

Mach4 CNC Controller
Lathe Programming Guide
10/26/22

The following are registered trademarks of Microsoft Corporation: Microsoft, Windows. Any other trademarks used in this manual are the property of the respective trademark holder.

Table of Contents

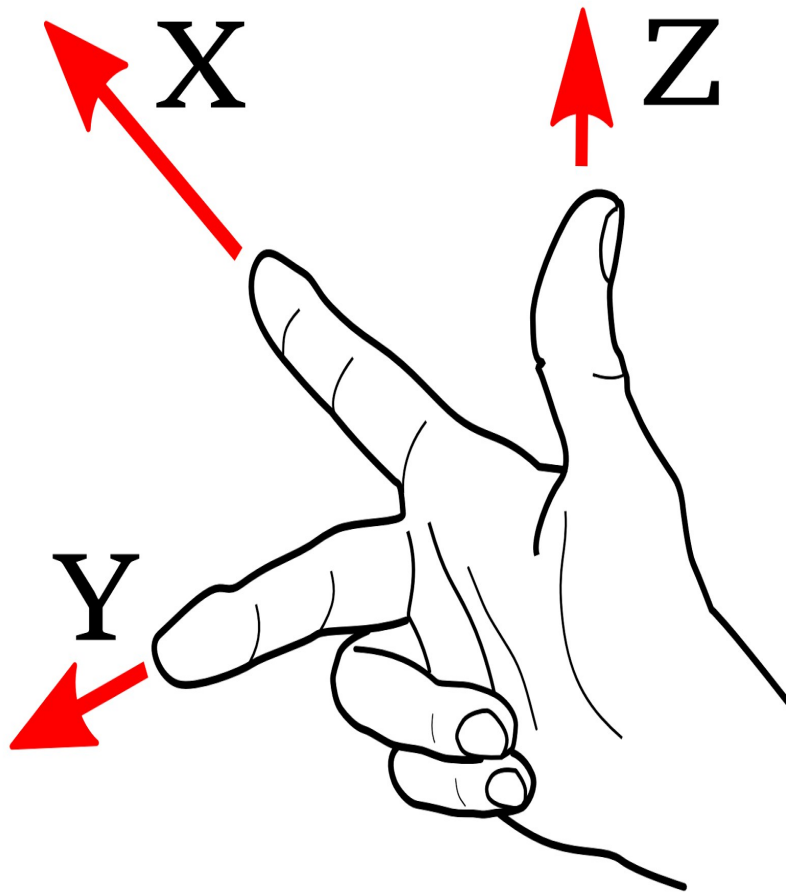
Chapter 1: Introduction.....	3
Glossary.....	4
Format.....	5
Chapter 2: G Codes.....	7
G Code Descriptions and Examples.....	10
Chapter 4: Hole Machining Canned Cycles.....	49
Drilling.....	51
Tapping.....	52
Boring.....	55
Chapter 5: Advanced Hole Machining Canned Cycles.....	57
Advanced Drilling.....	58
Advanced Tapping.....	61
Advanced Boring.....	64
Chapter 6: Tool Nose Radius Compensation.....	70
Chapter 7: M Codes.....	73
M Code Descriptions.....	73

Chapter 1: Introduction

Gcode is a special programming language that is interpreted by Computer Numerical Control (CNC) machines to create motion and other tasks. It is a language that can be quite complex at times and can vary from machine to machine. The basics, however, are much simpler than it first appears and for the most part follows an industry adopted standard. Mach4 has made a large leap closer to this standard.

An important point to remember when reading this manual: In describing motion of a machine it will always be described as tool movement relative to the work piece. In many machines the work piece will move in more axes than the tool; however the program will always define tool movement around the work piece. Axes directions follow the right hand rule, see figure A.

Figure A: Right hand rule.



Glossary

Block	A single line of G Code
Canned Cycle	Complex cycle defined by a single block of code
Dwell	Program pause with a duration defined by “P” in milliseconds. If “P” contains a decimal point, the dwell is in seconds.
EOB	End of block. Required at the end of every block of G Code. In Mach4 this is a carriage return
Feedrate	Velocity, set by F, at which an axis will move
Group	Collection of G codes that control the same function or mode, i.e. G90 and G91 positioning modes
Modal	Active until a code from the same group is called
Normal	A line perpendicular to a plane, pointing in the positive direction.
Origin	Point in a coordinate system where X, Y and Z are zero
RPM	Revolutions per minute
UPM	Units per minute (inches, millimeters, degrees, etc)
Word	A single word of G Code is a letter followed by a number. G01, X1.0, etc. are words

G	Preparatory function, G followed by a numerical code, specifies machining modes and functions
M	Miscellaneous function, M followed by a numerical code, defines program flow and can control auxiliary functions such as coolant. Can also perform machine specific functions and macros user or builder.
X, Y, Z, C	Absolute movement commands followed by a numerical value, define the end point of a motion command in the user coordinate system.
U, V, W, H	Incremental movement commands followed by a numerical value, define the distance and direction to the end point of a motion command.
S	Spindle speed, followed by numerical value of desired rpm or surface speed
T	Tool call, followed by next tool number and offset to be used
F	Followed by a numerical value to define the feedrate. The magnitude and value of which will be determined by the feed mode setting
P	Followed by a numerical value, specifies dwell time in milliseconds. (also used in other functions) If P contains a decimal point, then the time is in seconds.
N	Sequence numbers. Used for program organization and go to commands

Format

In writing Gcode programs there are some rules to be aware of as well as some general formatting guidelines that should be followed or at least considered.

The first part of any program should be a safe start up block. This line of code is used to make sure that some modes are disabled and others are set to their most common setting. An example safe start block would look like this:

```
G00 G18 G54 G40 G80
```

This block of code tells the machine that we want to be in rapid mode and using absolute position in the ZX plane of fixture offset 1. At the same time we want to cancel tool nose radius compensation and make sure any active canned cycles are canceled.

G00 – Rapid mode

G18 – ZX plane select

G54 – Fixture offset 1

G40 – Tool nose radius compensation cancel

G80 – Canned cycle cancel

It is recommended that this safe start block be used at the start of the program and also before or immediately following every tool change. It is common to restart a program from a tool change, having the safe start line there can greatly reduce the chance of a machine not acting as expected, the results of which can be aggravating at best and a crash at worst. The safe start block shown here is just an example. Every machine and every programmer are a little different and each will have their own start up block.

Jumping to the end of the program, there is not a lot required. Typically there will be a couple blocks of code to return the X and Z axes to a safe position. Shutting off the spindle and coolant or any other accessories is also a good idea here. The final block in a program is a program end code, most commonly M30 but there are other options. Make sure this final block is followed by an end of block. It is easy to forget this last EOB in a program for Mach because it is simply a carriage return and not always readily apparent. One way to make sure that there is always an EOB on your program end block is to follow it with %. Like this:

```
.  
.   
.   
M30  
%
```

This percent sign is a familiar symbol to CNC programmers in industry; however any symbol or character can be used as it will not be read by the control because of the program end before it. If there is no EOB following the percent sign it will not even show up in the program when loaded into Mach.

In between the start and end is the body of the program. There are a few rules here. Each block of code will contain a combination of words. Multiple G codes can be specified in a single block, however if more than one from the same modal group is specified the last one in the block will be valid, with the exception of group 00. Modal G codes stay active until another from the same group is called. For example; G01 is modal so it is not necessary to put it in consecutive blocks. Once active every successive positioning block will be in the G1 mode unless another code from group one is called (G00, G02, G03, etc.). All G codes not in group 0 behave this way. In the following example, the program on the right will result in the exact same motion as the program on the left. Once specified, the G1 will remain active for successive blocks.

G0 G54 G18 G40 G80	--	Safe start line
T0101	--	Tool change
S2500 M3	--	Start spindle
G99	--	Feed per revolution mode
G0 X0.0 Z1.0	G0 X0.0 Z1.0	Rapid to XY position
G1 Z0.0 F0.008	G1 Z0.0 F0.008	Move to position at feedrate
G1 X1.0	X1.0	Move to position at feedrate
G1 Z-2.0	Z-2.0	Move to position at feedrate
G1 X1.5	X1.5	Move to position at feedrate
G1 Z-4.0	Z-4.0	Move to position at feedrate
G0 G53 X0 Z0	G0 G53 X0 Z0	Rapid return X and Z to home position
M30	--	Program end and rewind

Only one M code can be specified in a single block. Same holds true for all other words.

Generally leading zeroes are not required in G Code. For example G01 and G1 are the same. The same holds true for M codes, position commands, feedrates, etc. When specifying values for position, feedrate, variables, etc., it is good practice to always use a decimal point and trailing zero, instead of X1 use X1.0. Although the decimal point is not required (in Mach X1 = X1.0) it is HIGHLY recommended.

Chapter 2: G Codes

Code	Group	Description	Modal	Page
G00	1	Rapid Move	Y	10
G01	1	Linear Feed Move	Y	10
G02	1	Clockwise Arc Feed Move	Y	11
G03	1	Counter Clockwise Arc Feed Move	Y	11
G04	0	Dwell	N	14
G09	0	Exact stop	N	15
G10	0	Overview / Tool Offset Setting	Y/N	16
G10 L2	0	Fixture Offset Setting	Y/N	18
G10 L3	0	Tool Life Management Setting	Y	20
...	0	G10 L3 Format 1 - Delete and Add	Y	21
...	0	G10 L3 Format 2 – Add or Change	Y	22
...	0	G10 L3 Format 3 – Delete Group	Y	23
G10 L20		Additional Fixture Offset Setting	Y/N	24
G11	0	G10 Cancel	N	25
G12	1	Clockwise Circle	Y	25
G13	1	Counter Clockwise Circle	Y	25
G17	2	XY Plane Select	Y	26
G18	2	ZX Plane Select	Y	26
G19	2	YZ Plane Select	Y	26
G20	6	Inch	Y	27
G21	6	Millimeter	Y	27
G28	0	Zero Return	N	28
G28.1	0	Home Axes	N	29
G30	0	2 nd , 3 rd , 4 th Zero Return	N	29
G31	1	Probe function	N	29
G32	1	Threading*	N	31
G34	1	Variable Lead Threading*	N	33
G35	1	Clockwise Circular Threading*	N	34
G36	1	Counterclockwise Circular Threading*	N	34
G40	7	Tool Nose Radius Compensation Cancel	Y	35
G41	7	Tool Nose Radius Compensation Left	Y	35
G42	7	Tool Nose Radius Compensation Right	Y	35
G50	0	Min/ Max Spindle Speed	N	36
G50	0	Local Coordinate System Setting	N	36
G50.1	11	Cancel Scaling	Y	37
G51.1	11	Scale Axes	Y	37
G52	0	Local Coordinate System Shift	Y	38
G53	0	Machine Coordinate System	N	39

G54	14	Fixture Offset 1	Y	39
G54.1	14	Additional Fixture Offsets	Y	39
G55	14	Fixture Offset 2	Y	39
G56	14	Fixture Offset 3	Y	39
G57	14	Fixture Offset 4	Y	39
G58	14	Fixture Offset 5	Y	39
G59	14	Fixture Offset 6	Y	39
G60	0	Unidirectional Approach	N	40
G61	15	Exact Stop Mode	Y	40
G64	15	Cutting Mode (Constant Velocity)	Y	40
G65	0	Macro Call	N	41
G66	12	Macro Modal Call	Y	41
G67	12	Macro Modal Call Cancel	Y	42
G73.1	9	High Speed Peck Drilling (Advanced)	Y	67
G74.1	9	Reverse Tapping (Advanced)	Y	69
G76	0	Thread Cutting Cycle (Two block style)	N	42
G76	0	Thread Cutting Cycle (One block style)	N	45
G76.1	9	Fine Boring (Advanced)	Y	70
G80	9	Canned Cycle Cancel	Y	54
G81.1	9	Hole Drilling (Advanced)	Y	64
G82.1	9	Spot Face (Advanced)	Y	65
G83	9	Face Drilling	Y	55
G83.1	9	Deep Hole Peck Drilling (Advanced)	Y	66
G84	9	Face Tapping	Y	58
G84.1	9	Tapping (Advanced)	Y	68
G84.2	9	Right Hand Rigid Tapping* (Advanced)	Y	69
G84.3	9	Left Hand Rigid Tapping* (Advanced)	Y	69
G85	9	Face Boring	Y	61
G85.1	9	Boring, Retract at Feedrate (Advanced)	Y	71
G86.1	9	Boring, Retract at Rapid (Advanced)	Y	72
G87	9	Side Drilling	Y	55
G87.1	9	Back Boring (Advanced)	Y	73
G88	9	Side Tapping	Y	60
G88.1	9	Boring, Manual Retract (Advanced)	Y	74
G89	9	Side Boring	Y	62
G89.1	9	Boring with Dwell (Advanced)	Y	75
G90.1	4	Arc Center Absolute Mode	Y	48
G91.1	4	Arc Center Incremental Mode	Y	48
G92	1	Simple Threading Cycle*	Y	49
G93	5	Inverse Time Feed	Y	50
G96	13	Constant Surface Speed*	Y	50
G97	13	Constant Speed	Y	50
G98	5	Feed Per Minute	Y	50
G98.1	10	Initial Point Return	Y	50

G99	5	Feed Per Revolution*	Y	52
G99.1	10	R Point Return	Y	52

* Implementation based on machine and control configuration

G Code Descriptions and Examples

G00 – Rapid move: Rapid moves are used to move from point to point in free space, not cutting material. These moves do not require a feed rate input as they take place at max velocity of the machine. For absolute positioning X and Z define the end point of the move in the user coordinate system. For incremental positioning U and W define the distance and direction to move from the current position.

Format: G00 X__ Z__

Example: Program a rapid move to X1.0, Z3.0. Then make an incremental move -0.5 in the X and -0.25 in the Z.

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 X1.0 Z3.0	Rapid to XZ position
U-0.5 W-0.25	Incremental move will end at X0.5, Z2.75
M30	Program end and rewind

G01 – Linear Feed Move: Linear feed moves are point to point moves in a straight line at a federate specified by F. The moves are interpolated so all axes in motion reach the end point at the same time. For absolute positioning X and Z define the end point of the move in the user coordinate system. For incremental positioning U and W define the distance and direction to move from the current position.

Format: G01 X__ Z__ F__.

Example: Program a feed move from X1.0, Z3.0 to X10.0, Z-1.0 at a feedrate of 15.0 UPM.

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G98	Feed per minute mode
G0 X1.0 Z3.0	Rapid to XZ position
G1 X10.0 Z-1.0 F15.0	Move to XZ position at feedrate
M30	Program end and rewind

G02/G03 – Arc Feed Move: Used to cut an arc at a federate specified by F. An arc is defined by its start and end points, its radius or center point, a direction, and a plane. Direction is determined by G02, clockwise, and G03, counterclockwise, when viewed from the plane's positive direction (If ZX plane is selected look down so that the X axis positive direction is pointing forward, and the Z axis positive direction is pointing to the right). See figure 2-1 for a graphic representation of the motion for G02. The start point is the current position of the machine. Specify the end point with X and Z or U and W. For absolute positioning X and Z define the end point

of the move in the user coordinate system. For incremental positioning U and W define the distance and direction to move from the current position.

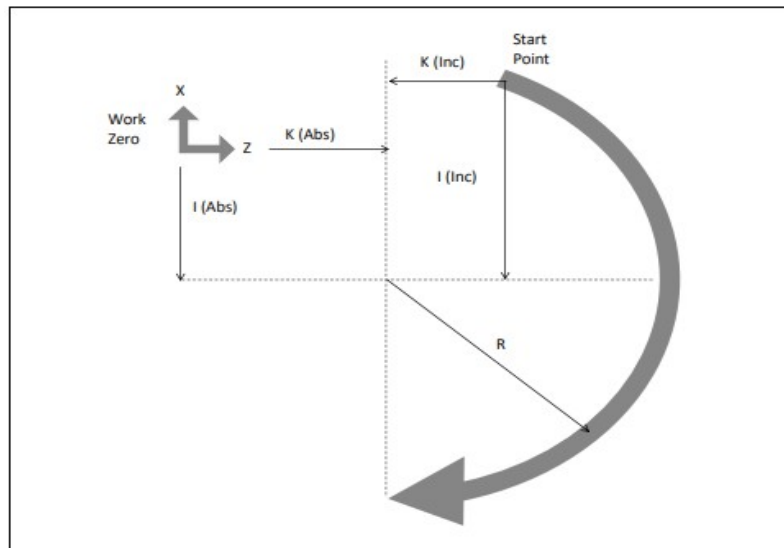


Figure 2-1: Motion of tool tip for G02 command.

Next is to specify the radius or the center point of the arc, only one or the other, not both.

- To specify the radius, use R and input the actual radius of the desired arc, see Format 2. When an arc is created knowing only start and end points and a radius there are two possible solutions, an arc with a sweep less than 180° and one with sweep greater than 180°. The sign of the radius value, positive or negative, determines which arc will be cut, see figure 2-2. A positive value for R cuts an arc with sweep less than 180°. A negative value for R cuts an arc with sweep greater than 180°.
- A more accurate and reliable way to define an arc is by specifying the center point, this is done with addresses I, and K, see Format 1. The center point must be defined in the current plane. I and K correspond to X and Z respectively. Mach has two settings for how I and K should be specified, absolute and incremental. This setting can be changed by G code, G90.1 and G91.1, or in the general tab in the Mach configuration. If arc center mode is set to incremental then I and K are the distance and direction from the start point to the center point of the arc, I is a radius value. If arc center mode is set to absolute then I and K are the absolute position of the arc center point in the current user coordinate system, I is a diameter value.

