COMPENSATION

G40, G41, G42 CUTTER COMPENSATION

To turn cutter radius compensation off, program G40 .
It is OK to turn compensation off when it is already off.
Cutter radius compensation may be performed only if the XY - plane is active.

To turn cutter radius compensation on left (i.e., the cutter stays to the left of the programmed path when the tool radius is positive), program G41 D~

To turn cutter radius compensation on right (i.e., the cutter stays to the right of the programmed path when the tool radius is positive), program G42 D
The D word is optional; if there is no D word, the radius of the tool currently in the spindle will be used. If used, the D number should normally be the slot number of the tool in the spindle, although this is not required. It is OK for the D number to be zero; a radius value of zero will be used.

G41 and G42 can be qualified by a P - word.
This will override the value of the diameter of the tool (if any) given in the current tool table entry.

It is an error if:
- the D number is not an integer, is negative or is larger than the number of carousel slots,

- the XY - plane is not active,

The behavior of the machining system when cutter radius compensation is ON is described in the chapter of Cutter Compensation. Notice the importance of programming valid entry and exit moves.

By Graham

I have been asked to try and clarify the cutter compensation debate (CC) that keeps raging on and on and on.........

The basics

Compensation is not a quick fix.

Cutter compensation is used to adjust the size of a cut area, this can be on the outside or the inside of a component or section of a component.

Compensation has to be planned into the job, it is not easy to adapt afterwards.

When programming the job the tool is programmed on the centre line of the cut.

When programming the part any inside radius can not be smaller than the radius of the cutter to be used.

The machine should never rapid with an active G41 or G42, Mach3 may allow this move just as you can drive your car off a cliff, its not advisable.

You should not change local (G52) or global (G54-G59 Etc.) work offsets while a G41 or G42 is active.

Cutter compensation should not be used for pocketing and area clearance.

How it works

To use CC we have to command a G41 or a G42, the one we use depends on which side of the line we are on and what direction we want to travel. The next consideration is how we move onto the cutter path, we can not rapid straight onto the line and start cutting, we have to create a lead in line. We also have to have lead out lines. This is why you have to pre-plan CC.

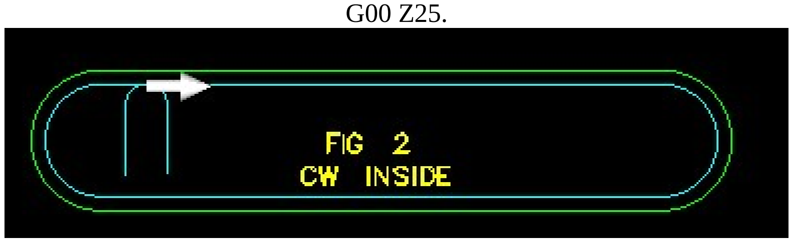
A lead line can be a straight line or an arc or a combination of both. Straight line moves are the easiest, arcs give the best blend.

The picture below shows 4 examples of how we can use CC, the top 2 show cutting on the outside of the green line. The bottom two are slots so are cutting on the inside of the green line.

The parts are 10mm wide and 50mm long giving 40 mm centres on the rads. The example code was written for the use of a 2mm dia end mill. Before anybody tells me this is not the best cutter for the job, I don't care.

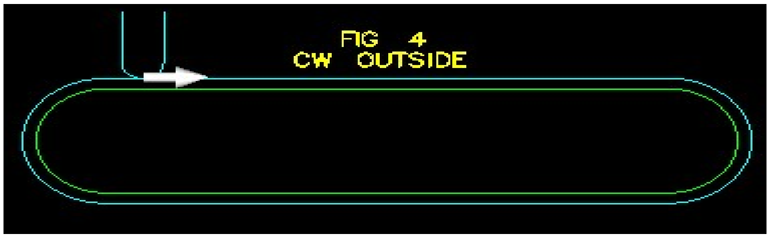
Graham.
%
G54 G00 G90 G43
T1 M6
S1500 M3

(FIG 1 - CCW INSIDE)
G00 X41.5 Y-2.5
G00 Z1.
G01 Z-2. F900.
G41 X42.5 Y2.5 F1800.
G03 X40. Y5. R2.5
G01 X0.
G03 X-5. Y0. R5.
X0. Y-5. R5.
G01 X40.
G03 X45. Y0. R5.
X40. Y5. R5.
X37.5 Y2.5 R2.5
G01 G40 X38.5 Y-2.5
G00 Z25.



(FIG 2 - CW INSIDE)
G00 X61.713 Y-2.5
Z1.
G01 Z-2. F900.
G42 X60.713 Y2.5 F1800.
G02 X63.213 Y5. R2.5
G01 X100.
G02 X105. Y0. R5.
X100. Y-5. R5.
G01 X60.
G02 X55. Y0. R5.
X60. Y5. R5.
G01 X63.213
G02 X65.713 Y2.5 R2.5
G01 G40 X64.713 Y-2.5
G00 Z25.


```
(FIG 3 - CCW OUTSIDE)
G00 X41.5 Y42.5
Z1.
G01 Z-2. F900.
G42 X42.5 Y37.5 F1800.
G02 X40. Y35. R2.5
G01 X0.
G03 X-5. Y30. R5.
X0. Y25. R5.
G01 X40.
G03 X45. Y30. R5.
X40. Y35. R5.
G02 X37.5 Y37.5 R2.5
G01 G40 X38.5 Y42.5
G00 Z25.
```




```
(FIG 4 - CW OUTSIDE)
G00 X61.074 Y42.5
Z1.
G01 Z-2. F900.
G41 X60.074 Y37.5 F1800.
G03 X62.574 Y35. R2.5
G01 X100.
G02 X105. Y30. R5.
X100. Y25. R5.
G01 X60.
G02 X55. Y30. R5.
X60. Y35. R5.
G01 X62.574
G03 X65.074 Y37.5 R2.5
G01 G40 X64.074 Y42.5
G00 Z25.
G91 G28 Y0 Z0
M30
%
```

