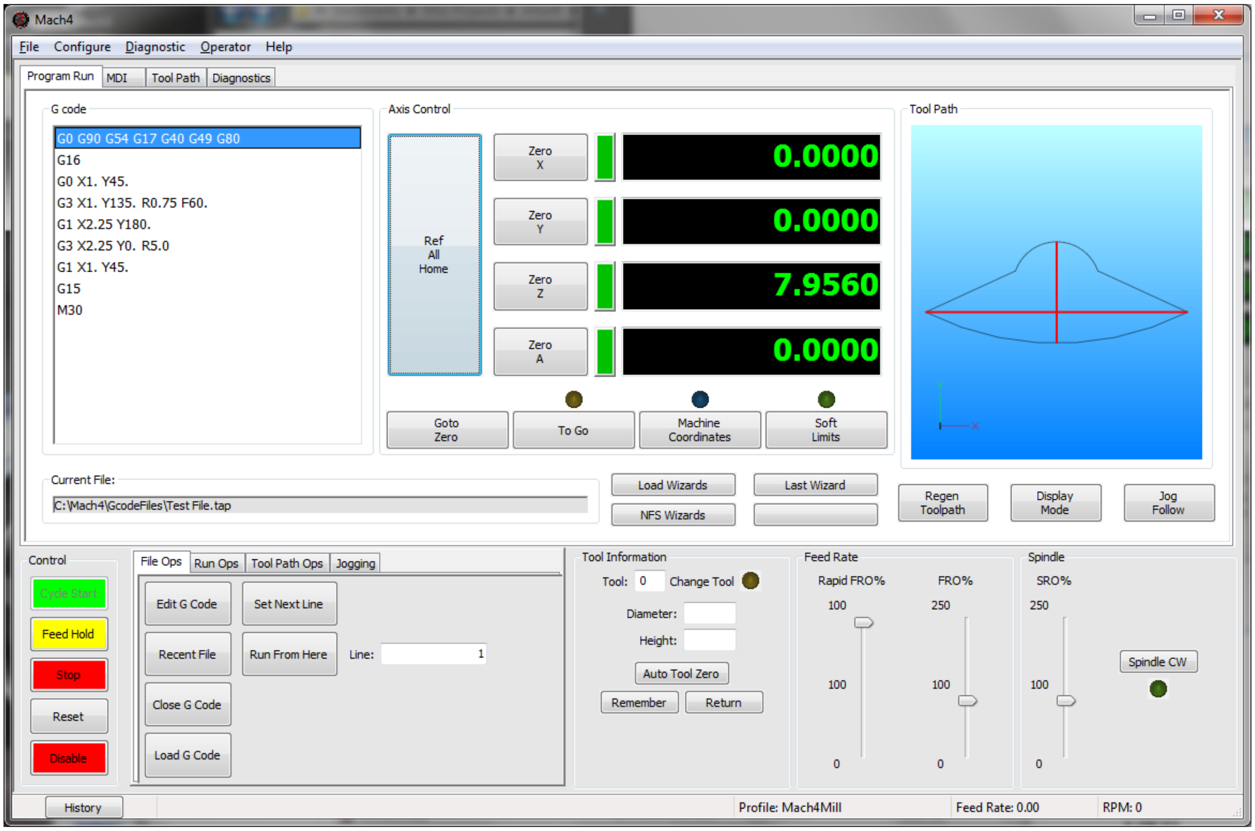
Mach4 CNC Controller

Lathe Programming Guide

Version 1.0

Mach4 Lathe is not yet completed - some listed gcodes are not implemented



```
G0 G90 G54 G17 G40 G49 G80
G16
G0 X1. Y45.
G3 X1. Y135. R0.75 F60.
G1 X2.25 Y180.
G3 X2.25 Y0. R5.0
G1 X1. Y45.
G15
M30
```

Current File: C:\Mach4\GcodeFiles\Test File.tap

Tool: 0

Line: 1

Profile: Mach4Mill

Feed Rate: 0.00

RPM: 0

Table of Contents

Chapter 1: Introduction

G Code 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.

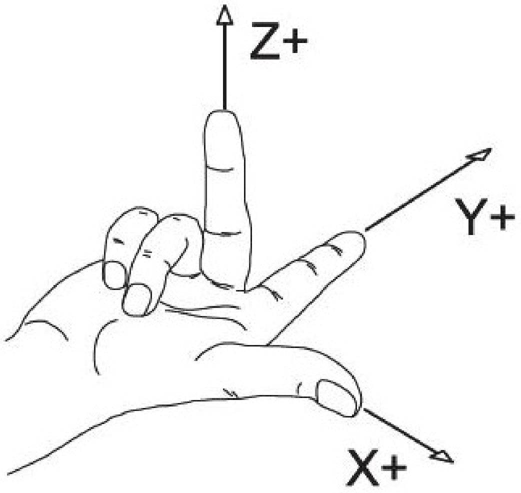


Figure A: Right hand rule

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 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 seconds. (also used in other functions)
N	Sequence numbers. Used for program organization and go to commands

Format

In writing G Code 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 cancelled.

G00 – Rapid mode

G18 – ZX plane select

G54 – Fixture offset 1

G40 – tool nose 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 just 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 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.

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 Code List

Code	Group	Description	Modal	Page
G00	1	Rapid Move	Y	7
G01	1	Linear Feed Move	Y	8
G02	1	Clockwise Arc Feed Move	Y	8
G03	1	Counter Clockwise Arc Feed Move	Y	8
G04	0	Dwell	N	12
G07.1	0	Cylindrical Interpolation	Y	12
G09	0	Exact stop	N	12
G10	0	Fixture and Tool Offset Setting	N	13
G12	1	Clockwise Circle	Y	16
G13	1	Counter Clockwise Circle	Y	16
G12.1	21	Polar Coordinate Interpolation	Y	17
G13.1	21	Polar Coordinate Interpolation Cancel	Y	17
G17	2	XY Plane Select	Y	18
G18	2	ZX Plane Select	Y	18

G19	2	YZ Plane Select	Y	18
G20	6	Inch	Y	20
G21	6	Millimeter	Y	20
G28	0	Zero Return	N	20
G30	0	2nd, 3rd, 4th Zero Return	N	22
G31	1	Probe function	N	22
G32	1	Threading*	N	22
G34	1	Variable Lead Threading*	N	24
G35	1	Clockwise Circular Threading*	N	25
G36	1	Counterclockwise Circular Threading*	N	25
G40	7	Cutter Compensation Cancel	Y	22
G41	7	Cutter Compensation Left	Y	27
G42	7	Cutter Compensation Right	Y	27
G50	0	Min/ Max Spindle Speed	N	27
G50.1	9	Cancel Scaling	Y	27
G51.1	9	Scale Axes	Y	28
G52	0	Local Coordinate System Shift	Y	28
G53	0	Machine Coordinate System	N	29
G54	12	Fixture Offset 1	Y	30
G54.1	12	Additional Fixture Offsets	Y	30
G55	12	Fixture Offset 2	Y	30
G56	12	Fixture Offset 3	Y	30
G57	12	Fixture Offset 4	Y	30
G58	12	Fixture Offset 5	Y	30
G59	12	Fixture Offset 6	Y	30
G60	0	Unidirectional Approach	N	30
G61	13	Exact Stop Mode	Y	30
G64	13	Cutting Mode (Constant Velocity)	Y	31
G65	0	Macro Call	N	31
G66		Macro Modal Call	Y	32
G67		Macro Modal Call Cancel	Y	32
G70	0	Finishing Cycle	N	36
G71	0	OD/ID Roughing Cycle	N	36
G72	0	Face Roughing Cycle	N	36
G73	0	Pattern Repeat Cycle	N	36
G73.1	16	High Speed Peck Drilling (Advanced)	Y	51
G74	0	Face Grooving Cycle	N	36
G74.1	16	Reverse Tapping (Advanced)	Y	53
G75	0	OD/ID Grooving/Cutoff Cycle	N	36
G76	0	Thread Cutting Cycle	N	36
G76.1	16	Fine Boring (Advanced)	Y	54

G80	16	Canned Cycle Cancel	Y	39
G81	16	Hole Drilling (Advanced)	Y	47
G82	16	Spot Face (Advanced)	Y	49
G83	16	Face Drilling	Y	40
G83.1	16	Deep Hole Peck Drilling (Advanced)	Y	50
G84	16	Face Tapping	Y	43
G84.1	16	Tapping (Advanced)	Y	52
G84.2	16	Right Hand Rigid Tapping* (Advanced)	Y	53
G84.3	16	Left Hand Rigid Tapping* (Advanced)	Y	53
G85	16	Face Boring	Y	45
G85.1	16	Boring, Retract at Feedrate (Advanced)	Y	55
G86	16	Boring, Retract at Rapid (Advanced)	Y	56
G87	16	Side Drilling	Y	41
G87.1	16	Back Boring (Advanced)	Y	57
G88	16	Side Tapping	Y	44
G88.1	16	Boring, Manual Retract (Advanced)	Y	59
G89	16	Side Boring	Y	46
G89.1	16	Boring with Dwell (Advanced)	Y	60
G90	1	OD/ID Cutting Cycle		
G90.1	4	Arc Center Absolute Mode	Y	33
G91.1	4	Arc Center Incremental Mode	Y	33
G92	1	Threading Cycle	Y	
G93	5	Inverse Time Feed	Y	34
G94	1	End Face Turning Cycle	Y	
G96	14	Constant Surface Speed*	Y	35
G97	14	Constant Speed	Y	35
G98	5	Feed Per Minute	Y	35
G98.1	10	Initial Point Return	Y	35
G99	5	Feed Per Revolution*	Y	35
G99.1	10	R Point Return	Y	36

* Implementation based on machine and control configuration

G Code Descriptions and Examples

Note:
For clarity this manual assumes a basic 2 axis lathe, X and Z. More advanced machines may have Y and C axes, or more as specified by the machine builder. Some examples and formats may omit the Y, A, B and C axes for clarity. They can be included in all motion moves.

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

G0 G54 G18 G40 G80	Safe start line
T1 M6	Tool change
S2500 M3	Start spindle
G0 X1.0 Z3.0	Rapid to XZ position
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
T1 M6	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 to the forward, and the Z axis positive direction is pointing to the right). See figure 2-1 for a graphic representation of the motion of a G02. The start point is the current position of the machine. Specify the end point with X and Z. 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.

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__

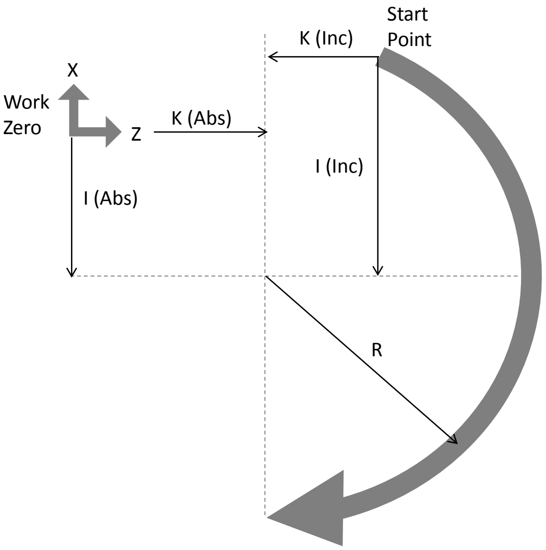


Figure 2-1: Path of tool point in circular and helical interpolation (G02).