Format 1: (G18) G02/03 X__ Z__ I__ K__ F__

Format 2: (G18) G02/03 X__ Z__ R__ F__

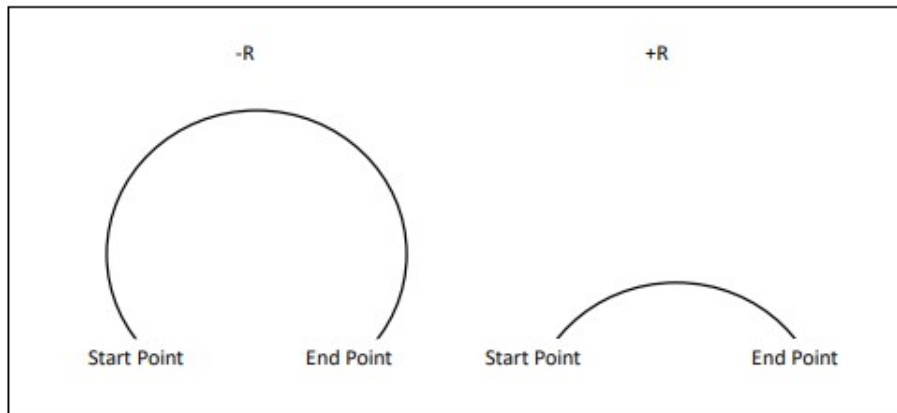


Figure 2-2: Difference between positive and negative values of R

Example: Program an arc centered at X1.0, Z-0.2 in the ZX plane with radius 0.2. Start point at X1.0, Z0.0 and sweep 90 degrees counter clockwise. Feedrate 0.006 UPR. (Arc center mode set to incremental)

Format 1:

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 X1.0 Z0.0	Position to X and Y start point
G99	Feed per revolution mode
G3 X1.4 Z-0.2 I0.0 K-0.2 F0.006	Arc move
G0 G53 X0.0	Return X to home
G0 G53 Z0.0	Return Z to home
M30	Program end and rewind

Format 2:

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 X1.0 Z0.0	Position to X and Y start point
G99	Feed per revolution mode
G3 X1.4 Z-0.2 R0.2 F0.006	Arc move
G0 G53 X0.0	Return X to home
G0 G53 Z0.0	Return Z to home
M30	Program end and rewind

Lathes with Y axis capability can also make arc moves in other planes. The function of G02 and G03 remains the same, but the format changes slightly to accommodate the additional axis and plane options. For specifying the arc center position use J for the Y axis component.

Format 1: (G17) G02/03 X__ Y__ I__ J__ F__

(G18) G02/03 X__ Z__ I__ K__ F__

(G19) G02/03 Y__ Z__ J__ K__ F__

Format 2: (G17) G02/03 X__ Y__ R__ F__

(G18) G02/03 X__ Z__ R__ F__

(G19) G02/03 Y__ Z__ R__ F__

Helical interpolations are also possible using G02 and G03; however a Y axis is required for the third coordinate of the end point, not all lathes will support this functionality. The third coordinate will define the height of the helix. See the following format for what this looks like in the XY plane:

Format 1: (G17) G02/03 X__ Y__ Z__ I__ J__ F__

Format 2: (G17) G02/03 X__ Y__ Z__ R__ F__

Example: Program a helix with radius 1.0 and center point 0.0, 0.0 in the X,Y plane, start point 0.0, .5, height 1.0 with initial Z at 0.0. Feedrate 10.0 UPM. Arc sweep should be 270° clockwise.

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 X0.0 Y.5	Position to X and Y start point
G0 Z.5	Activate tool offset 1 and move to Z rapid plane
G98	Feed per minute mode
G1 Z0.0 F10.0	Z plunge at feedrate
G2 X-.5 Y0.0 Z-1.0 I0.0 J-.5 F10.0	Helical interpolation
G0 Z.5	Retract Z to rapid plane
G0 G53 Z0.0	Return Z to home
M30	Program end and rewind

G04 – Dwell: A dwell is simply a pause in the program. The duration of the dwell is specified by P in milliseconds with no decimal point. If a decimal point is used, then P specifies seconds. No machine movement will take place during a dwell. No auxiliary codes will be turned off, i.e. if the spindle is on it will stay on, coolant will stay on, etc.

Format: G04 P__

Example: Program a 5 second dwell after positioning to X1.0, Z1.0 (using no decimal point to specify milliseconds).

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 X1.0 Z1.0	Rapid to XZ position
G4 P5000	Dwell for 5 seconds
M30	Program end and rewind

Example: Program a 5 second dwell after positioning to X1.0, Z1.0 (using decimal point to specify seconds).

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 X1.0 Z1.0	Rapid to XZ position
G4 P5.	Dwell for 5 seconds
M30	Program end and rewind

G09 – Exact Stop: G09 is a non-modal exact stop. Machine accelerations cause corners to be slightly rounded; when a true sharp corner is required G09 should be used. Although similar to G61 in function, G09 is not modal while G61 is. When G09 is included in a movement block, axis motion is decelerated to the end point of motion and the position is checked to be exactly as specified. This position check at the end of the move ensures that the machine actually reaches the desired position before moving onto the next block

Format: G01 G09 X__ Z__ F__

Example: Program a stepped shaft, first diameter is 1.0 inch for 2.0 inch length, second diameter is 1.5 inch for 2 inch length. Make a sharp corner at X1.5, Z-2.0. Feedrate 0.008 IPR.

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G99	Feed per revolution mode
G0 X0.0 Z1.0	Rapid to XY position
G1 Z0.0 F0.008	Move to position at feedrate
X1.0	Move to position at feedrate
Z-2.0	Move to position at feedrate
G9 X1.5	Move to position at feedrate, exact stop active
Z-4.0	Move to position at feedrate
G0 G53 X0 Z0	Return X and Z to home position
M30	Program end and rewind

Figure 9-1 shows what this shaft would look like, slightly exaggerated.

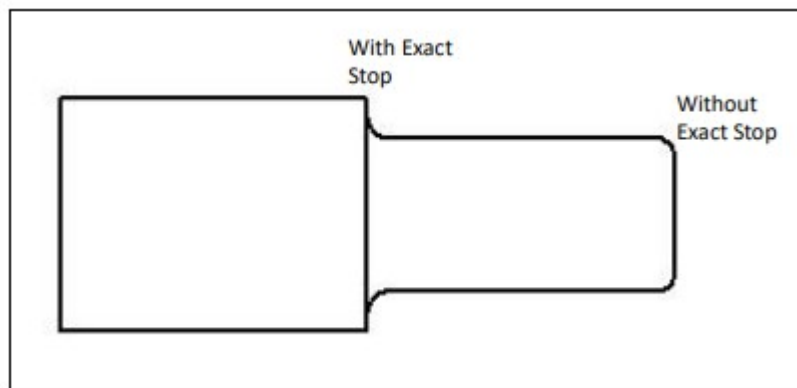


Figure 9-1: Corners with and without exact stop.

G10 – Overview: It is possible to set fixture and tool offsets in the program. This can be very useful for programming multiple fixtures that have known zero points, multi pallet machines, applying automatic compensation of tool wear, and many other situations that require changing offset values. G10 is also one of the least understood G codes and is therefore underutilized. Changing offset values in a program requires a bit of cautiousness, a mistake can easily result in ruined parts and damaged tools or machines. When used properly however, G10 can add flexibility and safety to a program and machine, especially with automation and lights out capacity or inexperienced operators.

G10 – Tool Offset Setting: G10 (without an L__ word) is used to set tool offsets in a program. Tool offset setting requires just as much care as setting of fixture offsets. Specifying in absolute causes the current value to be over written with the value in the G10 block, while incremental adds the value from the G10 block to the current value.

Format (Absolute): G10 P__ X__ Y__ Z__ R__ Q__ T__

Format (Incremental): G10 P__ U__ V__ W__ C__ Q__ T__

Or the modal version:

G10

P__ X__ Y__ Z__ R__ Q__ T__

P__ U__ V__ W__ C__ Q__ T__

...

G11

Not all values are required, if omitted that value simply will not be set. P__ specifies the offset that is to be modified. It also determines if the work shift offset is modified or if tool geometry offsets or tool wear offsets are modified. Specify P__ as follows:

- 0: work shift offset is modified.
- 1-99: The tool wear offset is modified.
- 10000+(1-99): The tool geometry offset is modified.

The remaining arguments specify the type and value of offset to be set.

X(U)	X offset
Y(V)	Y offset
Z(W)	Z offset
R(C)	Tool nose radius
Q	Imaginary Position
T	Tool tip direction

It is possible to use X__ Y__ Z__ R__ and U__ V__ W__ C__ in the same G10 block.

Example: Set X value of tool offset #5 to -5.0. Add .05 to Z axis wear offset #2.

G0 G54 G18 G40 G80	Safe start line
G10 P10005 X-5.0	Set X axis offset #5 to -5.0
G10 P2 W.05	Add .05 to Z wear for offset #2
M30	Program end and rewind

Example: Set X value of tool offset #5 to -5.0. Add .05 to Z axis wear offset #2 using the modal variant.

G0 G54 G18 G40 G80	Safe start line
G10	Start the G10 sequence
P10005 X-5.0	Set X axis offset #5 to -5.0
P2 W.05	Add .05 to Z wear for offset #2
G11	Cancels the G10 sequence
M30	Program end and rewind

G10 L2 – Fixture Offset Setting: G10 L2 is used to set fixture offsets in a program.

Starting with fixture offset setting the G10 block will look like the following:

Format (Absolute): G10 L2 P__ X__ Y__ Z__ C__

Format (Incremental): G10 L2 P__ U__ V__ W__ H__

Or the modal version:

G10 L2

P__ X__ Y__ Z__ C__

...

P__ U__ V__ W__ H__

...

G11

The value of P__ specifies which offset is being set. For the basic 6 fixture offsets P__ values are as follows:

Fixture offset (G__)	P__
Work Shift	0
54	1
55	2
56	3
57	4
58	5
59	6

The data for the fixture offset is set by X, Y, Z, C and U, V, W, H.

X(U)	X axis offset
Y(V)	Y axis offset
Z(W)	Z axis offset
C(H)	C axis offset

All values do not need to be specified, only the ones to be set. If, for example, a Z is not programmed it simply will not be changed in the fixture offset.

When specified in absolute mode (X, Y, Z, C) the values in the G10 line will be directly input into the fixture offset. When specified in incremental mode (U, V, W, H) the values will be **added** to the desired fixture offset. This is a major difference in functionality and care should be taken to make sure the correct arguments are used for the desired effect.

Example: Set G56 fixture offset to X-8.0, Y-3.0, Z-5.0, C45.0 in a program.

G0 G54 G18 G40 G80	Safe start line
G10 L2 P3 X-8.0 Y-3.0 Z-5.0 C45.0	Set G56 fixture offset values
M30	Program end and rewind

Example: Set G56 fixture offset to X-8.0, Y-3.0, Z-5.0, C45.0 in a program with the modal version.

G0 G54 G18 G40 G80	Safe start line
G10 L2	Start the G10 sequence
P3 X-8.0 Y-3.0 Z-5.0 C45.0	Set G56 fixture offset values
P3 U.1 V.1 W.1 H.1	Add .1 to G56 fixture offset values
G11	Cancel the G10 L2 sequence
M30	Program end and rewind

G10 L3 – Tool Life Management Setting: Tool life management data can be registered, changed, or deleted in a G code program. TLM is an Industrial only feature.

There are 3 G10 L3 program formats which provide for the following actions:

Function	Description
Delete all tool groups and register new data	Delete all existing tool group and tool data before adding new tool group and tool data. This effectively replaces the entire TLM data.
Add or change tool or group data	Add or change TLM data for a tool group.
Delete tool or group data	Delete TLM data for a tool group.
Specify the tool life count type	How tool life is counted for an individual group (time, frequency, and per cycle).
Specify the tool life value	The time (minutes) or cycle count value.

G10 L3 is always modal and multiple lines are used in the formats. All formats of G10 L3 will require G11 to cancel.

Tool groups are specified as positive integers from 1 to 99. A tool group is selected for a tool change with the T word by specifying a 2 digit group number and 99. For example, for tool group 1, T0199 would be used.

The tool life count type is specified with a 1 (cycle count) or 2 (time). The time based life count is best suited for cutting tools that can be characterized by the amount of cutting time they can endure and still remain viable. The cycle based life count is best suited for tools that are typically used in canned cycles that can typically be characterized by the number holes or bores they can endure and still remain viable.

The tool life value is a positive integer number that specifies the life of the tool (minutes or number of cycles the tool is used). It will count up from 0 to the life count specified by the group, at which point, the tool's life will expire.

Unlike a Fanuc, the only limit to the number of tools in a tool group is the maximum tool setting for the control, which is limited to 99 tools.

G10 L3 Format 1 - Register with deleting all groups: The G10 L3 line starts the sequence. The next line specifies the group P_, the life count value L_, and the optional life count type Q_. If Q is not specified, the value of bit 2 (0 == cycle or 1 == time) in system variable #6800 will determine the life count type. The next line(s) specify the tool number T_. T should be comprised of a 2 digit tool number plus a two digit offset number. The T_ line may be repeated for as many tools in the group as needed. Multiple tool groups can be populated by adding a new P_ L_ line sequence with its' own following T_ lines. The G10 function is then canceled with G11.

Format:

G10 L3

P_ L_ [Q_]

T_

...

P_ L_ [Q_]

T_

...

G11

Example: Delete all tool group information and add new data.

G10 L3	Start the register with delete sequence (No P_ given). This will delete all preexisting TLM data.
P1 L1000 Q1	Select group 1 with a 1000 life count and a cycle based life count type.
T0101	Add too1 1 to group 1 using offset from tool 1.
T0201	Add tool 2 but use tool 1 offset.
P2 L5000 Q2	Select group 2 with a 5000 life count with a time based life count type.
T0303	Add tool 3 to group 2.
T0404	Add tool 4 to group 2
G11	End G10 sequence.

G10 L3 Format 2 – Add or change tool group: The G10 L3 P1 line starts the sequence. The next line specifies the group P_, the life count value L_, and the optional life count type Q_. If Q is not specified, the value of bit 2 (0 == cycle or 1 == time) in system variable #6800 will determine the life count type. If the group did not exist, it is added. If the group exists the L_ and Q_ values will be updated with the specified values. The next lines specify the tool number T_. T should be comprised of a 2 digit tool number plus a two digit offset number. If the tool did not exist in the group, the tool will be added. If the tool exists in the group, the values will be updated with the specified values. The T_ line may be repeated for as many tools in the group as needed. Multiple tool groups can be populated by adding a new P_ L_ line sequence with its' own following T_ lines. The G10 function is then canceled with G11.

Format:

G10 L3 P1

P_ L_ [Q_]

T_

...

G11

Example: Add or change tool group information.

G10 L3 P1	Start the register or change tool group sequence.
P1 L1000 Q1	Select group 1 with a 1000 life count and a cycle based life count type. Assume that the group exists and that only the life count was updated.
T0101	Add tool 1 to group 1 using offset from tool 1.
T0201	Add tool 2 but use tool 1 offset. Assume that the tool already existed in the group, but was using offset 2.
P2 L5000 Q1	Select group 2 with a 5000 life count with a cycle based life count type. Assume that the group didn't exist therefore it will be added.
T0303	Add tool 3 to group 2.
T0404	Add tool 4 to group 2
G11	End G10 sequence.

G10 L3 Format 3 – Delete tool life data: The G10 L3 P2 line to starts the sequence. The next line specifies the group P_ to delete. Multiple groups can be specified for deletion on the following lines as needed. The G10 function is then canceled with G11.

Format:

G10 L3 P2

P__

...

P__

...

G11

Example: Delete selected tool groups.

G10 L3 P2	Start the delete tool life data sequence.
P1	Delete group 1.
P2	Delete group 2.
G11	End G10 sequence.

G10 L20 – Additional Fixture Offset Setting: Additional fixture offsets, G54.1 Pxx, can also be set using G10. Setting of these offsets is the same, except it is done using L20. The legacy additional fixture offsets (see fixture offset section of this manual for more details) can still be set with G10. The following table shows the additional fixture offset P number and its corresponding G10 P number as well as the legacy offsets. Note that G54.1 P1 is the same offset as G59 P7, so G10 L20 P1 and G10 L2 P7 both set the same offset values.

Starting with fixture offset setting the G10 L20 block will look like the following:

Format (Absolute): G10 L20 P__ X__ Y__ Z__ C__

Format (Incremental): G10 L20 P__ U__ V__ W__ H__

Or the modal version:

G10 L20

P__ X__ Y__ Z__ C__

...

P__ U__ V__ W__ H__

...

G11

The value of P__ specifies which offset is being set. Head shift is set using P0. The following table describes the use of P__ for the additional fixture offsets:

G54.1 P__	G10 L20 P__	Legacy G59 P__	Legacy G10 L2 P__
1	1	7	7
2	2	8	8
3	3	9	9
-	-	-	-
-	-	-	-
-	-	-	-
248	248	254	254

Example: Set G54.1 P5 fixture offset to X3.0, Y3.4, Z-10.0 in a program.

G0 G54 G18 G40 G80	Safe start line
G10 L20 P5 X3.0 Y3.4 Z-10.0	Set G54.1 P5 fixture offset values
M30	Program end and rewind

Example: Set G54.1 P5 fixture offset to X3.0, Y3.4, Z-10.0 in a program.

G0 G54 G18 G40 G80	Safe start line
G10 L20	Start the G10 L20 sequence.
P5 X3.0 Y3.4 Z-10.0	Set G54.1 P5 fixture offset values
P5 U.1 V.1 Z-.1	Add .1 to the previously set values
G11	Cancel the G11 sequence
M30	Program end and rewind

G11 – Cancel Modal G10: To end the G10 modal state a G11 must be called. G11 should be specified on its own line. The only G10 that is always modal is G10 L3 (Tool Life Management) setting.

Format: G11

G12/G13 – Circle Interpolation: These codes are used to cut a circle using the current position as the center point. A word, I, defines the radius of the circle. G12 will cut a circle in the clockwise direction and G13 will cut in the counterclockwise direction. It is also possible to cut a larger circular pocket by specifying Q for the start radius and P for the step over amount. This can be useful for cutting a circular pocket or an ID groove.

Format 1: G12/13 I__ F__

Format 2: G12/13 I__ Q__ P__ F__

See figure 12-1 for a graphic of the motion. The tool will move in the positive X direction

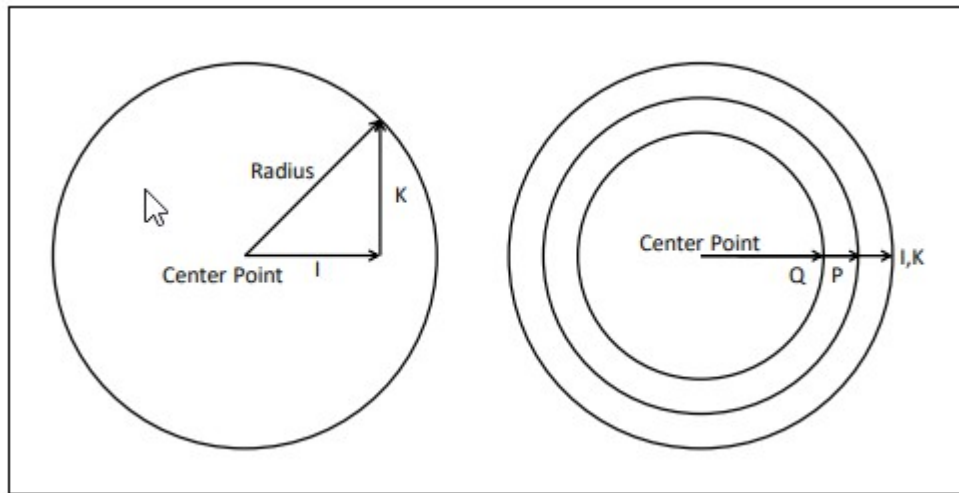


Figure 12-1: Tool motion during circular interpolation

Example: Cut a 1.0 inch radius circle centered at X1.5 Y0.25. Lead-in along the X axis.

