific to certain commands and functions. These sections are intended more as a reference guide, once the operator is confident with the basics of programming. Please note that not all the G and M Codes listed in this manual may apply to your particular machine. A full list of G and M codes that apply to your machine is included in your separate Denford CNC milling machine manual.

Naturally, this manual cannot "teach" the operator everything there is to know about programming. The subject is simply too vast to include it all. The content of this manual does, however, form a good basis from which to start learning and hopefully inspires confidence in using more technically structured documents.

1: What is CNC?

CNC (Computer Numerical Control) is the general term used for a system which controls the functions of a machine tool using coded instructions processed by a computer.

Example CNC Manufacturing Process.

The diagram on page 8 shows the main stages involved in producing a component on a CNC system:

- 1) A part program is written, using G and M codes. This describes the sequence of operations that the machine must perform in order to manufacture the component.

This program can be produced off-line, ie, away from the machine, either manually or with the aid of a CAD/CAM system.

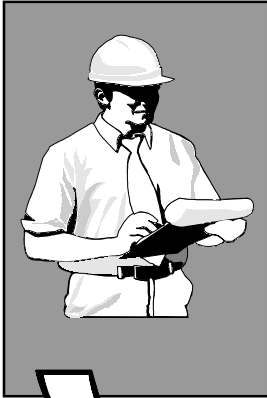
- 2) The part program is loaded into the machines computer, called the controller. At this stage, the program can still be edited or simulated using the machine controller keypad/input device.
- 3) The machine controller processes the part program and sends signals to the machine components directing the machine through the required sequence of operations necessary to manufacture the component.

The application of CNC to a manual machine allows its operation to become fully automated.

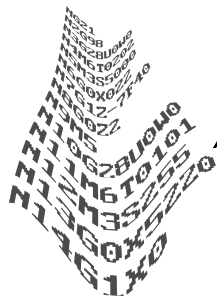
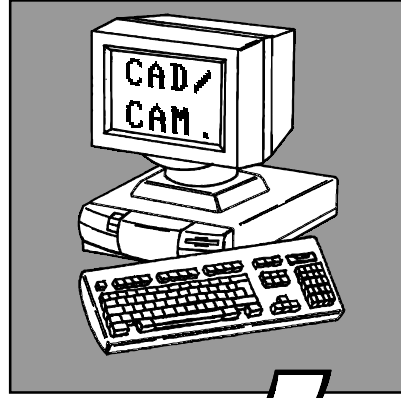
Combining this with the use of a part program enhances the ability of the machine to perform repeat tasks with high degrees of accuracy.

1: Example CNC Manufacturing Process

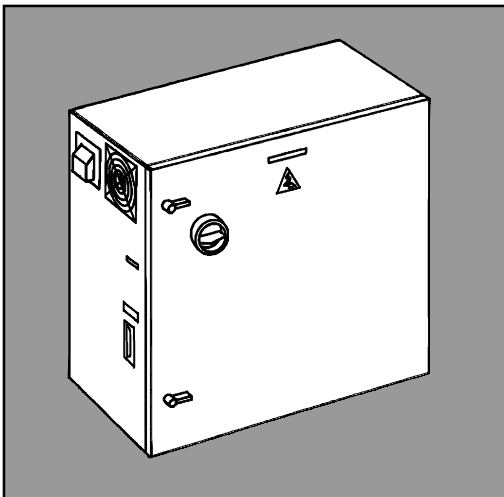
Human Programming (MDI - Manual Data Input).



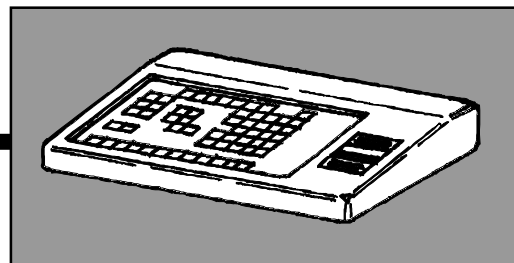
Computer Programming (CAD/CAM).



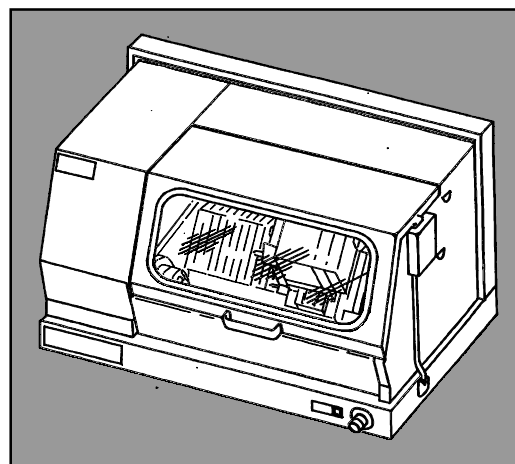
G & M Codes.



Machine Electrical Control Unit.



Machine Control Keypad.



Denford CNC Lathe.

Control Signals.

2: Composition of a Part Program

Part Program example -

(Lathe CAM Designer - ØØ5Ø.LCD)

(22/1Ø/1996)

(Mirac lathe (metric))

(Post fanucl:1.2 23 May 1994)

OØØ5Ø;

NØØ1Ø G21;

[BILLET X1Ø Z2Ø;

NØØ2Ø G98;

NØØ3Ø G28UØWØ;

NØØ4Ø M6TØ1Ø1;

NØØ5Ø M3S2546;

NØØ6Ø GØØX1ØZ2;

NØØ7Ø X12;

NØØ8Ø X1Ø;

NØØ9Ø GØ1Z-13.5F1Ø;

NØ1ØØ X12;

NØ11Ø GØØZ2;

NØ12Ø X8;

NØ13Ø M3S3183;

NØ14Ø GØ1Z-8.26;

NØ15Ø X1Ø;

NØ16Ø GØØZ2;

NØ17Ø X6;

NØ18Ø M3S4ØØØ;

NØ19Ø GØ1Z-6.26;

NØ2ØØ X8;

NØ21Ø X12;

NØ22Ø GØØZ2;

NØ23Ø GØ1ZØ;

NØ24Ø X4;

NØ25Ø Z-5;

NØ26Ø M3S2829;

NØ27Ø X9Z-1Ø;

NØ28Ø Z-14;

NØ29Ø X12;

NØ3ØØ GØØZ2;

NØ31Ø M5;

NØ32Ø G28UØWØ;

NØ33Ø M3Ø;

A Part Program is a list of coded instructions which describes how the designed component, or part, will be manufactured.

These coded instructions are called data - a series of letters and numbers. The part program includes all the geometrical and technological data to perform the required machine functions and movements to manufacture the part.

The part program can be further broken down into separate lines of data, each line describing a particular set of machining operations. These lines, which run in sequence, are called blocks.

A block of data contains words, sometimes called codes. Each word refers to a specific cutting/movement command or machine function. The programming language recognised by the CNC, the machine controller, is an I.S.O. code, which includes the G and M code groups.

Each program word is composed from a letter, called the address, along with a number.

Block example - NØ14Ø G1Z-8.26;

Word example - GØ1

Address example - G

2: Main Program Structure

The part program can contain a number of separate programs, which together describe all the operations required to manufacture the part.

The *Main Program* is the controlling program, ie, the program first read, or accessed, when the entire part program sequence is run. This controlling program can then call a number of smaller programs into operation. These smaller programs, called *Sub Programs*, are generally used to perform repeat tasks, before returning control back to the main program.

Normally, the controller operates according to one program. In this case the main program is also the part program.

Main Programs are written using I.S.O. address codes listed below.

Addresses -

N refers to the block number.

G refers to the G code (Preparatory function).

X refers to the absolute distance travelled by the slide tool in the X axis direction.

U refers to the incremental distance travelled by the slide tool in the X axis direction.

Z refers to the absolute distance travelled by the slide tool in the Z axis direction.

W refers to the incremental distance travelled by the slide tool in the Z axis direction.

F refers to the feed rate.

M refers to the M code (Miscellaneous function).

S refers to the spindle speed.

T refers to the tooling management.

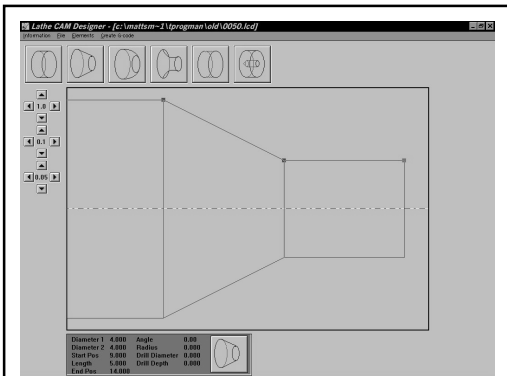
Each block, or program line, contains addresses which appear in this order :

N , G , X (or U) , Z (or W) , F , M , S , T ;

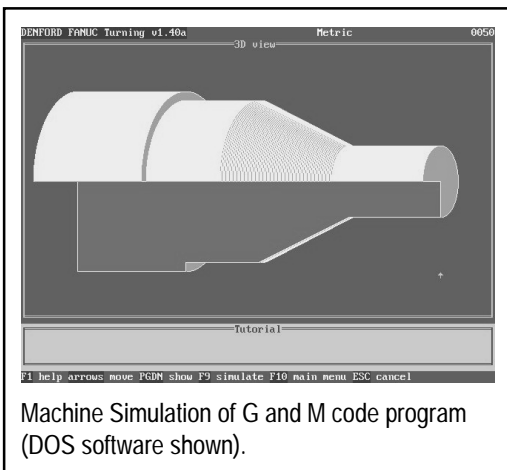
This order should be maintained throughout every block in the program, although individual blocks may not necessarily contain all these addresses.

2: Main Program Structure

The organisation of blocks of data within the program follows a layout. Again, it is recommended that the programmer keeps to this program layout.



The main program can be generated using Denfords LatheCAM Designer CAD/CAM software package, saved onto disk and transferred to the CNC. Alternatively, the main program can be manually entered into the CNC memory when the controller is set in Edit Mode.



Machine Simulation of G and M code program (DOS software shown).

- CAD/CAM software package information. — (Lathe CAM Designer - Ø050.LCD)
(22/10/1996)
(Mirac lathe (metric))
(Post fanuc:1.2 23 May 1994)
- Program number. — O0050;
- Units definition (Metric or Imperial) and billet size for simulation. — N0010 G21;
[BILLET X10 Z20;
N0020 G98;
N0030 G28U0W0;
N0040 M6T0101;
N0050 M3S2546;
N0060 G00X10Z2;
N0070 X12;
N0080 X10;
N0090 G01Z-13.5F10;
N0100 X12;
N0110 G00Z2;
N0120 X8;
N0130 M3S3183;
N0140 G01Z-8.26;
N0150 X10;
N0160 G00Z2;
N0170 X6;
N0180 M3S4000;
N0190 G01Z-6.26;
N0200 X8;
N0210 X12;
N0220 G00Z2;
N0230 G01Z0;
N0240 X4;
N0250 Z-5;
N0260 M3S2829;
N0270 X9Z-10;
N0280 Z-14;
N0290 X12;
N0300 G00Z2;
N0310 M5;
N0320 G28U0W0;
N0330 M30;
- Main program information. —
- Program end. —

The above listing shows an example program using the Denford programming system.

For the program to operate correctly on a genuine FANUC control, the CAD/CAM software information and billet size definitions must be removed from the listing.

2: Sub Program Structure

A program which contains fixed sequences or frequently repeated patterns may be entered into memory as a *Sub Program*, in order to simplify the main program.

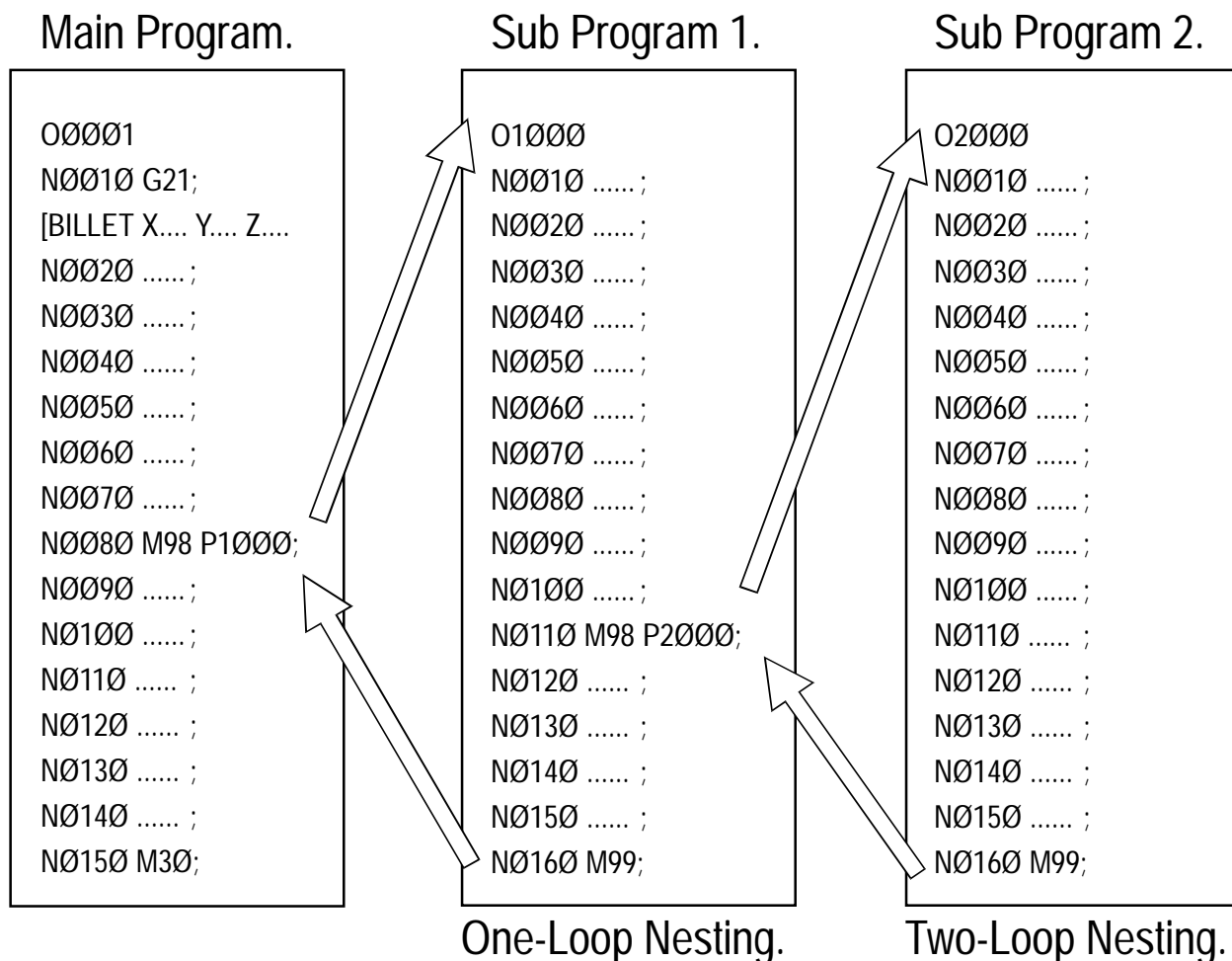
A sub program is entered into the machine controller memory in Edit Mode, in the same manner as the main program.

Differences between a sub and main program :

- 1) A sub program does not have a billet size definition at the top of the program listing.
- 2) A sub program is ended by the M99 code.

The sub program can be called into operation when the machine is set to run in Auto Mode. Sub programs can also call other sub programs into operation.

When the main program calls one sub program into operation, the process is called a one-loop sub program call. It is possible to program a maximum four-loop sub program call within the main program. Shown below is an illustration of a two-loop sub program call.



2: Sub Program Commands - Notes

Note 1.

A sub program must be saved to memory using a four digit number.

Note 2.

If cutter compensation is required on a tool and the co-ordinates for the tool are within the sub program, the cutter compensation must be applied and cancelled within the sub program.

Note 3.

To call a sub program the M98 code is used followed by P0000 (the number of the sub program required).

For example,

```
M98 P2000
```

This command is read call program number 2000.

Note 4.

A sub program call command (M98 P0000) can be specified along with a move command in the same block.

For example,

```
G01 X42.5 M98 P1000;
```

Note 5.

At the end of a sub program, the M99 code is entered. This returns control to the main program.

The M99 code will return control to the next block after the M98 sub program call block in the main program.

If the code M99 P0000 is entered, control will pass to the main program at a block with the N number equal to that of the P number stated after the M99 code.

For example,

```
M99 P0160
```

This command is read return to the main program at block number N0160.

2: Sub Program Repeat Command

A call command can be set to call a sub program repeatedly. This call can specify upto 999 repetitions of a sub program.

A sub program repeat command has the following format:

M98 P0000 0000

Sub program number.

Number of times the sub program is to be repeated.

Call command.

When the repetition is omitted, the sub program will be called once only.

For example,

M98 P100001

This command is read call the sub program number 0001 ten times.

2: Billet Definition

The *Billet Definition* is a feature which is only used in the Denford programming system.

It defines the size of the workpiece billet for use in the simulation sections of the Denford machine software. The billet definition command has no outcome on the actual machining of the part.

The billet definition command is written at the start of the main program. The previous block usually states the units of measurement to be applied to the billet dimensions, ie, G21 (Metric data input) or G20 (Imperial data input).

For example,

```
N0010 G21;
```

```
[BILLET X44.8 Z65.0;
```

These two commands are read.

- program line number 10 states that all units are to be measured in Metric,
- the billet is a bar with a diameter of 44.8mm and a length protruding from chuck/collet face of 65.0mm.

Note 1.

A program that has been written on a Denford control (or using Denford CAD/CAM post processor software) will not operate directly on a genuine FANUC machine. The simulation sections of the program are incompatible with the FANUC control.

For the program to run successfully, lines referring to the CAD/CAM software (at the beginning of the program) and the billet definition block must be deleted.

2: Program Numbering and Saving

The Denford system of program numbering relies on the programmer saving the program to disk or computer hard drive at the time of writing.

When saving a program using the Denford Desktop Tutor keypad, the program number can range from 1 to 99999999.

Writing the program on an offline system with a qwerty keyboard allows the programmer to save the program using letters and/or numbers.

Note 1.

Sub program numbers must be saved between the ranges 0001 to 9999, ie, using a four digit number. It is recommended that all programs saved have filenames also between these numbers.

Note 2.

Before saving a program to disk or hard drive, check that the program name you wish to use has not been used on another file. If the program is saved using a name identical to an old program file, the old program file will be overwritten.

Note 3.

Programs that need to be used at a later date on genuine FANUC controls must have their program number stated on the first program block. The format for inserting a FANUC compatible program number line is as follows:

O 0000
| |
| | Four digit program number.
| |
Address code O.

2: Program Block Numbering

A program is composed of several commands, each command instructing the machine to carry out a particular operation. Each command is a separate line of data within the program, called a *Block*.

One block is separated from another block using an end of block code, ie, effectively signifying the end of a program line. The Denford programming system uses a semicolon (;) as the end of block code.

A four digit sequence number can be specified (0001 - 9999) following the address code N, at the start of each block. The order of these block numbers is arbitrary and need not be consecutive. Block numbers can be specified for every program line, or just on program lines requiring them.

Note 1.

The block number must be written at the start of a program line when used.

Note 2.

It is recommended that all blocks are numbered using a four digit number which rises between each block in steps of 10. This allows the program to be edited at a later date, ie, new blocks can be inserted or deleted as required.

For example,

N 0010 ...

N 0020 ...

N 0030 ...

N 0040 ...

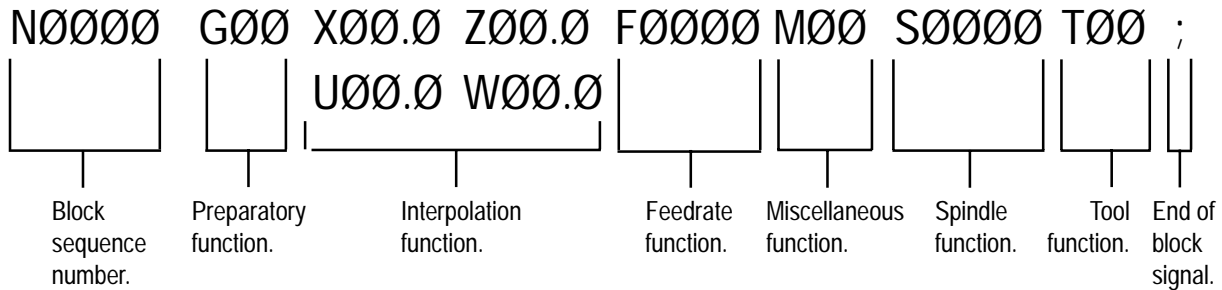
etc....

Note 3.

Even when block numbering is not a priority, it is useful to insert block numbers at important points in the program, such as tool change commands. This will help if a program search is used in the future.

2: Block Configuration

The sequence in which address codes appear in each block should remain consistent throughout the program. It is recommended that the order of these address codes follows the example shown below:



Note 1.

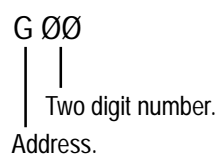
Each block may not necessarily contain all these items.

2: G Codes (Preparatory Functions)

Preparatory functions, called *G codes*, are used to determine the geometry of tool movements and operating state of the machine controller; functions such as linear cutting movements, drilling operations and specifying the units of measurement.

They are normally programmed at the start of a block.

A G code is made from the G address letter and a two digit number as follows,



2: Tool Movement

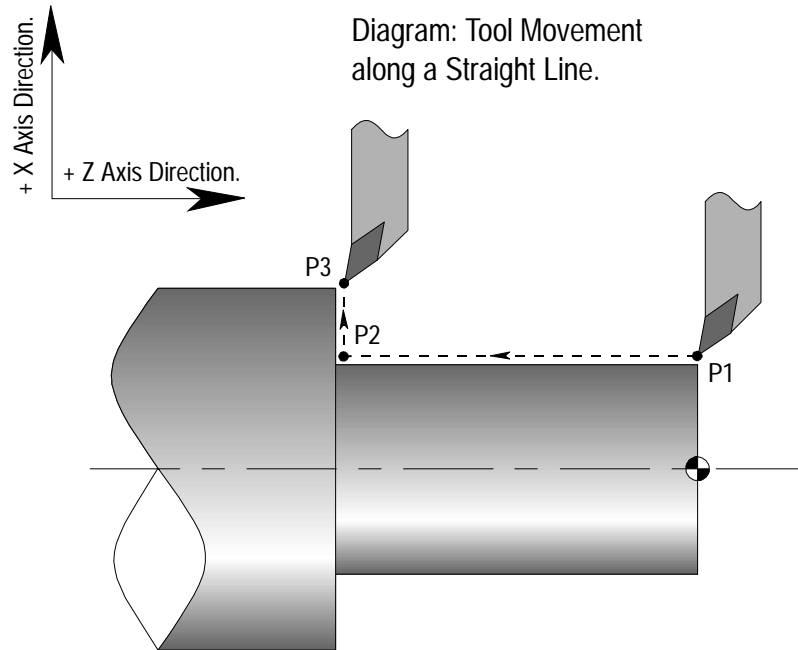
The tool moves along straight lines and arcs forming the workpiece shape.

a) Tool Movement along a Straight Line.