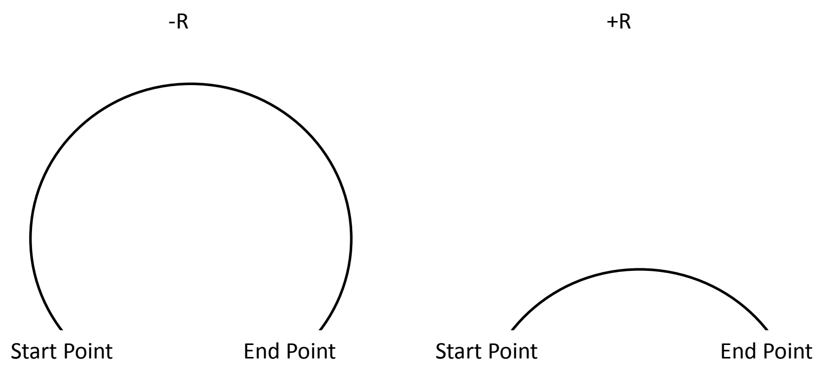


Figure 2-2: Difference between negative and positive values for R

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. 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. See figure 2-2 for the tool path.

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

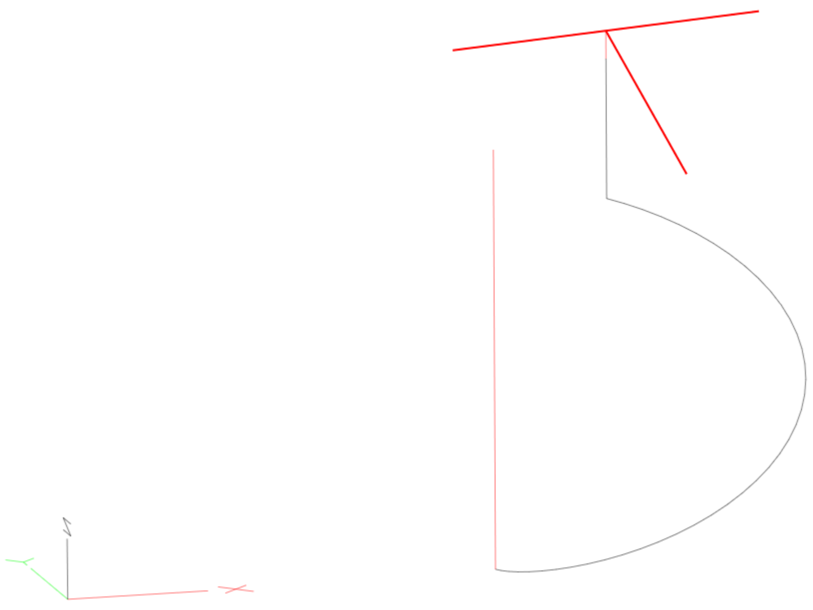


Figure 2-2: Helical interpolation example toolpath

G04 – Dwell: A dwell is simply a pause in the program. The duration of the dwell is specified by P in 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.

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

G07.1 – Cylindrical Interpolation: The cylindrical interpolation mode simplifies the programming of machined profiles around the circumference of a part, a cam profile for example. Machining of this type requires the lathe spindle to also be a positioning axis (normally C) and a rotating (live) tool in the turret or tool post. Lines and arcs can be programmed; however arcs can only be programmed using R for the radius. Arc center point, I, J, K programming is not supported.

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 sharp corners 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.

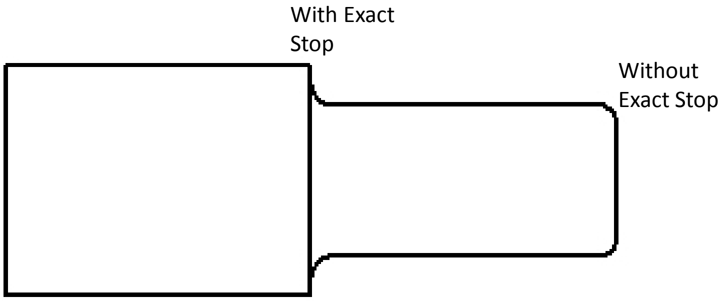


Figure 9-1: Shaft with sharp corner

G10 – Fixture and Tool Offset Setting: 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. 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.

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

Format (Absolute): G10 L2 P__ X__ Y__ Z__ A__ B__ C__
Format (Absolute): G10 L2 P__ U__ V__ W__ H__

L selects the function of the G10 block, different values will have different functions. L2 is the designation for fixture offset setting. 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__
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

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.

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

Work shift and head shift can also be set with G10. Work shift is set using G10 L2 P0. Head shift is set using G10 L20 P0. Notice the L20 for head shift. All other values are set in the same format as the other fixture offsets.

Tool offset setting requires just as much care as setting of fixture offsets. G90 and G91 affect the setting of offset values in the same way. 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.

For tool geometry offsets:
Format (Absolute): G10 L1 P__ X__ Y__ Z__ R__ Q__
Format (Incremental): G10 L1 P__ U__ V__ W__ C__ Q__

For tool wear offsets:
Format (Absolute): G10 L3 P__ X__ Y__ Z__ R__ Q__
Format (Incremental): G10 L3 P__ U__ V__ W__ C__ Q__

Again, not all values are required, if omitted that value simply will not be set. L1 specifies tool geometry offset setting, L3 specifies tool wear offset setting, P is again the offset number to be set (offset #1 = P1, offset #2 = P2, etc.). 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
T	Tool tip direction

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 L1 P5 X-5.0	Set X axis offset #5 to -5.0
G10 L3 P2 W.05	Add .05 to Z wear for offset #2
M30	Program end and rewind

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 R for the start radius and Q 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__ R__ Q__

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

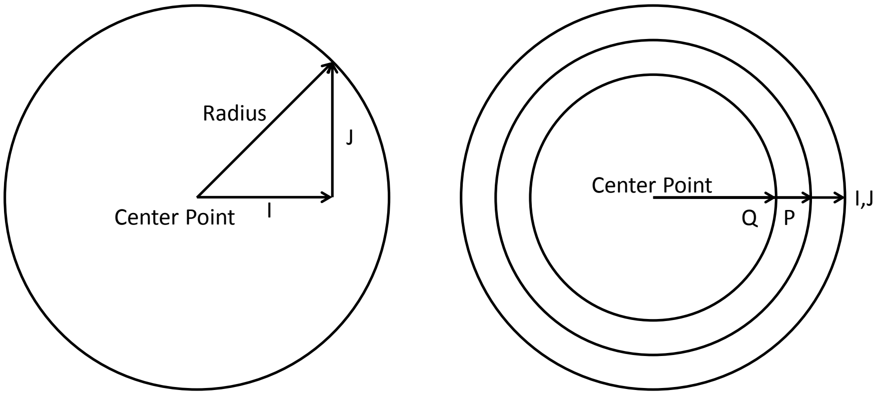


Figure 12-1: Tool motion during circle interpolation

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°)=.7071, Y=1*Sin(45°)=.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

G12.1/G13.1– Polar Coordinate Interpolation: Polar coordinate interpolation mode can be used in lathes with C axis and milling to machine profiles on the face of a part. This mode requires the lathe spindle to also be a programmable positioning axis (normally C) and a rotating (live) tool in the turret or tool post. Not all machines will have this capability.

Polar coordinate interpolation greatly simplifies programming of profiles on the part face. The coordinate system is as shown in figure 13-1. In this mode the C positions are programmed as a linear axis with radius values. The X axis is programmed as normal, in diameter values. So if figure 13-1 showed a 1.0 inch square, then the top right corner would have the coordinate X1.0, C0.5. Arcs and lines can be cut in this mode. Arcs are programmed using R or I, J to define the radius. In the G12.1 mode the current plane selection is invalid; the arcs are on a plane defined by the current linear axis and a virtual axis defined by the rotation axis. By default this is the linear axis is the X axis and the rotational axis is the C axis. With this setting I is the arc center in the X and J is the arc center in the C. The linear and rotation axes are set by parameters: #5460 and #5461. When the linear axis is the Y axis use J, K and when the Z axis is the linear axis use K, I.

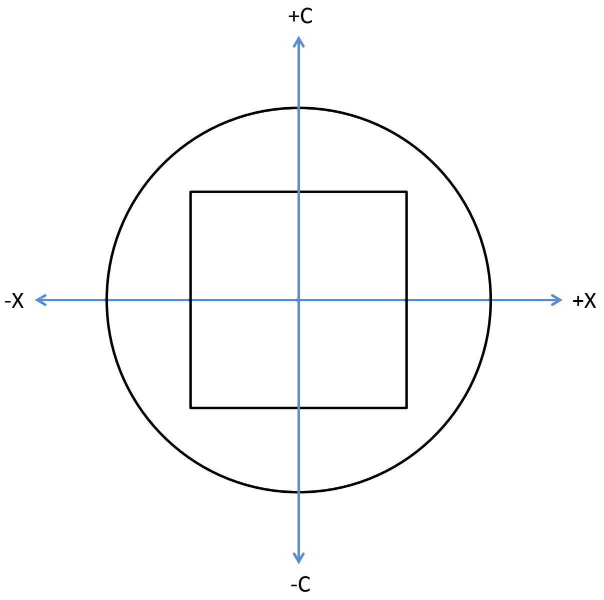


Figure 13-1: Polar coordinate interpolation programming coordinates

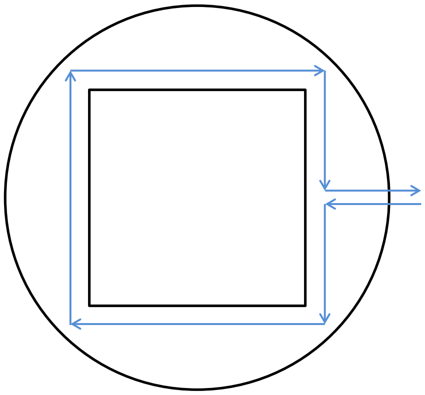


Figure 13-2: Polar coordinate interpolation example

Example: Program a 1.0 inch square on the face of part, Figure 13-2

G0 G18 G40 G80	Safe start line, absolute mode, XY plane
Mxx	Enable C axis
G28 H0.0	Send C axis to home position
T0202	Call Tool 2 Offset2
G0 X1.5 Z.1	Move to initial position
G98	UPM Mode
G1 Z-.25 F30.0	Feed Z into part
G12.1	Enable polar coordinate interpolation mode
G42 X1.0	Lead-in, enable tool radius compensation
C-0.5	Feed move
X-1.0	Feed move
C0.5	Feed move
X1.0	Feed move
C0.0	Feed move
G40 X1.5	Lead-out, disable tool radius compensation
G13.1	Disable polar coordinate interpolation mode
G0 G53 X0.0 Z0.0	Return to machine zero
M30	Program end and rewind

G17/G18/G19 – Plane Selection: Arcs, circles and drill cycles require the selection of a plane. The 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. The default plane selection is G18.

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.

Canned 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.

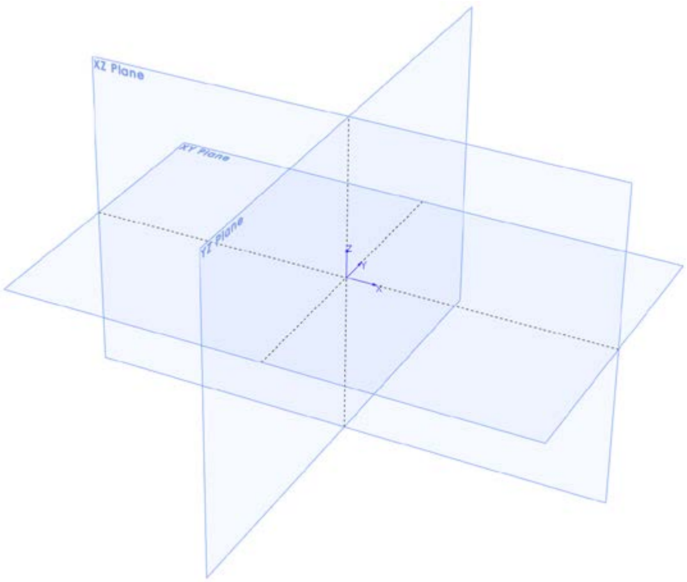


Figure 17-1: Planes

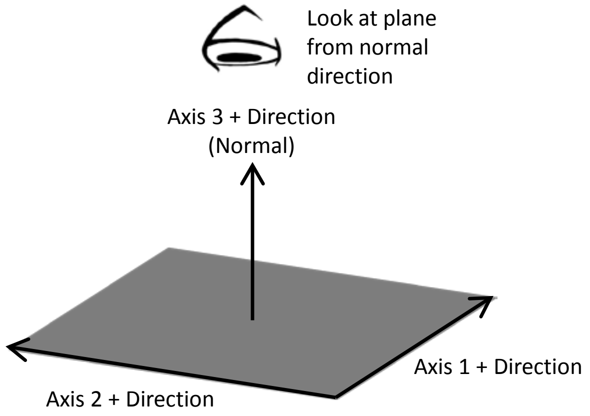


Figure 17-2: Plane orientation

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 DRO's, 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.