G0 G17 G40 G80	Safe start line, absolute mode, XY plane
G0 X1.5 Y.25	Move to initial position
G13 I1.0 F30.0	Cut circle
G0 G53 Z0.0	Z axis to machine zero
M30	Program end and rewind

Example: Cut the same circle but lead-in at 45°. ($X=1*\cos(45^\circ)=.7071$, $Y=1*\sin(45^\circ)=.7071$)

G0 G17 G40 G80	Safe start line, absolute mode, XY plane
G0 X1.5 Y.25	Move to initial position
G13 I0.7071 J0.7071 F30.0	Cut circle
G0 G53 Z0.0	Z axis to machine zero
M30	Program end and rewind

G17/G18/G19 – Plane Selection: All motion and positions lie in planes defined by the machines axes. In 3-dimensional space, three axes (X, Y and Z) define three available planes XY, ZX, and YZ, see figure 17-1. The third axis defines the top of the plane, this axis is also known as the normal, see figure 17-2. Selection of a

plane is done by specifying one of three G codes: G17 for XY, G18 for ZX and G19 for YZ. These are modal G codes and will stay active until another plane is selected or the system is reset. Arcs, circles and advanced drill cycles require the selection of a plane to fully define the motion. When in lathe mode, Mach4 will default to G18, ZX plane.

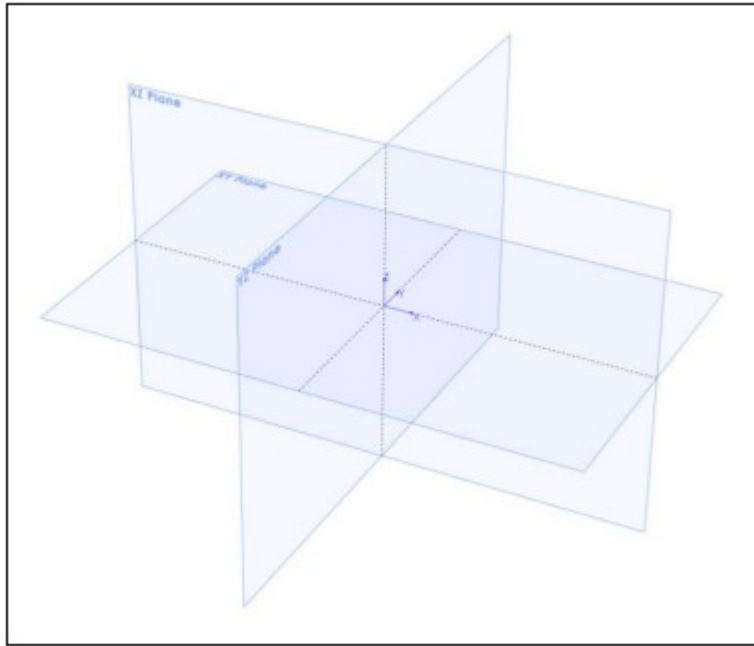


Figure 17-1: Planes

All arc and circular motion will take place on a single plane. Direction of motion, clockwise or counterclockwise, is as viewed from the normal direction, see figure 17-2.

Advanced drill cycles also require the selection of a plane. In this case all hole positions will be located in the selected plane and the normal axis will be the drilling axis. For example in the XY plane the Z axis is the drilling axis. The standard lathe drilling cycles do not require plane selection and will always operate in the ZX plane regardless of plane selection.

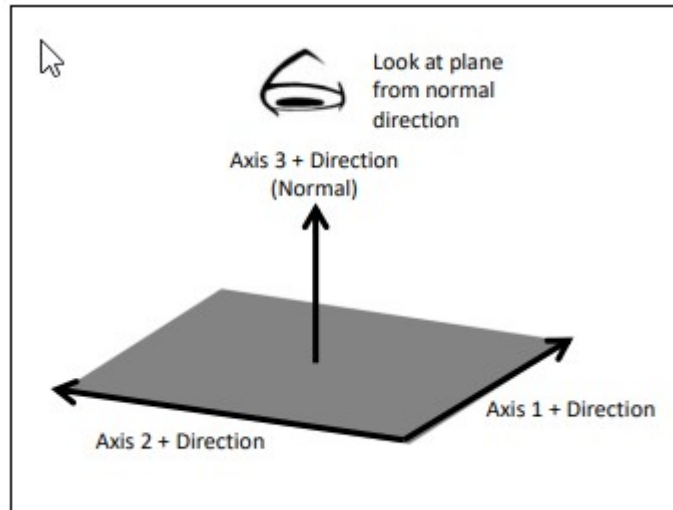


Figure 17-2: Plane normal direction

G20/G21 – Unit selection: Programming units are selected using G20 for inch and G21 for millimeter. Use these G codes to specify the units in the program only; the setting will not affect Mach configuration settings, or offsets.

G28 – Zero Return: This function is used to send one or more axes to the home position via an intermediate point. Exercise caution when using this function. If not fully understood the resulting motion may differ greatly from what is expected. When used correctly the intermediate point can be useful for avoiding obstacles in the way of a direct path to the home position, see figure 28-1.

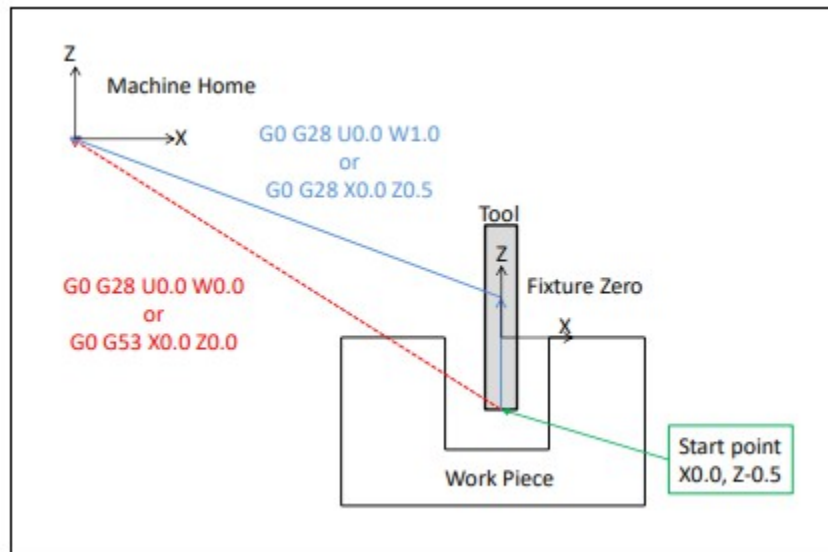


Figure 28-1: Zero return via intermediate point

Format: G28 X__ Z__ C__

Format 2: G28 U__ W__ H__

This is not a modal code and will only be active in the block in which it is specified. Following the G28 are the axes to be sent home. For example, to send the Z axis back to the home position program: G28 Z0. The value specified with the axis letter specifies the intermediate point.

Look at an example program:

G0 G54 G18 G40 G80	Safe start line
G0 X1.0 Z1.0	Rapid position to point
G28 Z0.0	Send Z axis home via point Z0
M30	Program end and rewind

Reading through the program there is a safe start up block then, the next line commands the machine to move to the point X1, Z1. in the coordinate system set by the G54 fixture offset. Now for the G28 block. This line of code, G28 Z0, gives instructions to send the Z axis to the home position via the point Z0. The motion will be as follows: First the Z axis will plunge to the point Z0 then return to home. If not intended this motion could result in a broken tool or scrapped part. To avoid this unintended motion use incremental moves of 0 as follows:

G28 W0

In this case the intermediate point is an incremental move of 0 inches resulting in no motion before the Z axis returns home.

G28.1 – Home Axis: G28.1 physically homes the specified axes. If more than one axis is specified, then the axes are homed in order specified by the control configuration. The values of the parameters are ignored except in the case of P. If P is specified, the corresponding out of band axis is homed. Valid values for P are P6 through P11. P can only be specified once.

Format: G28.1 X__ Y__ Z__ A__ B__ C__ P__

G30 – 1st, 2nd, 3rd, 4th Zero Return: G30 functions the same way as G28, moving the machine to a zero return point via an intermediate point. However, instead of sending the machine to the home position, G30 movement ends at a user definable 1st, 2nd, 3rd, or 4th zero return point, specified by P1, P2, P3, or P4 respectively. If P is omitted the 1st zero return point is selected. This is handy for tool changers that are not located at the home position, tool setters, or any number of other applications. Just as with G28, incremental commands of zero are recommended to avoid unintended motion.

Format: G30 P__ X__ Z__ C__

Format 2: G30 P__ U__ W__ H__

The zero return point is defined by # variables as follows:

Axis	P1 # Variables	P2 # Variables	P3 # Variables	P4 # Variables
X	5181	5351	5361	5371
Y	5182	5352	5362	5372
Z	5183	5353	5363	5373
A	5184	5354	5364	5374
B	5185	5355	5365	5375
C	5186	5356	5366	5376

The position values in the # variables can be set in a program, MDI mode or in the Regfile diagnostics window.

G31/G31.X – Probe function: Also known as skip function, G31 allows the use of part and tool probes. Multiple probes can be used, G31.0 (or G31) for probe 0, G31.1 probe 1, G31.2 probe 2 and G31.3 probe 3. Motion is defined along linear axes, in a similar format to G01, with a feedrate.

Format: G31 X__ Z__ F__

The machine will move toward the specified end point, at the same time it is looking for the probe input to be activated. When the probe input is activated the current position is recorded to # variables according to the table below and motion is stopped. The recorded position can then be used to calculate tool offsets, work offsets, measure parts, etc.

Axis	Absolute Position # Variables	Machine Position # Variables
X	5061	5071
Y	5062	5072
Z	5063	5073
A	5064	5074
B	5065	5075
C	5066	5076

G32 – Threading: Equal lead straight, tapered and scroll threads can be cutting using the thread cutting command. Spindle speed feedback from an encoder, index pulse, tachometer or other device is required for this operation. The syncing of the feed axis to the spindle speed creates an accurate thread; however, axis acceleration can cause variations in the thread lead especially at the start and end of the thread. To avoid these errors, program a slightly longer thread to give the axis time to accelerate. The amount of extra thread length will vary based on machine specifications.

Changes in spindle speed and feedrate will impact thread quality and accuracy. Using constant surface speed mode can also result in variations in thread lead when cutting tapered or scroll threads, use G97 constant RPM mode instead. During the threading move the spindle speed and feedrate overrides will be disabled and the machine will run at 100%. Feed hold will be delayed, if pressed the machine will stop at the end of the threading move.

Format: G32 X__ Z__ F__

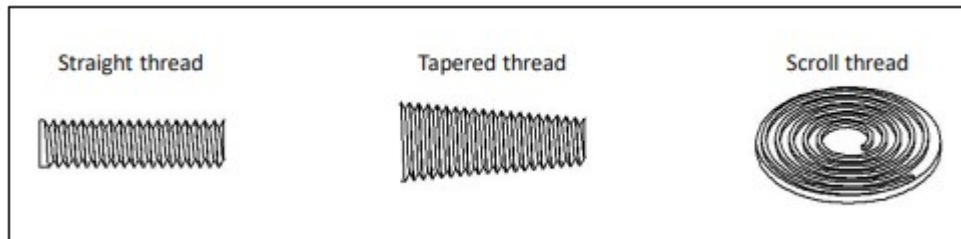


Figure 32-1: Basic thread types

The G32 threading cycle is a single linear move synced to the spindle speed. F specifies the lead or pitch of the thread when in the feed per revolution (G99) mode. For example a 20 TPI thread would have a pitch of .05 inches, so program F.05. By default the movement is basic linear move with synced feedrate. By default the lead is assumed to be along the major axis of the current plane selection; for G17 XY plane lead is along the X axis, G18 ZX plane is along the Z axis, G19 is along the Y axis.

Example: Thread a ¼-20 rod held in the spindle, 1" length of thread.

G0 G54 G18 G40 G80	Safe start line
G0 X1.0 Z0.1	Rapid position
G97 S1000 M3	Start spindle at 1000 RPM
G0 X0.22	Move to start position for rough
G32 X0.22 Z-1.0 F.05	Cut straight thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
X0.21	Move to start position for finish
G32 X0.21 Z-1.0 F.05	Cut straight thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
G53 X0.0 Z0.0 M5	Return home and stop spindle
M30	Program end and rewind

Threads can also be cut with a taper by adding the proper end point.

Example: Cut same thread as previous example with 0.03 taper.

G0 G90 G54 G18 G40 G80	Safe start line
G0 X1.0 Z0.1	Rapid position
G97 S1000 M3	Start spindle at 1000 RPM
G0 X0.22	Move to start position for rough
G32 X0.25 Z-1.0 F.05	Cut tapered thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
X0.21	Move to start position for finish
G32 X0.24 Z-1.0 F.05	Cut tapered thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
G53X0.0 Z0.0 M5	Return home and stop spindle
M30	Program end and rewind

G34 – Variable Lead Threading: Threads with a variable lead, or pitch, can be cut by specifying the start lead and the change in lead per revolution of the screw.

Format: X__ Z__ F__ K__

G32 threading and G34 variable lead threading are programmed and function the same way with the exception of address K. K specifies the change in thread lead per revolution. All requirements are the same, and not all machines will have this capability.

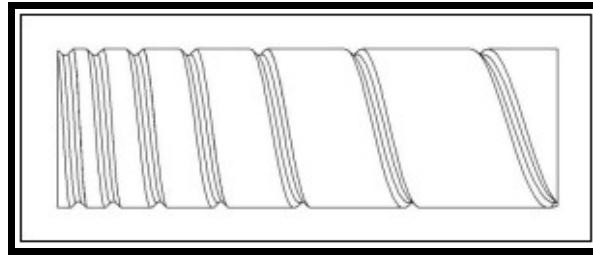


Figure 34-1: Variable lead thread

Example: Thread a ¼ rod held in the spindle, 1" length of thread. Start with 0.1 lead with a decrease in lead of 0.02 per revolution.

G0 G54 G18 G40 G80	Safe start line
G0 X1.0 Z0.1	Rapid position
G97 S1000 M3	Start spindle at 1000 RPM
G0 X0.22	Move to start position for rough
G32 X0.22 Z-1.0 F0.1 K-0.02	Cut variable lead thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
X0.21	Move to start position for finish
G32 X0.21 Z-1.0 F0.1 K-0.02	Cut variable lead thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
G53 X0.0 Z0.0 M5	Return home and stop spindle
M30	Program end and rewind

G35/G36 – Circular Threading: Circular threading allows cutting threads along the profile of an arc. These can be useful for a number of applications, including but not limited to worm gears and screws, see figure 35-1. The thread follows an arc, defined in the same way as an arc is programmed with G02/G03. The thread lead is along the major axis of the current plane selection; in the G17 XY plane the X axis is the major axis, G18 ZX plane the Z axis is the major axis, and G19 YZ plane the Y is the major axis. The orientation of the major axis also limits the range of the specified arc. Figure 35-2 shows the allowable range shaded in gray.

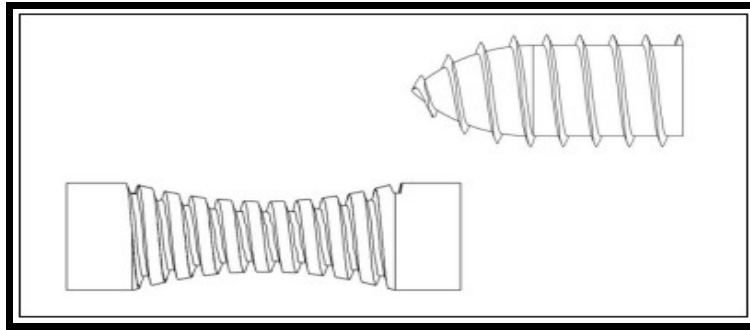


Figure 35-1: Circular thread examples

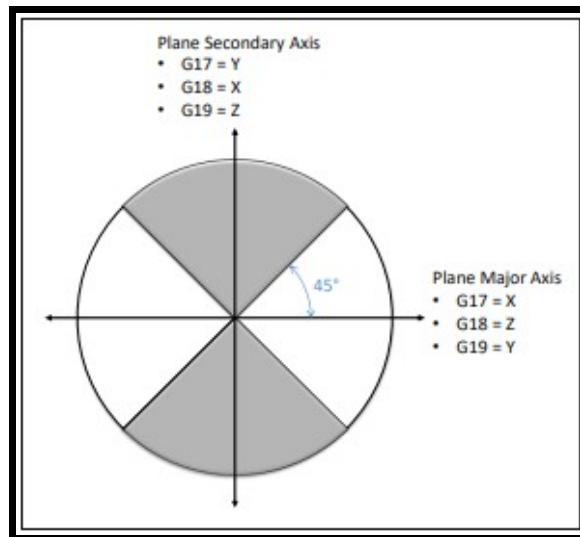


Figure 35-2: Circular thread allowable ranges

The programming format contains the same addresses as G02/G03 and G32; they perform the same function as well.

Format 1: (G17) G35/36 X__ Y__ I__ J__ F__ Q__

(G18) G35/36 X__ Z__ I__ K__ F__ Q__

(G19) G35/36 Y__ Z__ J__ K__ F__ Q__

Format 2: (G17) G35/36 X__ Y__ R__ F__ Q__

(G18) G35/36 X__ Z__ R__ F__ Q__

(G19) G35/36 Y__ Z__ R__ F__ Q__

Example: Thread a ¼ rod held in the spindle, 1” length of thread. Start with 0.1 lead with a decrease in lead of 0.02 per revolution.

G0 G54 G18 G40 G80	Safe start line
G0 X1.0 Z0.1	Rapid position
G97 S1000 M3	Start spindle at 1000 RPM
G0 X0.22	Move to start position for rough
G32 X0.22 Z-1.0 F0.1 K-0.02	Cut variable lead thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
X0.21	Move to start position for finish
G32 X0.21 Z-1.0 F0.1 K-0.02	Cut variable lead thread
G0 X0.3	Retract X axis
Z0.1	Retract Z axis
G53 X0.0 Z0.0 M5	Return home and stop spindle
M30	Program end and rewind

G40 – Tool Nose Radius Compensation Cancel: Cancels the tool nose radius compensation mode.

G41/G42 – Tool Nose Radius Compensation Left/Right: Enables tool nose radius compensation to the left (G41) or right (G42) of the cutter path by an amount specified by the tool nose radius and tip direction.