Program command format:

G01 Z _____ ; (P1 - P2)

G01 X _____ ; (P2 - P3)

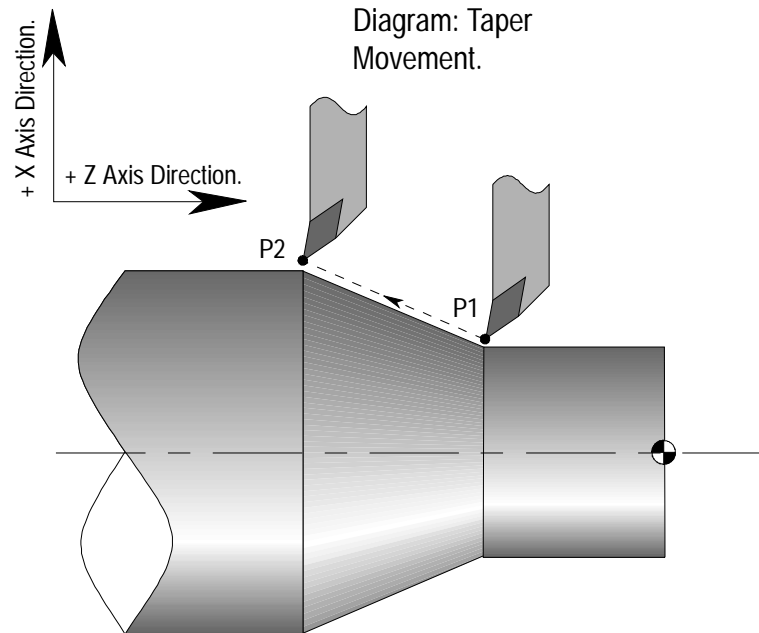


2: Tool Movement

b) Taper Movement.

Program command format:

G01 X _____ Z _____ ; (P1 - P2)



2: Tool Movement

c) Tool Movement along an arc.

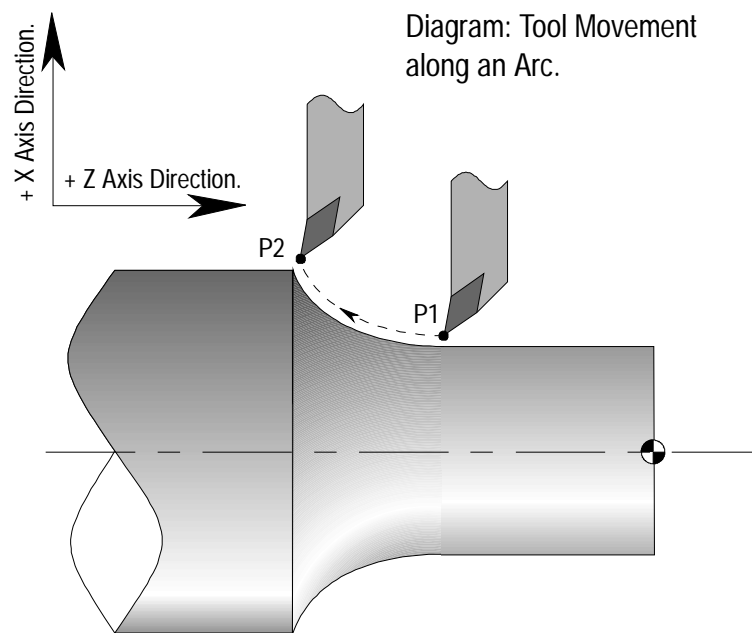
Program command format:

G02 X _____ Z _____ R _____ ; (P1 - P2)

or

G03 X _____ Z _____ R _____ ; (P1 - P2)

The function of moving the tool along straight lines and arcs is called the *Interpolation*. Symbols of the programmed commands G01, G02 and G03 are called the *Preparatory* functions and specify the type of interpolation conducted in the control unit.



2: Thread Cutting

Thread cutting of equal lead straight threads can be cut using G76 or G92 commands.

In G76 and G92, the lead is specified numerically following the address letter F.

2: Feed Function

The movement of the tool at a specified speed for cutting is called the Feedrate. The feedrate is defined using the address letter F followed by a numerical value. The two types of feedrate used are G98 (feedrate per minute) and G99 (feedrate per revolution).

In G20 Imperial units, the feedrate is defined by Inches per minute. In G21 Metric units, the feedrate is defined by Millimetres per minute.

For example:

(G20, G98)

G01 X _____ Z _____ F6 Inches per minute

(G20, G99)

G01 X _____ Z _____ F0.006 Inches per minute

(G21, G98)

G01 X _____ Z _____ F150 Millimetres per minute

(G21, G99)

G01 X _____ Z _____ F0.15 Millimetres per minute

2: M Codes (Miscellaneous Functions)

Miscellaneous functions, called *M codes*, are used by the CNC control to command on/off signals to the machine functions. ie, M03 - spindle forward (CW), M05 - spindle off, etc...

The functions allocated to lower M code numbers are constant in most CNC controls, although the higher M code number functions can vary from one make of controller to the next.

An M code is made from the M address letter and a two digit number as follows,

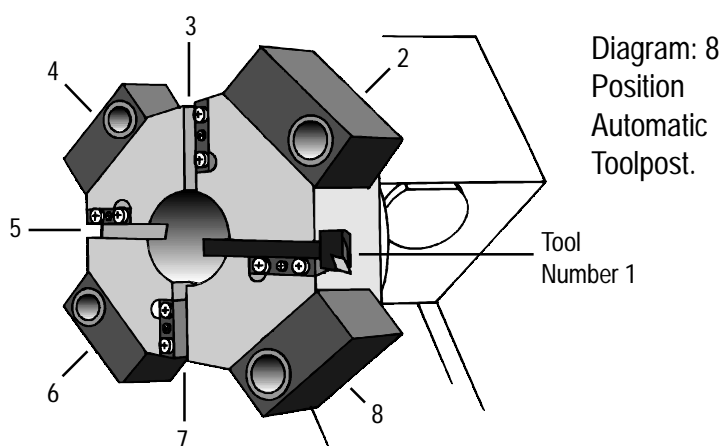
M 00
| |
| |
| Two digit number.
Address.

2: Spindle Speed Function (Cutting Speed)

The cutting speed can be specified using the spindle speed (written in the units, RPM, revolutions per minute) from the G97 code.

The cutting speed can also be set as a constant surface speed. In this case, even in examples such as taper cutting where the workpiece diameter is changing, the CNC changes the spindle speed so that the cutting speed maintains a constant value. This function is the G96 code, called Constant Surface Speed Control.

2: Tool Function



Tool profiles can be changed during a program using the tool function command for an automatic toolpost. By specifying a tool number, a signal is transmitted from the CNC to the machine tool, where upon the toolpost will index to the corresponding tool number.

The tool number is defined using the address letter T followed by a numerical value.

An automatic tool change is defined using the M06 code, called a *Miscellaneous* function, followed by the new tool number required.

For example,

M06 T1 (Tool number 1 and offset number 1)

M06 T01 (Tool number 1 and offset number 1)

M06 T0105 (Tool number 1 and offset number 5)

2: Tool Compensation (Tool Offset)

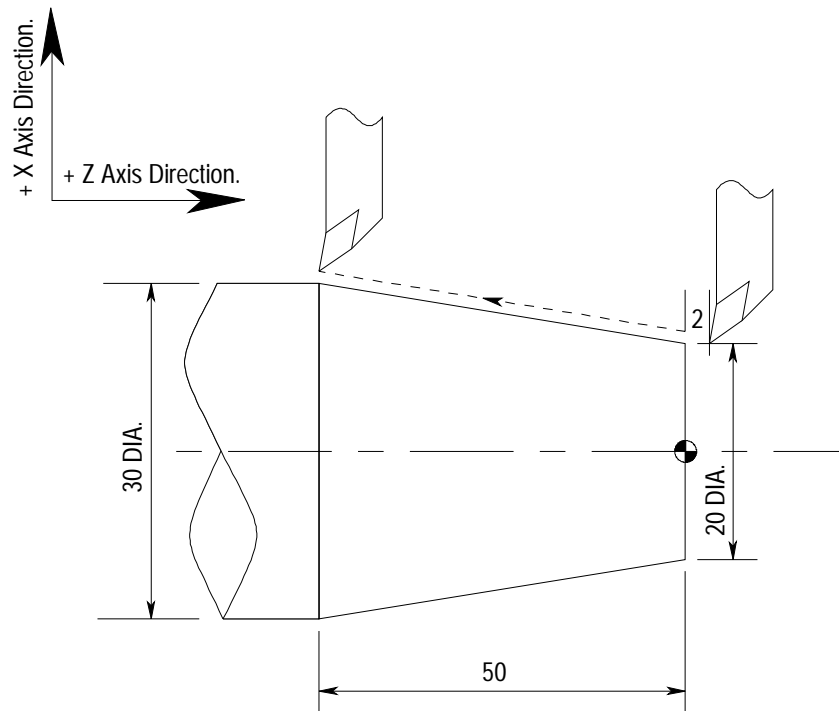
Generally, several different types of tool are required to machine a workpiece, all of different shapes and sizes (called profiles).

It would be very difficult to write a program that allowed for this difference in size between all the various tools. To account for this, the difference between the position of the tool nose and the workpiece datum is measured, in advance, for all the tools that will be used. Essentially, this means that the cutting paths for all the tools now coincide. The values are entered into the offset file.

2: Absolute and Incremental Co-ordinates

The addresses X and Z, within a program, relate to a co-ordinate position from the workpiece datum (the zero position).

The addresses U and W, within a program, relate to the individual axis movements required to reach the new position, from the last position reached by the tool. The address U refers to the X axis and the address W refers to the Z axis.



The example above shows four different ways that can be used to write a block within a program that defines a taper:

Absolute/Absolute

```
G01 X30 Z-50 F0.15 ;
```

Incremental/Incremental

```
G01 U10 W-52 F0.15 ;
```

Absolute/Incremental

```
G01 X30 W-52 F0.15 ;
```

Incremental/Absolute

```
G01 U10 Z-50 F0.15 ;
```

2: Optional Block Skip

When a forward slash mark (/) is followed by a block number (at the beginning of a block) and the block skip switch on the machine operator panel is set to "on", the block will be ignored in memory operation. When the block switch is set to "off", then the blocks indicated by the "/" marks will be considered as valid.

For example,
N3Ø X15 ;
/ N4Ø Z-2Ø ;
/ N5Ø X25 ;
/ N6Ø Z-4Ø ;
N7Ø

If the block skip switch is set to "on" in the above program example, then blocks indicated by the "/" mark are skipped.

Note 1.

A "/" mark must be specified at the start of the block. If it is placed elsewhere in the block, the information from the "/" mark to the ";" mark (the end of block mark) will be ignored, whilst the information before the "/" mark will be effective.

2: Tutorials and Comments

For DOS control software users:

If the program is written off-line with a qwerty keyboard, information relating to the program can be inserted within the program.

Tutorial information appear in the Tutorial dialog box of the machine controlling software (ie, the tutorial message "Now performing thread cutting cycle" could be written to appear when the thread cutting operation starts in the program).

Comments information only appear in the text of the program itself (ie, the comment "Tool 5 is thread cutting tool" could be inserted in the program for use as reference only).

3: G Codes (Preparatory Functions) - Introduction

Preparatory functions, called *G codes*, are used to determine the geometry of tool movements and operating state of the machine controller; functions such as linear cutting movements, drilling operations and specifying the units of measurement.

They are normally programmed at the start of a block.

A G code is made from the G address letter and a two digit number as follows,

G	00
	Two digit number.
Address.	

Modal and Non-modal G codes.

Additionally, G codes are split into two categories -

1) *Modal* (retained) G codes.

A modal G code, once programmed into a block, will affect any subsequent blocks of the program without having to be restated.

Additionally, modal G codes are further split into *groups* according to their task and function. A modal G code will remain active until another G code from the same group is programmed into a block, or it is cancelled.

2) *Non-modal* (one-shot) G codes.

A non-modal G code must be programmed into every block when it is required.

3: G Codes - Introduction Notes

Note 1.

Remember there are two types of G code.

A modal G code is retained in memory - it is effective until another G code from the same modal group is commanded.

A non-modal G code is one-shot - it is effective only within the block in which it is specified.

Note 2.

It is not necessary to enter a modal G code in repetitive blocks within a program.

For example:

If all movements are G01 (linear cutting command) then G01 is entered on the first block and omitted from all subsequent blocks. This G code will remain active until an interpolation change is commanded (using G00, G02 or G03).

Note 3.

The machine controller has the ability to interpret a maximum of four G codes in one single block of data. However, these G codes must be from separate modal groups. When two or more G codes from the same group are specified in the same block, the CNC will only make the last stated G code from that modal group effective.

Notes for G Codes Listing shown right

Note 1.

The G50 code will be a modal (retained) code if it is followed by the letter S.

A G50 code will be a non-modal (one-shot) code if it is followed by X(U) or Z(W).

Note 2.

G codes marked with an * are set at machine power up and when the machine is reset or the emergency stop button is activated.

3: List of G Codes supported by Denford CNC Controls

Note - Not all G codes apply to each machine.

G Code.	Group.	Type.	Function.
G00*	1	Modal	Positioning (Rapid Traverse)
G01	1	Modal	Linear Interpolation (Feed)
G02	1	Modal	Circular Interpolation CW
G03	1	Modal	Circular Interpolation CCW
G04	Ø	Non-modal	Dwell
G20	6	Modal	Inch Data Input
G21*	6	Modal	Metric Data Input
G28	Ø	Non-modal	Reference Point Return
G40*	7	Modal	Tool Nose Radius Compensation Cancel
G41	7	Modal	Tool Nose Radius Compensation Left
G42	7	Modal	Tool Nose Radius Compensation Right
G50	Ø	Non-modal	Work Co-ordinate Change/Max. Spindle Speed setting
G70	Ø	Non-modal	Finishing Cycle
G71	Ø	Non-modal	Stock Removal in Turning - X
G72	Ø	Non-modal	Stock Removal in Facing - Z
G73	Ø	Non-modal	Pattern Repeating
G74	Ø	Non-modal	Peck Drilling in Z Axis
G75	Ø	Non-modal	Grooving in X Axis
G76	Ø	Non-modal	Multiple Thread Cutting Cycle
G81	1	Modal	Deep hole drilling (No FANUC)
G90	1	Modal	Diameter Cutting Cycle A (Outer/Inner)
G92	1	Modal	Thread Cutting Cycle
G94	1	Modal	Cutting Cycle B (End Face Cycle)
G96	2	Modal	Assessed Surface Speed Control
G97*	2	Modal	Assessed Surface Speed Control Cancel
G98*	11	Modal	Feed Per Minute
G99	11	Modal	Feed Per Revolution

Code listing full and correct at the time of printing.

3: G Codes - G00 (Rapid Positioning/ Traverse)

The G00 code executes a non cutting movement, at a rapid feedrate, to a specific co-ordinate position in the working area (absolute co-ordinate movement) or when a certain distance from a previously stated position (incremental co-ordinate movement) is programmed.

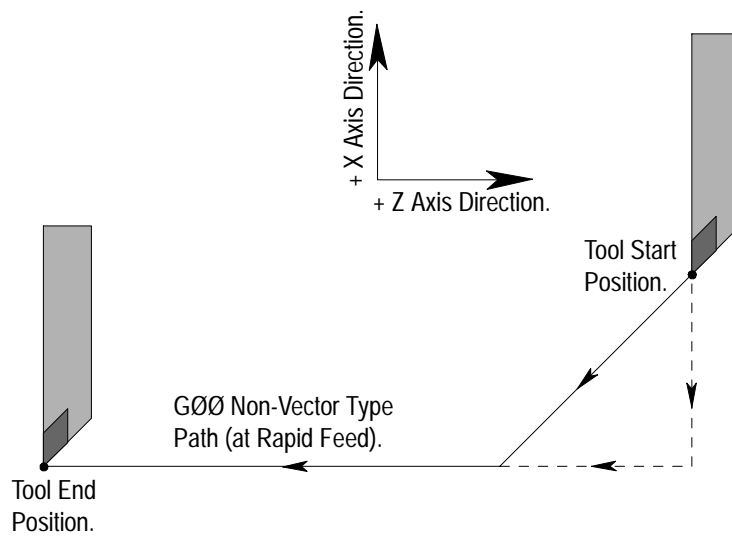
Note 1.

The rate of movement is set by the manufacturer of the machine tool. The rate of movement can be reduced from 100% to 0%, but only in increments of 10%, by using the feed override controls (see specific machine operating manual).

Note 2.

On instruction to perform the G00 command, the two slides (the X and Z axes) move completely independant of each other at a maximum feedrate, along a non-vector (sometimes called a non-linear) type path.

3: G Codes - G00 (Rapid Positioning/ Traverse)



In the above example, the G00 command will instruct both slides to begin moving, both at a maximum feedrate. When both slides begin moving the tool will appear to traverse diagonally, a composite movement of both axes moving together. When one axis reaches its finishing co-ordinate, the other axis will continue to move until it reaches its own finishing co-ordinate. This gives the impression that the tool "changes" direction.

Working example:

Program Block instructing a move to Position P1.

G00 X30.0 Z2.0

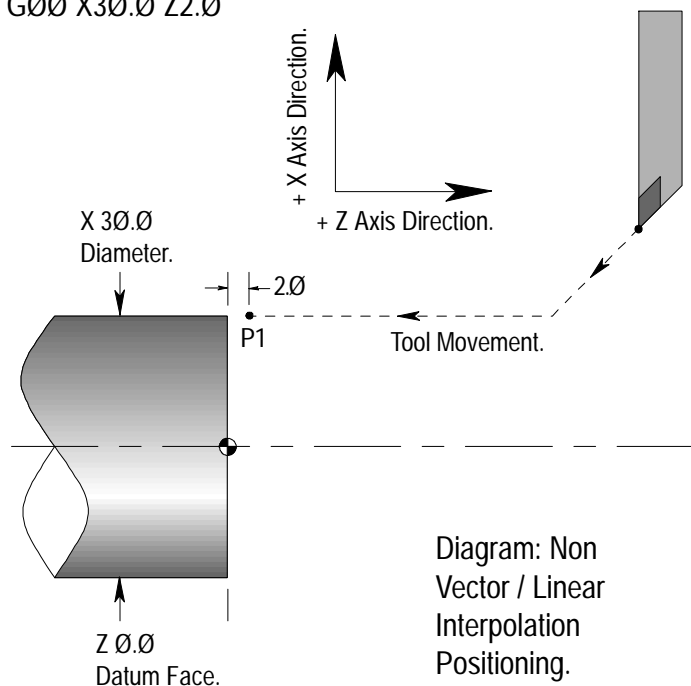


Diagram: Non Vector / Linear Interpolation Positioning.

3: G Codes - G00 (Rapid Positioning/ Traverse)

Note 3.

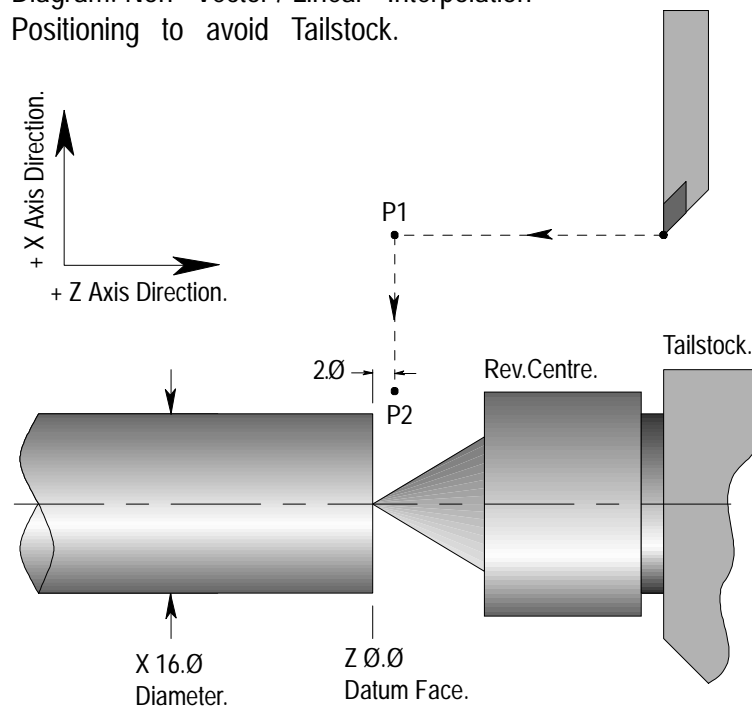
A problem with the non vector interpolation positioning system can occur if a tailstock is used. When the tailstock is set in a forward position, as shown in the diagram below, the tool may collide with the tailstock body or the revolving centre. To overcome this problem, the tool is sent to an intermediate position, P2, in order to clear the tailstock and then onto its final co-ordinate position, P1. This operation is carried out using a two blocks of commands.

Working example:

Part program at this point

```
G00 Z2.0 ;    (move to position P1)
      X16.0;   (move to position P2)
```

Diagram: Non Vector / Linear Interpolation Positioning to avoid Tailstock.



3: G Codes - G00 (Rapid Positioning/ Traverse)

Note 4.

The G00 code freezes the tool radius compensation, codes G41 and G42. If G41 or G42 are active when a G00 command is programmed, the tool radius compensation will not function again until a G01, G02 or G03 command is programmed.

Note 5.

The G00 code is modal and is therefore incompatible with G01, G02 and G03 codes in the same block.

Note 6.

A G00 code can be written into a program in two ways.
G00 or G0.

3: G Codes - G01 (Linear Interpolation)

The G01 code executes a cutting movement following a straight line, at a set feedrate.

A G01 command is written in the following format:

G01	X(orU)Ø.Ø	Z(orW)Ø.Ø	FØØØØ
Linear interpolation code.	X axis co-ordinate (X=absolute, U=incremental).	Z axis co-ordinate (Z=absolute, W=incremental).	F=Feedrate and ØØØØ=feedrate numerical value.

The feedrate value programmed into the G01 command is the actual feedrate along the proposed tool path, not the feedrate of the axis/slide.

On single axis moves (ie, the tool moves exactly parallel to the X axis direction or the Z direction), the slide will feed at the rate stated in the G01 command.

On two axis moves (ie, the tool is moving diagonally), both slides will have to operate exactly the same length of time in order to produce a single diagonal, or vector, move. The machine controller will calculate the separate feedrates for both the X and Z slides, enabling the actual vector feedrate to equal that stated in the G01 command.

The axis co-ordinate moves following a G01 command can be programmed in absolute values (relative to a set datum point) as X and Z, or incremental values (relative to the last stated co-ordinate in the program) as U and W.