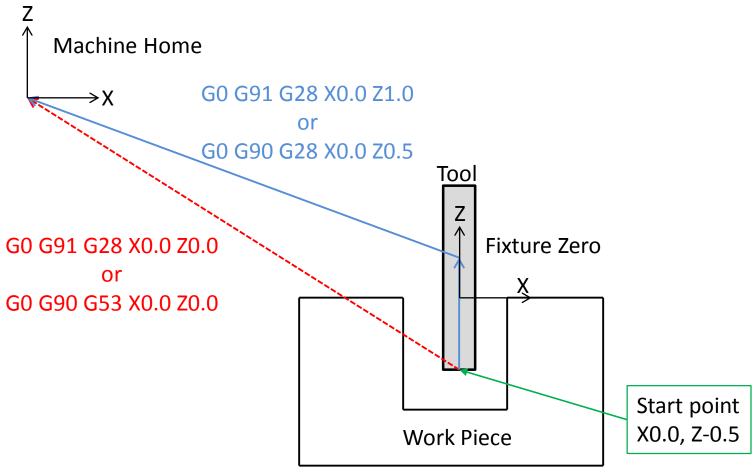


Figure 28-1: Zero return via intermediate point

Format: G28 X__ Y__ Z__ A__ B__ C__

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 G90 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 that sets the machine to absolute position mode. The next line causes 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 the common programming format is as follows:

G91 G28 Z0

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

G30 – 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 2nd, 3rd, or 4th zero return point, specified by P2, P3, or P4 respectively. If P is omitted the 2nd zero return point is selected. This is handy for tool changers that are not located at the home position or any number of other applications.

Format: G30 P__ X__ Y__ Z__ A__ B__ C__

The 2nd zero return point is defined by # variables as follows:

Axis	P2 # Variables	P3 # Variables	P4 # Variables
X	5301	5311	5321
Y	5302	5312	5322
Z	5303	5313	5323
A	5304	5314	5324
B	5305	5315	5325
C	5306	5316	5326

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

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 for probe 1, G31.1 probe 2, G31.2 probe 3 and G31.3 probe 4. Motion is defined along linear axes, in a similar format to G01, with a feedrate.

Format: G31 X__ Y__ 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	G31 # Variables	G31.1 # Variables	G31.2	G31.3
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. The compensate 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__ Y__ Z__ F__ Q__

The G32 threading cycle is a single linear move synced to the spindle speed. F specifies the lead or pitch of the thread. 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. Machine builders and advanced users have the added option to create custom M codes to specify the feed vector to create custom threads. Address Q specifies the start angle of the thread. This can be used for creating threads timed to a feature of the part or for cutting threads with multiple starts. This feature requires position feedback from the spindle, the accuracy of the feedback will determine the accuracy of the start angle. Not all machines will have this capability.

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

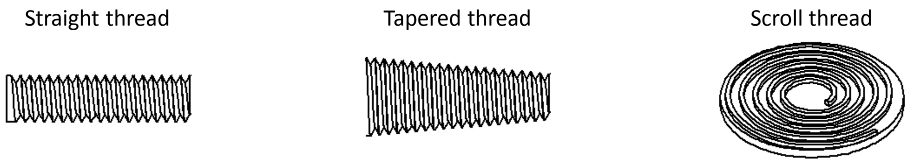


Figure 32-1: Basic thread types

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__ Y__ Z__ F__ K__ Q__

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.

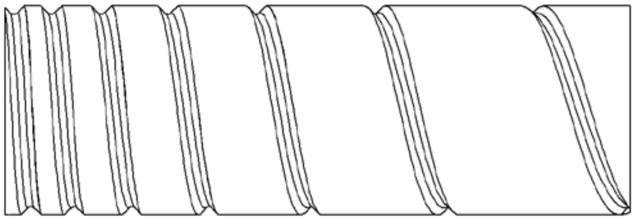


Figure 34-1: Variable lead thread

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 by 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.

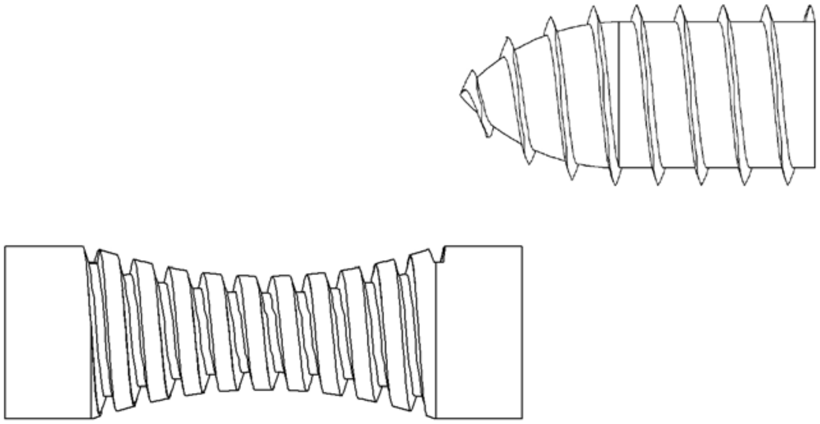


Figure 35-1: Circular thread examples

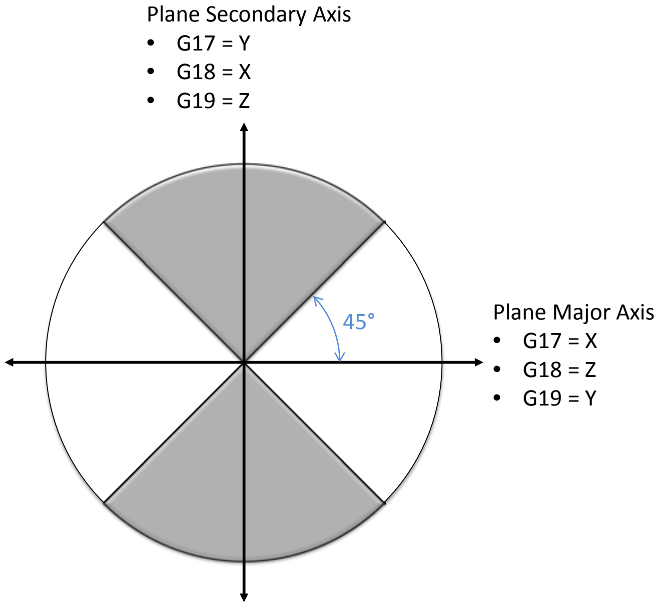


Figure 35-2: Circular thread arc allowable range

The programing format contains the same addresses as G02/G03 and G32; they perform the same functions 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 – Cutter Compensation Cancel: Cancels the cutter compensation mode.

G41/G42 – Cutter Compensation Left/Right: Enables cutter compensation to the left (G41) or right (G42) of the cutter path by an amount specified in the tool offset.

Format: G1 G42 X__ Y__ Z__ F__

For detailed information see the cutter 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, is a prime example of this. An unbalanced part can create harsh vibration when spun over a certain RPM. 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.

Format: G50 S__ Q__

In this format S specifies the maximum spindle speed. It is also possible to specify a minimum spindle speed using Q.

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__ Y__ Z__ A__ B__ 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. 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__ Y__ Z__ A__ B__ 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 cancelled
M30	Program end and rewind

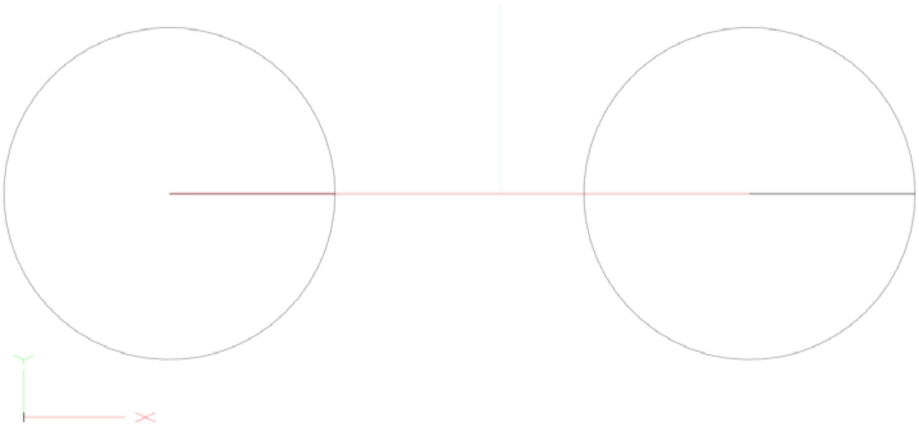


Figure 52-1: Example program tool path.

Once set, the setting will remain until cancelled by another G52 or the system is reset. As in the example above, a local coordinate system can be cancelled 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__ Y__ Z__ A__ B__ 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: Used to select the active fixture offset in a program. 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.

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.

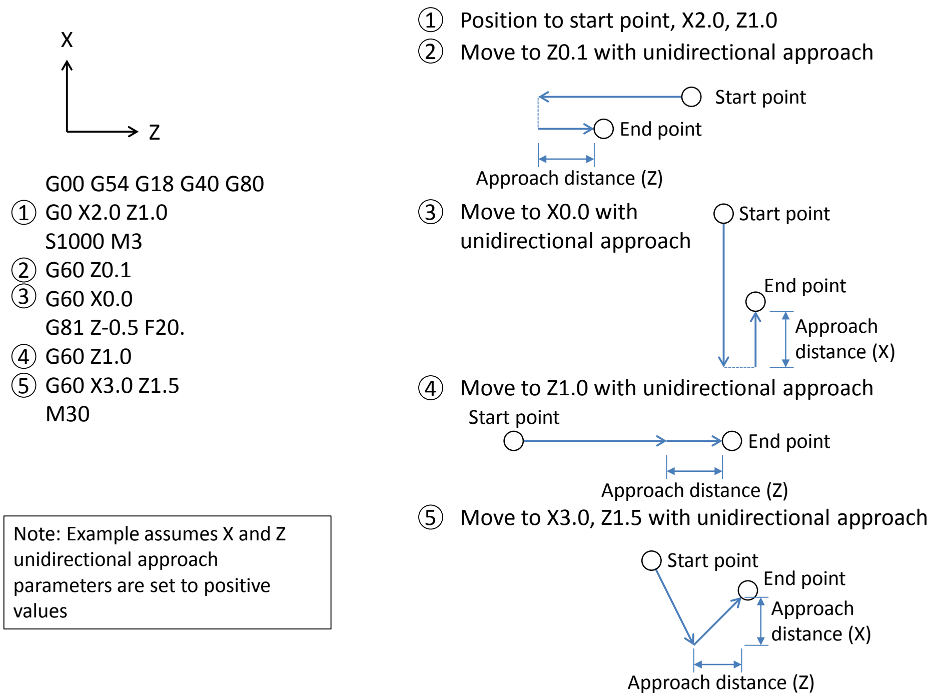
Please see the fixture offset section of this manual for more detail on setting up fixture offsets.