Format: G1 G42 X__ Z__ F__

For detailed information see the [tool nose radius compensation](#) section of this manual.

G50 – Minimum and Maximum Spindle Speed: In the G96 constant surface speed (CSS) mode the spindle RPM will vary based on current X position. The spindle speed will increase as the diameter gets smaller and decrease as the diameter gets bigger. Machining in this mode can produce superior finishes and accuracy. However, some set ups can only handle a certain RPM range. Castings, or similar near net shape stock, are a prime example of this. An unbalanced part can create harsh vibration when spun too fast. In cases like these the maximum RPM can be limited using G50 while still being able to take advantage of the benefits of CSS mode. However, in CSS mode excessive decelerations can occur when positioning the tool or moving away from the part for tool changes. In these cases the minimum speed can be very useful.

Format: G50 S__ Q__

In this format S specifies the maximum spindle speed and Q specifies the minimum speed.

G50 – Local Coordinate System Setting: The coordinate system can be set by issuing G50 in the program. This is equivalent to G92 in the mill interpreter. This function differs from G52 (Local Coordinate System Shift) in the way that it is specified. While G52 is specified with shift values, G50 is specified with the desired axis position. The affect is global and should be used with caution.

Format: G50 X__ Y__ Z__ A__ B__ C__

Using the above format specify a value for the desired axis. When G50 is specified the position DRO's are updated to the values specified. The local coordinate system setting will be canceled when a G92.1 is specified or the system is reset.

G0 G54 G18 G40 G80	Safe start line
G0 X4.0 Y0.0 Z1.0	Rapid move, current position X4, Y0, Z1
G50 X1.0 Y2.0 Z3.0	Set local coordinate system, current position X1, Y2, Z3
G92.1	Cancel local coordinate system, current position X4, Y0, Z1
M30	Program end and rewind

G50.1 – Scaling Cancel: Cancels scaling.

G51.1 – Scaling/Mirroring Function: When activated the scaling function multiplies all commanded positions by the specified scale factor. The DROs and offsets are not affected, but motion commanded from a program or the MDI screen is affected.

Format: G51 X__ Z__ C__

Specify the axis to be scaled and the desired scale factor. For example:

G0 G54 G18 G40 G80	Safe start line
G0 X4.0 Y0.0 Z1.0	Rapid position to point (X position is 4.)
G51 X2.0	Activate scaling on X axis (scale factor = 2)
G0 X5.0	Rapid position to point (X position is 10.)
G50	Cancel Scaling
G0 X5.0	Rapid position to point (X position is 5.)
M30	Program end and rewind

When scaling is active position moves will be calculated by multiplying the commanded position by the scale factor. In the example above the scale factor on the X axis is set to 2, then a move to X5.0 is commanded. The actual end position of this move will be $5 * 2 = 10$. So the X axis moves to 10.

Exercise caution when using scaling, the results can be unpredictable depending on program complexity. For example if G52 X2 Y4 is programmed followed by an arc move in the XY plane, the arc will NOT be scaled 2x in the X axis and 4x in the Y axis to obtain an ellipse. The start and end positions will be as expected, but the motion from one to the other may not be. Check and double check the tool path display before running the program.

To mirror a program, enter a negative scale value. For example:

G0 G90 G54 G18 G40 G80	Safe start line
G0 X4.0 Y0.0 Z1.0	Rapid position to point (X position is 4.)
G51 X-1.0	Mirror X axis (scale factor = 1)
G0 X5.0	Rapid position to point (X position is -5.)
G50	Cancel Scaling
G0 X5.0	Rapid position to point (X position is 5.)
M30	Program end and rewind

G52 – Local Coordinate System Shift: The local coordinate system setting is a programmable work shift. The setting is global; the entire system is shifted by the specified values. Every fixture offset will be affected, although the actual fixture offset values will not be changed.

Format: G52 X__ Z__ C__

To activate a local coordinate system with G52 use the above format. Setting of a local coordinate system is just like setting a fixture offset. In the G52 block specify the desired axes to set, and the value of the shift. For example (see figure 52-1 for the tool path):

G0 G54 G18 G40 G80	Safe start line
G0 X-4.0 Y0.0 Z1.0	Rapid position to point
G12 I2.0 F30.0	Cut circle with radius 2.0
G52 X7.0	Local Coordinate system active, X offset = 7
G0 X-4.0 Y0.0 Z1.0	Rapid to same start point
G12 I2.0 F30.0	Cut same circle with radius 2.0
G52 X0.0	Local coordinate system canceled
M30	Program end and rewind

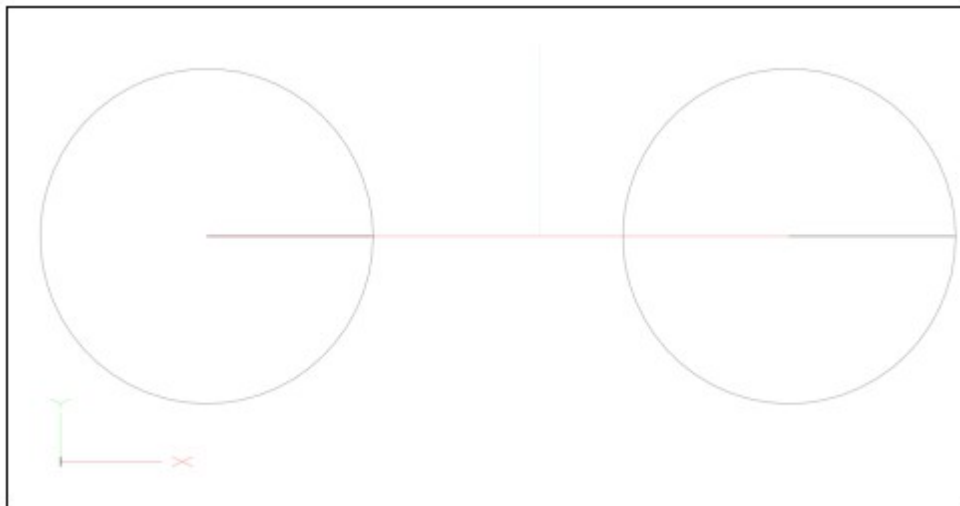


Figure 52-1: Coordinate system shift example program path

Once set, the setting will remain until canceled by another G52 or the system is reset. As in the example above, a local coordinate system can be canceled by specifying the axis with a value of zero. This effectively sets the local coordinate system shift to zero, giving it no effect.

G53 – Machine Coordinate System: Although the majority of machine positioning will take place in a user created coordinate system it is sometimes beneficial to program positions in the machine coordinate system. G53 is a non-modal, only active for the block in which it is specified, G code that allows the user to make positioning moves in the machine coordinate system. This can be useful for moving to a load/unload position at the end of a program or moving to a tool change location in a tool change macro. This is also a much safer way to move back to the machine home position, G53 X0 Y0 Z0, than using G28 as there is no intermediate position to be concerned with.

Format: G53 X__ Z__ C__

G0 G54 G18 G40 G80	Safe start line
G0 X4.0 Y0.0 Z1.0	Rapid position in G54
...	Body of program
G53 Z0.0	Return directly to Z home position
G53 X10.0 Y0.0	Move to load/unload position
M30	Program end and rewind

In the example above, the last two positioning moves are made in the machine coordinate system. These two blocks could be the same for every program in this machine.

G54-G59 – Fixture Offset: Fixture offsets are used to define the position of a work piece in the machine. In a basic 2 axis lathe the fixture offset will most commonly be used to define the position of the end of the work piece in the Z axis, see figure 54-1. The figure shows only one way of using the fixture offset, there are almost as many methods of fixture and tool offset setting as there are machinist/operators. Use a method that works the best for you and your machine. The G codes G54-59 are used to select the active fixture offset in a program. It is a modal call so the fixture offset will stay active until another is called or the system is reset. It is possible to use multiple fixture offsets in a single program.

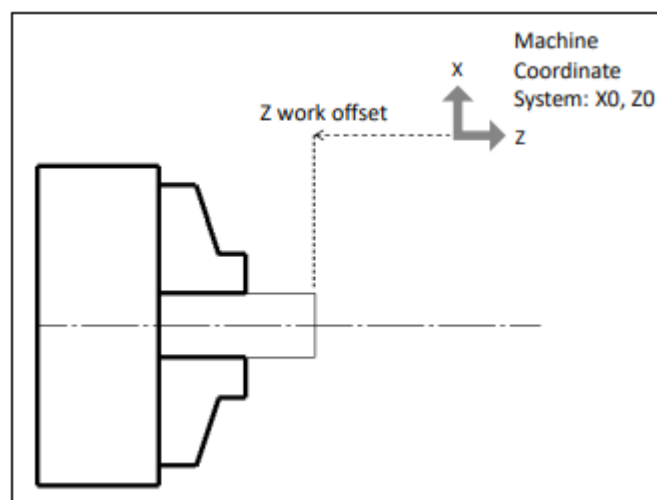


Figure 54-1: Fixture offset

G54.1 – Additional Fixture Offsets: Additional fixture offsets are provided for users with many fixtures/parts/setups. There are 248 additional offsets available.

Format: G54.1 P__

P specifies the number of the additional offset, 1 thru 248.

Previous version of Mach use G59 P7, P8 and so on. These legacy offsets can still be used in place of the G54.1. G59 P7 = G54.1 P1, G59 P8 = G54.1 P2, and so on. G54.1 is more consistent with industry machines.

G60 – Unidirectional Approach: In cases where mechanical backlash causes positioning errors unidirectional approach can be used to increase accuracy. G60 is a non-modal code, when specified in a movement block motion will move to the end point from a parameter defined distance and direction. The distance and direction of the approach movement is specified by setting values in # variables as shown in the following table:

Axis	# Variable
X	5440
Y	5441
Z	5442
A	5443
B	5444
C	5445

Format: G60 G0/G1 X__ Z__

When unidirectional approach is used in a drill cycle the Z axis motion is not affected. G76.1 and G87.1 boring cycles have a tool shift that is also not affected by the G60 unidirectional approach.

G61 – Exact Stop Mode: In exact stop mode the machine will decelerate to a complete stop at the end of each commanded move, see figure 9-1. This is a modal code, once activated it will remain on until canceled. For sharp corners and simple positioning this mode works well. However, when the code gets more complex the exact stop mode will produce jerky motion and witness marks on the work piece. For most turning jobs use G64.

G64 – Constant Velocity Mode: In constant velocity mode Mach will try to maintain feedrate even around sharp corners. As a result sharp corners will be slightly rounded and the machine may never reach the programmed point before a direction change. The magnitude of these position errors will be determined by the acceleration capability of the machine and the programmed feedrate. In most cases the error will be too small to notice or affect part function. Cutting will be faster and smoother with better finishes and no witness marks from stopping. This will be the active mode for the majority of machining time. It is modal and will be active until exact stop mode is activated.

G65 – Macro Call: Macros work like subprograms (see M98 on page 82) but allow values to be passed from the main program in the form of local variables. Macro programs can use these local variables passed to it to change part dimensions, select features, adjust feedrates, or anything else the user could need to change.

Format: G65 P____ A__ B__ C__ ...

The desired program number to be called is specified by P. The remaining arguments are determined by the macro program being called. The values of these arguments will be passed to local variables for use in the macro program. The available arguments and corresponding variables are shown in the table below:

Address	Variable	Address	Variable	Address	Variable
A	#1	I	#4	T	#20
B	#2	J	#5	U	#21
C	#3	K	#6	V	#22
D	#7	M	#13	W	#23
E	#8	Q	#17	X	#24
F	#9	R	#18	Y	#25
H	#11	S	#19	Z	#26

The G65 macro call is non-modal and has no option for repeating, the macro subprogram will be run only once per G65 call. For more information on macro programming and the availability and use of # variables see the Mach4 Macro Programming Guide.

G66 – Macro Modal Call: Sometimes it is useful to run the same macro in different positions (similar to drilling canned cycles) or with different parameters. G66 is a modal macro call, the macro is called and values passed in the same way as G65. Execution of the macro does not occur in the same block as the G66. Macro execution occurs at every position move following G66, prior to the G67.

Format: G66 P___ A__ B__ C__ ...
 G00 X__ Z__ C__
 G67

G0 G54 G18 G40 G80	Safe start line
G0 X4.0 Z1.0	Rapid position in G54
G66 P9100 A4.0 B5.0 R1.0	Modal macro call (macro program not executed)
G0 X3.0 Z0.0	Macro 9100 executed at X3.0 Z0.0
G0 X1.0 Z-2.0	Macro 9100 executed at X1.0 Z-2.0
G67	Modal macro call cancel
G53 Z0.0	Return directly to Z home position
G53 X10.0	Move to load/unload position
M30	Program end and rewind

More information on macro programming is available in the Mach4 Macro Programming Guide.

G67 – Macro Modal Call Cancel: Cancels the macro modal call.

G76 – Multiple Thread Cutting Cycle (two block type): The multiple thread cutting cycle simplifies the cutting of threads. This cycle is specified in two blocks and will rough and finish threads with or without lead outs and with different infeed options. By comparison, using G32 would require at least 3 blocks of code per pass.

Format: G76 P_____ Q__ R__ K__
 G76 X__ Z__ R__ P__ Q__ F__

First line:

- P: 6 digit number specifying the number of finish passes, length of lead out, and the angle of the threading tool. Each parameter is two digits. The number of finish passes can be specified from 01 to 99. The lead out is specified in the number of leads from 0.0 to 9.9 (00 to 99). Then the tool angle. For example: P053060 = 05 finish passes, 3.0 leads to lead out, and a 60 degree tool.
- Q: Specifies the minimum cutting depth. If the calculated depth of cut becomes less than this value, then the specified minimum is used.
- R: Finish allowance.
- K: Infeed type. There are 4 possible infeed selections, 1 to 4. The 4 options are combinations of two infeed types and two depth of cut types. The two infeed types are flank and alternate flank, see figure 76-1. In the flank infeed mode each depth of cut moves down at the angle for the thread, following the trailing flank of the tool. Alternate flank changes from leading to trailing flank for each pass. The two depth of cut types are constant volume and constant depth. In constant volume the depth of cut will get smaller (down to the minimum specified by Q) for each pass to maintain a constant volume of material being removed. This is usually better for tool life and thread quality. Constant depth is just that. Each depth of cut is the same down to the finish allowance.
 - 1 constant volume flank infeed
 - 2 constant volume alternate flank infeed
 - 3 constant depth flank infeed
 - 4 constant depth alternate flank infeed.

See Figure 76-1 for a diagram of the infeed types.

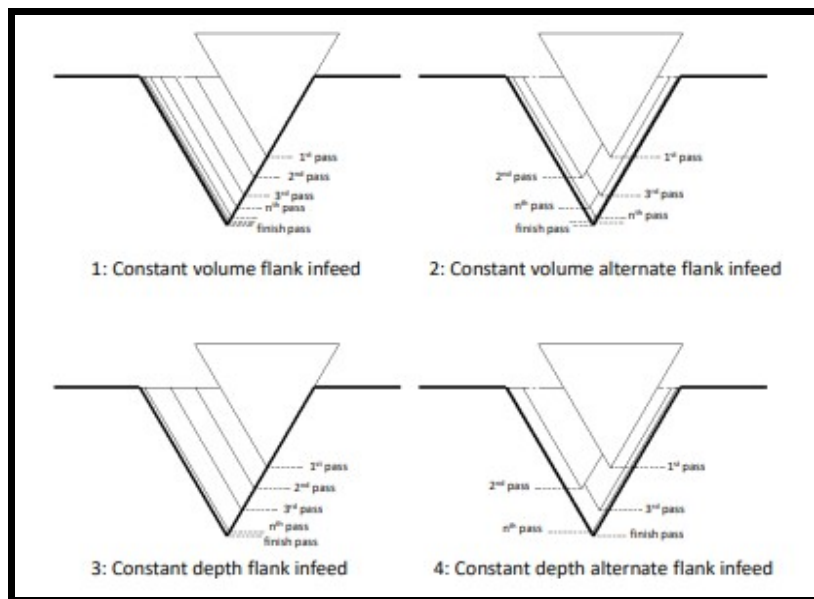
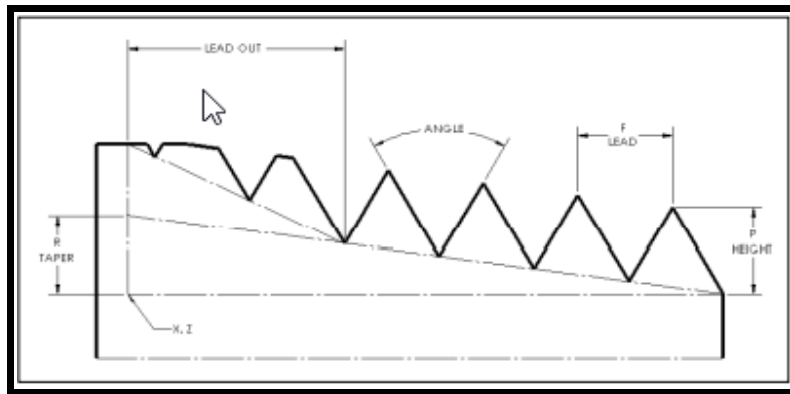


Figure 76-1: Threading infeed types

Second line:

- X, Z: Coordinates of the end point of the thread. Can also be specified as U and W for incremental distance and direction from the current position.
- R: Taper amount in the X axis, specified in radius. If omitted or 0 is specified a straight thread will be cut.
- P: Height of the thread in radius.
- Q: 1st depth of cut in radius.

F: Lead of thread



Threading arguments Figure 76-2: G76

Example: Cut a 1/2-20 straight thread 1 inch long, starting at Z0 with a 2 lead lead-out.

Setup:

- Finish passes, we'll do 2
- Leadout, 2.0 from the description
- Angle, 60 degrees is the most common
- Minimum depth of cut, we'll say 0.007 inch
- Finish allowance 0.005
- Alternate flank infeed at constant volume

So that gives us our first line of the G76 call: G76 P022060 Q0.007 R0.005 K1

- X finish diameter is 0.433
- Z finish position is -1.0
- We want a straight thread so omit R
- P is the height of the thread, 0.043
- First depth of cut will be 0.02
- Lead at 20 thread per inch is 0.05

Second line will be G76 X0.433 Z-1.0 P0.043 Q0.02 F0.05

G0 X1.0 Z0.1	Rapid position in G54
G76 P022060 Q0.007 R0.005 K1	First line of threading cycle
G76 X0.433 Z-1.0 P0.043 Q0.02 F0.05	Second line of threading cycle, cut threads
G53 Z0.0	Return directly to Z home position
G53 X10.0	Move to load/unload position
M30	Program end and rewind

To get thread dimensions, standard major/minor diameters, thread height, etc., consult the [Machinery's Handbook](#) or similar.