It is also possible to program one axis with an absolute value and the other axis with an incremental value (U-Z or X-W).

3: G Codes - G01 (Linear Interpolation)

Note 1.

The example below shows a G01 command being used to produce both straight and diagonal (vector) cutting moves. This can be achieved by programming the data using a variety of methods.

Working example:

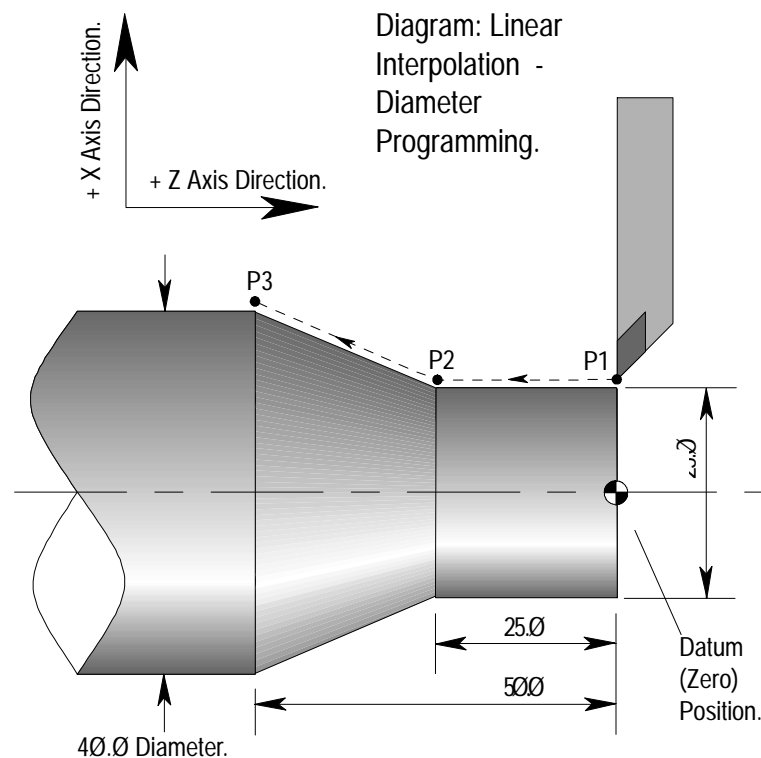
Metric units set previously (G99 mode).

Absolute. — $\left[\begin{array}{l} G01 \quad Z-25 \quad F0.1 ; \\ \quad \quad X40 \quad Z-50 ; \end{array} \right.$

Incremental. — $\left[\begin{array}{l} G01 \quad W-25 \quad F0.1 ; \\ \quad \quad U15 \quad W-25 ; \end{array} \right.$

Absolute and Incremental. — $\left[\begin{array}{l} G01 \quad Z-25 \quad F0.1 ; \\ \quad \quad U15 \quad Z-50 ; \\ \\ G01 \quad W-25 \quad F0.1 ; \\ \quad \quad X40 \quad W-25 ; \end{array} \right.$

In the above programs, both G01 and the F feedrate value are modal, ie, their commands continue onto the next block without having to be restated.



3: G Codes - G01 (Linear Interpolation)

Note 2.

The programmed feedrate can be varied from 0% to 150% by using the feed override controls (see specific machine operating manual).

Note 3.

If no feedrate is programmed on a G01 command, the machine controller will set a feedrate of F0.01 mm per rev. on G99 mode (Feed per revolution), or F10 mm per min. on G98 mode (Feed per minute).

Note 4.

If a block containing G01 and a feedrate F_ is executed and the following block contains a G00 command, the machine controller will retain this F_ value in its memory. The G00 command will usually set the feedrate to its maximum value. On overriding the G00 command with a block containing G01, G02 or G03, the original F_ value will be released from memory and become active.

Note 5.

The G01 code is modal and is therefore incompatible with G00, G02 and G03 codes in the same block.

Note 6.

A G01 code can be written into a program in two ways.
G01 or G1.

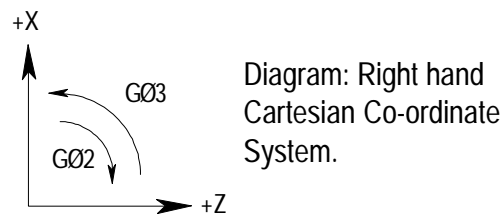
3: G Codes - G02 G03 (Circular Interpolation)

Data to be given	Command	Meaning
Rotation Direction	G02	Clockwise Direction (CW).
Rotation Direction	G03	Counterclockwise Direction (CCW).
End Point Position - Absolute Command	X,Z	Endpoint Position in the Work Co-ordinate Position.
End Point Position - Incremental Command	U,W	Distance from Start Point to End Point.
Distance from Start Point to Centre	I,K	Distance with Direction from Start Point to Arc Centre (always Radius Value and Incremental Value).
Radius of Arc	R	Radius of Arc (always a Radius Value).
Feedrate	F	Feedrate along the Arc.

The G02 code executes a cutting movement following a clockwise circular path, at a set feedrate.

The G03 code executes a cutting movement following an anticlockwise circular path, at a set feedrate.

The definitions of clockwise (G02) and anticlockwise (G03) are fixed according to the system of co-ordinates in the diagram below.



When programming arcs, the end point of the arc is specified by the address X, Z when stated as an absolute position, or U, W when stated as an incremental position. This arc end point is sometimes referred to as the target position.

Using an absolute position (X, Z), the value is the dimension of the end point of the arc in relation to the datum position of the component.

Using an incremental position (U, W), the co-ordinate of the end point is the distance the tool moves from the start position of the arc. This can either be a plus (+) value or a minus (-) value depending on the movement of the machine slides in relation to the start position.

3: G Codes - G02 G03 (Circular Interpolation)

The format to program a circular interpolation in cartesian co-ordinates using a known radius for the arc is as follows :

To follow a clockwise circular path,

Absolute G02 X Z R

Incremental G02 +/-U +/-W R

To follow an anticlockwise circular path,

Absolute G03 X Z R

Incremental G03 +/-U +/-W R

where,

G02 or G03 defines the direction of the circular interpolation.

X or +/-U defines the co-ordinate value of the arcs end point along the X axis.

Z or +/-W defines the co-ordinate value of the arcs end point along the Z axis.

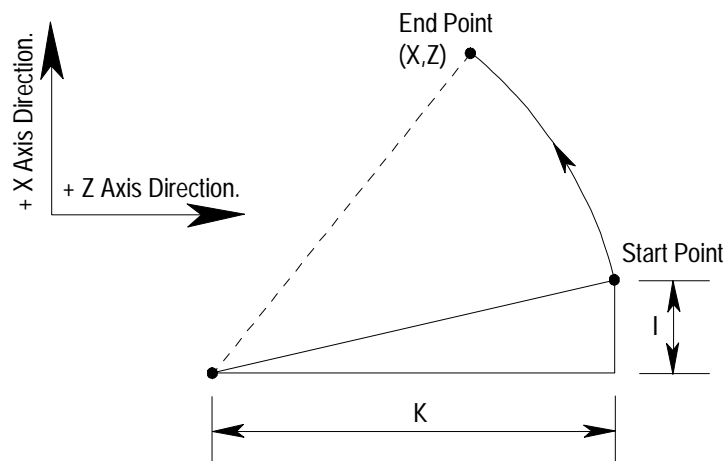
R defines the length of the arcs radius.

I and K

To program an arc when only the arc centre is given (the radius is unknown) use the address letters I and K.

I relates to the address X and is the incremental value and direction (+/-) from the start point of the arc in the X axis to the arc centre (see diagram below).

K relates to the address Z and is the incremental value and direction (+/-) from the start point of the arc in the Z axis to the arc centre (see diagram below).

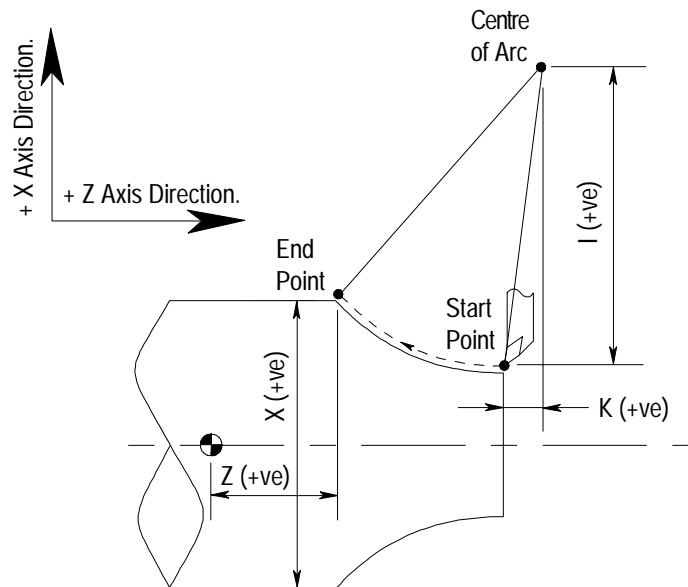


3: G Codes - G02 G03 (Circular Interpolation)

G02 X ___ Z ___ I ___ K ___ F ___ ;

(Absolute programming)

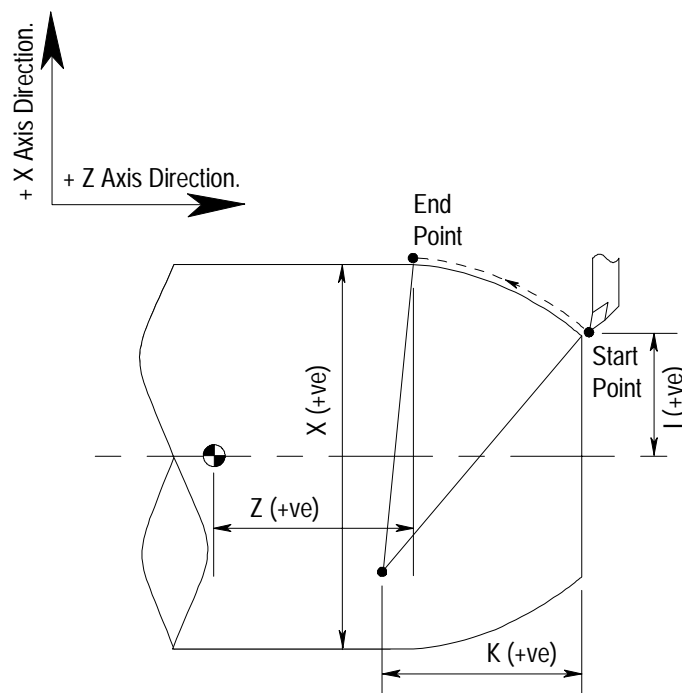
(Diameter programming)



G03 X ___ Z ___ I ___ K ___ F ___ ;

(Absolute programming)

(Diameter programming)

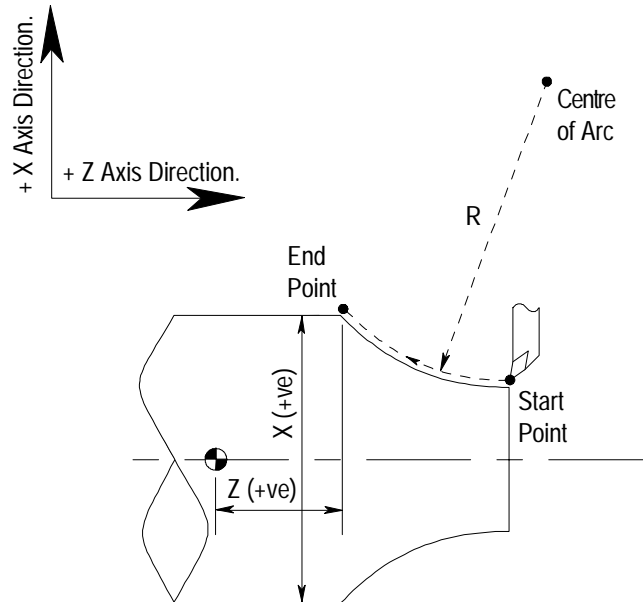


3: G Codes - G02 G03 (Circular Interpolation)

G02 X ____ Z ____ R ____ F ____ ;

(Absolute programming)

(Diameter programming)



Note 1.

When programming arcs using the address R (arc radius), the value of R must be equal to, or greater than half the longest distance travelled by either axis.

Note 2.

For an arc greater than 180 degrees, two separate arcs have to be programmed.

Note 3.

A G02 code can be written into a program in two ways.

G02 or G2.

A G03 code can be written into a program in two ways.

G03 or G3.

3: G Codes - G04 (Dwell)

The G04 code is used to enter a set time delay into the program.

The dwell value is programmed using the address letters X (time in seconds), U (time in seconds) or P (time in 1/1000 seconds), followed by a number indicating this dwell value.

For example:

G04 X1.0 ;

This command is read perform a dwell of 1 second duration.

G04 U1.5 ;

This command is read perform a dwell of 1.5 seconds duration.

G04 P2000 ;

This command is read perform a dwell of 2 seconds duration.

Note 1.

A decimal point cannot be used with the address P.

Note 2.

The dwell is performed at the start of the block in which it is programmed.

Note 3.

The dwell begins when the commanded feedrate of the previous block reaches zero. For the slides to remain motionless in terms of revolutions, the current r.p.m. must be known and the length of the dwell based upon that speed.

Note 4.

The maximum value of a dwell time is 9999 seconds.

Note 5.

G04 is a non-modal G code. It is only active in the block in which it is programmed.

Note 6.

A G04 code can be written into a program in two ways.

G04 or G4.

3: G Codes - G20 G21 (Inch/Metric Data Input)

The machine controller can be programmed in either Inch unit input (G20) or Metric unit input (G21). The standard format for a CNC part program is to write the G20 or G21 code in the first block of the program.

G code. Type. Units. Lowest input value.

G20 Inch Inch 0.0001 inch

G21 Metric Millimetre 0.001 mm

The unit systems of the following items are changed depending on whether G20 or G21 is set.

- 1) Positioning commands (X and Z).
- 2) Incremental movement distances.
- 3) Feedrates commanded by the F code.
- 4) Offset values.

Note 1.

The status of G20 or G21 in the machine controller is dependant on the option that is saved to the disc.

Note 2.

A G20 code must not be changed for a G21 code (or vice versa) during the program.

Note 3.

When switching between G20 and G21, the offsets must be set according to the units of measurement being used.

Note 4.

G 20 and G21 are both modal G codes within the same modal group.

3: G Codes - G28 (Reference Point Return)

The reference point is a fixed position on the machine, to which the tool can be moved.

On machines fitted with Denford turning software, this point is also used as the Home position, the point used by the machine to set the limits of movement for the X and Z slides.

A G28 code instructs the tool to automatically move to this reference point.

A G28 command is written in the following format :

```
G28 X(U) Z(W)
```

X(U) and Z(W) can be used to indicate an intermediate point, through which the tool will pass, before continuing to the reference point.

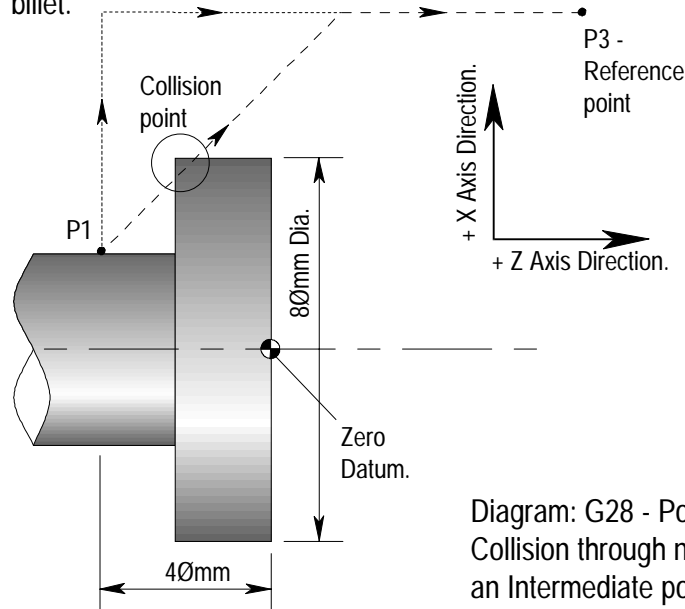
This intermediate point allows the tool to be programmed to follow a more "predictable" path, keeping it sufficiently clear from any part of the machine or billet it could hit when moving to the reference point.

The move to any intermediate point and the reference point are performed at a rapid traverse rate, using a non-vector (non-linear) type path, ie, the tool may appear to "change" direction due to the non-vector type positioning being used.

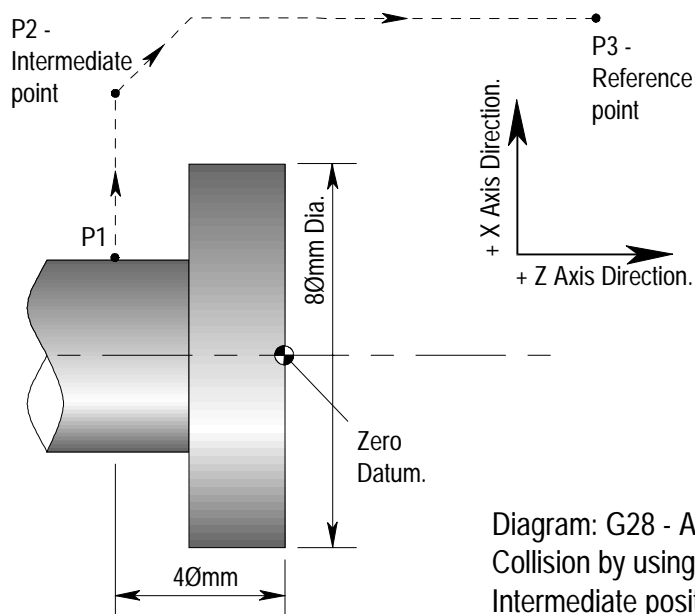
3: G Codes - G28 (Reference Point Return)

Note 1.

The diagram below shows how the tool could collide with the billet when manoeuvring towards the reference point. This is a result of the non-vectorised movements forcing the tool to follow a path which "cuts" through the edge of the billet.



To avoid this collision, the tool is sent on a path which includes the additional, or intermediate, point P2. The intermediate point is used to allow the tool to move completely clear from the billet, before continuing onto the reference point, P3, shown below.



Working example:

Block required to perform this operation would read,

```
G28 X100.0 Z-40.0 ;
```

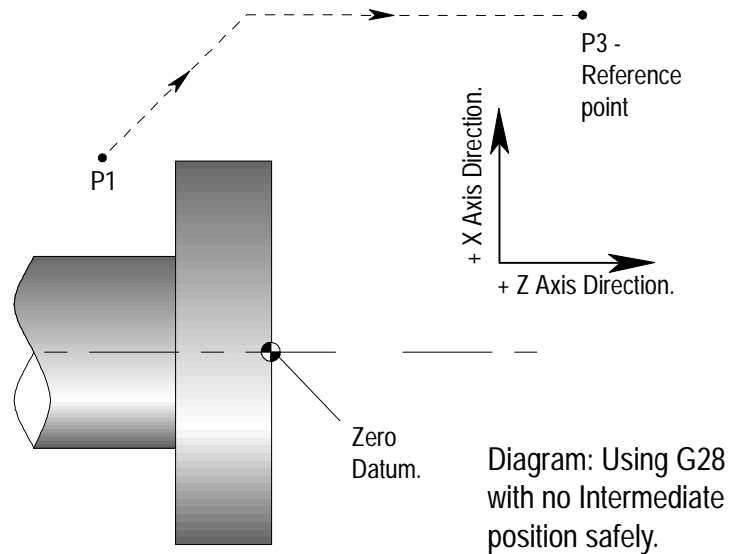
3: G Codes - G28 (Reference Point Return)

Note 2.

In the diagram below, the tool is in a position (P1) where no collision is possible. The intermediate point, in this case, is not required, so the block can be written as follows:

G28 U0 W0 ;

The intermediate point co-ordinates are still stated, using the incremental addresses U and W but with both their values set to zero, indicating no axis movement. Therefore, the tool will move from point P1 to the reference point, P3, along a non-vector type path.

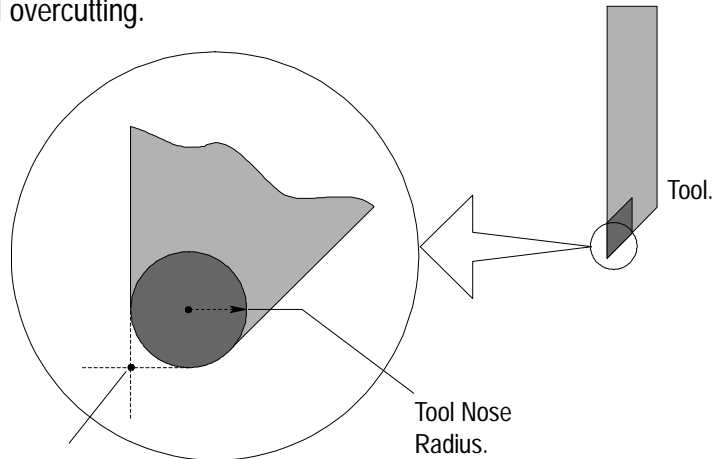


Note 3.

G28 is a non-modal G code. It is only active in the block in which it is programmed.

3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

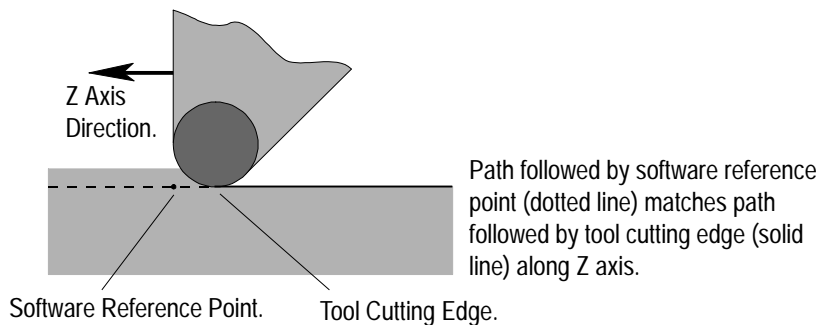
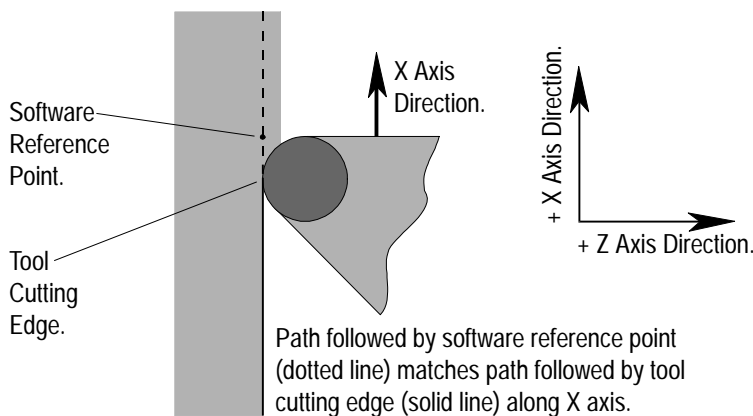
The collection of G40, G41 and G42 codes allow the machine controller to produce arcs and tapers on the billet, very accurately, by automatically accounting for the radius shape on the tool nose. Without tool nose compensation, the profile cut by the tool nose would be subject to undercutting and overcutting.