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, see figure 60-1. 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__ Y__ Z__

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



```
   G00 G54 G18 G40 G80
①  G0 X2.0 Z1.0
   S1000 M3
②  G60 Z0.1
③  G60 X0.0
   G81 Z-0.5 F20.
④  G60 Z1.0
⑤  G60 X3.0 Z1.5
   M30
```

Note: Example assumes X and Z unidirectional approach parameters are set to positive values

① Position to start point, X2.0, Z1.0

② Move to Z0.1 with unidirectional approach

③ Move to X0.0 with unidirectional approach

④ Move to Z1.0 with unidirectional approach

⑤ Move to X3.0, Z1.5 with unidirectional approach

Figure 60-1: 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, or in side milling with arcs, the exact stop mode will produce jerky motion and witness marks on the work piece. For most milling 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 68) 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, but G66 remains active until cancelled. The block(s) following the G66 can contain new positions and variables to run the macro again.

Format: G66 P____ A__ B__ C__ …
A__ B__ C__ …
G67

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

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

G70-G76 – Multiple Repetitive Cycles: Programming the roughing and finishing of turned profiles can be made easier with the use of multiple repetitive cycles. See the Multiple Repetitive Cycles section of this manual for detailed information.

G70	Finishing Cycle	
G71	Rough Turning	
G72	Rough Facing	
G73	Pattern Repeat	
G74	Face Groove	
G75	OD/ID Groove	
G76	Multiple Thread Cutting	

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.

G80	Canned Cycle Cancel	G80
G81	Drilling	G81 X_ Y_ Z_ R_ F_
G82	Spot Face	G82 X_ Y_ Z_ R_ P_ F_
G83	Deep Hole Peck Drilling	G83 X_ Y_ Z_ R_ Q_ F_
G84	Tapping	G84 X_ Y_ Z_ R_ F_

G85	Boring, Retract at Feed, Spindle On	G85 X_ Y_ Z_ R_ F_
G86	Boring, Retract at Rapid, Spindle Off	G86 X_ Y_ Z_ R_ F_
G87	Back Boring	G87 X_ Y_ Z_ R_ I_ J_ F_
G88	Boring, Manual Retract	G88 X_ Y_ Z_ R_ P_ F_
G89	Boring, Dwell, Retract at Feed, Spindle On	G89 X_ Y_ Z_ R_ P_ F_

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.

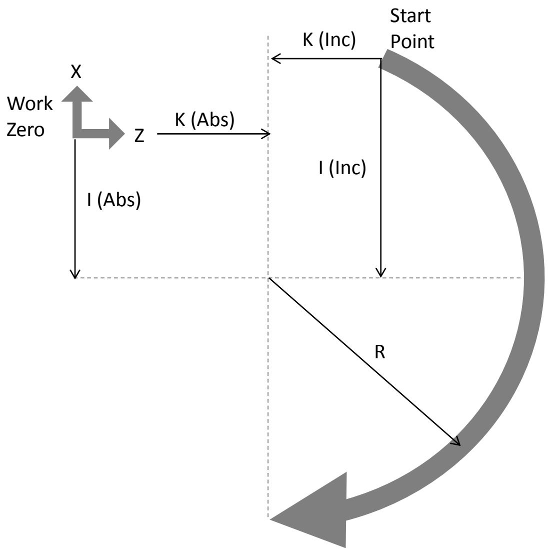


Figure 2-1: Path of tool point in circular and helical interpolation (G02).