Some precautions:

1. When threading the feed override is not effective. Threading moves will always take place at 100% feed.
2. Feed hold will NOT pause execution during a threading move. Motion will be stopped after the tool retracts from the current cut.
3. Leave room for the machine to accelerate before starting into the cut or variations in thread pitch could occur.
4. It is not suggested to use CSS mode during threading, use constant RPM (G97).

G76 – Multiple Thread Cutting Cycle (one block type): The multiple thread cutting cycle simplifies the cutting of threads. This cycle is specified in two blocks and will rough and finish threads with or without lead outs and with different infeed options. By comparison, using G32 would require at least 3 blocks of code per pass.

Format: G76 X__ Z__ I__ K__ D__ A__ P__ F__

- X, Z: Coordinates of the end point of the thread. Can also be specified as U and W for incremental distance and direction from the current position.
- I: Taper amount in the X axis, specified in radius. If omitted or 0 is specified a straight thread will be cut.
- K: Height of the thread in radius.
- D: 1st depth of cut in radius.
- A: The tool angle.
- P: Infeed type. There are 4 possible infeed selections, 1 to 4. The 4 options are combinations of two infeed types and two depth of cut types. The two infeed types are flank and alternate flank, see figure 76-1. In the flank infeed mode each depth of cut moves down at the angle for the thread, following the trailing flank of the tool. Alternate flank changes from leading to trailing flank for each pass. The two depth of cut types are constant volume and constant depth. In constant volume the depth of cut will get smaller (down to the minimum specified by Q) for each pass to maintain a constant volume of

material being removed. This is usually better for tool life and thread quality. Constant depth is just that. Each depth of cut is the same down to the finish allowance.

- 1 constant volume flank infeed
- 2 constant volume alternate flank infeed
- 3 constant depth flank infeed
- 4 constant depth alternate flank infeed.

See Figure 76-1 for a diagram of the infeed types.

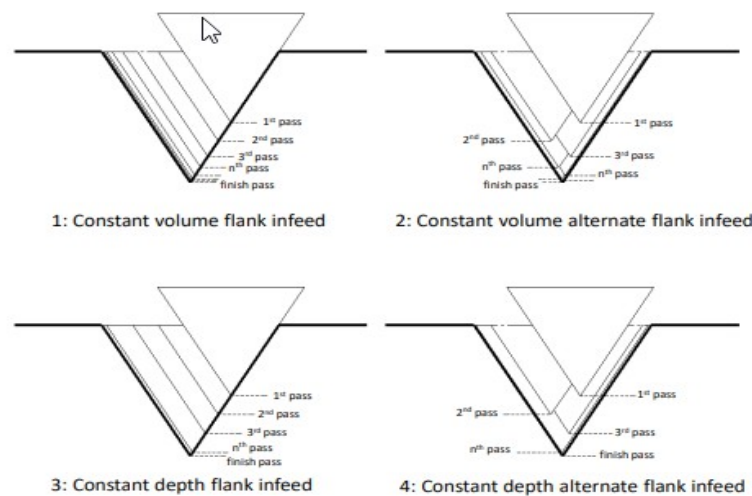


Figure 76-1: Threading infeed types

- F: Lead of thread

The leadout (thread finishing) is controlled by M23 to turn on leadout and M24 to turn off leadout. The leadout is specified in the number of leads from 0.0 to 9.9 (00 to 99 integer value) in parameter 5144.

Example: Cut a 1/2-20 straight thread 1 inch long, starting at Z0 with a 2 lead lead-out.

Setup:

- Z finish position is -1.0
- X finish diameter is 0.433
- Leadout, 2.0 from the description
- A 60 degree tool angle, the most common
- Alternate flank infeed at constant volume
- We want a straight thread so omit I or specify I0

- The height of the thread is 0.043
- First depth of cut will be 0.02
- Lead at 20 thread per inch is 0.05

So that gives us our line of the G76 call: G76 X0.433 Z-1.0 A60 I0 K0.043 D0.02 P2 F0.05

G0 G54 G18 G40 G80	Safe start line
M23	Turn on lead out
#5144=20	Lead out over two thread leads
G0 X1.0 Z0.1	Rapid position in G54
G76 X0.433 Z-1.0 A60 I0 K0.043 D0.02 P2 F0.05	Threading cycle
M24	Turn off lead out
G53 Z0.0	Return directly to Z home position
G53 X10.0	Move to load/unload position
M30	Program end and rewind

To get thread dimensions, standard major/minor diameters, thread height, etc., consult the [Machinery's Handbook](#) or similar.

Some precautions:

1. When threading the feed override is not effective. Threading moves will always take place at 100% feed.
2. Feed hold will NOT pause execution during a threading move. Motion will be stopped after the tool retracts from the current cut.
3. Leave room for the machine to accelerate before starting into the cut or variations in thread pitch could occur.
4. It is not suggested to use CSS mode during threading, use constant RPM (G97).

G80-G89 – Hole Machining Canned Cycles: Canned cycles are special G codes used to simplify programming. See the [Hole Machining Canned Cycles](#) section of this manual for detailed information.

Standard Hole Machining Cycles	
G80	Canned Cycle Cancel
G81	Drilling
G82	Spot Face
G83	Deep Hole Peck Drilling
G84	Tapping
G85	Boring, Retract at Feed, Spindle On
G86	Boring, Retract at Rapid, Spindle Off
G87	Back Boring
G88	Boring, Manual Retract

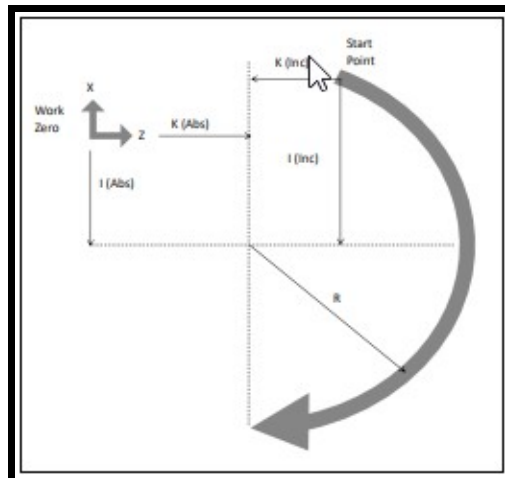
G89	Boring, Dwell, Retract at Feed, Spindle On
-----	--

G73.1-G89.1 – Advanced Hole Machining Canned Cycles: These canned cycles provide additional functionality over the standard lathe cycles. See the [Advanced Hole Machining Canned Cycles](#) section of this manual for detailed information.

Advanced Hole Machining Cycles	
G73.1	High Speed Peck
G74.1	Reverse Tapping
G76.1	Fine Boring
G81.1	Drilling
G82.1	Spot Face
G83.1	Deep Hole Peck Drilling
G84.1	Tapping
G84.2	Right Hand Rigid Tapping
G84.3	Left Hand Rigid Tapping
G85.1	Boring, Retract at Feed, Spindle On
G86.1	Boring, Retract at Rapid, Spindle Off
G87.1	Back Boring
G88.1	Boring, Manual Retract
G89.1	Boring, Dwell, Retract at Feed, Spindle On

G90.1/G91.1 – Absolute/Incremental Arc Center Mode: This setting affects arcs when programmed in the I, J, K format. In absolute arc center mode the I, J, K values designate the position of the arc center in the user coordinate system. In incremental arc center mode the I, J, K values designate the distance and direction to the arc center from the start point. See figure 2-1 for a graphical description.

Figure 2-1: Arc center types



Example: Program an arc centered at 1.0, 1.0 in the ZX plane with radius 2. Start point at 1.0,3.0 and sweep 90 degrees counter clockwise. Program two times, once in incremental arc center mode and once in absolute arc center mode.

G0 G54 G18 G40 G80	Safe start line
G91.1	Incremental arc center mode
T0101	Tool change
G0 X1.0 Z3.0	Position to X and Z start point
G3 X5.0 Z1.0 I0.0 K-2.0 F10.0	Arc move
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G0 G54 G18 G40 G80	Safe start line
G90.1	Absolute arc center mode
T0101	Tool change
G0 X1.0 Z3.0	Position to X and Z start point
G3 X5.0 Z1.0 I1.0 K1.0 F10.0	Arc move
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Notice the difference in the I values of the example programs. Both programs will produce the same arc.

G92 – Simple Threading Cycle: G92 is used for simple threading, however, multiple passes for threading are possible by specifying the X locations of additional passes on following blocks. Straight threads are made by specifying X, Z, and F. By adding a R value, a pipe or taper thread is cut. The amount of taper is referenced from the target. That is, R is added to the value of X at the target. The thread leadout can be controlled with either M23 (leadout on) or M24 (leadout off) issued before the G92. If M23 is used, the leadout chamfer is controlled by parameter 5130 (G92 CHAMFER PITCH).

Parameter 5130: An integer specifying .1 pitch increments. The default is 10, which is 1.0 pitch.

Format: G50 X__ Z__ R__ F__

G00 G18 G20 G40 G80 G99	Safe start line
G50 S1000	Limit spindle to 1000 RPM
G97 S500 M03	CSS off, spindle CW
G00 G54 X0 Z0.25	Rapid to 1 st position
M08	Coolant on
M23	Two axis chamfer exit
X1.2 Z.2	Rapid to clear position
G92 X.980 Z-1.0 F0.0833	Begin threading cycle
X.965	2 nd pass
X.955	3 rd pass
X.945	4 th pass
X.935	5 th pass
X.925	6 th pass
X.917	7 th pass

X.910	8 th pass
X.905	9 th pass
X.901	10 th pass
X.899	11 th pass
G00 G53 X0 M09	X home, coolant off
G53 Z0 M05	Z home, spindle off
M30	Program end and rewind

G93 – Inverse Time Feed: Inverse time feed is most commonly used for machine movement containing at least one rotary axis, however that is not the only application. Instead of specifying a velocity a time to complete the movement is specified. The following formula is used to determine F:

When inverse time feed is active an F word is required in every block of code containing a feed move.

G96 – Constant Surface Speed: Spindle speed can be specified two ways. One is constant surface speed. In this mode the surface speed will be kept constant by varying the spindle RPM based on the current cutting diameter. Maintaining a constant surface speed results in superior surface finishes and more consistent parts. Surface speed is specified in surface units per minute. In the inch mode this is surface feet per minute, in millimeter mode it is surface meters per minute.

G97 – Constant RPM: In this mode the spindle speed is specified in revolutions per minute. Spindle RPM is maintained regardless of current cutting diameter.

G98 – Feed per Minute: Sets the machine to feed per minute mode. Specify the desired feedrate in units/minute, in the imperial system this is inches per minute, in the metric system this is millimeters per minute. In this mode the feedrate is modal and not required in all feed move blocks.

G98.1 – Initial Point Return: Specifies that an advanced drilling canned cycle end at the initial Z level. The machine will also return to the initial point before a rapid move to the next position. Initial point return is useful for avoiding steps in parts or fixture clamps without adding a significant amount of cycle time. See figure 981-1.

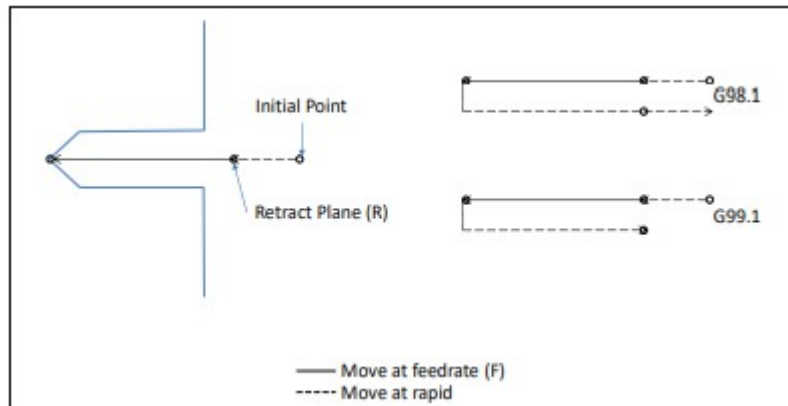


Figure 981-1: Initial and Retract point return for advanced drilling cycles

G99 – Feed per Revolution: In this mode the feed rate is specified in units/revolution; for every revolution of the spindle the machine will move the specified units. By specifying the feedrate in this way if the spindle speed varies the chip load of the tool is kept constant. This can provide and increase in surface finishes and tool life. Feed per rev mode requires RPM feedback from the spindle.

G99.1 – R Point Return: Specifies that an advanced drilling canned cycle end at the programmed R level, see figure 981-1. When drilling holes in a flat plane, G99.1 can be used to reduce excessive machine movement decreasing cycle time.

Chapter 4: Hole Machining Canned Cycles

Standard Hole Machining Cycles	
G80	Canned Cycle Cancel
G83	Face Drilling
G84	Face Tapping
G85	Face Boring
G87	Side Drilling
G88	Side Tapping
G89	Side Boring

Canned cycles are used to reduce program complexity. For example peck drilling a 1 inch hole with .1 inch peck depth would use 30 lines of regular code, but with a canned cycle this same hole can be completed in just 2 lines of code. More importantly if multiple holes are required only 1 extra line of code per hole is needed. There are a variety of canned cycles for different hole types including drilling, boring, and tapping.

There are two sets of hole machining canned cycles available for use, the standard lathe set and an advanced set. This section focuses on the standard lathe cycles.

Axis positions can be programmed in the absolute or incremental modes; specify X, Y, Z, C for absolute and U, V, W, H for incremental. All active machine axes are allowed for positioning, some axes have been left out of examples for clarity.

The basic format of a canned cycle is as follows:

Absolute:

Gcc **X**xx **Z**zz **Q**qq **R**rr **P**pp **L**ll **F**ff

Xxn

G80

Incremental:

Gcc **U**uu **W**ww **Q**qq **R**rr **P**pp **L**ll **F**ff

Uun

G80

cc	Number of the desired canned cycle (i.e. 83, 84, 85, etc).
xx	X position of the center point of the first hole with respect to the current work zero point
uu	Distance and direction along X axis to center point of first hole from the current position
zz	Z position of bottom of hole with respect to the current work zero point
ww	Distance and direction along Z axis, from point R, to bottom of hole
qq	Peck increment if deep hole drilling, always positive
rr	Retract plane, always programmed in incremental and always in radius values
pp	Dwell, in milliseconds, at bottom of hole
ll	Number of repetitions

ff	Feedrate
xn	Position of nth hole X axis, same rules as applied to xx
yn	Position of nth hole Y axis, same rules as applied to yy

Please note that not all arguments will appear in all cycles, and there are a couple special cases that will be discussed.

G80 – Canned Cycle Cancel: To end a canned cycle a G80 must be called. G80 should be specified on its own line to avoid any unintended movements. For example:

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial point
X0.0	Move to X position
G81 G99.1 X0.0 Z-1.0 R.25 F0.01	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Drilling

G83 – Face Drilling: The face drilling cycle allows drilling holes in the face of a part; the Z axis is always the drilling axis. Shallow holes can be straight drilled, or with the addition of address Q, deeper holes can be peck drilled. For straight drilling cycle use format 1, for deep hole peck drilling cycle use format 2. See figure 83-1 for a graphic of the tool motion.

Format 1: G83 X__ Z__ R__ P__ L__ F__

Format 2: G83 X__ Z__ R__ Q__ P__ L__ F__

X – Position of hole

Z – End point of hole

R – Retract plane

Q – Peck depth

P – Dwell time

L – Number of repetitions

F – Feedrate

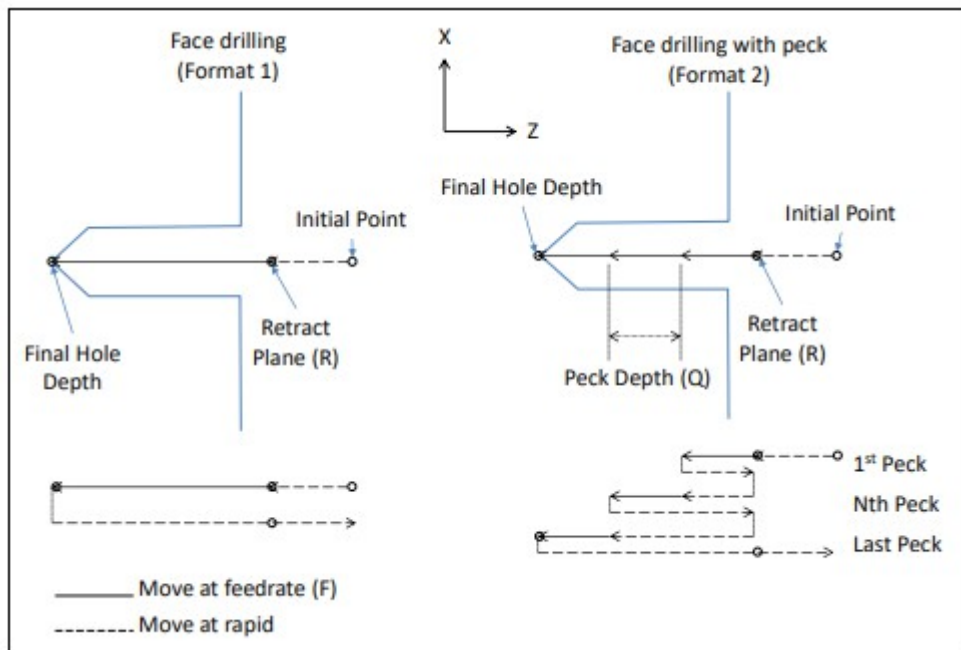


Figure 83-1: Motion of tool point in G83 drilling cycle

Example: Drill a 0.201 inch diameter hole, located at X0.0, Z0.0 to a depth of 1.0 inch with 0.25 inch pecks. Spot to a depth of 0.1. Use initial point 0.5, retract plane 0.1.

G0 G54 G18 G40 G80 G99	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z0.5	Move to Z initial point
X0.0	Move to X position
G83 X0.0 Z-0.1 R-0.4 F0.01	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
G0 G54 G18 G40 G80 G99	Safe start line
T0202	Tool change
G97 S3000 M3	Start spindle
G0 Z0.5	Move to Z initial point
X0.0	Move to X position
G83 X0.0 Z-1.0 R-0.4 Q0.25 F0.008	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G87 – Side Drilling: The side drilling cycle allows drilling holes in the OD of a part; the X axis is always the drilling axis. Shallow holes can be straight drilled, or with the addition of address Q deep holes can be peck drilled. For straight drilling cycle use format 1, for deep hole peck drilling cycle use format 2. See figure 87-1 for a graphic of the tool motion.