Software Reference Point (Imaginary Tool Nose) - This is the point used by the software to follow the programmed path line.

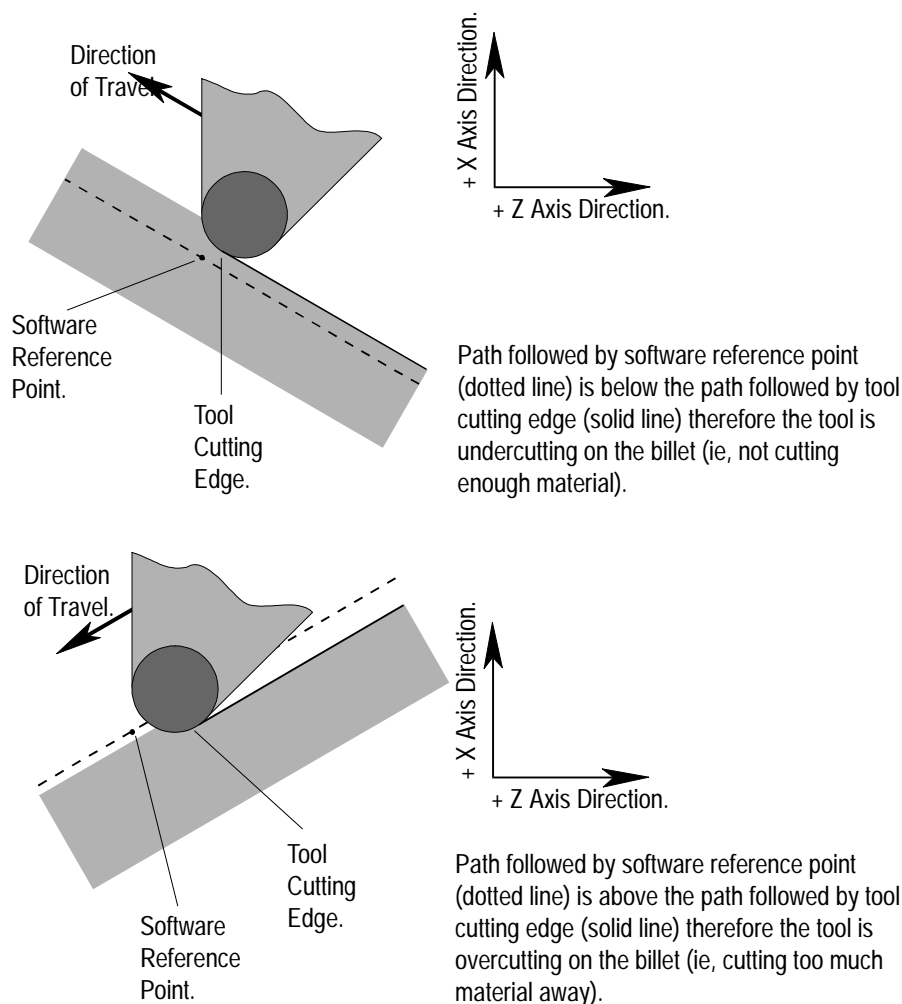
Diagram: Enlarged view showing Tool Nose Radius.

Tool nose radius compensation is not required if the tool cuts parallel to the X and Z axes, as shown below. The software reference point and the cutting edge of the tool both follow the same path.



3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

When the tool starts to cut diagonally, creating tapers and arcs, the tool with either undercut or overcut, according to the direction in which it is travelling, as shown in the two diagrams below.



Imaginary Tool Nose.

The Software Reference point on the tool is also referred to as the Imaginary Tool Nose.

Its main purpose is to simplify the process of setting offsets by allowing the tool be touched onto known points on the X and Z axes.

When cutter compensation is used, the CNC control calculates the displacement required between the imaginary tool nose and the actual cutting edge of the tool. This makes the tool follow the contour path set in the part program more accurately. This distance will vary according to the direction and angle of tool movement and is also based upon the radius of the tool tip, as shown above.

3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

Tool Radius.

To find the correct amount of nose radius for the tool tip, refer to the labelling on the original tool delivery box.

Below is an example of a tool tip code box, from ISO 1832 - 1985 :

C	N	M	G	12	Ø4	12	F	N
1	2	3	4	5	6	7	8	9



In the above tip code box, the amount in box number 7, translates to a radius of 1.2mm.

3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

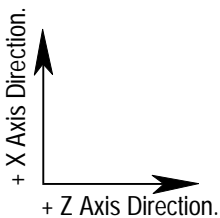
Direction of Imaginary Tool Nose.

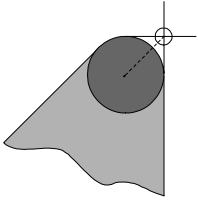
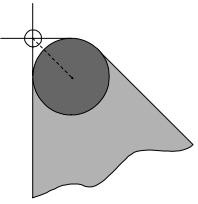
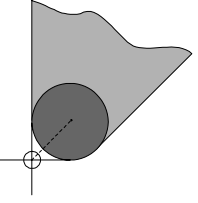
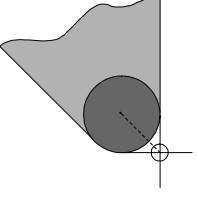
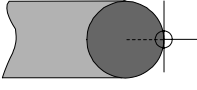
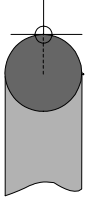
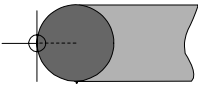
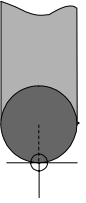
The direction of the tool nose refers to its orientation in respect to the billet position, ie, which way the tool is pointing... for example, up and towards the left. Each tool direction has a number, from 0 - 9, entered into the Tool Offsets table next to the letter "T". In addition to "T", the tool length (X and Z offset) must also be entered together with a tool tip radius value "R".

The table below shows the tool directions for numbers 1 - 8.

Table showing Tool Nose Direction Numbering.

Axes Direction for Tool Nose Numbering Table, shown right.



 <p>Imaginary Tool Nose No. 1 (Left hand Boring)</p>	 <p>Imaginary Tool Nose No. 2 (Right hand Boring)</p>
 <p>Imaginary Tool Nose No. 3 (Left hand Turning)</p>	 <p>Imaginary Tool Nose No. 4 (Right hand Turning)</p>
 <p>Imaginary Tool Nose No. 5</p>	 <p>Imaginary Tool Nose No. 6</p>
 <p>Imaginary Tool Nose No. 7</p>	 <p>Imaginary Tool Nose No. 8</p>

3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

Entering the Tool Nose Radius, Direction and Length values.

The machine controller requires three main pieces of information, in order to calculate the cutter compensation for each tool. These three pieces of information, entered into the tools offset table, are:

- 1) X and Z, referring to the tool offset values (accounting for the different length of each tool).
- 2) R, referring to the tool nose radius.
- 3) T, referring to the direction of the tool nose (the tool orientation).

A maximum of 16 complete offset values can be listed in the table.

The table below shows an example set of offsets.

	X Length Offset	Z Length Offset	Tool Radius	Tool Direction
Tool	X	Z	R	T
1	-138.06	-79.454	Ø.4	3
2	-121.85	-83.185	Ø.8	2
–	–	–	–	–
–	–	–	–	–
15	-125.45	-50.335	Ø.Ø	Ø
16	-127.35	-75.252	Ø.4	4

Note

From the eight tool direction numbers available to indicate the orientation of the imaginary tool nose, the numbers 2, 3 and 4 correspond to:

- 2 - Boring bar cutting towards the chuck.
 - 3 - Standard right hand knife tool cutting towards the chuck.
 - 4 - Standard left hand knife tool cutting away from the chuck.
-

3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

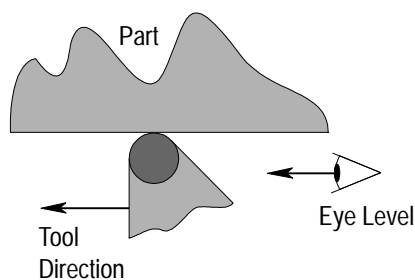
Work Position and Movement Command.

When tool nose radius compensation is required in a CNC program, the position of the billet in respect to the tool must be specified using the table below.

G code	Direction	Tool Path
G40	Cancel	Movement along programmed path
G41	Left hand	Movement on the left hand side of the programmed path
G42	Right hand	Movement on the right hand side of the programmed path

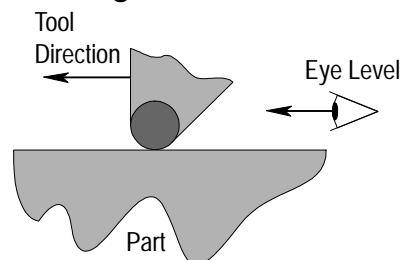
The two diagrams below illustrate the direction of compensation codes G41 and G42, in relation to your eye level.

G41 - Left Hand



The tool is positioned on the left hand side of the part, as seen following the direction of movement, from behind the tool.

G42 - Right Hand

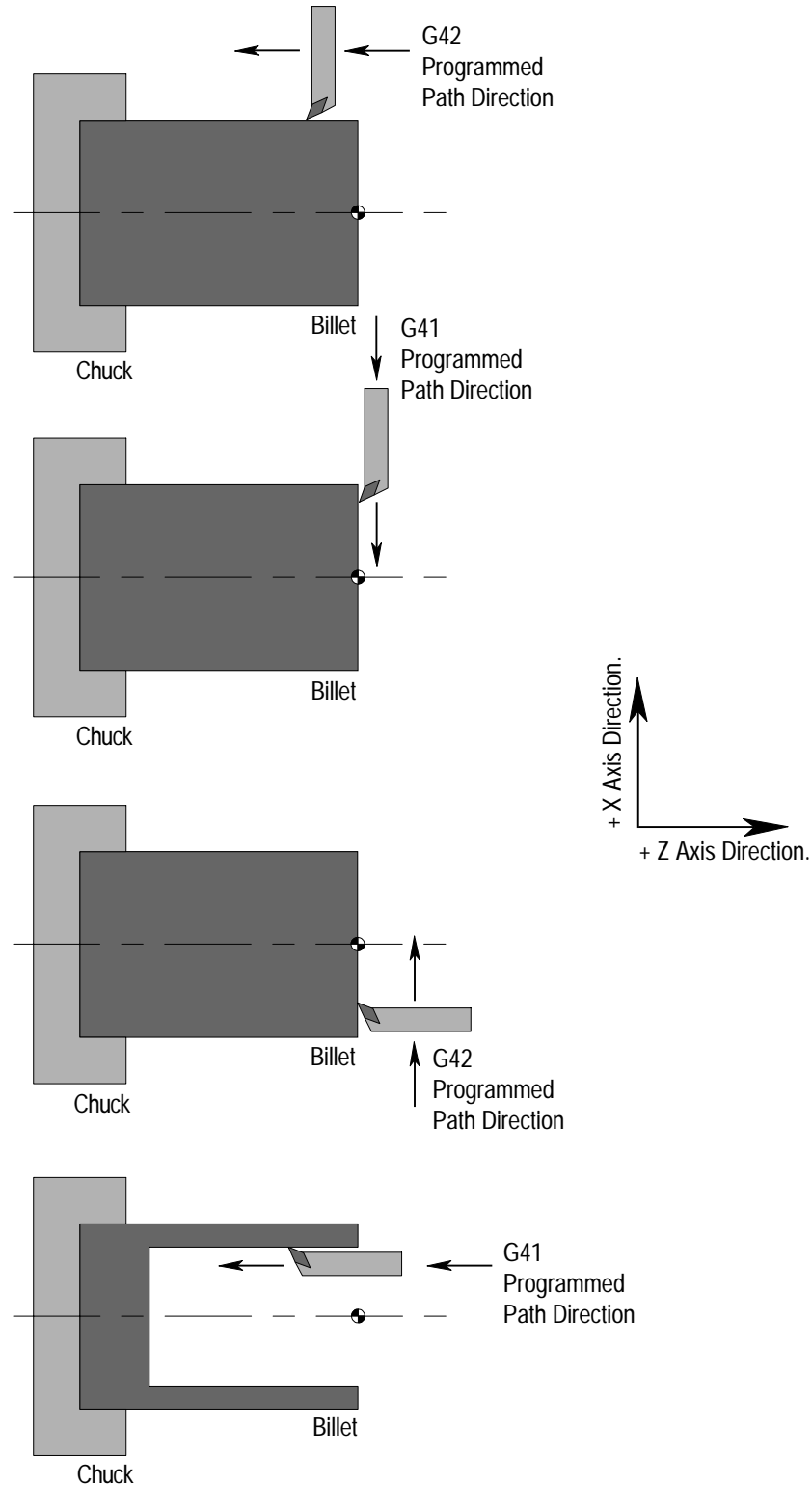


The tool is positioned on the right hand side of the part, as seen following the direction of movement, from behind the tool.

3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

Work Position and Movement Command.

The four diagrams below illustrate the direction of compensation codes G41 and G42, in working situations.



3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

Tool Nose Radius Compensation Start-up (G41 - G42).

The operation instructing a machine to switch to tool nose radius compensation mode is called the *start-up* block, or *ramping on* block. The start-up block is used to allow the tool time to change from moving along the programmed path line to following either side of the programmed path line.

The start-up block should satisfy the following points :

- 1) A G41 or G42 code must be contained in the block, or specified in the previous block.
- 2) A G01 X... Z... move is specified in the block and the distance of the linear move must be greater than the tool nose radius.
- 3) The tool nose radius value, "R", entered into the tool offsets table must not be 00.
- 4) The correct tool direction (orientation) number, "T", must be entered into the tool offsets table.

Note 1.

A G02 or G03 circular interpolation command cannot be specified in the start-up block.

Note 2.

In tool nose radius compensation start-up, two blocks are read into the CNC control. The first block is performed and the second block is entered and held in memory.

In subsequent compensation moves, two blocks are read in advance, so the CNC control has the block currently being performed and the next two blocks in memory.

This is because tool nose radius compensation always needs to know what happens in the move following the one being currently performed. The CNC control can plan ahead to calculate the correct end position for the current move, that will also be the correct start position allowing for cutter compensation, for the next move.

Note 3.

The codes G40, G41 and G42 are modal, belonging to the same modal family. They are incompatible with each other on the same block.

3: G Codes - G40 G41 G42 (Tool Nose Radius Compensation)

Cancellation of Tool Nose Radius Compensation (G40).

The G40 code is used to cancel Tool nose radius compensation.

A G40 command can only be performed in a block in which a linear move (ie, G00, G01, G28) is programmed.

Following a radius or taper performed in compensation mode, it is recommended that the two next blocks both contain straight (linear) moves before a G40 cancellation code can be programmed.

These straight (linear) moves need only be slightly larger than the tool nose radius.

The CNC control enters compensation cancel mode automatically when :

- 1) the machine power is first switched on.
- 2) the reset button on the CRT/MDI controller panel is pushed.
- 3) a program is forced to end by performing an M02 or M30 command.

The following part program for a finishing pass shows the recommended method for start-up and cancellation:

```
[BILLET X32 Z55  
O 0048  
N0010 G99 G97 G21 S3000 M03 ;  
N0020 M06 T0505 ;  
N0030 G00 X33 Z2 ;  
N0040 G01 Z00 F0.3 ;  
N0050 X-1 F0.1 ;  
N0060 G00 X12 Z1 ;  
N0070 G01 G42 X10 Z0 F0.2 ; (Start-Up)  
N0080 G03 X15.86 Z-2.355 R3.0 F0.05 ;  
N0090 G01 X29 Z-32 F0.1 ;  
N0100 Z-50 ;  
N0110 X33 ;  
N0120 G28 G40 U0 W0 ; (Cancellation)  
N0130 M30 ;
```

3: G Codes - G5Ø (Multiple Command Functions - 1. Clamping Max. Spindle Speed)

The G5Ø code is interpreted by the machine controller in three different ways, according to the wording that follows G5Ø. These commands are:

- 1) Clamping Maximum Spindle Speed.
- 2) Co-ordinate System Setting.
- 3) Work Co-ordinate System Shift.

1) Clamping Maximum Spindle Speed.

The figure following G5Ø S_ _ _ _ specifies the maximum spindle speed for constant surface speed control, measured in r.p.m.

When the spindle speed in the constant surface speed control reaches the figure specified by the program (G5Ø S_ _ _ _), the spindle is clamped and this speed becomes the maximum value.

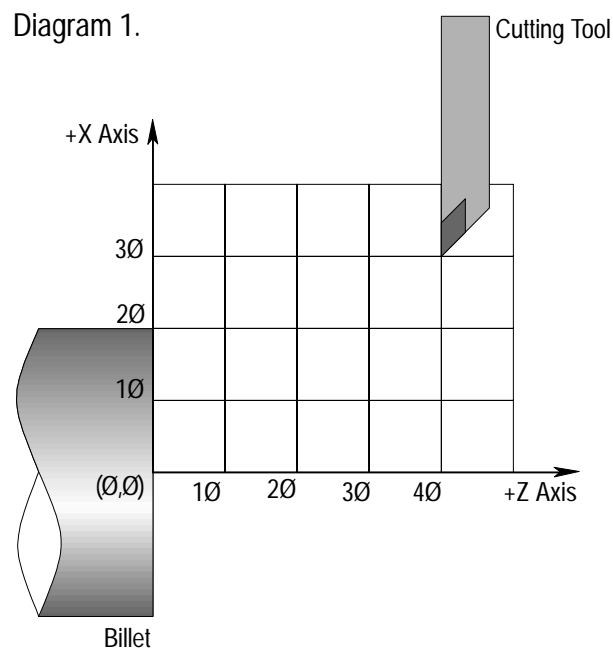
3: G Codes - G50 (Multiple Command Functions - 2. Co-ordinate System Setting)

2) Co-ordinate System Setting.

The G50 code can be used to shift the X and Z axes along to a different position, as if they were an "imaginary" piece of graph paper.

This is done by setting a specific co-ordinate value that you want the machine controller to register as the current position of the tool. The tool will then "think" it is at this specific co-ordinate, although it has not actually moved.

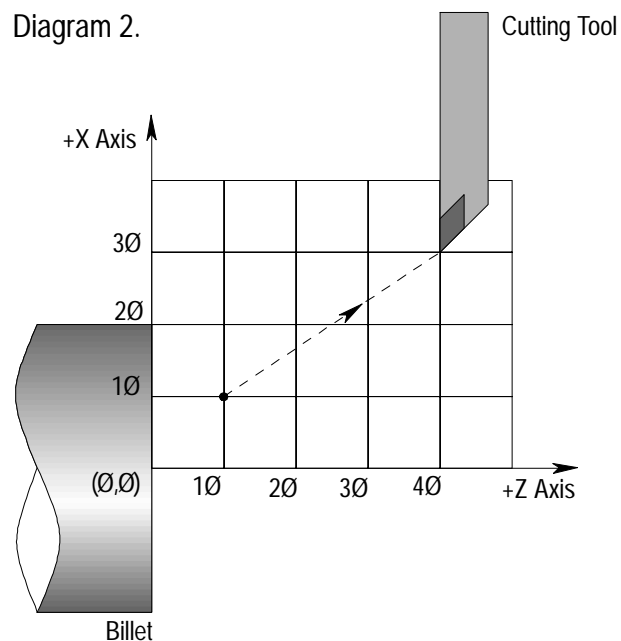
Recap - it is the grid positions on the "imaginary" graph paper for the X and Z axes that move, not the actual tool.



In diagram 1, the zero datum is set at the face end of the billet, along its centreline, simply to make the diagrams and graphs easier to follow. Note that the zero datum may be set at another position on your machine.

The tool is currently at the position Z=40, X=30. Position (0,0) is the zero datum.

3: G Codes - G50 (Multiple Command Functions - 2. Co-ordinate System Setting)



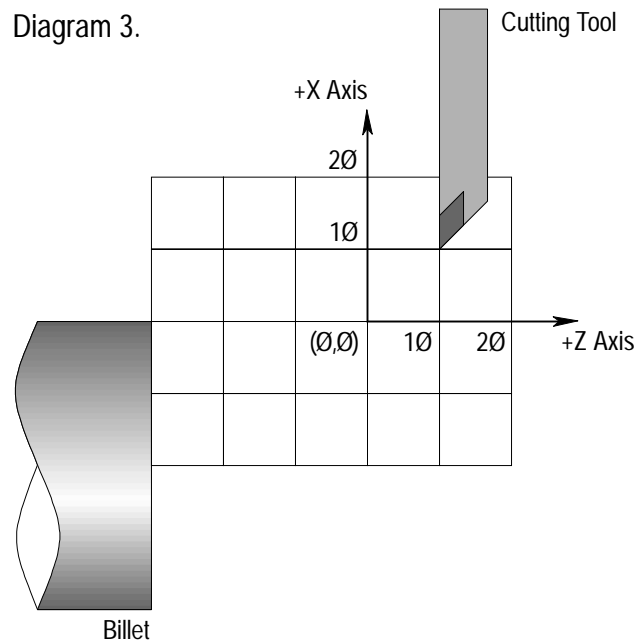
In diagram 2, a command is programmed instructing the entire co-ordinates grid (the graph paper) to shift so that the tool position will line up with the co-ordinates Z=10, X=10.

Note that the command instructs the grid to move, not the tool.

G50 X10 Z10

This command is read move the entire co-ordinates grid so that the position Z=10, X=10 is now at the current position of the tool.

3: G Codes - G50 (Multiple Command Functions - 2. Co-ordinate System Setting)



In diagram 3, the tool position now reads $Z=1Ø$, $X=1Ø$, even though the tool is still exactly the same distance away from the billet as it was in diagram 1.

Recap :

General G50 command layout for co-ordinate system setting using absolute values :

G50 X..... Z.....

where,

X and Z values represent the required tool position co-ordinates.

This command is read set the current tool position to the X..... and Z..... values.

All future absolute dimensions are now relative to a "new" zero datum position.

3: G Codes - G50 (Multiple Command Functions - 3. Co-ordinate System Shift).

3) Co-ordinate System Shift..

The G50 code can also be used to shift the co-ordinate axes (grid) using incremental values (U and W) rather than absolute (X and Z) values.

General G50 command layout for co-ordinate system setting using incremental values:

G50 U..... W.....

This command is read instruct the co-ordinate grid to move so that the tool position becomes X +/-U and Z +/-W.

3: G Codes - G70 (Finishing Cycle)

After part profile rough cutting has been completed, using the G71, G72 or G73 codes, the G70 code can be used to perform a finishing cut/pass.