Example: Program an arc centered at 1.0, 0.0 in the XY plane with radius 2. Start point at 3.0,0.0 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
S2500 M3                            Start spindle
G0 X3.0 Y0.0                        Position to X and Y start point
G43 H1 Z.5                          Activate tool offset 1 and move to Z rapid plane
G1 Z0.0 F10.0                       Z plunge at feedrate
G3 X1.0 Y2.0 I-2.0 J0.0 F10.0       Arc move
G0 Z.5                              Retract Z to rapid plane
G0 G53 Z0.0                         Return Z to home
M30                                 Program end and rewind
```

G0 G54 G18 G40 G80	Safe start line
G90.1	Absolute arc center mode
T0101	Tool change
S2500 M3	Start spindle
G0 X3.0 Y0.0	Position to X and Y start point
G43 H1 Z.5	Activate tool offset 1 and move to Z rapid plane
G1 Z0.0 F10.0	Z plunge at feedrate
G3 X1.0 Y2.0 I1.0 J0.0 F10.0	Arc move
G0 Z.5	Retract Z to rapid plane
G0 G53 Z0.0	Return Z 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 – Local Coordinate System Setting: The coordinate system can be set by issuing G92 in the program. This function differs from G52 (Local Coordinate System Shift) in the way that it is specified. While G52 is specified with shift values, G92 is specified with the desired axis position. The affect is global and should be used with caution.

Format: G92 X__ Y__ Z__ A__ B__ C__

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

G92 was used for fixture offset setting before fixture offsets were available. It is recommended that the fixture offsets be used instead of using G92. The offset amount of the G92 setting is not immediately known by the user, because of this the results can be unpredictable when fixture offsets and G92 are combined.

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:

$$F = \frac{1}{time\ (min)} = \frac{feedrate\ (units/min)}{distance\ (units)}$$

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 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 a 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.

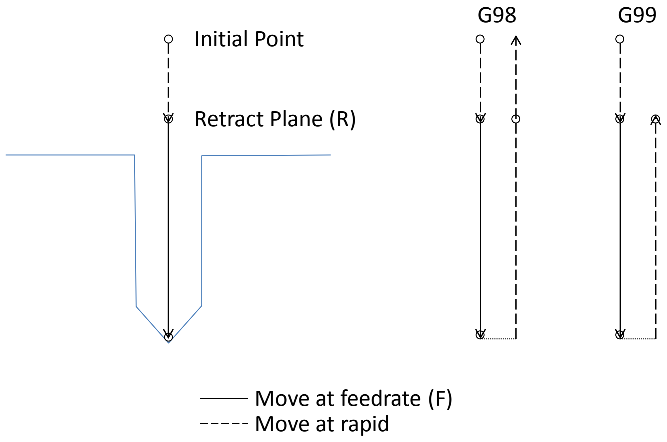


Figure 981-1: Initial and R point return setting.

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 a canned cycle end at the programmed R level, see figure 981-1. When drilling holes in a flat plate, G99 can be used to reduce excessive machine movement decreasing cycle time.

Chapter 3: Multiple Repetitive Cycles

Multiple repetitive cycles are useful for reducing program complexity. By defining only the finished part shape in G code it is possible to rough turn and face, finish turn, thread and groove a part. Roughing passes are automatically calculated using the specified part shape and stock diameter.

G70 – Finishing Cycle: After roughing a part using the G71, G72 or G73 the part can be finished using the G70 cycle.

Format: G70 A__ P__ Q__

A – Program number of the program containing the finished part shape. If omitted the current program is used.

P – Sequence (N) number of the first block of the finished part shape.

Q – Sequence (N) number of the last block of the finished part shape.

G71 – OD/ID Roughing Cycle:

G72 – Face Roughing Cycle:

G73 – Pattern Repeat Cycle:

G74 – Face Grooving Cycle:

G75 – OD/ID Grooving Cycle:

G76 – Thread Cutting Cycle: Used to thread a part with only 2 lines of Gcode

Format Line 1: G76 P__|__|__ Q__ R__ K__

P–Three part word each section will be divided as (m) (r) (a) in their respective order.

m- Number of repetitive finish passes (1 to 99)

r-The amount of chamfer for the machine to pull out of the thread. The r value will be multiplied by the pitch to calculate the position to start pulling out. An r value of 05 (0.5) will cause the thread to pull out in half a pitch.

a-Angle of the thread to be cut

Example: 2 Spring passes, retract in 1 pitch, 60 Deg thread would be represented as: P021060

Q–Minimum cutting depth when doing a threading pass

R–Finish pass allowance

K–Cutting method

K = 0: Flank in-feed constant volume removal for each pass

K = 1: Alternate flank in-feed constant volume removal for each pass

K = 2: Flank in-feed constant depth removal for each pass (each pass same as first pass)

K = 3: Alternate flank in-feed constant depth removal for each pass (each pass same as first pass)

Format Line2: G76 X__ Z__ R__ P__ Q__ F__

X–End Diameter of thread

Z–End Position of thread

R–Difference in radius from start to end of thread (straight thread r=0)

P–Height of thread

Q–Depth of First cut

F–Lead of thread

Chapter 4: Hole Machining Canned Cycles

Standard Hole Machining Cycles		
G80	Canned Cycle Cancel	G80
G83	Face Drilling	G83 X_ Z_ R_ Q_ P_ F_
G84	Face Tapping	G84 X_ Z_ R_ P_ F_
G85	Face Boring	G85 X_ Z_ R_ P_ F_
G87	Side Drilling	G87 Z_ X_ R_ Q_ P_ F_
G88	Side Tapping	G88 Z_ X_ R_ P_ F_
G89	Side Boring	G89 Z_ X_ R_ P_ F_

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

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 competed 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. In the standard set the drilling direction is set by the code called, the current plane selection does not determine the drilling axis. For the face machining cycles the Z axis will always be the drilling axis and for side machining cycles the X axis will always be the drilling axis. In the advanced cycles the current plane selection determines the drilling axis.

For the advanced cycles 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).

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.

Axis positions can be programed 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:

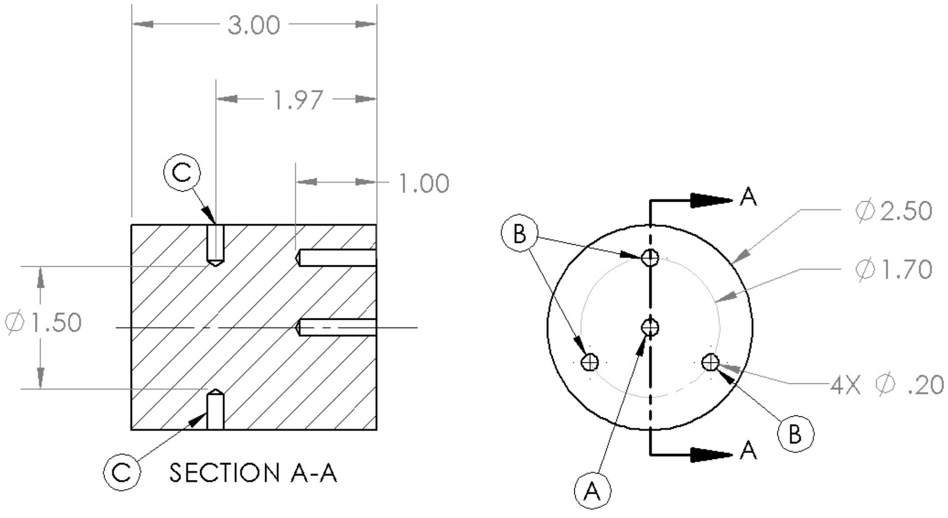
Gcc **G98.1/99.1 X**xx **Z**zz **Q**qq **R**rr **P**pp **L**ll **F**ff
Xxn
G80

Incremental:
Gcc **G98.1/99.1 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 seconds, 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.

Figure 102: Example hole pattern



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 deep 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

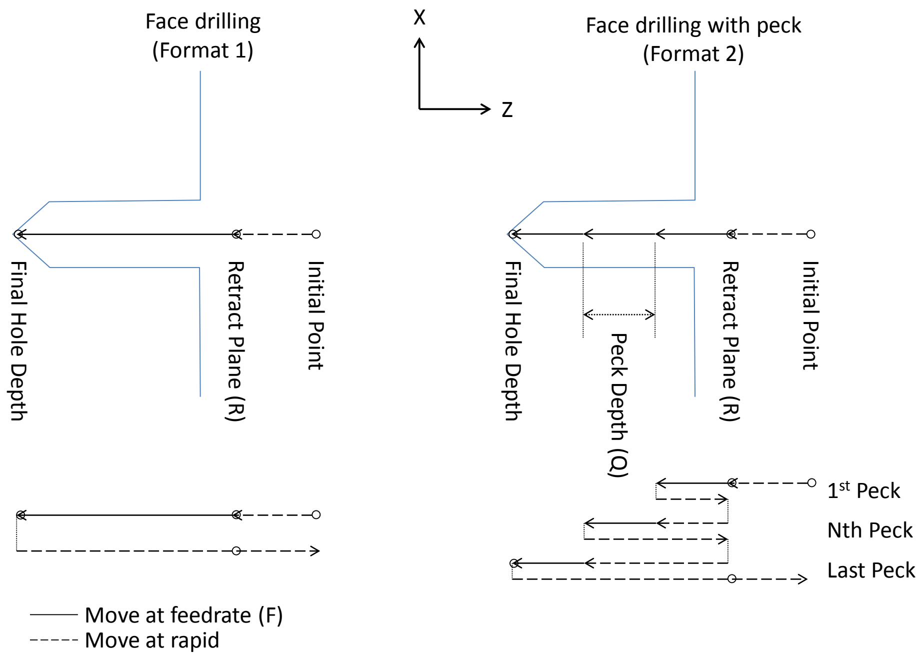


Figure 83-1: Motion of tool point for G83 cycle.

Example: Spot drill hole A from figure 102 and then drill to depth.

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
G83 G99.1 X0.0 Z-0.1 R.25 F0.01	Drill cycle start
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
G0 G54 G18 G40 G80	Safe start line
T0202	Tool change
S3000 M3	Start spindle
G0 Z1.0	Move to Z initial point
X0.0	Move to X position
G83 G99.1 X0.0 Z-1.0 R0.25 Q0.20 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

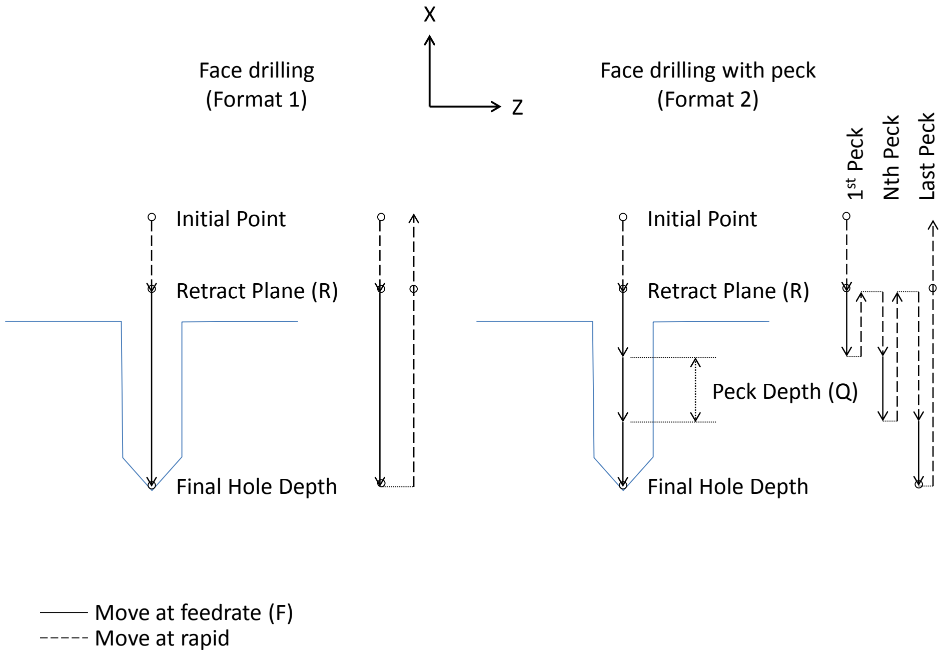


Figure 87-1: Motion of tool point for G87 cycle.