Format 1: G87 Z__ X__ R__ P__ L__ F__

Format 2: G87 Z__ X__ R__ Q__ P__ L__ F__

Z – Position of hole

X – End point of hole

R – Retract plane

Q – Peck depth

P – Dwell time

L – Number of repetitions

F – Feedrate

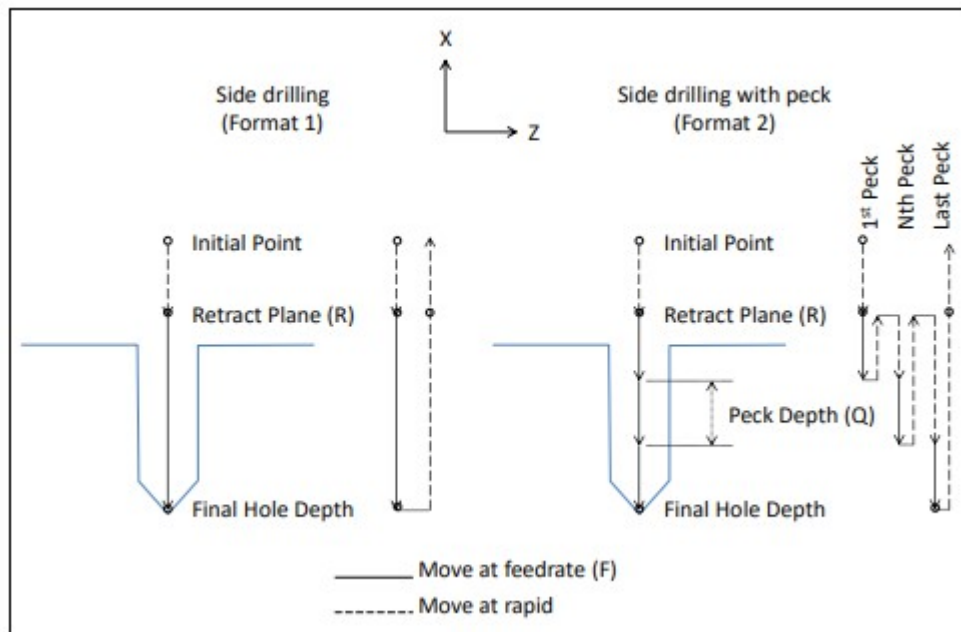


Figure 87-1: Motion of tool point in G87 drilling cycle

Example: Drill a 0.201 inch diameter hole, located at X2.0, Z-1.0, C0.0 to a depth of 1.0 inch with 0.25 inch pecks. Use initial point 3.0, retract plane 2.2.

G0 G54 G18 G40 G80 G99	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z-1.0	Move to Z position
X3.0 C0.0	Move to X and C initial position
G87 Z-1.0 X0.0 R-0.8 Q0.25 F0.008	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Tapping

G84 – Face Tapping: A cycle for creating threaded holes in the face of a part using a tap; the Z axis is always the tapping axis. For the tapping cycle the spindle is started in the clockwise direction. When the bottom of the hole is reached the spindle reverses for retraction. Changes to feedrate or spindle speed mid cycle can be damaging to the tool and work piece, for this reason the feedrate and spindle speed overrides are disabled. The machine will run at 100% override for the duration of the cycle. Feed hold is also disabled during the cycle. If feed hold is pressed motion will stop when the tool reaches the retract point.

Tapping requires synchronization of the spindle speed and feed rate to create the correct thread form. The feedrate can be programmed in either feed per minute mode, G98, or feed per revolution mode, G99. In the feed per rev mode the commanded feedrate will be simply the pitch of the thread. Metric threads are classified with the thread pitch, i.e. M8x1.25mm thread has a 1.25mm pitch. Unified threads are classified by threads per inch which requires a bit of calculation to get the pitch, don't worry it's easy. Simply divide 1 inch by the TPI. For a ¼-20 fastener we would calculate $1/20 = .05$, this is the pitch. The catch is, to use feed per rev requires some form of rpm feedback from the machine, not every machine will have this. For the machines without feedback the tapping cycle can be programmed in feed per min mode (G98). This method requires a little more math to obtain the correct feedrate based on spindle rpm and pitch of the thread. The equation looks like this: $RPM * Pitch = IPM$. To tap that ¼-20 hole at 1500 RPM we first need to calculate the pitch, remember $1/TPI = Pitch$, so $1/20 = .05$. Now we calculate the feed per min as $1500 * .05 = 75 IPM$. It is important to note that if the spindle speed is changed, the feed per minute must also be changed to match. For this reason, if the machine is capable of feed per rev programming use it.

Format: G84 X__ Z__ R__ P__ L__ F__

X – Position of hole
 Z – End point of hole
 R – Retract plane
 P – Dwell time
 L – Number of repetitions
 F – Feedrate

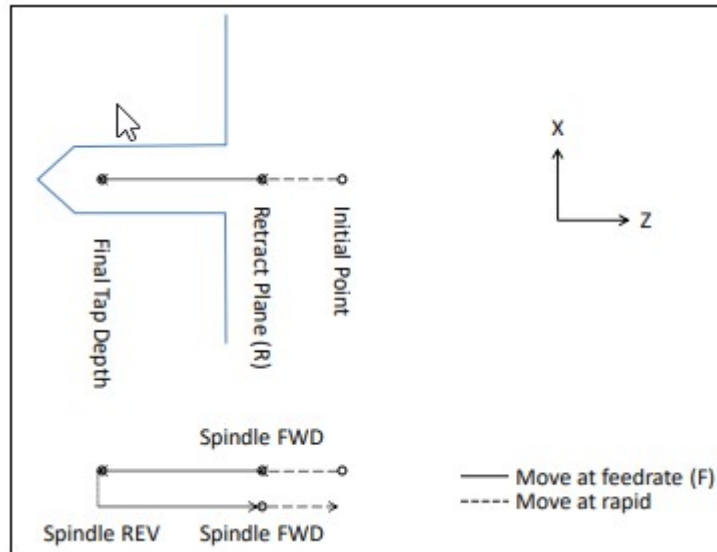


Figure 84-1: Motion of tool point in G84 tapping cycle

Example:

Tap the hole from the G83 example to a depth of .500 with a 1/4-20 tap using feed/min.

$$1/\text{TPI} = 1/20 = .05 \text{ Pitch}$$

$$\text{RPM} * \text{Pitch} = 1000 * .05 = 50.0 \text{ UPM}$$

G0 G54 G18 G40 G80	Safe start line
T0202	Tool change
G97 S1000 M3	Start spindle
G0 Z0.5	Move to Z initial point
X0.0	Move to X position
G98	Feed per minute mode
G84 X0.0 Z-0.5 R-0.4 F50.0	Tap cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G88 – Side Tapping: Same functionality as G84 except the X axis is the tapping axis allowing the tapping of holes on the OD of a part.

Format: G88 Z__ X__ R__ P__ L__ F__

Z – Position of hole

X – End point of hole

R – Retract plane

P – Dwell time

L – Number of repetitions

F – Feedrate

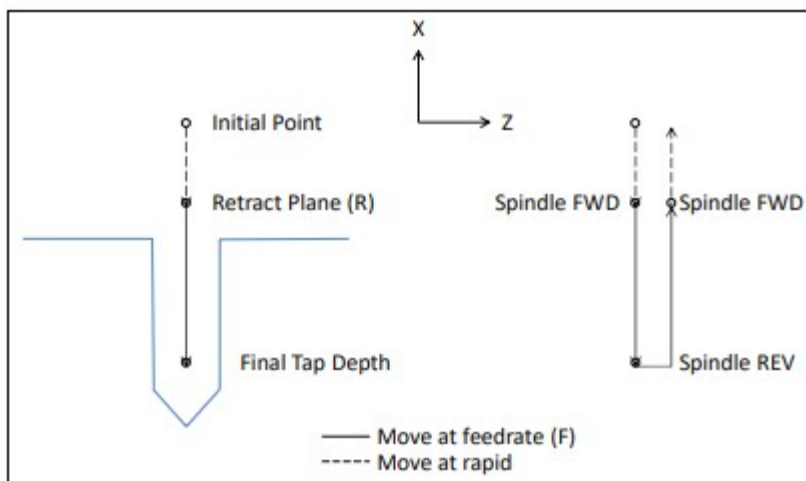


Figure 88-1: Motion of tool point in G88 tapping cycleExample:

Tap the hole from the G87 example to a depth of 0.375 with a ¼-20 tap using feed/rev.

$1/\text{TPI} = 1/20 = .05$ Pitch

G0 G54 G18 G40 G80	Safe start line
T0202	Tool change
G97 S1000 M3	Start spindle
G0 Z-1.0	Move to Z position
X3.0 C0.0	Move to X and C initial position
G99	Feed per minute mode
G84 Z-1.0 X1.25 R-0.8 F0.05	Tap cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Boring

G85 – Face Boring: Boring a hole provides superior size and position accuracy as well as much finer finishes. Generally a boring bar is used of either an adjustable or non-adjustable type. When the boring cycle is activated the machine will drill down to the desired depth at the specified feedrate. The retract move will be at twice the programmed feedrate. The G85 cycle is for boring holes in the face of a part, the Z axis will always be the drilling axis.

Format 1: G85 X__ Z__ R__ P__ L__ F__

X – Position of hole

Z – End point of hole

R – Retract plane

P – Dwell time

L – Number of repetitions

F – Feedrate

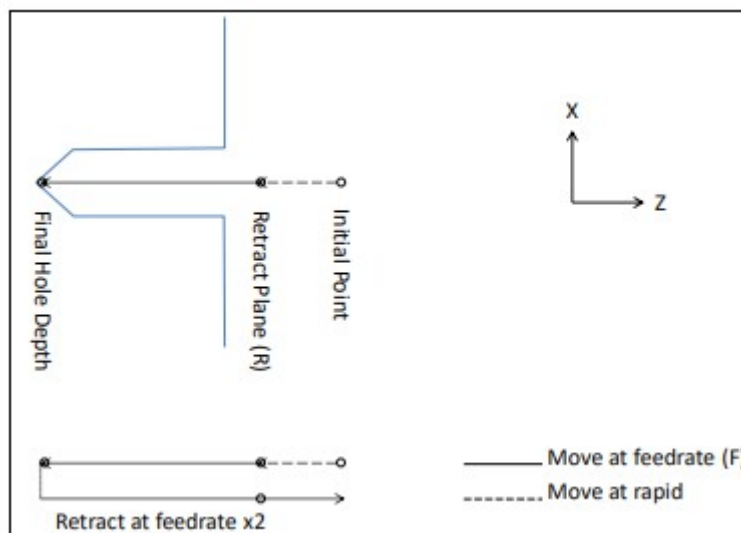


Figure 85-1: Motion of tool point in G85 boring cycle

Example: Bore the hole from the G83 example to a depth of 0.250.

G0 G54 G18 G40 G80	Safe start line
T0303	Tool change
G97 S750 M3	Start spindle
G0 Z0.5	Move to Z initial point
X0.0	Move to X position
G85 X0.0 Z-0.250 R-0.4 F0.003	Bore cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G89 – Side Boring: The side boring cycle is used to bore holes in the OD of a part; the X axis will always be the boring axis. All other parameters and functions are the same as the G85 face boring cycle.

Format 1: G89 Z__ X__ R__ P__ L__ F__

Z – Position of hole

X – End point of hole

R – Retract plane

P – Dwell time

L – Number of repetitions

F – Feedrate

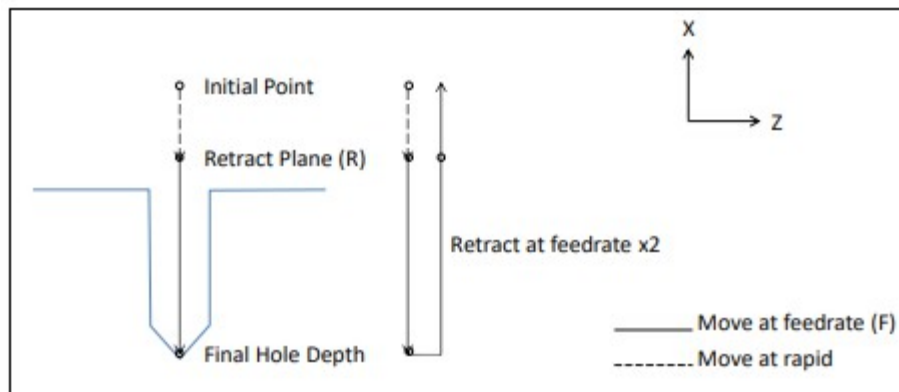


Figure 89-1: Motion of tool point in G89 boring cycle

Example: Bore the hole from the G87 example to a depth of 0.250.

G0 G54 G18 G40 G80	Safe start line
T0303	Tool change
G97 S750 M3	Start spindle
G0 Z-1.0	Move to Z position
X3.0 C0.0	Move to X and C initial position
G89 Z-1.0 X1.5 R-0.8 F0.003	Bore cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Chapter 5: Advanced Hole Machining Canned Cycles

Advanced Hole Machining Cycles	
G73.1	High Speed Peck
G74.1	Reverse Tapping
G76.1	Fine Boring
G81.1	Drilling
G82.1	Spot Face
G83.1	Deep Hole Peck Drilling
G84.1	Tapping
G84.2	Right Hand Rigid Tapping
G84.3	Left Hand Rigid Tapping
G85.1	Boring, Retract at Feed, Spindle On
G86.1	Boring, Retract at Rapid, Spindle Off
G87.1	Back Boring
G88.1	Boring, Manual Retract
G89.1	Boring, Dwell, Retract at Feed, Spindle On

The advanced hole machining canned cycles are provided for increased functionality over the standard cycles. These advanced canned cycles are functionally the same as the milling canned cycles, which provide more options such as high speed peck, absolute R point return, more boring cycles rigid tapping and more. When compared to the standard lathe cycles there are some variances that will be covered in this section.

Positioning will take place in the active plane, and the drilling operation will be on the normal axis. For example in G17 (XY Plane), hole position will be on the XY plane and the drilling axis will be Z. In G18 (ZX Plane) positioning will be on the ZX plane and the drilling axis will be Y. Specifying G19 (YZ Plane) selects the X axis for the drilling axis and positioning will be on the YZ plane. For the purposes of this manual all advanced cycle examples and definitions will be in the XY plane (G17).

Just as with the standard cycles, positions can be specified using absolute (X, Y, Z...) or incremental (U, V, W...) commands. The R point is always absolute position unlike in the standard canned cycles.

When the canned cycle is complete the machine will return to a final position in the drilling axis, either the initial point or the R point as determined by the G98.1/G99.1 setting.

Absolute:

Gcc G98.1/99.1 Xxx Zzz Qqq Rrr Ppp Lll Fff

Xxn

G80

Incremental:

Gcc G98.1/99.1 Uuu Www Qqq Rrr Ppp Lll Fff

Uun

G80

cc	Number of the desired canned cycle (i.e. 81.1, 84.1, 85.1, etc).
xx	X position of the center point of the first hole with respect to the current work zero point
uu	Distance and direction along X axis to center point of first hole from the current position
zz	Z position of bottom of hole with respect to the current work zero point
ww	Distance and direction along Z axis, from point R, to bottom of hole
qq	Peck increment if deep hole drilling, always positive
rr	Retract plane,
pp	Dwell, in milliseconds, at bottom of hole
ll	Number of repetitions
ff	Feedrate
xn	Position of nth hole X axis, same rules as applied to xx
yn	Position of nth hole Y axis, same rules as applied to yy

Please note that not all arguments will appear in all cycles, and there are a couple special cases that will be discussed.

Advanced Drilling

G81.1 – Drilling: This is a straight drilling cycle. The tool moves to position, feeds to the bottom of the hole then rapid retracts to either the R point or the initial point. This is an advanced canned cycle, this means that the drilling axis is determined by the current plane selection. See figure 811-1 for a graphic of the tool motion.

Format: G81.1 X__ Y__ Z__ R__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

L – Number of repetitions

F – Feedrate

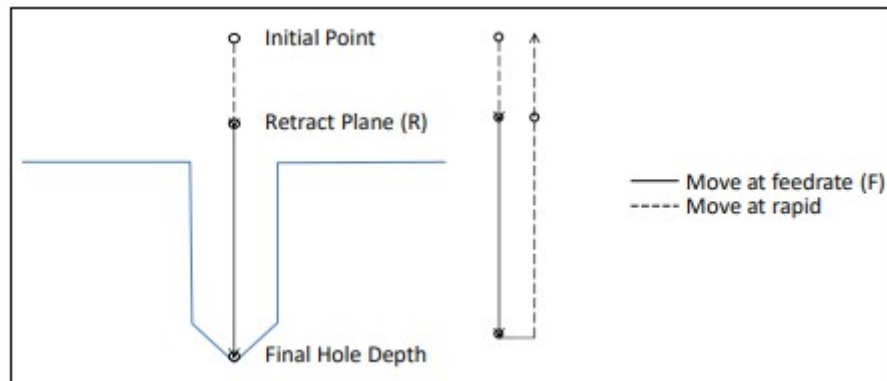


Figure 811-1: Motion of tool point for G81.1 drilling cycle

Example: Create a program to drill a hole 0.5 inch deep in the face of a part at X0.0, Z0.0.

G0 G54 G17 G40 G80	Safe start line, select XY plane
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z0.5	Move to Z initial point
X0.0	Move to X position of hole
G81.1 G99.1 X0.0 Z-0.5 R.1 F0.01	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G82.1 – Spot Face: Spot face adds the ability to dwell at the bottom of the hole for a specified amount of time. The actual tool motion is the same as a G81 cycle, however with the dwell it is possible to attain better accuracy and finish at the bottom of the hole. This is useful for chamfering, counter boring, and spot facing.

Format: G82.1 X__ Y__ Z__ R__ P__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

P – Dwell

L – Number of repetitions

F – Feedrate

Example:

Create a chamfering program for the hole from the G81.1 example, dwell for .2 seconds.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z1.0	Move to Z initial point
X0.0	Move to X position of hole
G82.1 G99.1 X0.0 Z-0.125 R.1 P0.2 F0.01	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G83.1 – Peck Drilling: Peck drilling is a cycle used for drilling deep holes. The cycle allows for breaking and clearing of chips and better application of coolant by fully retracting the tool from the hole between pecks. This retract move and plunge to previous depth are rapid moves, each peck is a feed move at the specified feed rate. See figure 831-1 for a graphic of the tool motion.