A G70 finishing pass command is written in the following format:

G70 P ns Q nf

where,

ns = The sequence number of the start block for the finishing pass.

nf = The sequence number of the last block for the finishing pass.

Note 1.

The two functions feedrate, F, and spindle speed, S, commanded between the sequence numbers ns and nf will only be effective in the G70 operation.

Note 2.

When the finishing pass is ended by reading the nf sequence number, the tool returns to the position it was originally at before the G70 was commanded.

Note 3.

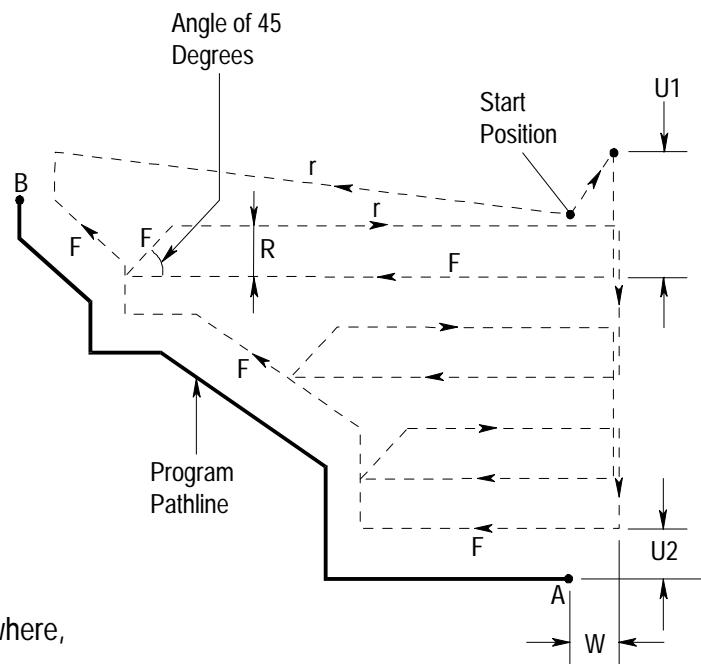
In the blocks between the sequence numbers ns to nf, no sub programs can be called into operation.

3: G Codes - G71 (Stock Removal in X Axis)

The G71 code commands a multiple repetitive cycle, sometimes referred to as a canned cycle.

Canned cycles are used within a CNC program to simplify programming, since only the dimensions describing the required component profile are required. The CNC control will then generate the roughing cuts needed to make this component profile, from within its own memory.

Below is a simplified diagram of the Movements of a G71 canned cycle :



where,

R = Rapid Traverse

F = Feed

U1 = Depth of cut in X axis

U2 = Finishing allowance in X axis

W = Finishing allowance in Z axis

3: G Codes - G71 (Stock Removal in X Axis)

A G71 command is written in the following format:

G71 U (1) R _ ;

G71 P *ns* Q *nf* U (2) W _ F _ ;

where,

U (1) = Depth of cut in the X axis (Radius value).

R = Escaping amount (Retract).

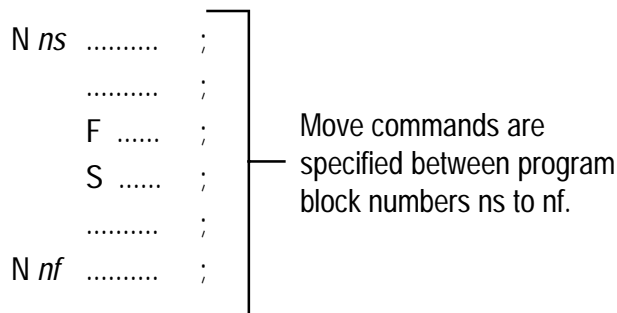
ns = Sequence number of the first block of the programmed finished shape.

nf = Sequence number of the last block of the programmed finished shape.

U (2) = Distance and direction of the finishing allowance in the X axis (Diameter value).

W = Distance and direction of the finishing allowance in the Z axis.

F = Feedrate for Roughing.



Note 1.

Any F (Feed rate) or S (Spindle speed, rpm) commands between *ns* to *nf* block numbers will only be effective on the finishing pass (G70).

Note 2.

Prior to writing the command G71 within the program, the tool is positioned at the start point.

3: G Codes - G71 (Stock Removal in X Axis)

In the preceding diagram, the sequence of tool moves in the G71 cycle is as follows:

Move 1 - From the start position, the tool retracts away from the start point. The distance it moves equates to U2 along the X axis and W along the Z axis, at a rapid feed rate.

Move 2 - The tool moves the distance U1, along the X axis towards the billet centreline, at a feedrate corresponding to the G code in the *ns* block (G01 or G00).

Move 3 - The tool moves along the Z axis towards the spindle, to a point that the CNC control has calculated in relation to the finished profile. The feedrate used is stated in the second block of G71.

Move 4 - At this point the tool retracts in the X and Z axes away from the work, at an angle of 45 degrees. The escaping amount (ie, the retracting distance), R, is stated in the first block of the G71. A rapid traverse feedrate is used.

Move 5 - The tool then retracts along the Z axis, at a rapid feedrate until it reaches the start point of that same diameter.

Move 6 - From this point, move numbers 2, 3, 4 and 5 are repeated by the CNC control unit, until the programmed shape between *ns* and *nf* has been roughed out.

Move 7 - After all the roughing passes are completed (ie, a rough component shape is visible), the CNC control will perform a single roughing profile pass at the G71 stated feedrate. When this single pass is complete the tool will retract to the original starting position of the G71. The next block will then be read into the CNC control.

Move 8 - If the next block contains G70 P *ns* Q *nf* , the CNC control will use the same tool to perform a finishing pass at a feedrate and spindle speed contained between *ns* and *nf*.

Should a different tool be required to perform the finishing pass, the block after *nf* would instruct the machine to move to a tool change position. The next block would instruct the machine to change to the tool required for the finishing pass. The next block would move the tool back to the original G71 starting position. At this point, the finishing pass block (G70 P *ns* Q *nf*) would be read into the CNC control. When the finishing pass ends, the tool will retract to the original G71 start position.

3: G Codes - G71 (Stock Removal in X Axis)

Note 1.

In canned cycle G71, only an X axis move is permitted in the first block of the finished profile (the *ns* block).

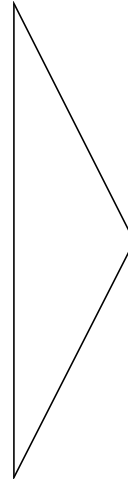
Note 2.

Only G00, G01, G02, G03 and G04 can be programmed between *ns* and *nf* blocks.

Four cutting patterns can be used with G71:

- 1) Outside turning tool moving towards the spindle.
- 2) Outside turning tool moving away from the spindle.
- 3) Internal turning tool moving towards the spindle.
- 4) Internal turning tool moving away from the spindle.

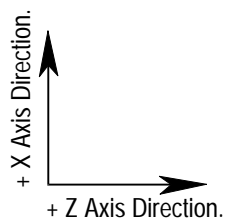
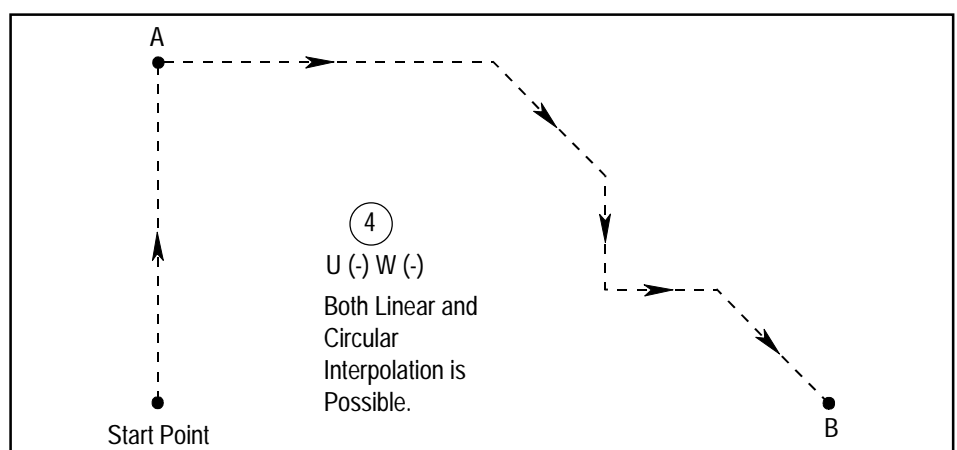
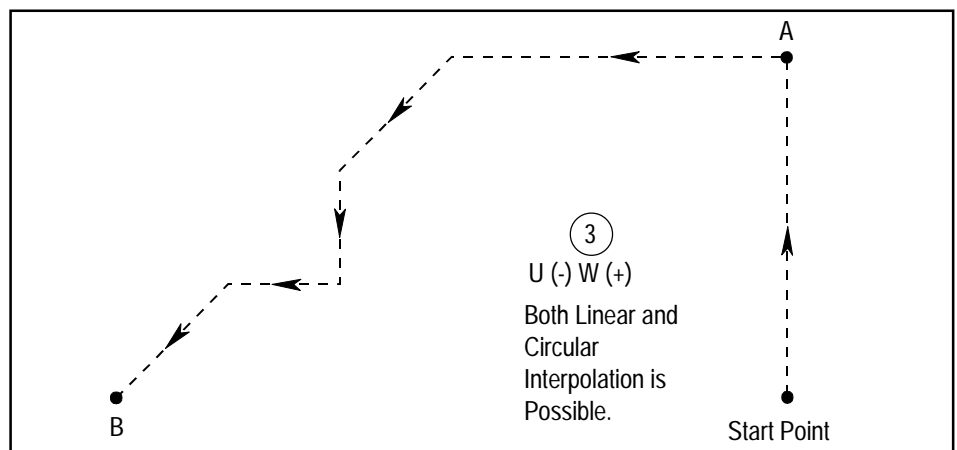
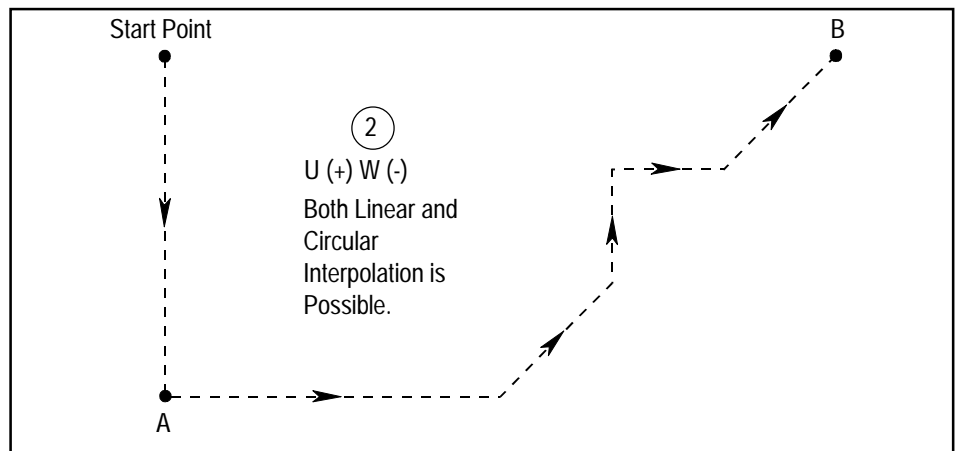
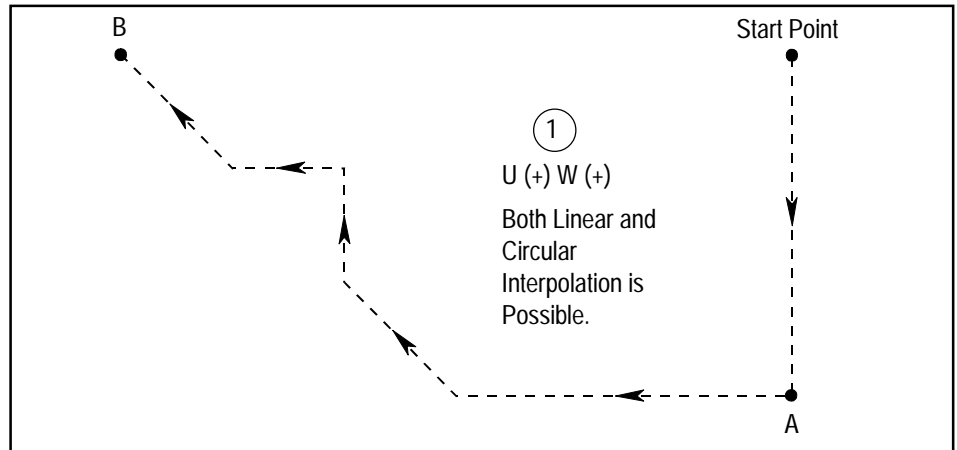
These 4 cutting patterns are illustrated opposite.



Note 1.

Only a steadily increasing, decreasing or parallel toolpath between A and B is permitted, depending on the cutting pattern.

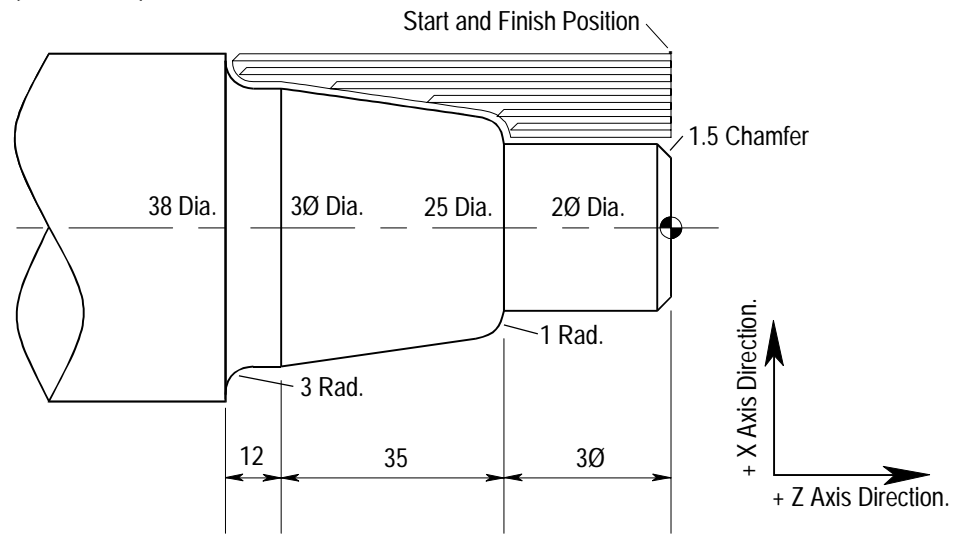
3: G Codes - G71 (Stock Removal in X Axis)



The plus and minus signs of U and W refer to the for the material that must be left on the billet to allow a clean finishing cut.

3: G Codes - G71 (Stock Removal in X Axis)

Example of programming an external profile using a multiple repetitive cycle (G70, G71)



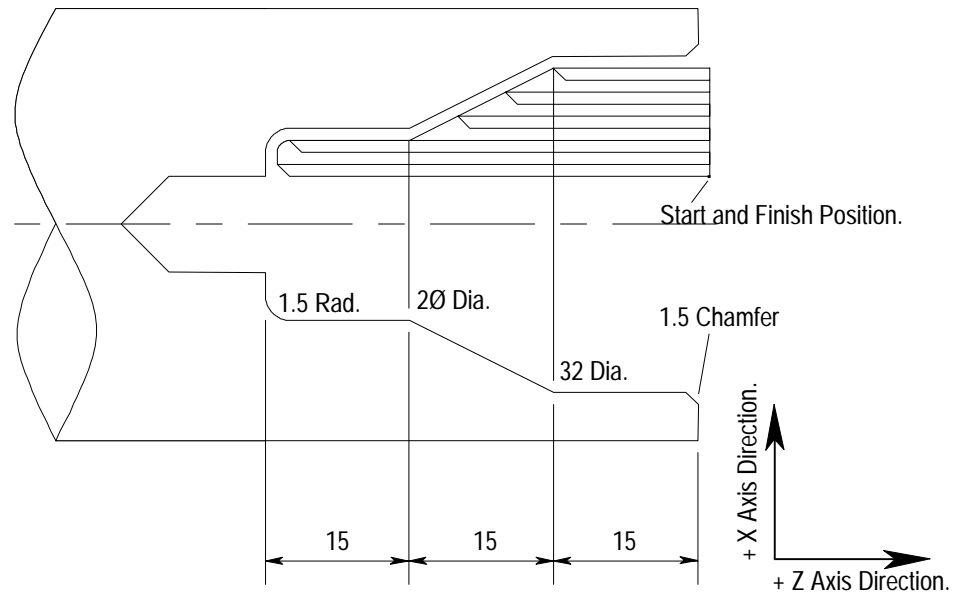
(Diameter Designation, Metric Input).

```

N0040 ..... ;
N0050 G00 X38.0 Z2.0 ; (Start and Finish Position)
N0060 G71 U1.5 R0.5 ;
N0070 G71 P80 Q170 U2.0 W0.1 F0.15 ;
N0080 G01 X16.0 F0.3 ; (ns)
N0090 Z0.5 ;
N0100 X20.0 Z-1.5 F0.05 ;
N0110 Z-30.0 F0.075 ;
N0120 X23.0 ;
N0130 G03 X25.0 Z-24.0 R1.0 ;
N0140 G01 X30.0 Z-65.0 ;
N0150 Z-74.0 ;
N0160 G02 X36.0 Z-77.0 R3.0 ;
N0170 G01 X38.5 ; (nf)
N0180 G70 P80 Q170 ; (Finishing Pass)
  
```

3: G Codes - G71 (Stock Removal in X Axis)

Example of programming an internal profile using a multiple repetitive cycle (G70, G71)



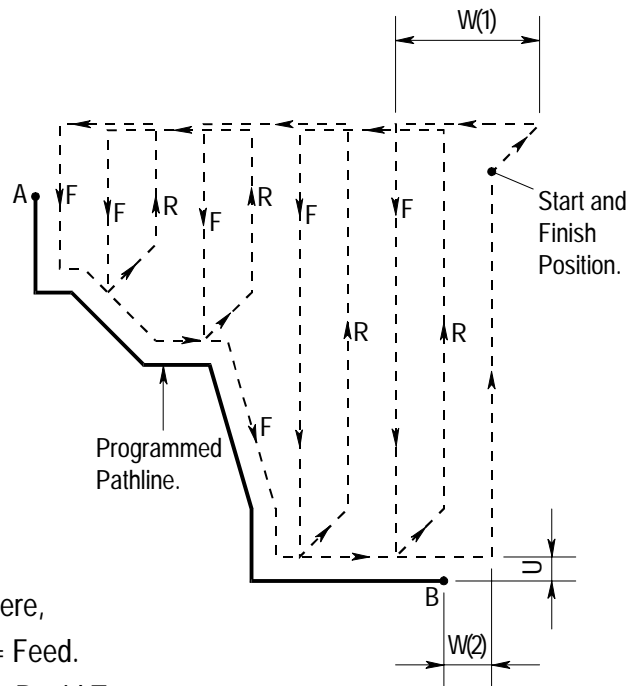
(Diameter Designation, Metric Input).

```

N0060 ..... ;
N0070 G00 X16.0 Z2.0 ;           ( Start and Finish Position )
N0080 G71 U1.25 R0.5 ;
N0090 G71 P100 Q170 U-1.5 W0.07 F0.075 ;
N0100 G01 X36.0 F0.3 ;           ( ns )
N0110 Z0.5 ;
N0120 X32.0 Z-1.5 F0.05 ;
N0130 Z-15.0 F0.07 ;
N0140 X20.0 Z-30.0 ;
N0150 Z-43.5 ;
N0160 G03 X17.0 Z-45 R1.5 F0.05 ;
N0170 G01 X15.8 ;               ( nf )
N0180 G70 P100 Q170 ;          ( Finishing Pass )
    
```

3: G Codes - G72 (Stock Removal in Facing)

The G72 cycle is similar to the G71 except that the cut is applied in the Z axis. The diagram below shows the tool movements for a G72 move:



where,

F = Feed.

R = Rapid Traverse.

W(1) = Depth of cut.

W(2) = Finishing cut in Z axis.

U = Finishing cut in X axis.

A G72 command is written in the following format :

G72 W (1) R _ ;

G72 P ns Q nf U _ W (2) F _ ;

where,

W (1) = Depth of cut in the Z axis.

R = Escaping amount (Retract).

ns = Sequence number of the first block of the programmed finished shape.

nf = Sequence number of the last block of the programmed finished shape.

U = Distance and direction of the finishing allowance in the X axis.

W (2) = Distance and direction of the finishing allowance in the Z axis (Diameter value).

F = Feedrate for Roughing.

3: G Codes - G72 (Stock Removal in Facing)

Note 1.

Only a Z axis move is allowed in the ns block.

Note 2.

If a finishing pass (ie, a single cut following the same pathline) is required, the block `G70 P ns Q nf` would be read into the CNC control after the G72 command has been completed. This method is similar to the finishing pass on a G71 command.

Four cutting patterns can be used with G72:

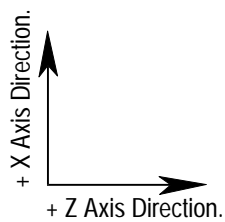
- 1) Internal facing with the Z axis cut moving towards the spindle.
- 2) Internal facing with the Z axis cut moving away from the spindle.
- 3) External facing with the Z axis cut moving towards the spindle.
- 4) External facing with the Z axis cut moving away from the spindle.

These 4 cutting patterns are illustrated on the next page.