Example: Spot drill holes C from figure 102 and then drill to depth.

G0 G54 G18 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z-1.97	Move to Z initial point
X3.0 C0.0	Move to X and C position
G87 G99.1 Z1.97 C0.0 X2.30 R3.0 F0.01	Drill cycle start
C180.0	Drill 2nd hole
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
G0 G54 G18 G40 G80	Safe start line
T0202	Tool change
S3000 M3	Start spindle
G0 Z-1.97	Move to Z initial point
X3.0 C0.0	Move to X and C position
G87 G99.1 Z-1.97 C0.0 X1.50 R3.0 Q0.20 F0.008	Drill cycle start

C180.0	Drill 2nd hole
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

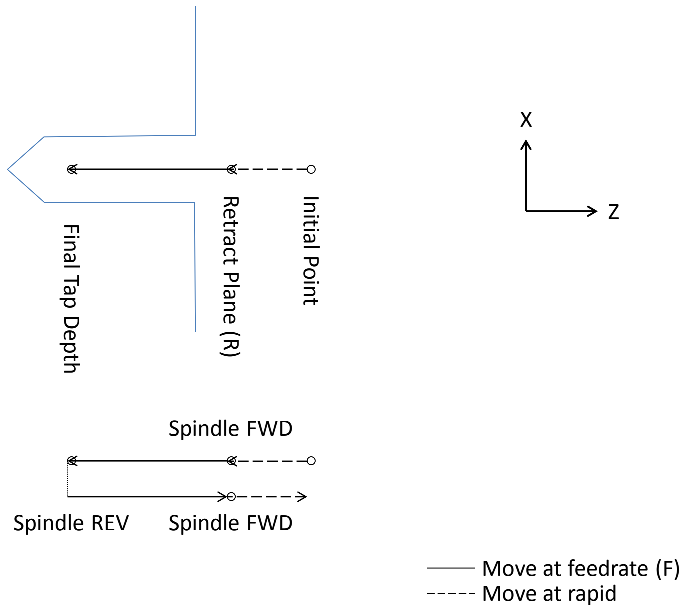


Figure 84-1: Motion of tool point for G84 cycle.

Example:
Create the program to tap hole A shown in figure 102 to a depth of .500 with a ¼-20 tap using feed/min.

1/TPI = 1/20 = .05 Pitich
RPM*Pitch = 1000*.05 = 50.0 UPM

G0 G54 G18 G40 G80	Safe start line
T0202	Tool change
S1000 M3	Start spindle
G0 Z1.0	Move to Z initial point
X0.0	Move to X position
G98	Feed per minute mode
G84 G99.1 X0.0 Z-0.5 R.25 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

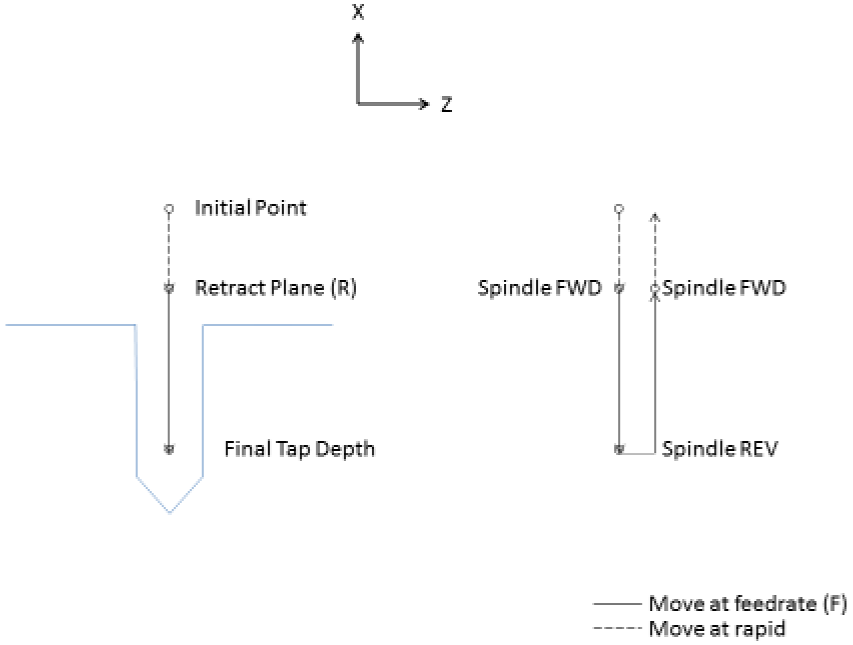


Figure 88-1: Motion of tool point for G88 cycle.

Example:
Create the program to tap holes C shown in figure 102 to a depth of 0.375 with a ¼-20 tap using feed/min.

1/TPI = 1/20 = .05 Pitich
RPM*Pitch = 1000*.05 = 50.0 UPM

G0 G54 G18 G40 G80	Safe start line
T0202	Tool change
S1000 M3	Start spindle
G0 Z-1.97	Move to Z position
X3.0 C0.0	Move to X and C position
G98	Feed per minute mode
G84 G99.1 Z-1.97 C0.0 X1.75 R3.0 F50.0	Tap cycle start
C180.0	Tap second hole
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

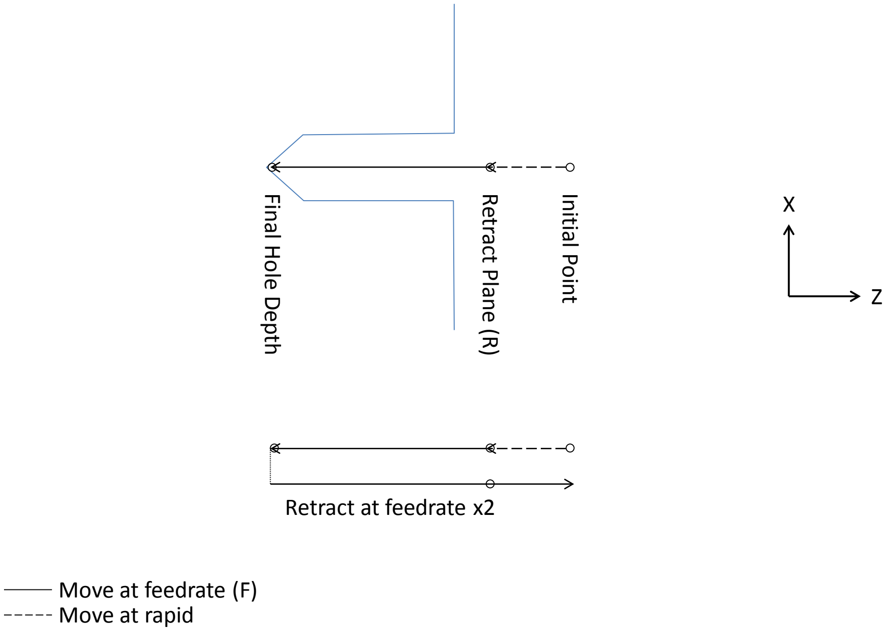


Figure 85-1: Motion of tool point for G85 cycle.

Example: Bore hole A from figure 102 to a depth of 0.250.

G0 G54 G18 G40 G80	Safe start line
T0303	Tool change
S750 M3	Start spindle
G0 Z1.0	Move to Z initial point
X0.0	Move to X position
G85 G99.1 X0.0 Z-0.250 R.25 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

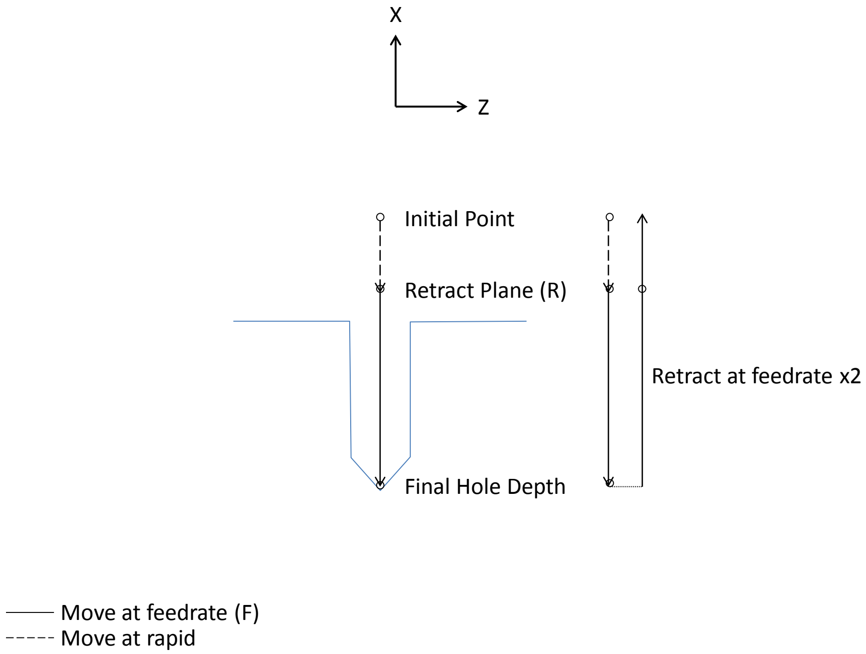


Figure 89-1: Motion of tool point for G89 cycle.

Example: Bore holes C from figure 102 to a depth of 0.250.

G0 G54 G18 G40 G80	Safe start line
T0303	Tool change
S750 M3	Start spindle
G0 Z-1.97	Move to Z initial point
X3.0 C0.0	Move to X and C position
G89 G99.1 Z1.97 C0.0 X2.00 R3.0 F0.003	Bore cycle start
C180.0	Bore 2nd hole
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Advanced Drilling

G81 – 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 81-1 for a graphic of the tool motion.

Format: G81 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

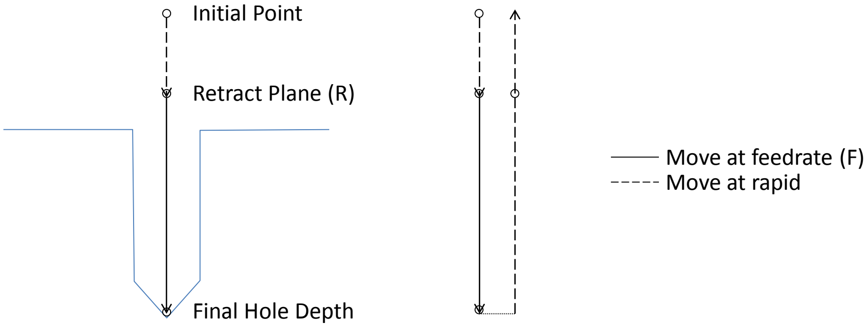


Figure 81-1: Motion of tool point for G81 cycle.

Example:
Create the program for the B holes shown in figure 102.

G0 G54 G17 G40 G80	Safe start line, select XY plane
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial point
X1.70 C0.0	Move to X and C position of first hole
G81 G99.1 X1.70 C0.0 Z-1.0 R.25 F0.01	Drill cycle start
C120.0	Drill second hole
C240.0	Drill third hole
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

Example: Drill holes C from figure 102.

G0 G54 G19 G40 G80	Safe start line, select YZ plane
T0101	Tool change
S2500 M3	Start spindle
G0 Z-1.97	Move to Z initial point
X3.0 C0.0	Move to X and C position
G81 G99.1 Z-1.97 C0.0 X1.50 R3.0 F0.008	Drill cycle start
C180.0	Drill 2nd hole
G80	Canned cycle cancel
G0 G53 X0.0 Z0.0	Return to home
M30	Program end and rewind

G82 – 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 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 B holes shown in figure 102, dwell for .2 seconds.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial point
X1.70 C0.0	Move to X and C position of first hole
G82 G99.1 X1.70 C0.0 Z-0.125 R.25 P0.2 F0.01	Drill cycle start
C120.0	Drill second hole
C240.0	Drill third hole
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

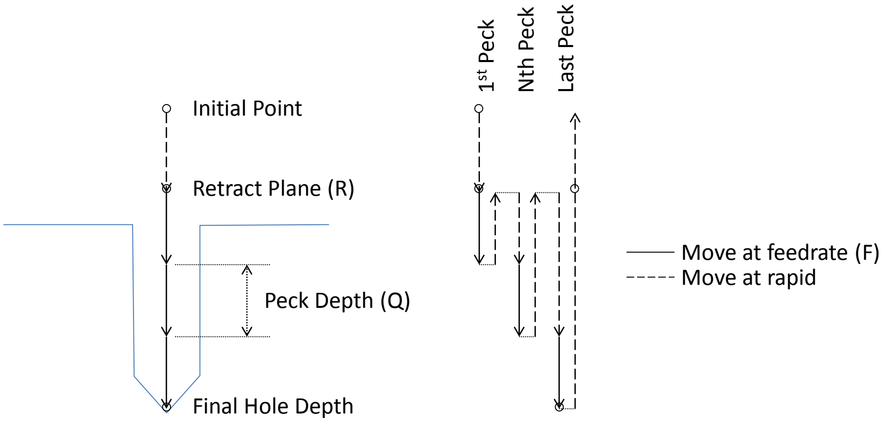


Figure 831-1: Motion of tool point for G83.1 cycle.

Example:
Create the program for the B holes shown in figure 102 using a .1 peck depth.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial point
X1.70 C0.0	Move to X and C position of first hole
G83.1 G99.1 X1.70 C0.0 Z-1.0 R.25 Q0.1 F0.01	Drill cycle start
C120.0	Drill second hole
C240.0	Drill third hole

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 .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

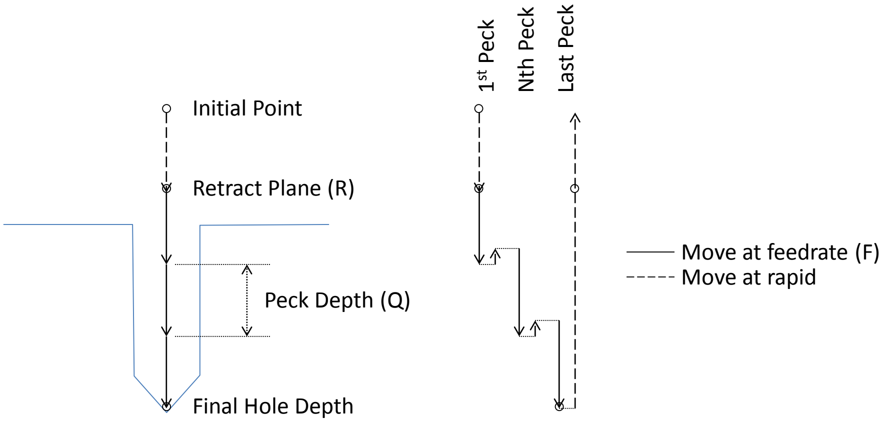


Figure 731-1: Motion of tool point for G73.1 cycle.

Example:
Create the program for the holes shown in figure 102 using a .05 peck depth.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial point
X1.70 C0.0	Move to X and C position of first hole
G73.1 G99.1 X1.70 C0.0 Z-1.0 R.25 Q0.05 F0.01	Drill cycle start

C120.0	Drill second hole
C240.0	Drill third hole
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

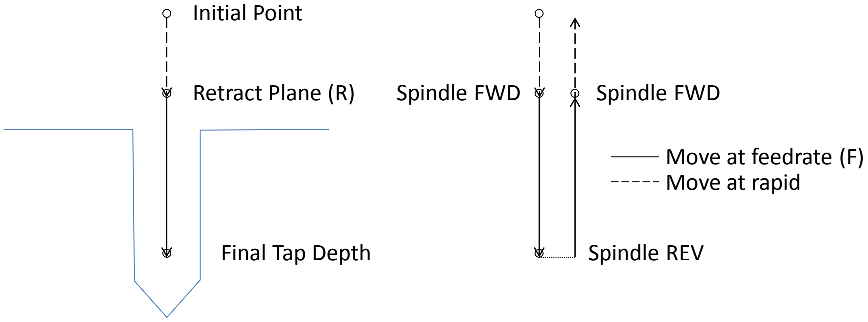


Figure 841-1: Motion of tool point for G84.1 cycle.

Example:
Create the program to tap the holes shown in figure 102 to a depth of .500 with a ¼-20 tap using feed/min.

1/TPI = 1/20 = .05 Pitich
RPM*Pitch = 1000*.05 = 50.0 UPM

G0 G54 G17 G40 G80	Safe start line
T0202	Tool change
S1000 M3	Start spindle
G0 Z1.0	Move to Z initial point
X1.70 C0.0	Move to X and C position of first hole
G98	Feed per minute mode
G84.1 G99.1 X1.70 C0.0 Z-1.0 R.25 F50.0	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 84-1 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 seconds
L – Number of repetitions
F – Feedrate
J – Spindle speed for retract

As will 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 the holes shown in figure 2 to a depth of .500 with a 1/4-20 tap using feed/rev. Tap at 1000 RPM, retract at 2000 RPM

G0 G54 G17 G40 G80	Safe start line
T0202	Tool change

S1000 M3	Start spindle
G0 Z1.0	Move to Z initial point
X1.70 C0.0	Move to X and C position of first hole
G99	Feed per revolution mode
G84.1 G99.1 X1.70 C0.0 Z-1.0 R.25 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 seconds
L – Number of repetitions
F – Feedrate

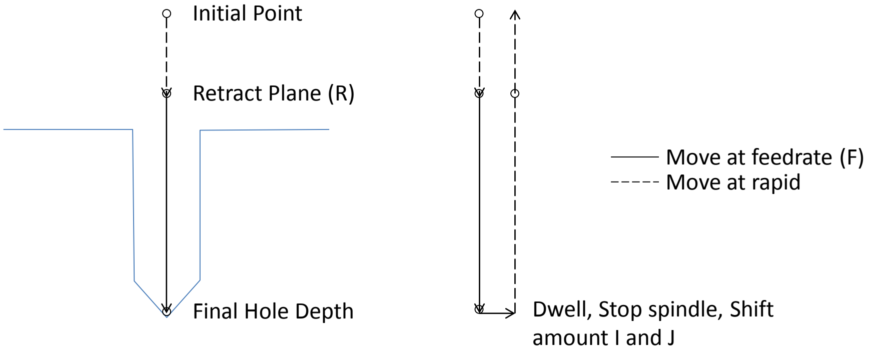


Figure 761-1: Motion of tool point for G76.1 cycle

After feeding to the bottom of the hole, the machine with 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:

INSERT MANUAL M19 MACRO PROGRAM HERE

Example:
Create the program to fine bore the B holes shown in figure 102.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z point
X1.7 C0.0	Move to X and C point of first hole
G76.1 G99.1 X1.7 C0.0 Z-1.0 R.25 I-.025 F10.0	Fine bore cycle, shift X-.025 at bottom
C120.0	Bore second hole
C240.0	Bore third hole
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