Format: G83.1 X__ Y__ Z__ Q__ R__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

Q – Peck amount

R – Retract plane

L – Number of repetitions

F – Feedrate

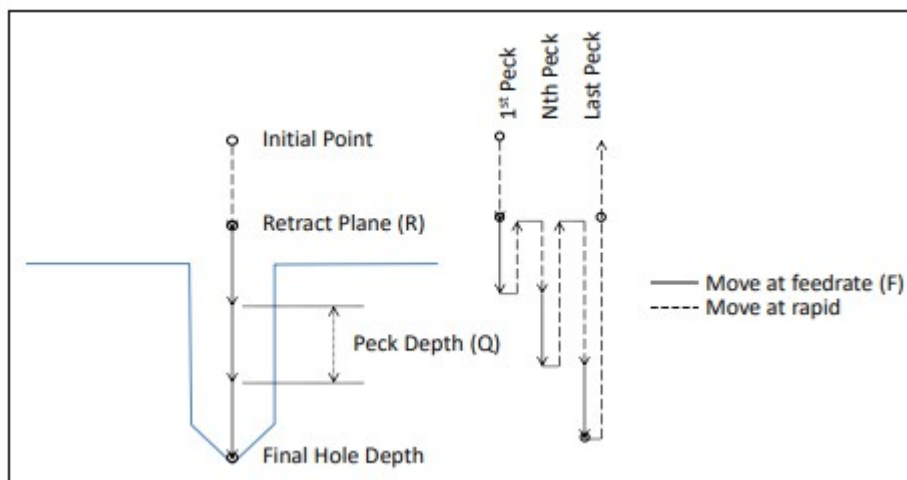


Figure 831-1: Motion of tool point in G83.1 drilling cycle

Example: Peck drill a hole in the OD of a part at Z-1.5, starting on a 2.0 inch diameter and drilling a depth of 0.75 inches with a 0.125 peck depth.

Note the plane selection to use the X axis as the drilling axis.

G0 G54 G19 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z-1.5	Move to Z position of hole
X3.0	Move to X initial point
G83.1 G99.1 X0.5 Z-1.5 R2.2 Q0.125 F0.01	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G73.1 – High Speed Peck: In materials that produce long stringy chips a high speed peck cycle can be used to break them up. Unlike the G83.1 cycle that retracts completely out of the hole after each peck, the G73.1 cycle only retracts 0.100 inch. See figure 731-1. This short retract helps to reduce cycle times when a complete retract is unnecessary.

Format: G73.1 X__ Y__ Z__ Q__ R__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

Q – Peck amount

R – Retract plane

L – Number of repetitions

F – Feedrate

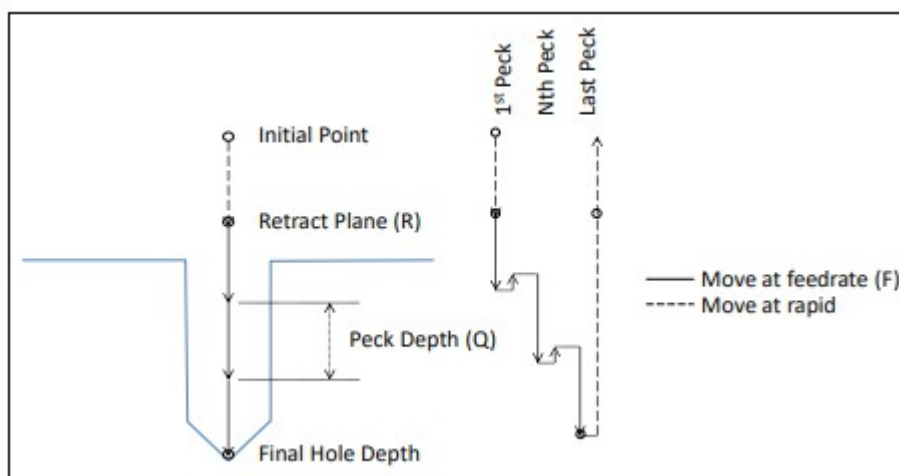


Figure 731-1: Motion of tool point in G73.1 drilling cycle

Example: Peck drill a hole in the OD of a part at Z-1.5, starting on a 2.0 inch diameter and drilling a depth of 0.75 inches with a 0.125 peck depth.

Note the plane selection to use the X axis as the drilling axis.

G0 G54 G19 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z-1.5	Move to Z position of hole
X3.0	Move to X initial point
G73.1 G99.1 X0.5 Z-1.5 R2.2 Q0.125 F0.01	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Advanced Tapping

G84.1 – Right Hand Tapping: The tapping cycle is used to create threaded holes using a tap. Set up of the tapping cycle is the same as with G84, and the same restrictions and warnings apply. This cycle is meant to be used with a tapping head. Because it this is an advanced cycle the tapping axis is determined by the current plane selection.

Format: G84.1 X__ Y__ Z__ R__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

L – Number of repetitions

F – Feedrate

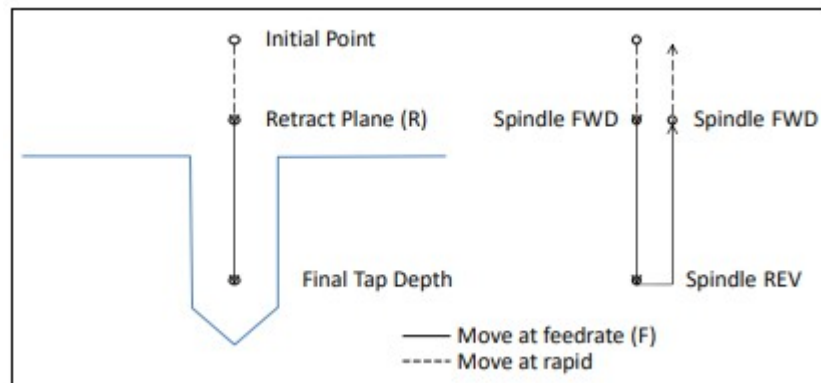


Figure 841-1: Motion of tool point in G84.1 tapping cycle

Example: Create a program to tap 3 holes in the face of a part to a depth of 0.500 with a ¼-20 tap, at a 1.75 inch bolt circle diameter.

$1/\text{TPI} = 1/20 = .05$ Pitch

G0 G54 G17 G40 G80	Safe start line
T0202	Tool change
G97 S1000 M3	Start spindle
G0 Z0.5	Move to Z initial point
X1.75 C0.0	Move to X and C position of first hole
G99	Feed per rev mode
G84.1 G99.1 X1.75 C0.0 Z-0.5 R.15 F0.05	Tap cycle start
C120.0	Tap second hole
C240.0	Tap third hole
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G74.1 – Left Hand Tapping: Left hand tapping is the same as right hand tapping (G84.1) except that it will cut left hand threads. The spindle must be started in the reverse direction before calling the G74.1 cycle.

G84.2/G84.3 – Right and Left Hand Rigid Tapping: Rigid tapping can be performed on capable machines. As the name implies the tap is held rigidly in the spindle, no tension/compression style tapping holder is required. Holding the tap in this manner requires the machine to have precise control of spindle speed, axis feed, and precise feedback of spindle RPM. The tapping axis will be electronically geared to the spindle RPM. Use G84.2 for right hand tapping and G84.3 for left hand tapping. See figure 841-1 for a graphic of the motion.

Format: G84.2/84.3 X__ Y__ Z__ R__ P__ L__ F__ J__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

P – Dwell in milliseconds

L – Number of repetitions

F – Feedrate

J – Spindle speed for retract

As with other tapping cycles the feedrate and spindle speed overrides are disabled and set to 100% for the duration of the cycle. Feed hold is also disabled for the duration of the cycle. If the feed hold button is pushed during a tapping cycle motion will stop when the machine reaches the retract position.

Example: Create the program to tap three holes in the OD of a 2.0 inch bar, evenly spaced, to a depth of .500 with a 1/4-20 tap using feed/rev. Locate the holes at Z-2.0 Tap at 1000 RPM, retract at 2000 RPM.

Note the plane selection to tap with the X axis.

G0 G54 G19 G40 G80	Safe start line
T0202	Tool change
G97 S1000 M3	Start spindle
G0 Z-2.0	Move to Z position
X3.0 C0.0	Move to X and C initial position of first hole
G99	Feed per revolution mode
G84.1 G99.1 X1.0 C0.0 Z-2.0 R2.2 F0.05 J2000	Tap cycle start
C120.0	Tap second hole
C240.0	Tap third hole
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return Z to home
M30	Program end and rewind

Advanced Boring

G76.1 – Fine Boring: The fine boring cycle allows the user to stop the spindle and move the tool away from the wall before retracting. This allows for a rapid retract without leaving a scratch on the wall.

Format: G76.1 X__ Y__ Z__ R__ I__ J__ P__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

I – X shift distance and direction (Radius value)

J – Y shift distance and direction

P – Dwell in milliseconds

L – Number of repetitions

F – Feedrate

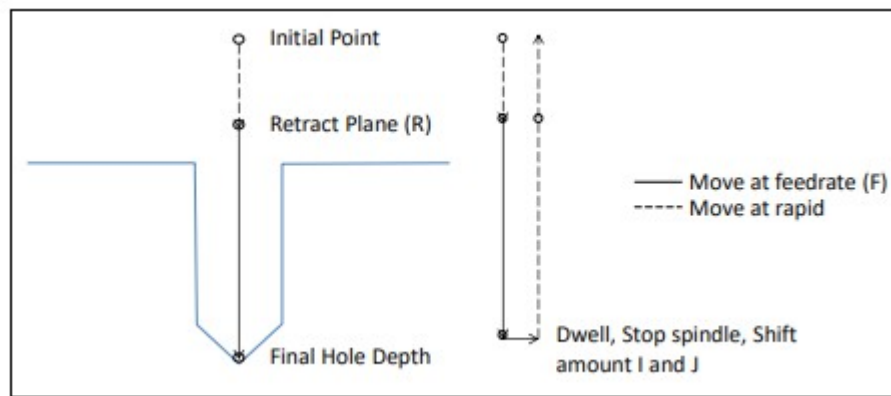


Figure 761-1: Motion of tool point in G76.1 boring cycle

After feeding to the bottom of the hole, the machine will pause for the specified dwell time, then the spindle will stop in the orient position before making the shift move defined by I and J. In machines with a spindle orient function called by M19 this will all be automatic. However, many machines are not capable of orienting the spindle so the orientation must be done manually. Make an M code macro program M19.mcs to command a spindle stop and an M0 mandatory program stop. This will allow the operator to manually orient the spindle before the shift move is made. Here is an example of the M19 macro for the manual orient:

Example: Create a program to bore a 1.0 inch deep hole in the center of the face of the workpiece.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z0.5	Move to Z initial point
X0.0	Move to X position of hole
G98	Feed per minute
G76.1 G99.1 X0.0 Z-1.0 R0.1 I-.025 F10.0	Fine bore cycle, shift X-.025 at bottom
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G85.1 – Boring, Feedrate Retract: G85.1 is a straight boring cycle, most commonly used for boring or reaming. The retract move is at the programmed feedrate with the spindle on.

Format: G85.1 X__ Y__ Z__ R__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

L – Number of repetitions

F – Feedrate

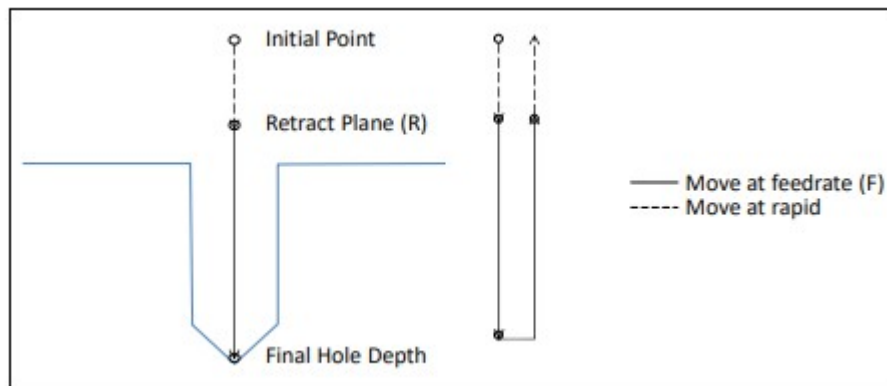


Figure 851-1: Motion of tool point in G85.1 boring cycle

Example: Create a program to ream a 0.25 inch deep hole in the OD of a 1 inch diameter workpiece, located at Z-0.5.

G0 G54 G19 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z-0.5	Move to Z position
X2.0	Move to X initial point
G99	Feed per rev
G85.1 G99.1 X0.5 Z-0.5 R1.2 F10.0	Start boring cycle
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G86.1 – Boring, Rapid Retract: G86 is a straight boring cycle. Before retracting from the hole the spindle is stopped. The retract is then performed at the rapid rate. This will leave a scratch or multiple scratches where the cutting edges are in contact with the wall.

Format: G86 X__ Y__ Z__ R__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

L – Number of repetitions

F – Feedrate

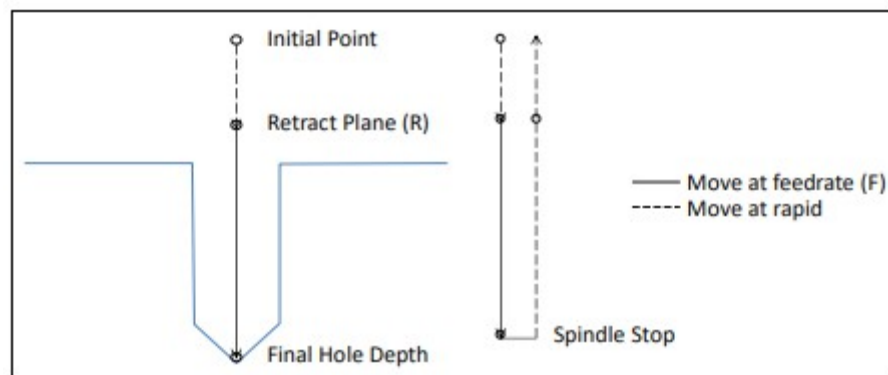


Figure 861-1: Motion of tool point in G86.1 boring cycle

Example: Create a program to bore 3 holes in the face of a part, 1.0 inch deep evenly spaced on a 2.0 inch bolt circle diameter.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z0.5	Move to Z initial position
X2.0 C0.0	Move to X and C positions of first hole
G86.1 G99.1 X2.0 C0.0 Z-1.0 R0.1 F0.004	Boring cycle start
C120.0	Bore second hole
C240.0	Bore third hole
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return Z to home
M30	Program end and rewind

G87.1 – Back Boring: Back boring is a useful cycle for spot facing, counter boring or chamfering the back side of a part. At the start of the cycle the spindle will be stopped at the orient position and the tool offset from the hole center by the distance and direction defined by I and J. For machines that do not have the ability to orient the spindle, see the M19 macro example in the G76.1 cycle description. The tool can then be positioned to the R point below the work piece. Once at the R point the tool will be positioned at the hole center and the spindle started to perform the back boring operation. When the specified Z point is reached the machine will orient the spindle, offset by amount I, J and retract back to the initial point. In this cycle the R point will always be below the work piece, not a good point to end the cycle. For that reason this canned cycle will always return to the initial point, it is not possible to specify a G99.1 R point return.

Format: G87.1 X__ Y__ Z__ R__ I__ J__ P__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

I – X shift distance and direction (Radius value)

J – Y shift distance and direction

P – Dwell in milliseconds

L – Number of repetitions

F – Feedrate

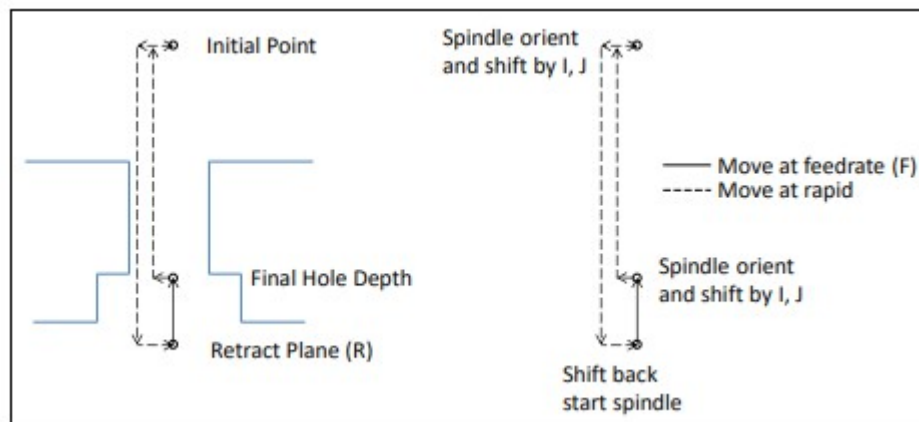


Figure 871-1: Motion of tool point in G87.1 boring cycle

Example: Create a program to back bore a thru hole in the center of the face of a workpiece. Thru depth 1.0 inch, back bore depth 0.150 inch.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z0.5	Move to Z initial point
X0.0	Move to X position of hole
G87.1 G98.1 X0.0 Z-0.85 R-1.05 I-.10 F0.008	Back boring cycle start (Shift -.1 in X axis)
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G88.1 – Boring, Manual Retract: This boring cycle features a manual retract. At the bottom of the hole the specified dwell is performed, then the spindle is stopped and the program paused. The operator can then manually retract the tool from the hole. After retracting the tool cycle start is pressed to continue program operation.

Format: G88.1 X__ Y__ Z__ R__ P__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

P – Dwell in milliseconds

L – Number of repetitions

F – Feedrate

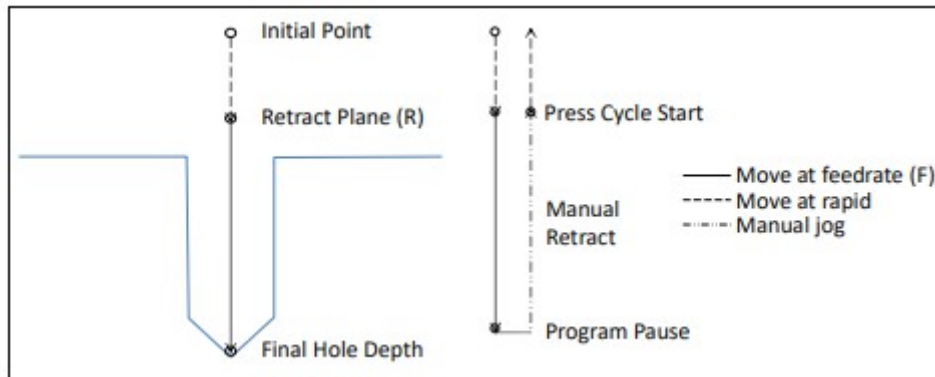


Figure 881-1: Motion of tool point in G88.1 boring cycle

Example: Create a program to bore a 0.4 inch deep hole located at Z-1.25 in the OD of a 4.0 inch diameter part.