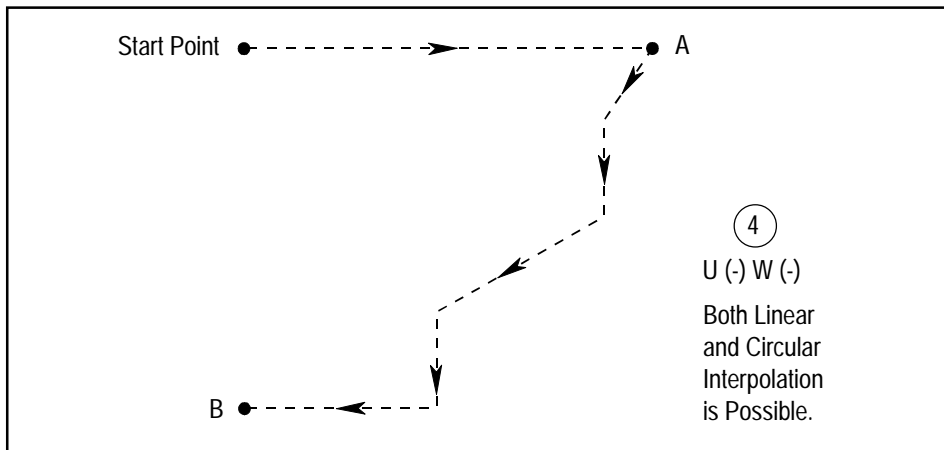
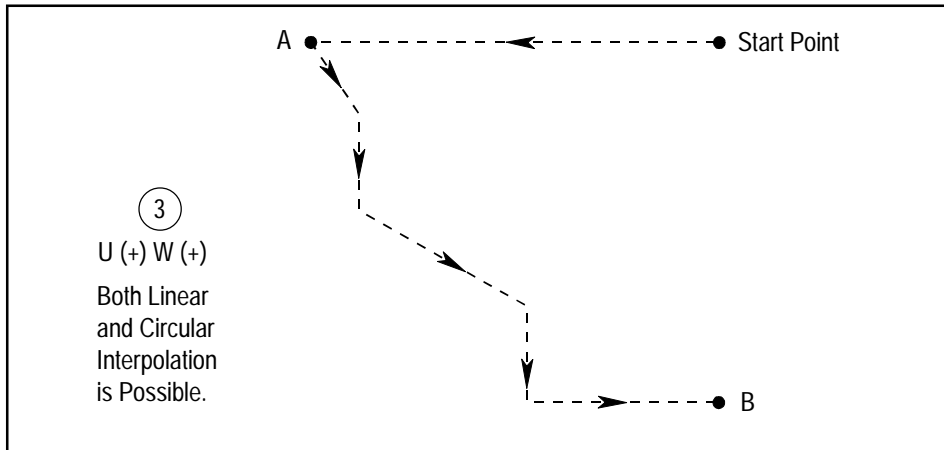
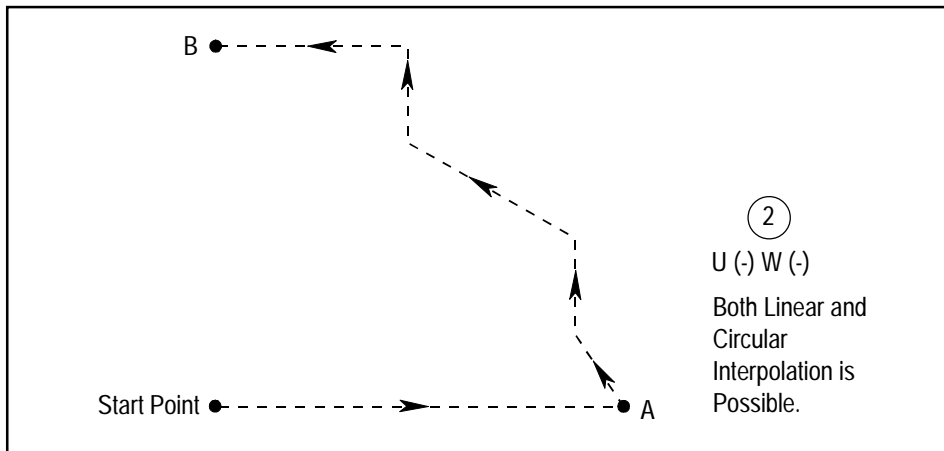
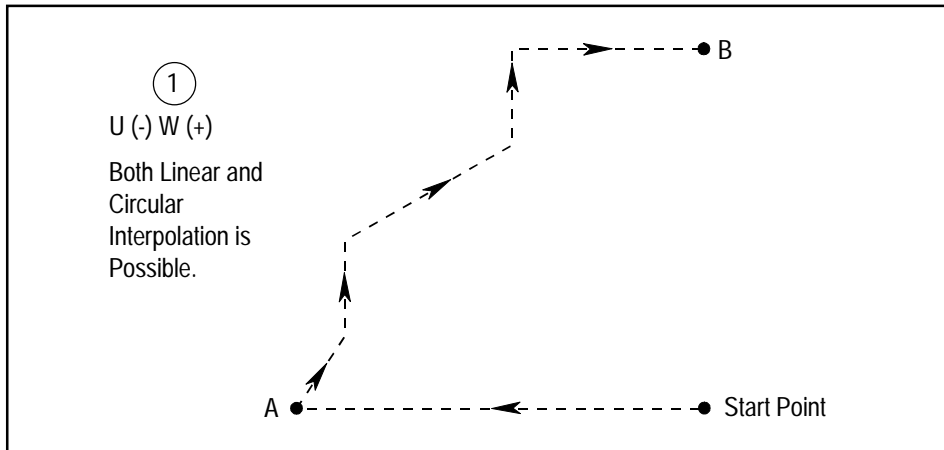
Note 1.

Only a steadily increasing, decreasing or parallel toolpath between A and B is permitted, depending on the cutting pattern.

3: G Codes - G72 (Stock Removal in Facing)

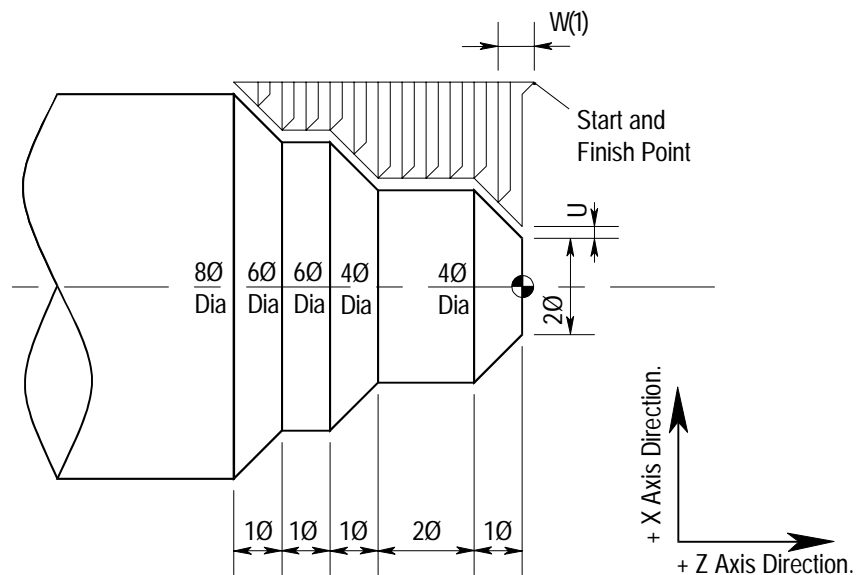


The plus and minus signs of U and W refer to the for the material that must be left on the billet to allow a clean finishing cut.



3: G Codes - G72 (Stock Removal in Facing)

Example of programming an external profile using a multiple repetitive cycle (G70, G72)



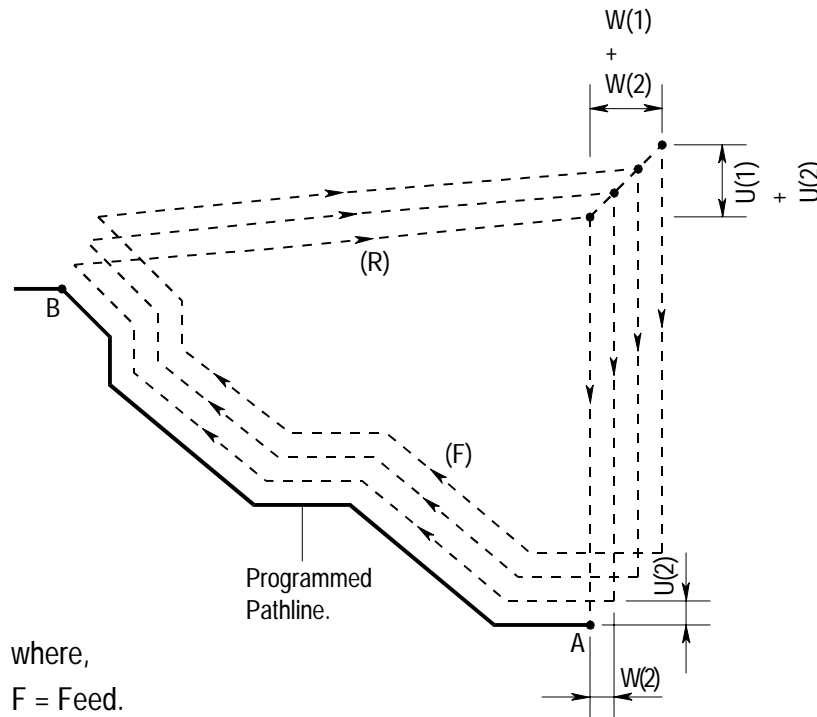
(Diameter Designation, Metric Input).

```

N0040 ..... ;
N0050 G00 X81.0 Z2.0 ;           ( Start and Finish Position )
N0060 G72 W2.0 R0.5 ;
N0070 G72 P80 Q150 U0.5 W1.0 F0.1 ;
N0080 G00 Z-60.0 ;             ( ns )
N0090 G01 X80.0 F0.2 ;
N0100 X60.0 W10.0 F0.075 ;
N0110 W10.0 ;
N0120 X40.0 W10.0 ;
N0130 W20.0 ;
N0140 X18.0 W11.0 ;
N0150 W1.0 ;                   ( nf )
N0160 G70 P80 Q150 ;          ( Finishing Pass )
N0170 ..... ;
  
```

3: G Codes - G73 (Pattern Repeating)

The function of the G73 canned cycle is to permit the cutting of a programmed profile repeatedly. It is mainly used for machined parts where the rough shape has already been formed by either rough machining, forging or casting.



where,

F = Feed.

R = Repeat patterns.

U (1) = Distance and direction of relief in the X axis direction (Radius value).

W (1) = Distance and direction of relief in the Z axis direction.

A G73 command is written in the following format :

G73 U (1) W (1) R _ ;

G73 P ns Q nf U (2) W (2) F _ ;

where,

U (1) = Depth of cut in the X axis (Radius value).

W (1) = Depth of cut in the Z axis.

R = Escaping amount (Retract).

ns = Sequence number of the first block of the programmed finished shape.

nf = Sequence number of the last block of the programmed finished shape.

U (2) = Distance and direction of the finishing allowance in the X axis (Diameter value).

W (2) = Distance and direction of the finishing allowance in the Z axis (Diameter value).

F = Feedrate for Roughing.

3: G Codes - G73 (Pattern Repeating)

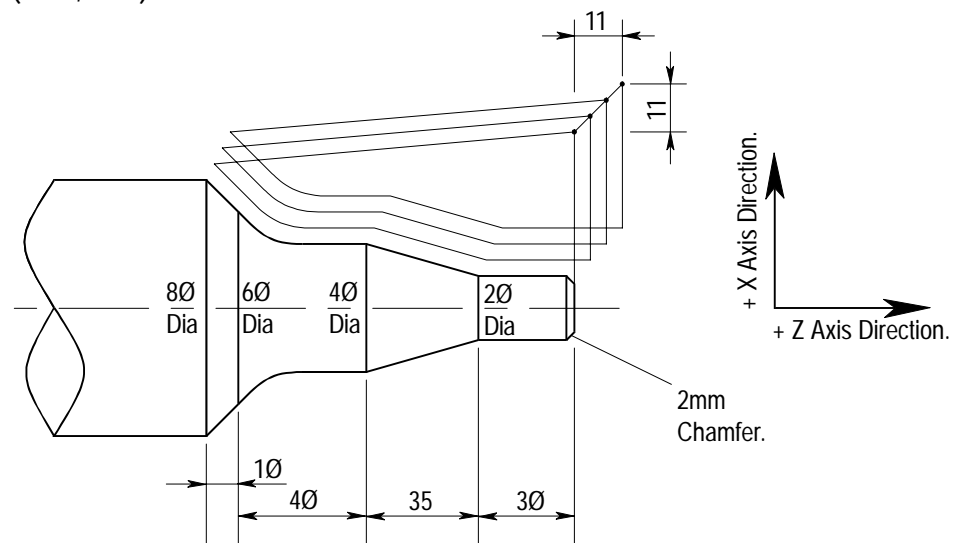
Note 1.

An X and Z axis move must be programmed in the *ns* block.

Note 2.

The four cutting patterns used by G71 can also be used by G73.

Example of programming an external profile using a multiple repetitive cycle (G70, G71).



(Diameter Designation, Metric Input).

```
N0050 ..... ;
N0060 G00 X82.0 Z2.0 ;
N0070 G73 U10.0 W10.0 R3.0 ;
N0080 G73 P90 Q150 U2.0 W1.0 F0.1 ;
N0090 G00 X15.0 Z0.5 ; (ns)
N0100 G01 X20.0 Z-2.00 F0.05 ;
N0110 Z-30.0 F0.075 ;
N0120 X40.0 Z-65.0 ;
N0130 Z-95.0 ;
N0140 G02 X60.0 Z-105.0 R10.0 ;
N0150 G01 X82.0 Z-116.0 ; (nf)
N0160 G70 P90 Q150 ;
N0170 ..... ;
```

3: G Codes - G74 (End Face Peck Drilling Cycle)

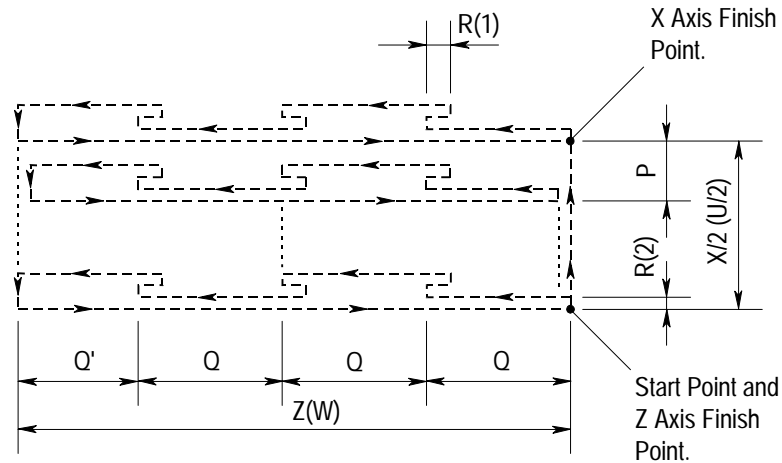
The G74 code instructs the machine to perform a peck drilling cycle. The centreline of the drill runs parallel to the Z axis, ie, the drill will make holes in the face end of the billet. Peck drilling refers to the method used for removing the material, ie, the the drill will move a set amount into the material, then withdraw to allow swarf and scrap drop out, before cutting further into the material.

The G74 code allows two different methods of drilling:

- 1) A tipped U-drill can drill into a billet, then move along the X axis and repeat the operation several times if required, ie, it can be used to drill holes off-centre.
 - 2) A standard drill can also be used. If the words X _ , P _ and R _ are omitted from the G74 command, a hole will be generated by pecking in stages (each stage measuring a depth of Q _) to a total depth of Z _ .
-

3: G Codes - G74 (End Face Peck Drilling Cycle)

The following diagram illustrates the movements of a G74 command if used in conjunction with a U-drill. Chip breaks / Peck cycles are useful for limiting the size of the waste swarf.



A G74 command is written in the following format :

G74 R (1) ;

G74 X (U) Z (W) P _ Q _ R (2) F _ ;

where,

R(1) = Peck Return amount.

X (U) = Diameter of the bore if stepover is used (ie, stepping along the X axis to repeat the peck cycle).

Z (W) = Depth of the bore.

P = Stepover in the X axis measured in micron's (without sign).

Q = Pecking depth in the Z axis measured in micron's (without sign).

F = Feedrate.

Note 1.

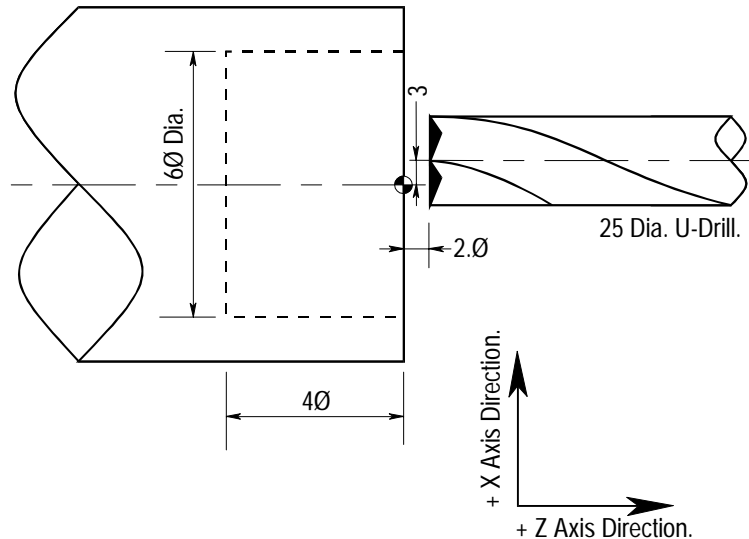
G74 is a chip break cycle (ie, peck drilling), not a deep hole drilling cycle where the drill retracts clear of the hole when it pecks.

Note 2.

In a G74, the absolute values of X and Z may be replaced with the incremental values U and W.

3: G Codes - G74 (End Face Peck Drilling Cycle)

Example of programming an end face peck drilling cycle (G74).



(Diameter Designation, Metric Input).

```
N0070 ..... ;  
N0080 G00 X3.0 Z2.0 ;  
N0090 G74 R0.5 ;  
N0100 G74 X35.0 Z-40.0 P4000 Q10000 R0.5 F0.2 ;  
N0110 ..... ;
```

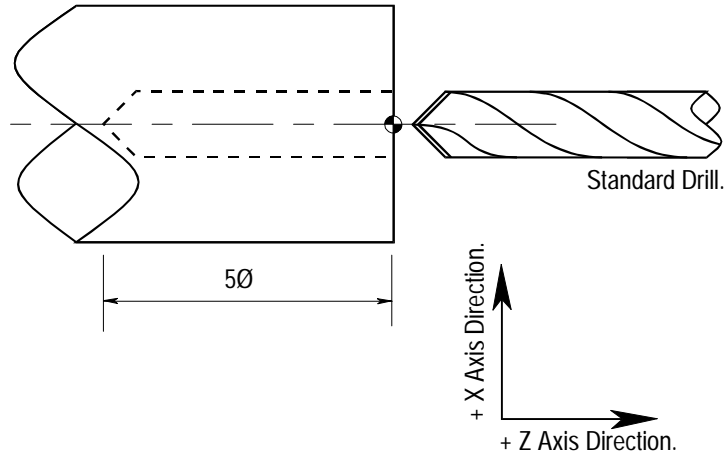
In the above program, the drill is positioned at X3.0 Z2.0. From this point, the G74 command is read into the CNC control. The drill moves forward 10mm (the Q value) and then retracts 0.5mm (the R(1) value). These movements are repeated until the full depth is reached (the Z (W) value).

At this point, the drill moves in a minus direction along the X axis (towards the spindle centreline) 0.5mm (the R (2) value), then it retracts at a rapid traverse rate to the start position in the Z axis. From this point, the drill moves in a positive direction along the X axis 4mm (the P value) and the sequence of peck drilling moves is repeated until the X axis diameter and Z axis depth stop is reached. The tool will repeat this operation of peck drilling each cut-in section until the correct hole diameter is achieved.

After all the moves have been completed, the drill is retracted back to the start position in the Z axis and the last cut-in section in the X axis.

3: G Codes - G74 (End Face Peck Drilling Cycle)

Example of programming G74 as a plain drilling cycle.



(Diameter Designation, Metric Input).

```
N0070 ..... ;  
N0080 G00 X0.0 Z2.0 ;  
N0090 G74 R0.5 ;  
N0100 G74 Z-50.0 Q1000 F0.15 ;  
N0110 ..... ;
```

In the above program, the drill is positioned at a point in the X and Z axis and the G74 is read into the CNC control. The drill moves forward 10mm (the Q value) and then retracts back 0.5mm (the R (1) value). These moves are repeated until the Z axis depth is reached. The drill then moves back at a rapid traverse rate to the start position and the next block is read into the CNC control.

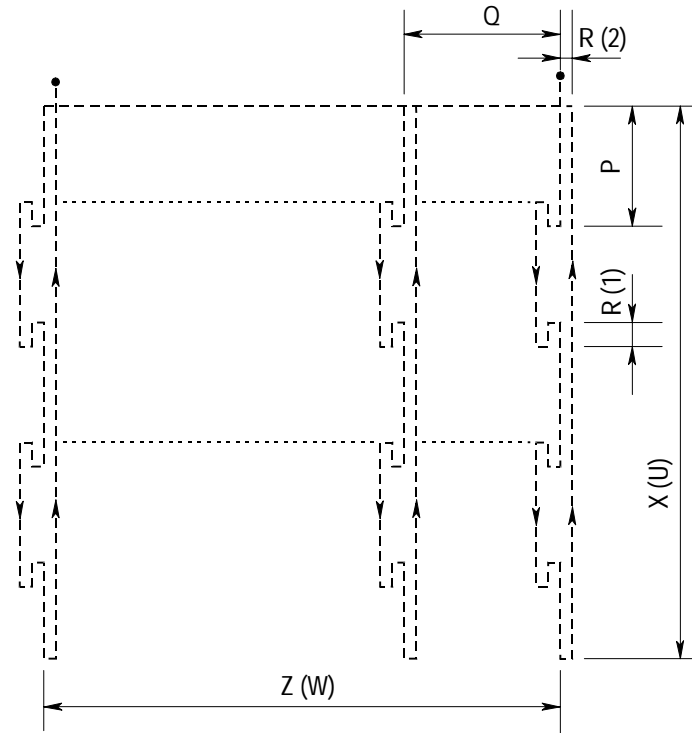
The following terms do not apply to this example since no X axis stepover is required; X (U) _ ; P _ ; R (2).

Note 1.

The Q value does not have to divide into the total Z depth equally.

3: G Codes - G75 (Outer Diameter/Internal Diameter Drilling and Grooving Cycle)

The G75 command permits drilling and grooving in the X axis. A chip break cycle is also possible - chip breaking is the same as drilling in pecking stages to limit swarf size.



A G75 command is written in the following format :

G75 R (1) ;

G75 X (U) Z (W) P _ Q _ R (2) F _ ;

where,

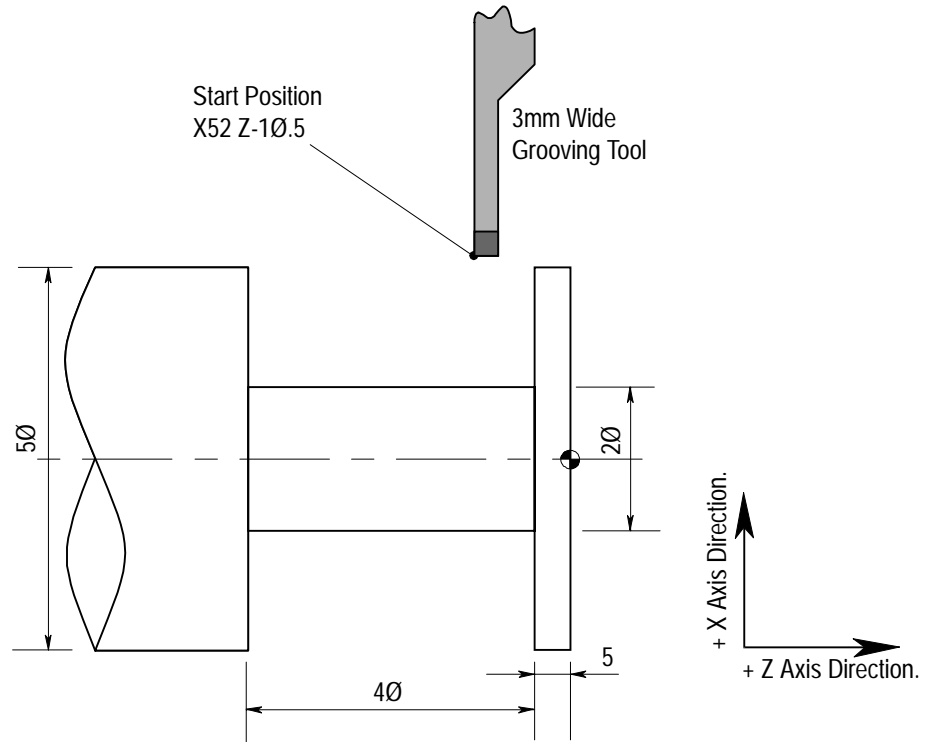
the definitions of R (1) , X (U) , Z (W) , P _ , Q _ , R (2) and F _ are the same as G74.