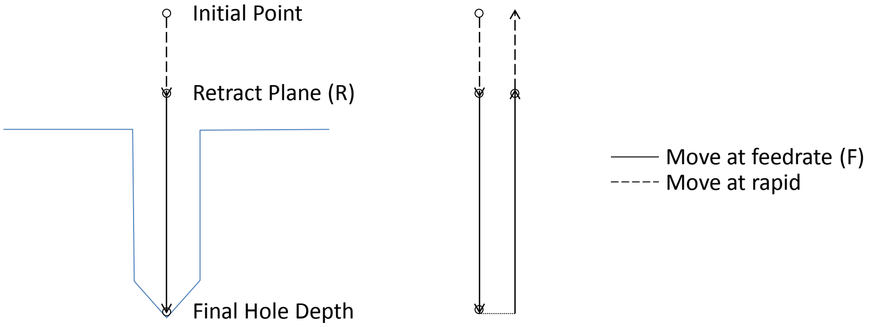


Figure 851-1: Motion of tool point for G85.1 cycle

Example:
Create the program to bore the B holes shown in figure 102.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial position
X1.70 C0.0	Move to X and C positions of first hole
G85.1 G99.1 X1.7 C0.0 Z-1.0 R.25 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

G86 – 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

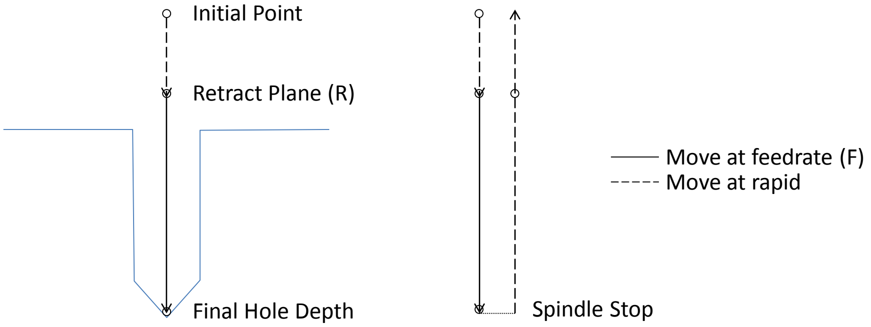


Figure 86-1: Motion of tool point for G86 cycle

Example:
Create the program to bore the B holes shown in figure 102.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial position
X1.70 C0.0	Move to X and C positions of first hole
G86 G99.1 X1.7 C0.0 Z-1.0 R.25 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: G87.1 is a back boring cycle. This 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 the end the cycle. For that reason this canned cycled 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 seconds
L – Number of repetitions
F – Feedrate

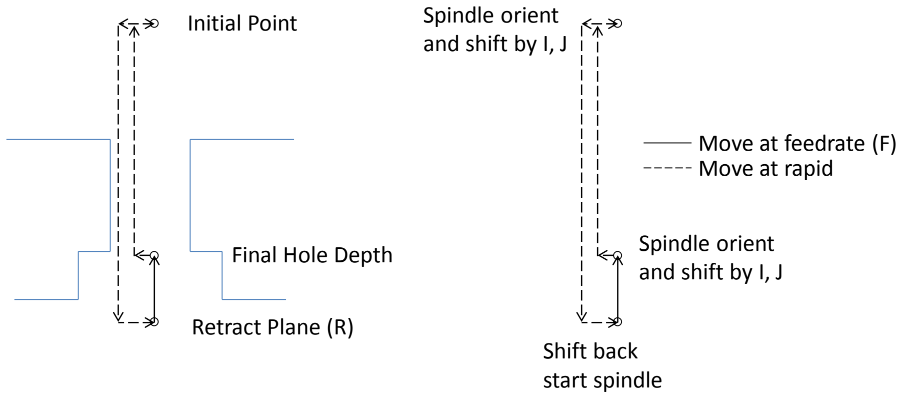


Figure 871-1: Motion of tool point for G87.1 cycle

Example:
Create the program to back bore a thru hole at the A position shown in figure 102. Thru depth 1 inch, back bore depth .150 inch.

G0 G54 G17 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 of hole
G87.1 G98.1 X0.0 Z-0.85 R-1.05 I-.10 F10.0	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 seconds
L – Number of repetitions
F – Feedrate

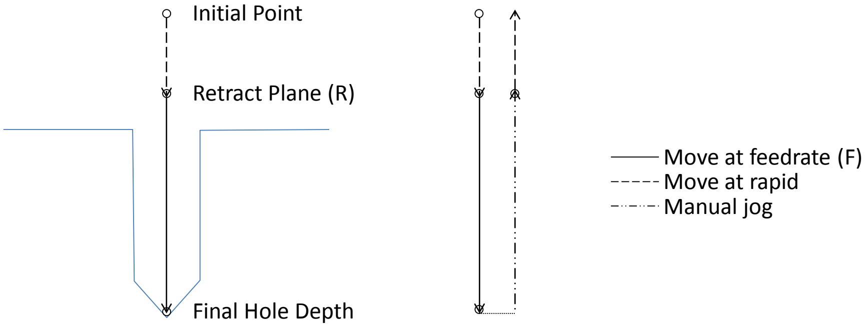


Figure 881-1: Motion of tool point for G88.1 cycle.

Example:
Create the program to bore the holes shown in figure 102.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial position
X1.70 C0.0	Move to X and C positions of first hole
G88.1 G99.1 X1.7 C0.0 Z-1.0 R.25 F0.004	Boring cycle start, pause for manual retract
C120.0	Bore second hole, pause for manual retract
C240.0	Bore third hole, 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 seconds
L – Number of repetitions
F – Feedrate

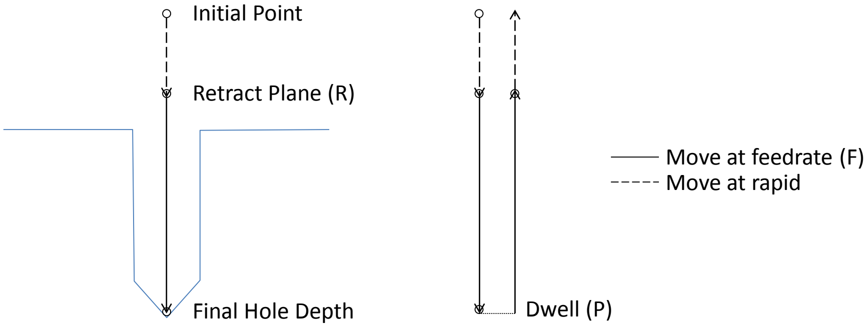


Figure 891-1: Motion of tool point for G89.1 cycle

Example:
Create the program to bore the holes shown in figure 102, dwell for 1 second.

G0 G54 G17 G40 G80	Safe start line
T0101	Tool change
S2500 M3	Start spindle
G0 Z1.0	Move to Z initial position
X1.70 C0.0	Move to X and C positions of first hole
G89.1 G99.1 X1.7 C0.0 Z-1.0 R.25 P1.0 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

Chapter 5: Cutter Compensation

Cutter compensation provides the user with the ability to adjust the tool path for variations in tool cutter diameter. It can be used in two ways. First, when programming by hand, without the aid of CAM (Computer Aided Manufacturing) software, it is much easier to program the actual part dimensions, part line programming. This saves the programmer for having to calculate to correct path at the center of the tool when the edge is doing the cutting. When given a proper diameter offset, cutter compensation will make the appropriate tool path adjustments to cut the part correctly. Essentially the machine does the math for the programmer. Second, with the more widespread use of CAM systems the tool path is already adjusted for the tool diameter and the part should, in theory, be cut perfectly to size. In practice however, there are many factors that determine the finished size of a machined part, cutter and machine deflection, machine positioning accuracy, cutter diameter variations, etc. Cutter compensation allows for fine tuning the tool path, and adjustment of part dimensions, without having to change the program itself.

There are two G codes used to enable cutter compensation. G41 offsets the tool to the left of the tool path and G42 offsets the tool to the right of the tool path. This is only true for positive diameter offset values. If negative offset is specified the offset direction will be reversed, see figure 41-1. There are two ways to call the offset value with G41 and G42.

Format 1: G00/G01 G41/G42 D__ X__ Y__ F__
Format 2: G00/G01 G41/G42 P__ X__ Y__ F__

Use D to call a diameter offset from a specific tool offset number. For example, D2 will use the diameter offset value of tool offset number 2. An alternative is to use P. The value specified with P will be the actual offset value. For example, P.05 will offset the tool path .05 to the left or right.

Cutter compensation only works on the two in plane axes, so for G17 (XY plane) the X and Y axes are affected by the comp, G18 ZX and G19 YZ.

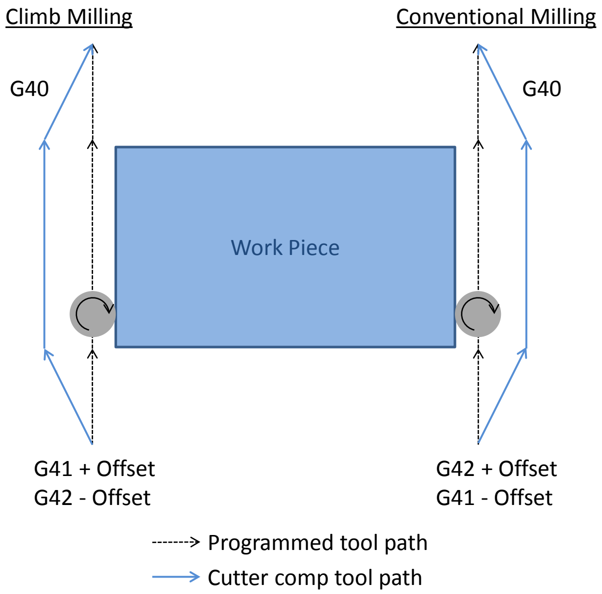


Figure 41-1: Cutter compensation offset direction.

There are two types of milling, which style is used determines how the tool should be compensated. The two types are climb milling and conventional milling, see figure 41-1. Conventional milling is the standard in manual machines, but with CNC it is possible and recommended to climb mill when possible. This manual will assume the tool will always be climb milling. With that assumption if G41 is used for outside features, the profile of a part, then a positive offset will make the part bigger and a negative offset will make the part smaller. For inside features, a hole, G42 will make the hole bigger with a positive offset and smaller with a negative offset, see figure 41-2. This method tends to make the offset values more intuitive for the operator. If conventional milling is used, this description will be reversed.

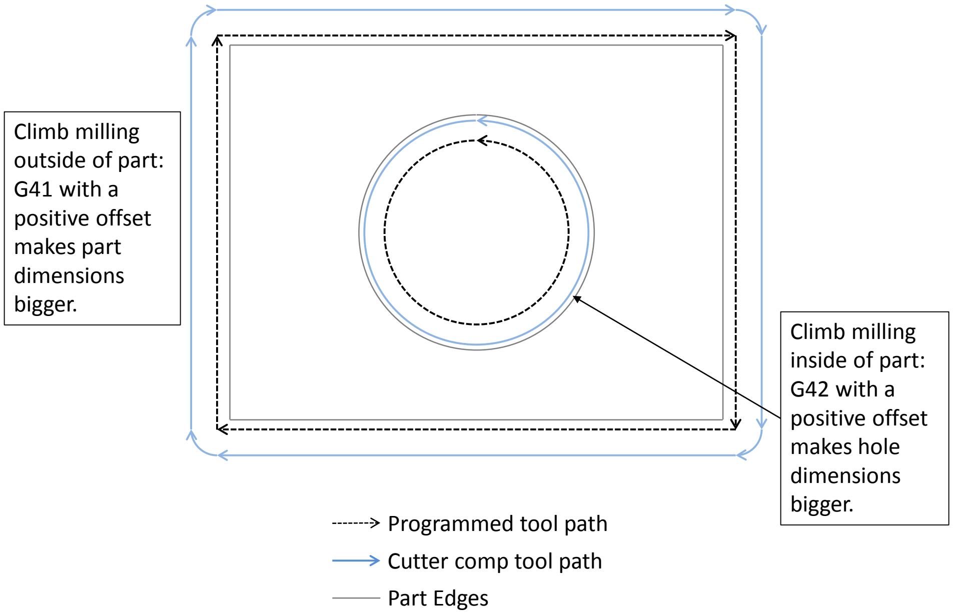

Climb milling outside of part: G41 with a positive offset makes part dimensions bigger.
Climb milling inside of part: G42 with a positive offset makes hole dimensions bigger.
Programmed tool path
Cutter comp tool path
Part Edges

Figure 41-2: Cutter compensation OD and ID features.

Cutter comp should be enabled on a linear lead-in move; an error will be produced if cutter comp is enabled in a block with an arc. If there is a value other than zero in the offset then this lead-in move may not be parallel to the programmed path, see figure 41-1, 41-6 and 41-7. The end point of the start block is the point at a 90° angle to the movement in the next block and at the offset distance. See figure 41-3 for examples. This linear move must be longer than the offset amount, if it is not there will be an error. Also, if there are any segments of the tool path shorter than the offset amount a gouge is likely, see figure 41-4. The tool path display in Mach will display the actual tool path with comp, check to make sure there are no gouges or abnormalities before running a part.

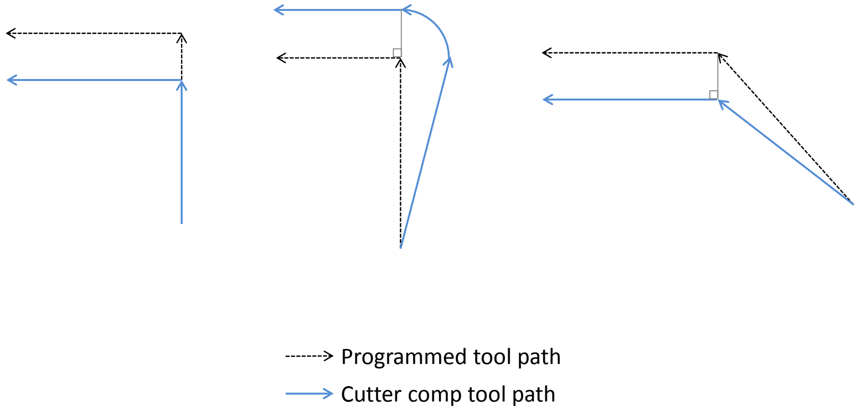
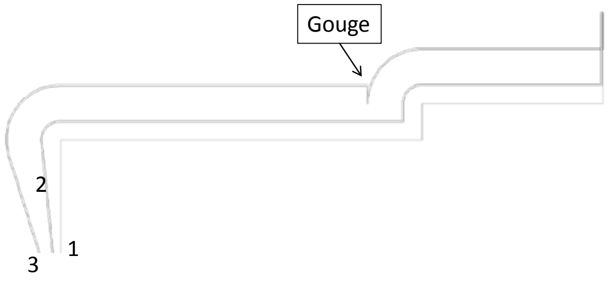


Figure 41-3: Cutter compensation start blocks.