G0 G54 G19 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z-1.25	Move to Z position
X5.0	Move to X initial position
G88.1 G99.1 X3.2 Z-1.25 R4.2 F0.004	Boring cycle start, pause for manual retract
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return Z to home
M30	Program end and rewind

G89.1 – Boring, Dwell & Feedrate Retract: Same function as G85 with the addition of a dwell at the bottom of the hole.

Format: G89.1 X__ Y__ Z__ R__ P__ L__ F__

X, Y – Position of hole in XY plane

Z – End point of hole

R – Retract plane

P – Dwell in milliseconds

L – Number of repetitions

F – Feedrate

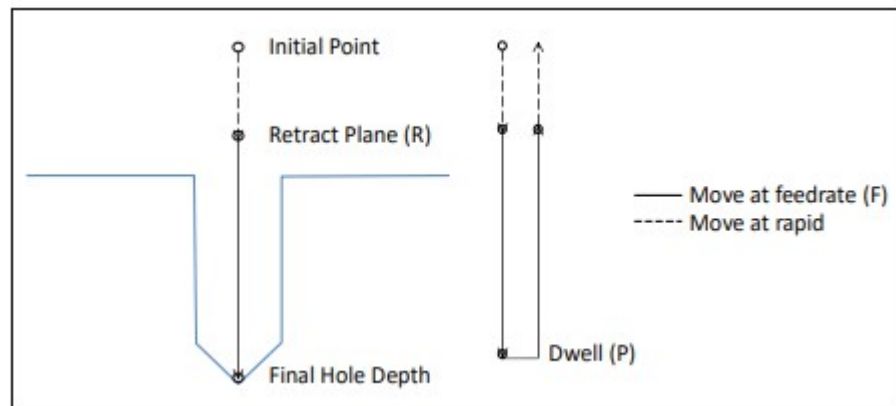


Figure 891-1: Motion of tool point in G89.1 boring cycle.

Example: Create a program to bore a 1.0 inch deep hole in the face of a workpiece. Dwell for 1.0 second at the bottom.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
G97 S2500 M3	Start spindle
G0 Z0.5	Move to Z initial position
X0.0	Move to X position of hole
G89.1 G99.1 X0.0 Z-1.0 R0.1 P1.0 F0.004	Boring cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return Z to home
M30	Program end and rewind

Chapter 6: Tool Nose Radius Compensation

Lathe tools generally have a slightly rounded nose instead of a sharp point, see figure 41-1. This round nose creates an offset between the programmed point and the actual cutting point when cutting tapers and arcs. Tool nose radius compensation allows the programmer to program the actual shape of the part, leaving the calculation of this offset amount to the machine control. Many modern Computer Aided Manufacturing (CAM) systems will automatically compensate for this offset in the G code file, but when manually programming a simple profile, tool nose radius compensation is a must have.

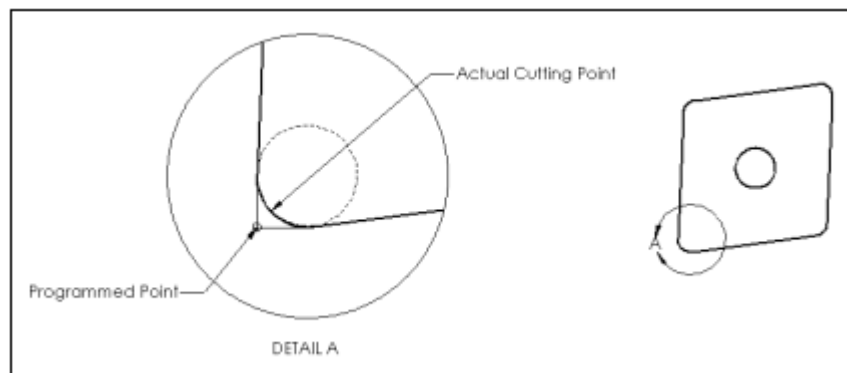


Figure 41-1: Difference between programmed and actual cut points

Profiles parallel to the X or Z axis, a simple cylinder, are not subject to this offset. When turning a straight surface, the offset between the programmed point and the actual cutting point is zero because they lie on the same line. See figure 41-2. However, when angles or arcs are introduced, the offset starts to change and over or under cutting will occur, see figure 41-3.

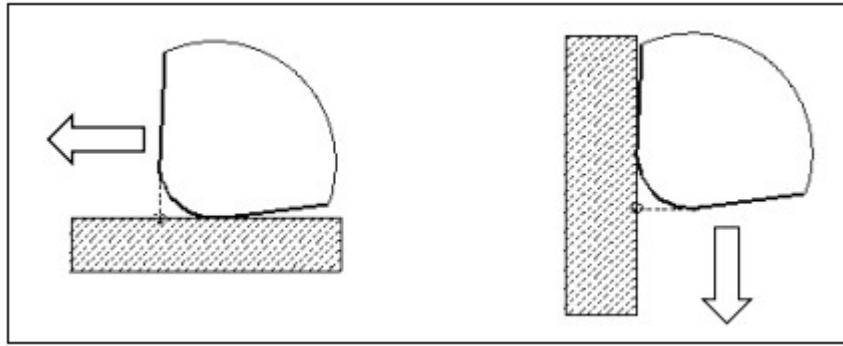


Figure 41-2: Straight turning and facing

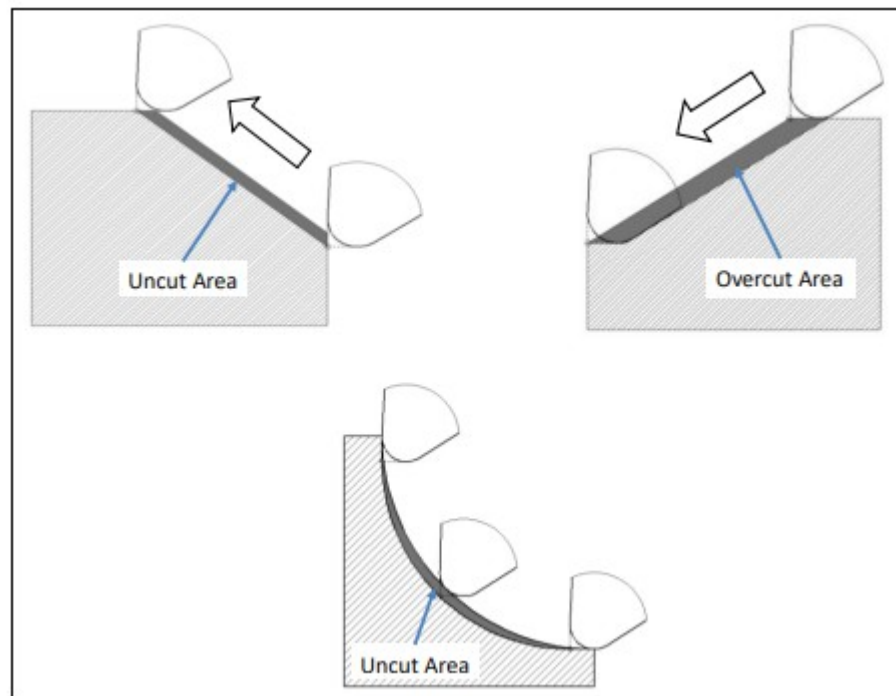


Figure 41-3: Tool nose radius effect of tapers and arcs

For the tool nose radius compensation to work correctly, it does require good input from the operator. An accurate value for the nose radius needs to be input and also the tip direction must be specified. If these values are not correct, the resulting tool path could amplify the negative effects of no radius compensation. The nose radius and tip direction are input in the desired tool offset under “Tip rad” and “Tip” respectively. The nose radius of the cutter, if purchased, can usually be found on the box or in the manufacturer's specs. Many inserts are classified with a standard ISO numbering system that contains the nose radius size. For example a CNMG 332 insert has a $2/64$ radius, or 0.0313 inch, a DCGT 331 has a $1/64$ radius, or 0.0156 inch. Check the nose radius information and input it into the “Tip rad” for the desired tool.

The tool tip direction is determined by how the tool is positioned in the machine, see figures 41-4 and 41-5.

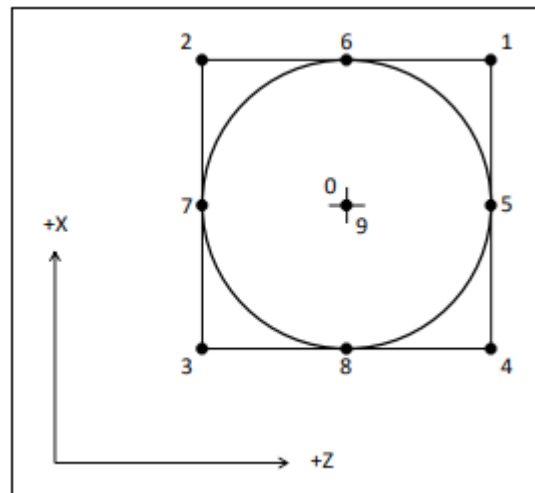


Figure 41-4: Tool tip number orientation

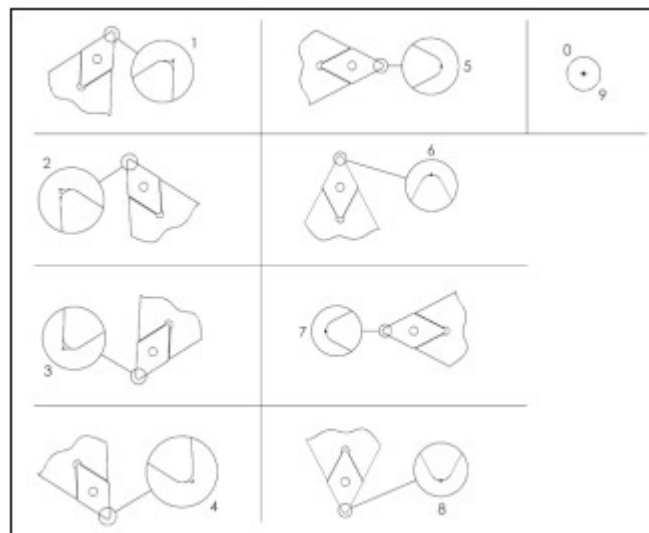


Figure 41-5: Tool tip orientation

For most OD turning tools the tool tip will be 2 or 3 depending on if it is a front or rear turret tool.

G41 and G42 are used to activate tool nose radius compensation. The offset used will be the offset number specified in the last T code called. G41 will offset to the left of the path and G42 will offset to the right of the path, see figure 41-6. When activating G41 or G42, it is always recommended to activate on a lead-in perpendicular to the path and a distance longer than the radius. This will greatly reduce the chances of a gouge or unexpected result. Also, nose radius compensation may only be activated on a linear move, no arcs.

Format: G00/01 G41/42 X__ Z__ F__

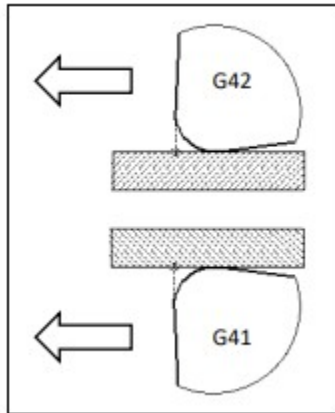


Figure 41-6: Nose radius compensation direction

Example program: To show how to use tool nose radius comp, let's make a quick program. This will be a simple OD profile with a couple tapers and an arc. Because we are using tip compensation we'll be able to program the part dimensions as shown, no extra calculations necessary. See figure 41-7 for part dimensions.

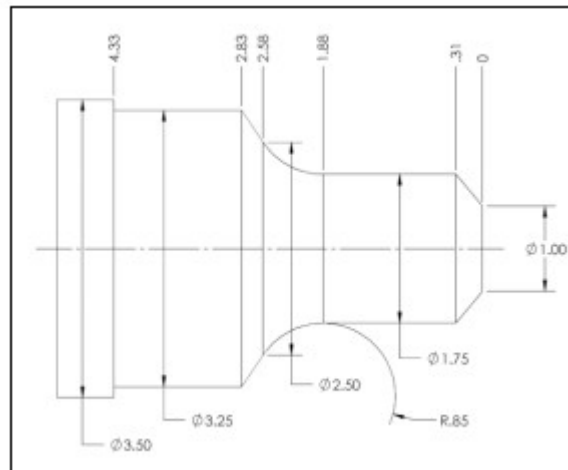


Figure 41-7: Example program part dimensions

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change, select tool 01, offset 01.
G96 S1000 M3	Start spindle
G0 Z0.0	Move to Z to face part
X3.7	Move to X start position
G1 X-0.035 F0.010	Face part (move past center line a bit more than the nose radius)
G0 U0.1 W0.05	Retract off the face
G0 X1.5 Z0.1	Position to start lead-in
G01 G42 X1.0 F0.025	Activate tool nose radius compensation
X1.0 Z0.0 F0.015	Move to start of cut
X1.75 Z-0.31 F0.008	Start cut
Z-1.88	Cut
G02 X2.5 Z-2.58 I0.85	Cut radius
G01 X3.25 Z-2.83	Cut 2 nd taper
G01 Z-4.33	Finish cut
G01 G40 X3.75	Lead-out and cancel tool nose radius compensation
G0 G53 X0.0 Z0.0 M5	Return home and stop spindle
M30	End program and rewind

Chapter 7: M Codes

Code	Description	Page
M00	Mandatory Program Stop	81
M01	Optional Program Stop	81
M02	Program End	81
M03	Spindle Forward/Clockwise	81
M04	Spindle Reverse/Counterclockwise	81
M05	Spindle Stop	81
M07	Mist Coolant On	82
M08	Flood Coolant On	82
M09	All Coolant Off	82
M19	Spindle Orient	82
M23	Single Line G76 Leadout On	82
M24	Single Line G76 Leadout Off	82
M30	Program End and Rewind	82
M98	Subprogram Call	82
M99	Return From Subprogram / Rewind	83
M???	Custom Macro M Codes	83

M Code Descriptions

M00 – Mandatory Program Stop: To pause a program at any point specify an M00. All motion will be halted and the spindle stopped. To resume operation press the cycle start button.

M01 – Optional Program Stop: Program pauses same as M00, but only when the optional stop switch is turned on. This allows the operator to stop and check program points when needed, but also to run without stops.

M02 – Program End: Ends the program at the M02 block. All program operation will be ended and defaults (G54, G17, G90, etc.) reset. The program will not rewind and if the cycle start button is pressed program operation will begin with the next block following the M02.

M03 – Spindle Forward/Clockwise: Turns the spindle on in the forward direction at the designated speed. M03 is usually combined with an S word to define the speed. If an S word is not specified the last spindle speed is used.

M04 – Spindle Reverse/Counterclockwise: Turns the spindle on in the reverse direction at the designated speed. M04 is usually combined with an S word to define the speed. If an S word is not specified the last spindle speed is used.

M05 – Spindle Stop: Stops spindle rotation.

M07 – Mist Coolant On: Turns on the mist coolant output.

M08 – Flood Coolant On: Turns on the flood coolant output.

M09 – All Coolant Off: Turns off all coolant outputs.

M19 – Spindle Orient: Although this code is not built into Mach and is completely user defined, it is required to orient the spindle for some machining cycles, G76 and G87 for example. Tool changers usually will require a spindle orientation, and although this orientation could be any M code defined by the user or even built into the M6, M19 is recommended as it is an industry standard M code.

M23 – Thread Leadout On: Turns on thread lead out for the single line G76 threading cycle.

M24 – Thread Leadout Off: Turns off thread lead out for the single line G76 threading cycle.

M30 – Program End and Rewind: Ends the current program and rewinds back to the beginning. Pressing the cycle start button will start program execution from the first block. All defaults (G54, G17, G90, etc.) will be reset.

M98 – Subprogram Call: Subprograms are external programs referenced by the current running program. When called program execution will continue in the subprogram. This allows the user to reduce program length and complexity by providing the ability to easily repeat sections of code, run the same section of code in multiple locations or in multiple fixture offsets, the possibilities are limited only by the programmer. To call a subprogram command M98 with the program number as shown.

Format: M98 P____ Q__ L__

P specifies the number of the program to be called. This is a four digit integer number. When the M98 is read Mach scans the current file for a block containing the program number in the following form:

Format: O1234

Note that the letter “O” is used to specify the program number 1234, NOT the number “0”. Program execution will continue with the block following the O number. For this method the subprogram should be below the end of the current program:

(MAIN PROGRAM)	Main program header
G0 G90 G54 G18 G40 G80	Safe start line
G0 X4.0 Y0.0 Z1.0	Rapid move
T1 M6	Tool change to tool 1
G0 X10.0 Y-5.0 Z1.0	Rapid move
M98 P1000	Call subprogram number 1000
M30	Program end and rewind
O1000 (SUB PROGRAM BEGIN)	Subprogram number

There are limitations to this method; mainly the subprogram must be pasted into every program that it is to be called from. Mach also allows a subprogram to be called from an external file. This way, multiple programs can call the same subprogram, without having the program in the main file. If a change needs to be made to the subprogram it only needs to be done in one file, not every file in which the sub is called. If the control does not find the program number in the current file it will then search for it in the Mach4\Subroutines directory. This time it will be searching filenames. The files in this directory should be named with the program number as follows:

Format: O____

Note the letter “O” followed by four numbers, and no file extension; O1234 not O1234.txt. When the program is loaded this file will be found and loaded into the memory, the tool path display will reflect what is in the subprogram. When running the file the M98 block will be read and program execution will continue with the first block of the subprogram file.

The arguments Q and L are optional. Q specifies the sequence number in the subprogram to start at. If Q is omitted then execution will start at the beginning of the sub program; see figure 198-1. L is the number of repetitions. For example, if L=2 the subprogram is run two times before execution of the main program continues. If L is omitted the program will run only once.

M99 – Return From Subprogram / Rewind: To return to the main program from a subprogram an M99 is used.

Format: M99 P__

M99 specified in a subprogram will return to the main program at the block immediately following the M98 subprogram call. The addition of a P in the M99 block will return to a desired sequence number in the main program. See figure 199-1.

If M99 is specified in a main program (not in a subprogram) a program rewind is performed. Execution will continue from the first block of the program.

Custom M Codes: Every machine is a little different and has different functions and options. Custom M codes allow the user to create simple programming calls for complex processes, activating or deactivating outputs, reading inputs, performing math, or whatever else the user and machine could require. These M codes are programmed in Lua and placed in the macros folder under each profile. Every profile can have its own unique set of M codes. The file name must be the macro to be called. For example to make a tool change macro, M6, add a Lua script with the filename M6.mcs to the macros folder in the desired profile. When an M6 is called in a program or in MDI the macro will be executed.