Note 1.

When G75 is used for grooving, the retract move (R(2)) at the base of the groove may be impossible, unless a clearance amount has been provided. If no clearance amount has been left, the R(2) value must be zero or omitted from the block.

3: G Codes - G75 (Outer Diameter/Internal Diameter Drilling and Grooving Cycle)

Example of programming a G75 command for grooving.



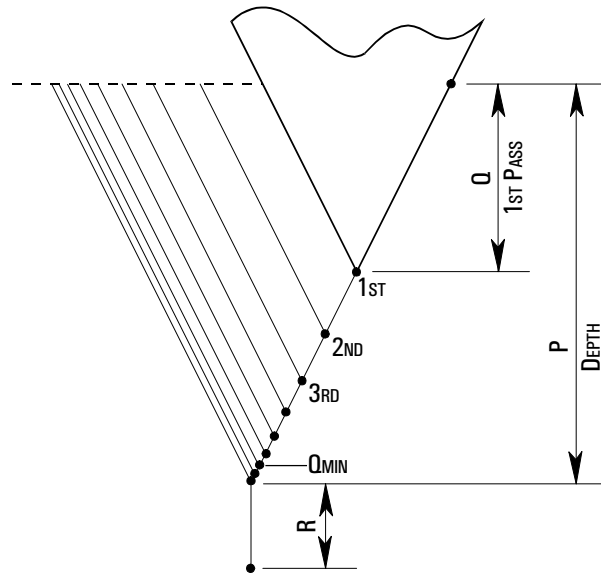
(Diameter Designation, Metric Input).

```
N0050 ..... ;  
N0060 G00 X52.0 Z-8.0 ;  
N0070 G01 X20.0 F0.075 ;  
N0080 G00 X52.0 ;  
N0090 W-2.5 ;  
N0100 G75 R0.5 ;  
N0110 G75 X20.0 Z-40.0 P5000 Q2500 R0.5 F0.075 ;  
N0120 ..... ;
```

3: G Codes - G76 (Multiple Thread Cutting Cycle)

The G76 command contains, within two blocks, all the information required to generate a standard thread form and pitch.

A G76 uses one edge cutting to reduce the load on the tool tip.



A G76 command is written in the following format :

G76 P (A) / (B) / (C) Q (Min) R _ ;

G76 X(U) _ Z(W) _ P (DEP) Q (1st) F _ ;

where,

P (A) = Number of thread finishing passes (1 to 99).

P (B) = The chamfer amount. This is the angle at which the tool leaves the billet, at the end of the thread cutting cycle.

P (C) = The angle of the tool tip (80° , 60° , 55° , 30° , 29° and 0°).

Note - (A), (B) and (C) are all specified at the same time by the address P, eg, P036060 = number of cuts is 03, chamfer amount of 60 and tool angle of 60° .

Q (Min) = Minimum cutting depth (in microns). When the depth of the cut calculated by the CNC control becomes less than this limit, the cutting depth is clamped at this minimum value.

R _ = Finishing allowance. This is the final, or finishing, cuts applied to the thread. The number of stages to complete this finishing allowance is determined by the value of P(A), ie, the value of R divided by the P(A) number of finishing passes equals the value of each finishing allowance stage.

X(U) _ = The end position of the thread in the X axis (the core diameter).

Z(W) _ = The end position of the thread in the Z axis.

P (DEP) = The depth of the thread as a radius value (in microns).

Q (1st) = The depth of the first pass as a radius value (in microns).

F _ = The size of the thread pitch.

3: G Codes - G76 (Multiple Thread Cutting Cycle)

Note 1.

When incremental dimensions are used, their signs (+ or -) are defined as follows :

U and W = Plus/Minus (determined by the direction of the tool path).

R = Plus/Minus (determined by the direction of the tool path).

P = Plus (always).

Q = Plus (always).

Note 2.

Four symmetrical patterns can be considered depending on the sign (plus or minus) of the X and Z axis movements.

Note 3.

It is possible to cut internal threads with the G76 command.

Note 4.

Thread cutting is repeated along the same tool path from rough cutting through to the final finishing cut, so the spindle speed must remain constant. The G96 command for constant surface speed must not be used when the thread cutting cycle is active, otherwise the pitch of thread could be incorrectly machined.

Note 5.

When possible, allow a 5mm run-in at the start of the thread pass, to allow for any lag in the machine drive system, etc... Without a sufficient run-in the start of the thread could be machined with an incorrect pitch.

Note 6.

The feedrate override on the CNC control panel will be ineffective, ie, it is set at a fixed value of 100% during the entire thread cutting cycle.

Note 7.

Although the spindle override feature is not disabled whilst a thread is being cut, it should not be activated since an incorrect thread pitch will be generated.

3: G Codes - G76 (Multiple Thread Cutting Cycle)

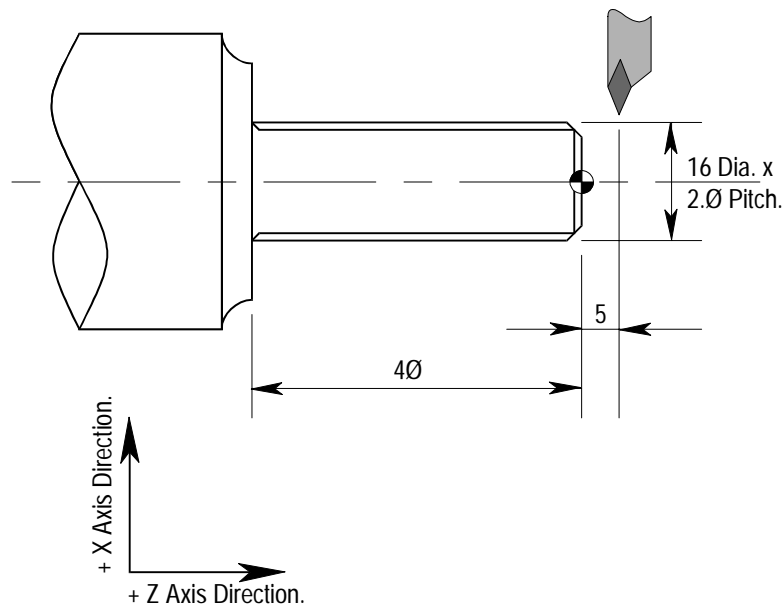
Note 8.

The cycle stop key will not operate during a thread cutting cycle. The thread cutting operation can only be stopped using either the [RESET] key on the CNC controller panel or [EMERGENCY STOP] button.

Note 9.

The machine can be set to read one block at a time by pressing the [SINGLE BLOCK] key on the CNC controller panel. When operating in single block mode, one press of the [CYCLE START] key will activate one complete threading pass (cut on, thread pass, cut off and rapid back). If single block mode is activated during a threading operation, the tool motion will stop at the beginning (cut on) of the next complete threading pass.

Example of programming a multiple thread cutting cycle (G76).

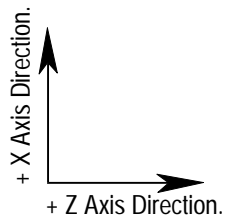
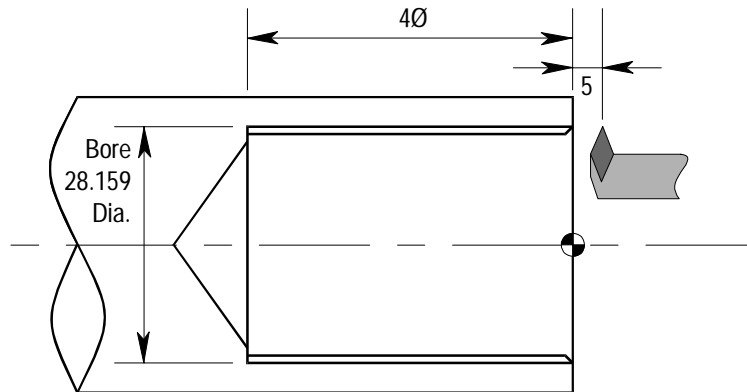


(Diameter Designation, Metric Input).

```
N0090 ..... ;  
N0100 G00 X16.25 Z5.0 ;  
N0110 G76 P030060 Q150 R0.03 ;  
N0120 G76 X13.546 Z-40.0 P1226 Q300 F2.0 ;  
N0130 ..... ;
```

3: G Codes - G76 (Multiple Thread Cutting Cycle)

Example of programming an internal thread cutting cycle (G76).



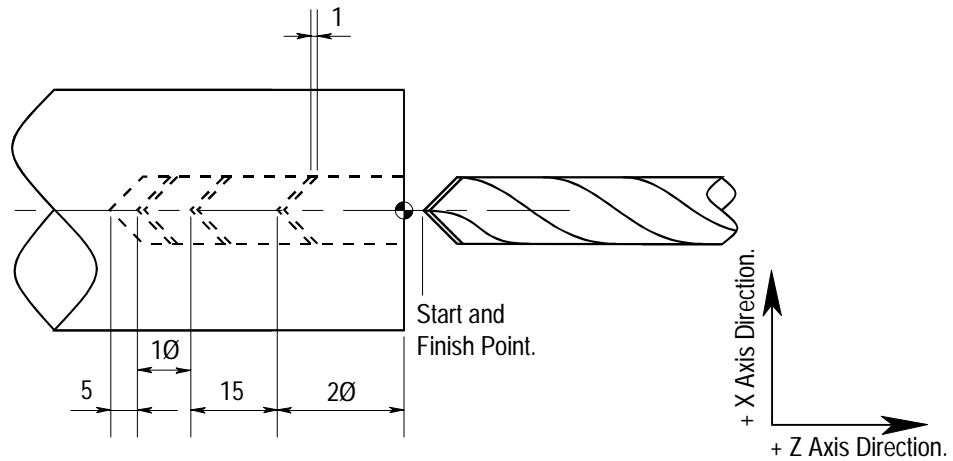
(Diameter Designation, Metric Input).

```
N0090 ..... ;  
N0100 G00 X27.9 Z5.0 ;  
N0110 G76 P040060 Q100 R0.04 ;  
N0120 G76 X30.0 Z-40.0 P920 Q200 F1.5 ;  
N0130 ..... ;
```

3: G Codes - G81 (Deep Hole Drilling Cycle)

The command G81 performs a deep drilling operation, where the drill retracts completely out of the hole at each peck. The definition of one of these operations is called a one pass cycle.

If a repetition of the cycle is required, only the values that change need to be entered into the next block, ie, the Z depth.



(Diameter Designation, Metric Input).

Absolute.

```
N0050 ..... ;
N0060 G00 X0.0 Z2.0 ;
N0070 G81 Z-20.0 F0.1 ;
N0080 Z-35.0 ;
N0090 Z-45.0 ;
N0100 Z-50.0 ;
N0110 G..... ;
```

(Diameter Designation, Metric Input).

Incremental.

```
N0050 ..... ;
N0060 G00 X0.0 Z2.0 ;
N0070 G81 W-22.0 F0.1 ;
N0080 W-37.0 ;
N0090 W-47.0 ;
N0100 W-52.0 ;
N0110 G..... ;
```

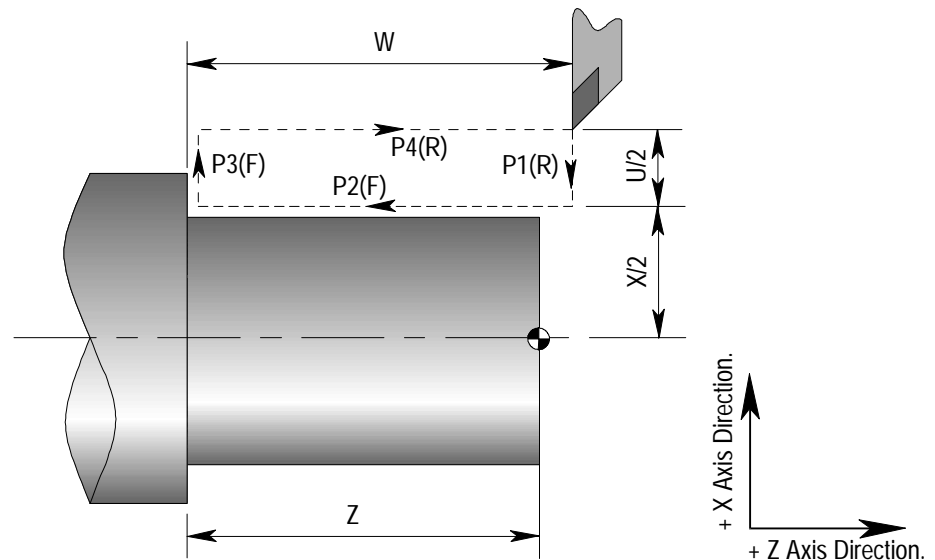
In the above program, the drill is positioned at the start point and G81 Z-20.0 F0.1 is read into the machine controller. The drill then cuts into the billet to the stated Z position (Z-20.0) and then rapid traverses back to the starting point.

At this point, the next block is read into the machine controller (Z-35.0). The drill rapid traverses forward to within 1mm of the previous cut (Z-19.0) where cutting starts for this pass. These moves continue (cut-in, rapid out to start, rapid back to within 1mm), until the last block containing a Z axis block has been completed. The next block containing a different G code will cancel the G81 command.

3: G Codes - G90 (Outer Diameter/Internal Diameter Cutting Cycle)

The command G90 performs a one pass cutting cycle, where the cut is applied in the X axis. Also, by using the command R __ within the G90 block, tapers can be generated.

If the one pass move needs to be repeated, only the values that change (ie, movement dimensions) need to be entered in the next block.



A G90 command for straight line cutting is written in the following format:

G90 X (U) __ Z (W) __ F __ ;

where,

F __ = Feedrate.

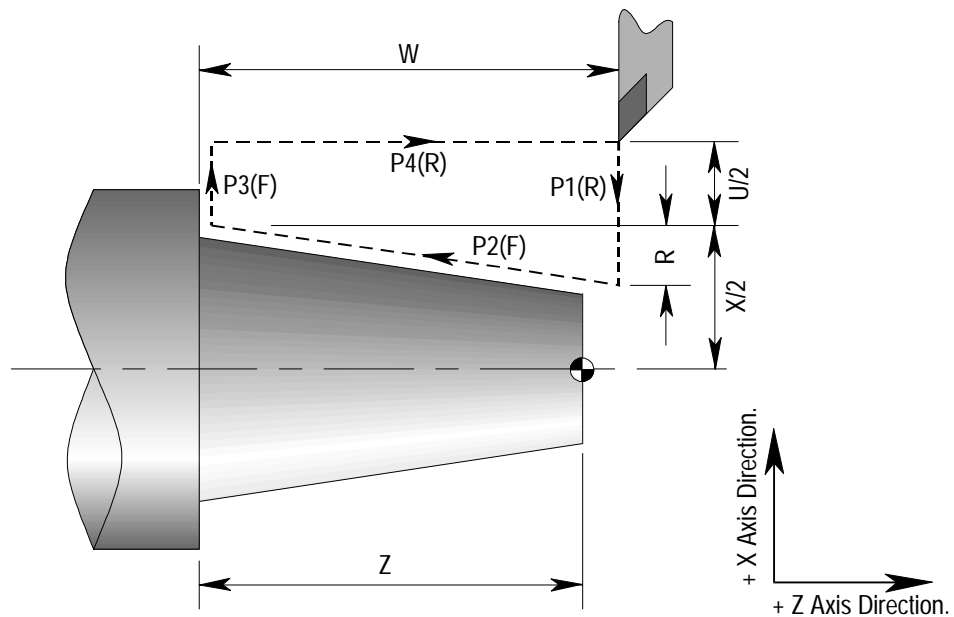
Note 1.

When programming using incremental dimensions (U, W), the direction of the tool movement along paths "P1" and "P2" will define the signs of U and W (plus or minus). In the above program, both U and W are minus.

Note 2.

The G90 straight cut command can be used for both internal and external cutting operations.

3: G Codes - G90 (Outer Diameter/Internal Diameter Cutting Cycle)



A G90 command for taper cutting is written in the following format:

G90 X (U) __ Z (W) __ R __ F __ ;

where,

R __ = Dimension defining taper angle.

F __ = Feedrate.

Note 1.

The sign of R depends on the cutting direction of path "P1" - in the above program, R is entered as a minus value.

Note 2.

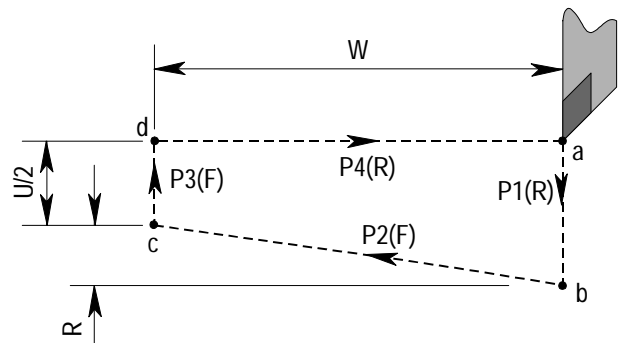
When programming using incremental dimensions (U, W), the direction of the tool movement along paths "P1" and "P2" will define the signs of U and W (plus or minus). In the above program, both U and W are minus.

Note 3.

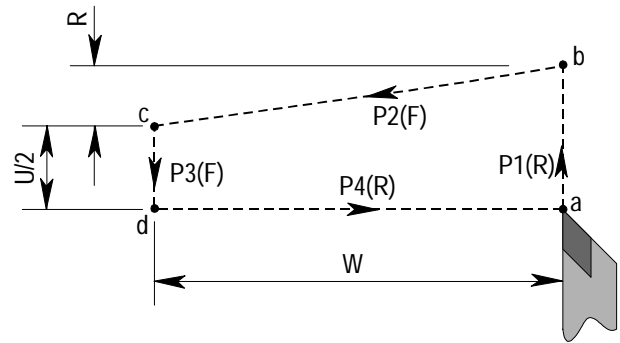
The G90 taper cut command can be used for both internal and external cutting operations.

3: G Codes - G90 (Outer Diameter/ Internal Diameter Cutting Cycle)

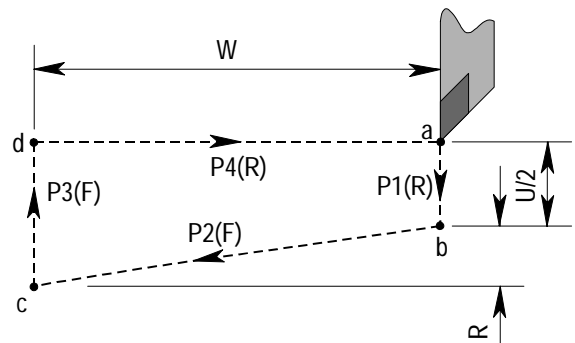
Sign Designations for the Path shown below are:
U=Minus sign ;
W=Minus sign ;
R=minus sign.



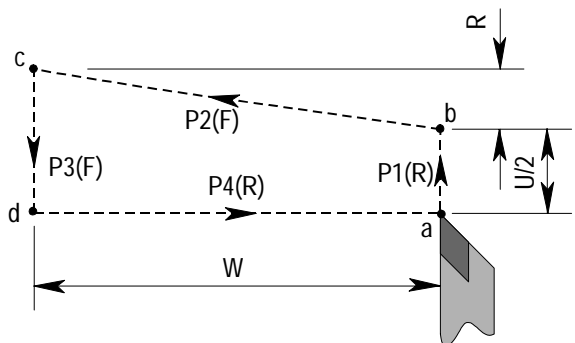
Sign Designations for the Path shown below are:
U=Plus sign ;
W=Minus sign ;
R=Plus sign.



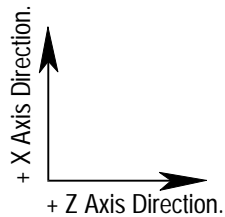
Sign Designations for the Path shown below are:
U=Minus sign ;
W=Minus sign ;
R=Plus sign.



Sign Designations for the Path shown below are:
U=Plus sign ;
W=Minus sign ;
R=Minus sign.



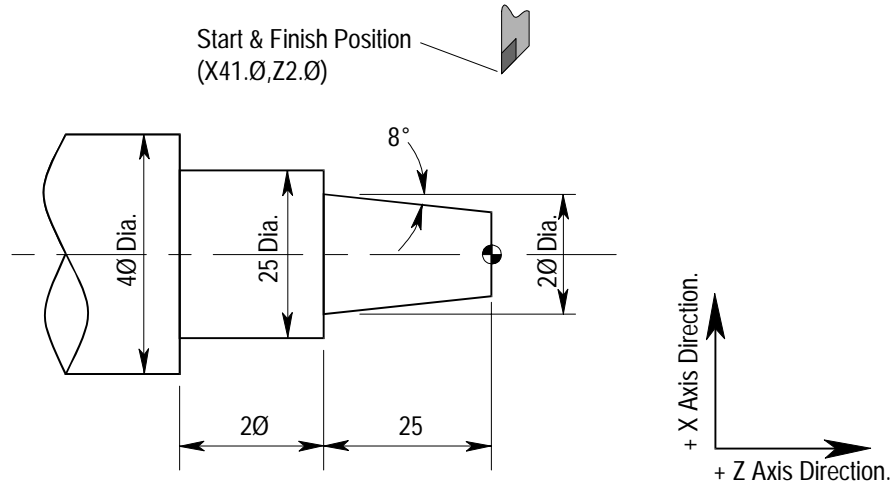
The Four Diagrams opposite show the Relationship of the signs of the following values for G90 Taper cutting, U, W and R.



Co-ordinate Positions at the four points labelled on each Diagram are as follows:
a= (X,Z)
b= (X+U+R,Z)
c= (X+U,Z+W)
d= (X,Z+W)

3: G Codes - G90 (Outer Diameter/ Internal Diameter Cutting Cycle)

Example of programming G90 for external cutting.