```
Tool Path 1: No cutter comp
G00 G90 G54 G17 G40 G49 G80
G0 X0 Y0 Z1.
S1000 M3

G1 G41 P0 Y.5 F25.
X1.5
Y.6
X2.0
G40 Y1.25

M30
```

```
Tool Path 2: Comp radius .05
G00 G90 G54 G17 G40 G49 G80
G0 X0 Y0 Z1.
S1000 M3

G1 G41 P0.05 Y.5 F25.
X1.5
Y.6
X2.0
G40 Y1.25

M30
```

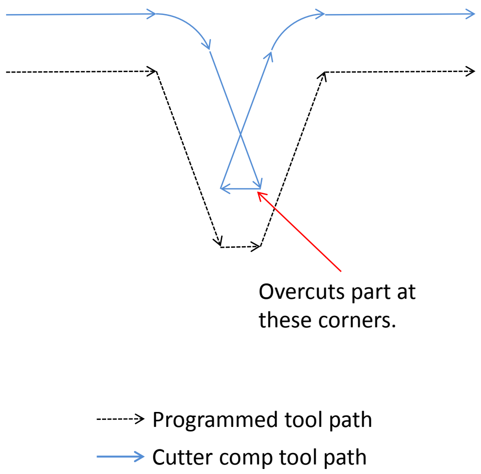
```
Tool Path 3: Comp radius .15
G00 G90 G54 G17 G40 G49 G80
G0 X0 Y0 Z1.
S1000 M3

G1 G41 P0.15 Y.5 F25.
X1.5
Y.6
X2.0
G40 Y1.25

M30
```

Figure 41-4: Cutter compensation radius and gouge.

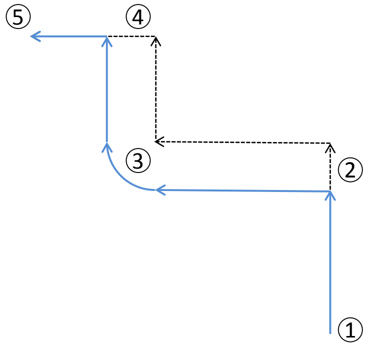
Figure 41-4 shows three tool paths generated by the same program with different cutter compensation offset values. The gouge is created when an offset value greater than the step of .10 is input. When this happens the compensated tool path over laps and is reversed, causing a gouge. Another example of this is in small grooves as shown in figure 41-5.



Overcuts part at these corners.

Figure 41-5: Cutter compensation in narrow groove.

Cutter compensation is cancelled by specifying G40 in the program, or when the control is reset. When G40 is specified it should be on a lead-out move following the same rules as when compensation is enabled. The path will be determined as shown in figure 41-3. Errors will occur if the lead-out distance is smaller than the offset amount or if G40 is specified on a block containing an arc move.

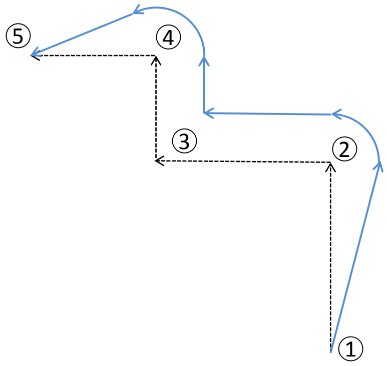


```
   G00 G90 G54 G17 G40 G49 G80
①  G0 X1.5 Y-.75 Z1.
   S1000 M3
   G1 Z-.25 F30.
②  G41 P.10 Y0
③  X.5
④  Y1
⑤  G40 X0
   G0 G53 Z0
   M30
```

--------> Programmed tool path

——> Cutter comp tool path

Figure 41-6: G41 cutter compensation path.



```
   G00 G90 G54 G17 G40 G49 G80
①  G0 X1.5 Y-.75 Z1.
   S1000 M3
   G1 Z-.25 F30.
②  G42 P.10 Y0
③  X.5
④  Y1
⑤  G40 X0
   G0 G53 Z0
   M30
```

--------> Programmed tool path

——> Cutter comp tool path

Figure 41-7: G42 cutter compensation path.

Chapter 6: M Code List

Code	Description	Page
M00	Mandatory Program Stop	67
M01	Optional Program Stop	67
M02	Program End	67
M03	Spindle Forward/Clockwise	67
M04	Spindle Reverse/Counterclockwise	67
M05	Spindle Stop	67
M07	Mist Coolant On	68
M08	Flood Coolant On	68
M09	All Coolant Off	68
M19	Spindle Orient	68
M30	Program End and Rewind	68
M48	Enable Feed/Speed Overrides	68
M49	Disable Feed/Speed Overrides	68
M98	Subprogram Call	68
M99	Return From Subprogram / Rewind	70
M???	Custom Macro M Codes	71

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.

M06 – Tool Change: Tells the machine to change to a tool number specified by T. The T word must appear in the same block as the M06 or above it in the program. For example:

G0 G90 G54 G18 G40 G80	Safe start line
G0 X4.0 Y0.0 Z1.0 T1	Rapid move, pre call tool 1
M6	Tool change to tool 1
G0 X10.0 Y-5.0 Z1.0	Rapid move
T2 M6	Tool change to tool 2
M30	Program end and rewind

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.

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.

M48 – Enable Feedrate/Spindle Speed Overrides: Enables the feedrate and spindle speed overrides after an M49 has been issued.

M49 – Disable Feedrate/Spindle Speed Overrides: Disables the feedrate and spindle speed overrides. This can be beneficial in production environments and also in programs or macros that are sensitive to feedrate and spindle speed by preventing accidental overrides.

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:

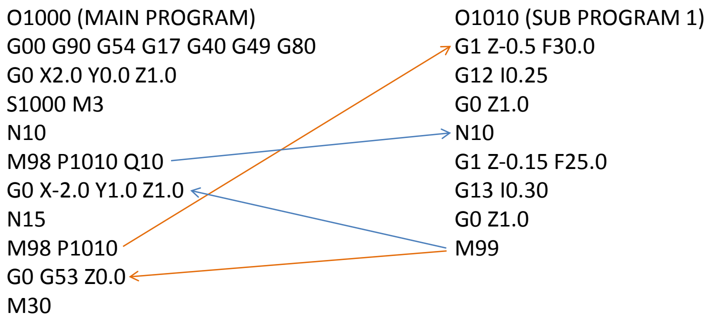
(MAIN PROGRAM)	Main program header
G0 G90 G54 G18 G40 G80	Safe start line
G0 X4.0 Y0.0 Z1.0	Rapid move
T1 M6	Tool change to tool 1
G0 X10.0 Y-5.0 Z1.0	Rapid move
M98 P1000	Call subprogram number 1000
M30	Program end and rewind
O1000 (SUB PROGRAM BEGIN)	Subprogram number
G12 I-0.25	Cut circle
M99	Return to main program

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.



```
O1000 (MAIN PROGRAM)
G00 G90 G54 G17 G40 G49 G80
G0 X2.0 Y0.0 Z1.0
S1000 M3
N10
M98 P1010 Q10
G0 X-2.0 Y1.0 Z1.0
N15
M98 P1010
G0 G53 Z0.0
M30
```

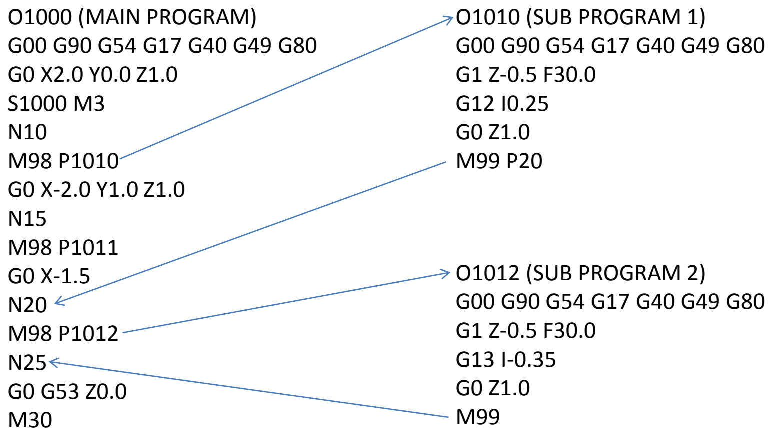
```
O1010 (SUB PROGRAM 1)
G1 Z-0.5 F30.0
G12 I0.25
G0 Z1.0
N10
G1 Z-0.15 F25.0
G13 I0.30
G0 Z1.0
M99
```

Figure 198-1: Subprogram call

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.



```
O1000 (MAIN PROGRAM)
G00 G90 G54 G17 G40 G49 G80
G0 X2.0 Y0.0 Z1.0
S1000 M3
N10
M98 P1010
G0 X-2.0 Y1.0 Z1.0
N15
M98 P1011
G0 X-1.5
N20
M98 P1012
N25
G0 G53 Z0.0
M30
```

```
O1010 (SUB PROGRAM 1)
G00 G90 G54 G17 G40 G49 G80
G1 Z-0.5 F30.0
G12 I0.25
G0 Z1.0
M99 P20
```

```
O1012 (SUB PROGRAM 2)
G00 G90 G54 G17 G40 G49 G80
G1 Z-0.5 F30.0
G13 I-0.35
G0 Z1.0
M99
```

Figure 199-1: Return from sub program

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.