(Diameter Designation, Metric Input).

```
N0050 ..... ;  
N0060 G00 X41.0 Z2.0 ;  
N0070 G90 X35.0 Z-45.0 F0.1 ;  
N0080 X30.0 ;  
N0090 X25.0 Z-25.0 ;  
N0100 X20.0 ;  
N0110 X20.0 R-3.794 ;  
N0120 G..... ;
```

3: G Codes - G92 (Thread Cutting Cycle)

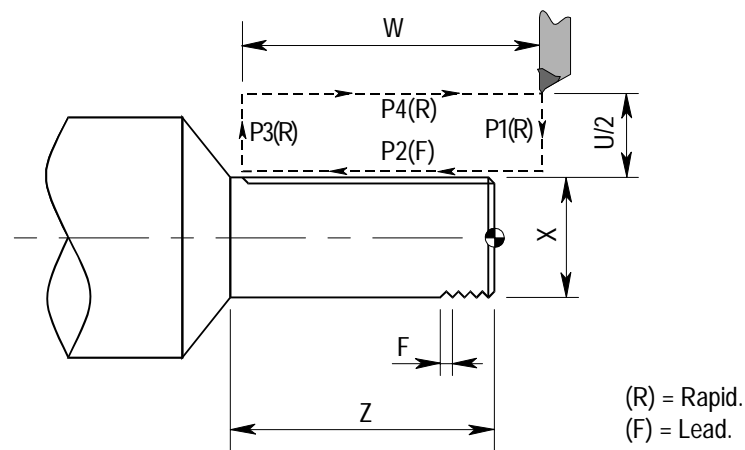
The G92 command performs a one pass threading cycle. Only the X (U) axis moves need to be entered in subsequent blocks, after the original read-in of the G92 command.

A G92 command for straight thread cutting is written in the following format :

G92 X (U) __ Z (W) __ F Lead ;

where,

F Lead = Threading lead feedrate.



Note 1.

G92 can be used to cut both internal and external threads.

Note 2.

Plunge cutting is used with G92 to generate a thread, from the first pass to the last finishing pass.

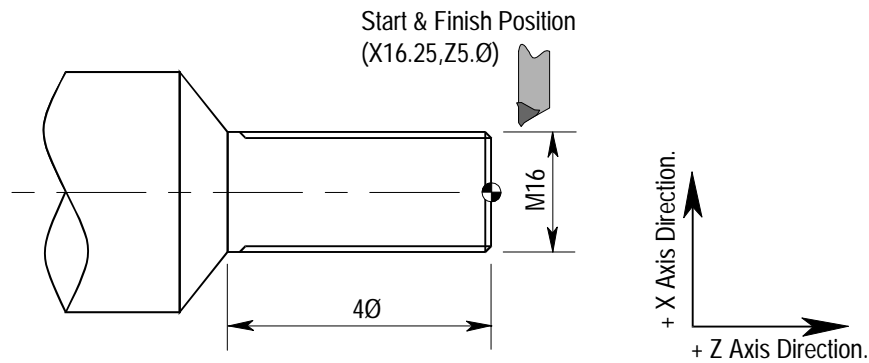
Plunge cutting involves the tool approaching the billet at 90 degrees to its surface, rather than approaching the billet at a run-in angle. Effectively, this means greater stresses are placed on the tool since both edges will be cutting. As the tool cuts further into the material, more surface area is in contact with the tool tip.

Note 3.

The same notes on thread cutting in G76 regarding speed, run-in, feed hold etc. are also relevant to G92.

3: G Codes - G92 (Thread Cutting Cycle)

Example of thread cutting using G92.



(Diameter Designation, Metric Input).

```
N0040 ..... ;  
N0050 G00 X16.25 Z5.0 ;  
N0060 G92 X15.6 Z-40.0 F2.0 ;  
N0070 X15.2 ;  
N0080 X14.8 ;  
N0090 X14.5 ;  
N0100 X14.2 ;  
N0110 X13.9 ;  
N0120 X13.9 ;  
N0130 X13.546 ;  
N0140 X13.546 ;  
N0150 G..... ;
```

(Spring Pass)

3: G Codes - G94 (End Face Turning Cycle)

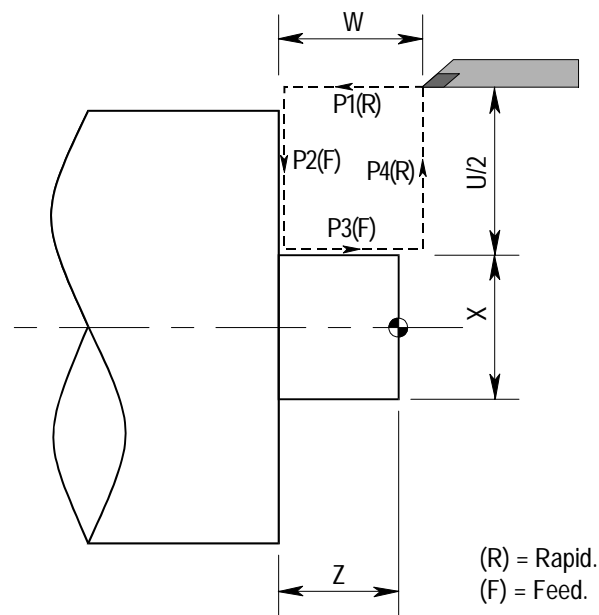
The G94 command performs a one pass face cutting cycle, where the cut is applied in the Z axis. Also, by using the word R __ within a G94 command, tapers can be generated. If a repartition of the move is required, only the values that change need to be entered into the next block.

A G94 command for a face cutting cycle is written in the following format:

G94 X(U) __ Z(W) __ F __ ;

where,

F __ = Feedrate.



Note 1.

When operating in Single Block Mode, each move of the tool requires one press of the [CYCLE START] button.

Note 2.

G94 can be used for internal and external cutting.

3: G Codes - G94 (End Face Turning Cycle)

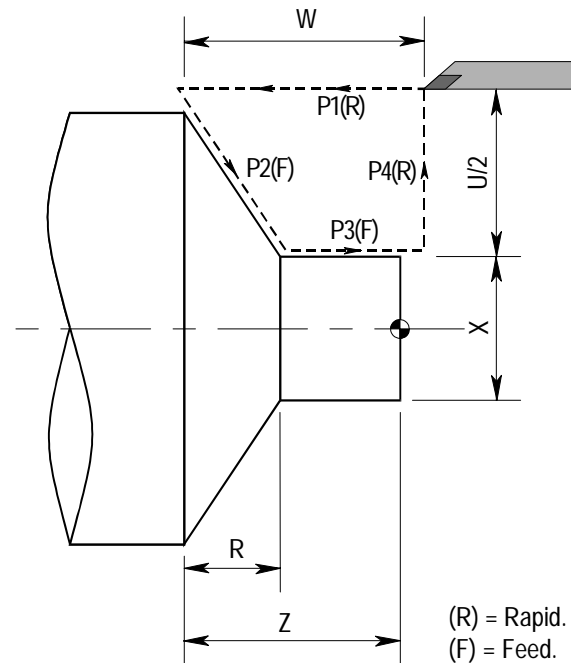
A G94 command for a taper face cutting cycle is written in the following format:

G94 X(U) __ Z(W) __ R __ F __ ;

where,

F __ = Feedrate.

R __ = Defines the size and angle of the taper.

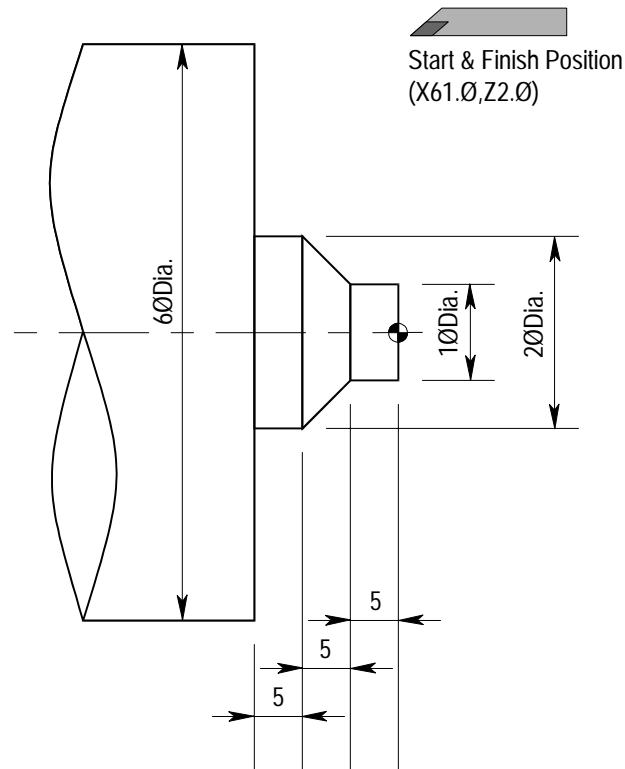


Note 1.

When programming using incremental positioning, the signs (plus or minus) of U, W and R will depend on the directions of paths "P1" and "P2", in a similar way to the methods shown for the command G90. In the above example, U, W and R are all minus values.

3: G Codes - G94 (End Face Turning Cycle)

Example of programming a face cutting cycle using G94.



(Diameter Designation, Metric Input).

```
N0050 ..... ;  
N0060 G00 X61.0 Z2.0 ;  
N0070 G94 X20.0 Z-2.5 F0.1 ;  
N0080 Z-5.0 ;  
N0090 Z-7.5 ;  
N0100 Z-10.0 ;  
N0110 Z-12.5 ;  
N0120 Z-15.0 ;  
N0130 G00 X20.0 Z5.0 ;  
N0140 G94 X10.0 Z2.5 R-2.5 ;  
N0150 Z0.0 ;  
N0160 Z-2.5 ;  
N0170 Z-5.0 ;  
N0180 ..... ;
```

3: G Codes - G96 (Constant Surface Speed Control)

If surface speed (relative speed between tool and billet) is set after the address S __ , the spindle speed is calculated so that the surface speed is always the specified value in relation to the tool position.

The units used will depend on whether the machine is operating using metric or imperial measurements.

The units of surface speed are as follows:

Input Unit.	Surface Speed Unit.
Metric - Millimetre.	Metres per Minute, M/Min.
Imperial - Inches.	Feet per Minute, Feet/Min.

A G96 command for surface speed control is written in the following format:

G96 S __ ;

where,

S __ = Surface speed (M/Min or Feet/Min)

Note 1.

The programmed surface speed can be altered during the running of the program, using the spindle override control on the machine operating panel. The range of adjustment varies from 50% to 120%, with 100% being the programmed speed.

Note 2.

When constant surface speed control is used, the work co-ordinate system setting and co-ordinate system shift (ie, the software datum point) must be set so that the centre of rotation meets the Z axis (ie, at X=0.0).

Note 3.

The G50 code defining clamping of maximum spindle speed must be used in conjunction with G96, so as not to overrun the safe maximum spindle speed of the chuck.

Note 4.

The G96 code is modal. It will remain active until G97, M02 or M30 codes are programmed into the same block, or the Emergency Stop or Machine Reset is activated.

3: G Codes - G97 (Spindle Speed in Rev/Minute)

The G97 command allows a spindle speed written in the units, revs per minute, to be entered into the machine controller. All subsequent spindle speeds are defined in revs per minute, after the original read-in of the command G97. If a change of spindle speed is required within a program, only the S __ value needs to be entered.

A G97 command for spindle speed control is written in the following format :

G97 S __ ;

where,

S __ = Spindle Speed, written in the format revs per minute.

Note 1.

The programmed spindle speed can be altered during the running of the program, using the spindle override control on the machine operating panel. The range of adjustment varies from 50% to 120%, with 100% being the programmed speed.

Note 2.

The G97 code is modal. It will remain active until G96, M02 or M30 codes are programmed into the same block, or the Emergency Stop or Machine Reset is activated.

Note 3.

The machine controller will assume G97 is active, in the following situations:

- Initially, when the machine is switched on.
 - After an M02 code is programmed.
 - After an M30 code is programmed.
 - After the Emergency Stop button is used.
 - After the machine has been Reset.
-

3: G Codes - G98 (Per Minute Feed)

The G98 command allows a feedrate written in the units, millimetres per minute or inches per minute, to be entered into the machine controller. All subsequent feedrates are defined in the chosen units, after the original read-in of the command G98. If a change of feedrate is required within a program, only the F __ value needs to be entered.

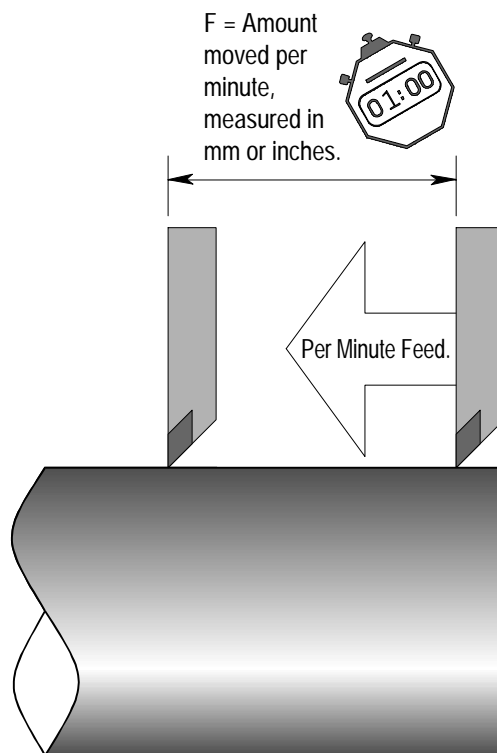
The units used will depend on whether the machine is operating using metric or imperial measurements.

A G98 feedrate command is written in the following format:

G98 F __ ;

where,

F __ =Feedrate, written in the format millimetres per minute or inches per minute.



3: G Codes - G98 (Per Minute Feed)

Note 1.

The programmed feedrate can be altered during the running of the program, using the feedrate override control on the machine operating panel. The range of adjustment varies from 0% to 150%, with 100% being the programmed speed.

Note 2.

The G98 code is modal. It remains active until G99, M02, M30 Emergency Stop or Machine Reset is activated.

Note 3.

The machine controller will assume G98 is active, in the following situations:

- Initially, when the machine is switched on.
 - After an M02 code is programmed.
 - After an M30 code is programmed.
 - After the Emergency Stop button is used.
 - After the machine has been Reset.
-

3: G Codes - G99 (Per Revolution Feed)

The G99 command allows a feedrate written in the units, millimetres per revolution or inches per revolution, to be entered into the machine controller. All subsequent feedrates are defined in the chosen units, after the original read-in of the command G99. If a change of feedrate is required within a program, only the F __ value needs to be entered.

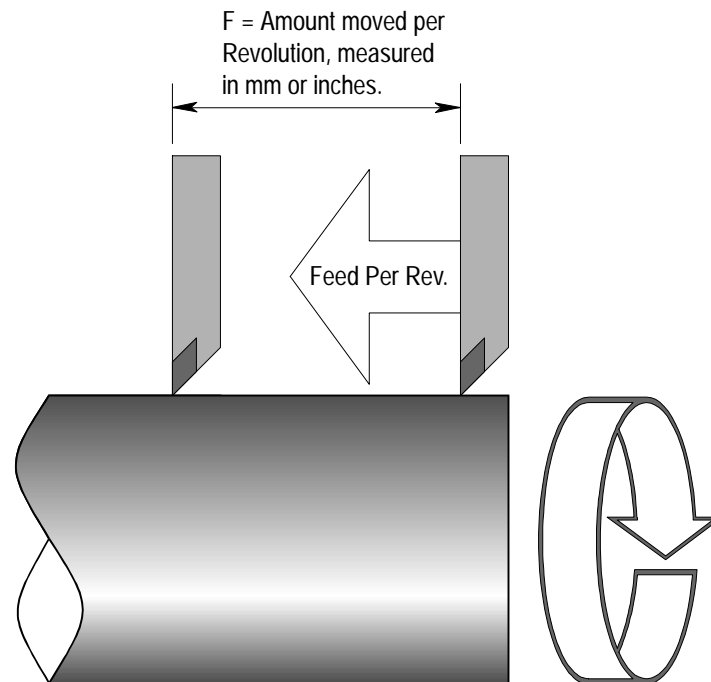
The units used will depend on whether the machine is operating using metric or imperial measurements.

A G99 feedrate command is written in the following format:

G99 F __ ;

where,

F __ =Feedrate, written in the format millimetres per revolution or inches per revolution.



Note 1.

The programmed feedrate can be altered during the running of the program, using the feedrate override control on the machine operating panel. The range of adjustment varies from 0% to 150%, with 100% being the programmed speed.

Note 2.

The G99 code is modal. It remains active until G98, M02, M30 Emergency Stop or Machine Reset is activated.

4: M Codes (Miscellaneous Functions) - Introduction

Miscellaneous functions, called *M codes*, are used by the CNC control to command on/off signals to the machine functions. ie, M03 - spindle forward (CW), M05 - spindle off, etc...

The functions allocated to lower M code numbers are constant in most CNC controls, although the higher M code number functions can vary from one make of controller to the next.

An M code is made from the M address letter and a two digit number as follows,

M 00
| |
| |
| | Two digit number.
| |
| | Address.

Notes for M Codes Listing shown on next page

Note 1.

M codes marked with an * are executed at the end of a block, ie, after axis movement.

Note 2.

Only one M code can be programmed within each block. If more than one M code is programmed, the machine controller will only perform the last stated M code.

4: List of M Codes supported by Denford CNC Controls

Note - Not all G codes apply to each Denford CNC machine.

M code.	Function.
M00*	Program stop
M01*	Optional stop
M02*	End of program
M03	Spindle forward (clockwise)
M04	Spindle reverse (counter clockwise)
M05*	Spindle stop
M06	Automatic tool change
M08	Coolant on
M09*	Coolant off
M10	Chuck open
M11	Chuck close
M13	Spindle forward and coolant on
M14	Spindle reverse and coolant on
M25	Tailstock quill extend
M26	Tailstock quill retract
M30	Program stop and reset
M38	Door open
M39	Door close
M40	Parts catcher extend
M41	Parts catcher retract
M62	Auxiliary output 1 on
M63	Auxiliary output 2 on
M64	Auxiliary output 1 off
M65	Auxiliary output 2 off
M66*	Wait for auxiliary output 1 on
M67*	Wait for auxiliary output 2 on
M76	Wait for auxiliary output 1 off
M77	Wait for auxiliary output 2 off
M98	Sub program call
M99	Sub program end and return

Code listing full and correct at the time of printing.

4: M Codes - M00 (Program Stop)

M00 - Program Stop.

When the machine controller reads the code M00 within a block, it halts the program. The [CYCLE START] key must be pressed to allow the program to continue.

4: M Codes - M01 (Optional Stop)

M01* - Optional Stop.

The M01 code performs the same function as the M00 code, except the machine controller only recognises the signal to halt the program if the optional [STOP] input key is activated.

4: M Codes - M02 (End of Program)

M02* - End of Program.

This code indicates the end of a program and performs a general reset function on the machine controller, ie, the CNC reverts to its initial state. The code also acts as an M05.

4: M Codes - M03 (Spindle Forward)

M03 - Spindle Forward (Clockwise).

The clockwise direction of the spindle is determined by viewing from the back of the machine headstock, along the Z axis towards the tailstock.

The spindle start command is activated at the beginning of the block in which it is programmed, ie, before any axis movement occurs.

4: M Codes - M04 (Spindle Reverse)

M04 - Spindle Reverse (Counter Clockwise).

An M04 code acts in the same way as an M03 code, only the spindle rotates in the opposite direction.

4: M Codes - M05 (Spindle Stop)

M05 - Spindle Stop.

The M05 code, to stop the spindle rotating, is activated at the end of the block in which it is programmed, ie, after any axis movement.

4: M Codes - M06 (Automatic Tool Change)

M06 - Automatic Tool Change.

This code activates the machine turret and is followed by the code T____, instructing it to move to the stated tool number.

For example :

M06 T0303 ;

This command is read change automatically from the current tool number to tool number 3.

4: M Codes - M08 (Coolant On)

M08 - Coolant On.

This code switches the coolant pump on.

4: M Codes - M09 (Coolant Off)

M09 - Coolant Off.

This code switches the coolant pump off.

4: M Codes - M10 (Chuck Open)

M10 - Chuck Open.

This code will open the jaws of a power chuck.

4: M Codes - M11 (Chuck Close)

M11 - Chuck Close.

This code will close the jaws of a power chuck.

4: M Codes - M13 (Spindle Forward and Coolant On)

M13 - Spindle Forward and Coolant On.

This code combines the functions of M03 and M08 together. The M05 code will stop both the spindle and coolant.

4: M Codes - M14 (Spindle Reverse and Coolant On)

M14 - Spindle Reverse and Coolant On.

This code performs the same function as M13 but the spindle rotates in the opposite direction.

4: M Codes - M25 (Tailstock Quill Extend)

M25 - Tailstock Quill Extend.

This code will drive the tailstock quill out (extend). The spindle has to be stopped (using the M05 code) to allow the machine controller to activate an M25 code.

4: M Codes - M26 (Tailstock Quill Retract)

M26 - Tailstock Quill Retract.

This code will drive the tailstock quill in (retract). The spindle has to be stopped (using the M05 code) to allow the machine controller to activate an M25 code.

4: M Codes - M30 (Program Stop and Reset)

M30* - Program Stop and Retract.

This code stops the program running, ie, it signals the end of the program.

Control is then reset back to the beginning of this program.

If the M30 code is followed by a block number, the program will be reset back to the stated block number.

For example :

M30 P0140 ;

This command is read stop the program running and reset it back to block number 140.

The M30 code also acts as an M05 and M09.

4: M Codes - M40 (Parts Catcher Extend)

M40 - Parts Catcher Extend.

This code drives the parts catcher out to beneath the part prior to parting off.

4: M Codes - M41 (Parts Catcher Retract)

M41 - Parts Catcher Retract.

This code drives the parts catcher back to its parking position after parting off.

4: M Codes - M62, M63, M64, M65, M66, M76, M77 (Auxiliary Output Functions)

M62 - Auxiliary Output 1 On.

M63 - Auxiliary Output 2 On.

M64 - Auxiliary Output 1 Off.

M65 - Auxiliary Output 2 Off.

M66 - Wait for Auxiliary Output 1 On.

M67 - Wait for Auxiliary Output 2 On.

M76 - Wait for Auxiliary Output 1 Off.

M77 - Wait for Auxiliary Output 2 Off.

These codes allow a signal to be sent from the machine controller to a different device, such as a robot, then wait for a return signal instructing that the device has completed its function.

4: M Codes - M98 (Sub Program Call)

M98 - Sub Program Call.

This code will cause the machine controller to jump across from the main program to read a different program in its memory (called a sub program).

4: M Codes - M99 (Sub Program End and Return)

M99 - Sub Program End and Return.

On the last line of a sub program, the code M99 is entered. This reverts control back to the main program.

If an M99 code is programmed at the end of a main program, a continuous loop will be established.

If an M99 code is followed by a block number, P____, control will return to the program line with the same number as stated in P____.