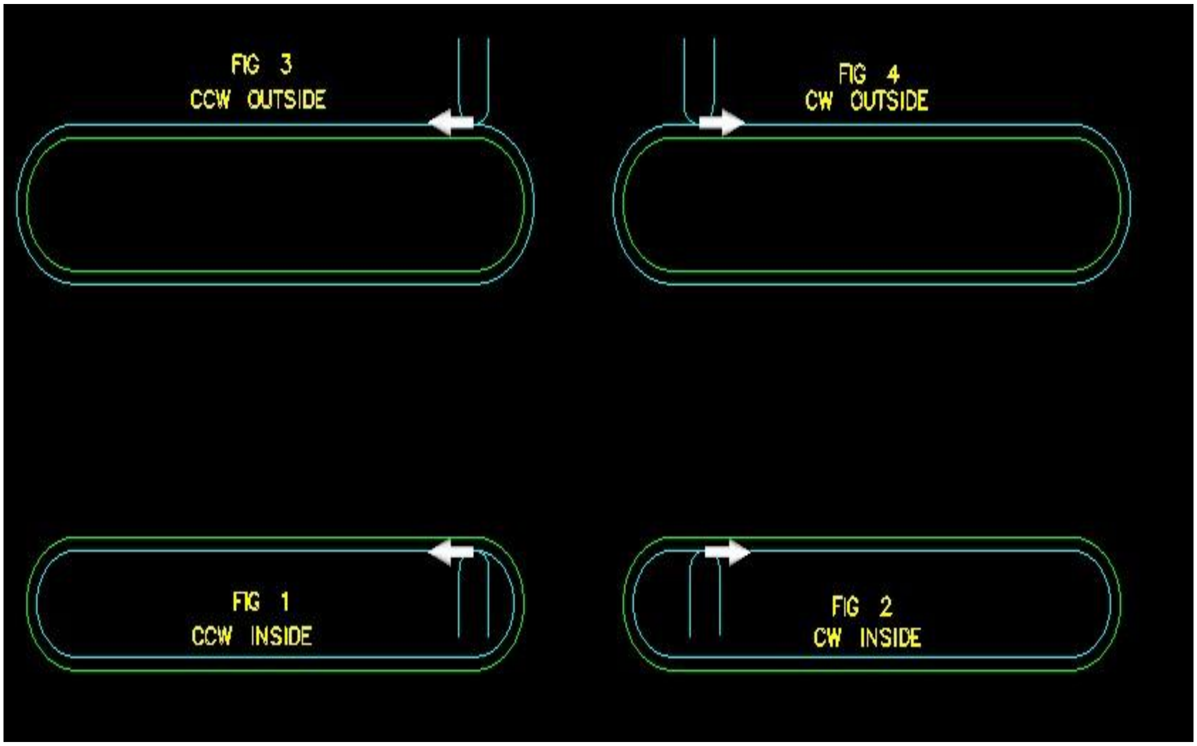


FIG 3
CCW OUTSIDE
FIG 4
CW OUTSIDE
FIG 1
CCW INSIDE
FIG 2
CW INSIDE

GER 21 WROTE

You don't need to set up the tool table to use comp, but you can if you like. There are two ways to use it. (Actually 3)

1) Call comp and specify the tool radius using the P word. For a 1/8" diameter bit, you'd use G41 P0.0625 or G42 P0.0625.

2) Call comp and specify the tool # in the tool table using the D word. This will use the radius of the tool in the table. To use comp with tool #3, use G41 D3 or G42 D3.

3) Basically the same as #2, but don't call any tool at all. Mach3 will use the diameter of the current tool (based on the diameter in the tool table). Just use G41 or G42.

To use comp properly, you need a lead in move where the comp is applied. Without comp, the center of the tool follows the g-code. When you call G41/G42, comp is applied during the move from the previous location to the one following the G41/G42. I like to use G41/G42 on it's own line, but you can combine it with the leadin move. Both of these will do the same thing:

```
G1 X0 Y0
G42 D1
G1 X1 Y1
G1 X2 Y1
```

is the same as

```
G1 X0 Y0
G1 G42 D1 X1 Y1
G1 X2 Y1
```

Once comp is applied, it remains in effect until you call G40. Depending on the move following the G40, comp can be either gradually or abruptly removed. You can use a leadout move, or you can raise the tool above the work and then remove the comp.

Left(G41) or right (G42) is pretty easy to figure out. Imagine walking along the toolpath in the cut direction. G42 will move the tool to the right side of the path, G41 will move it to the left side of the path. If you have a circular CCW toolpath, G42 will cut outside the toolpath, G41 will cut inside the toolpath. One thing that can be tricky is climb vs conventional cutting. Which method you're using will dictate whether you need G41/G42, as it determines the direction of the toolpath. The circular example above is using conventional cutting

Here's a little sample program that will show you how it works.

Here's a little sample program that will show you how it works.

```
G40
M3
G0 Z0.1250
G0 X1.4629 Y1.2013 Z0.1250
G1 X1.4629 Y1.2013 Z0.0000 F25
G42P0.25
G1 X2.6956 Y2.0815 Z-0.2500 F50
G1 X8.2664 Y2.0815 Z-0.2500
G1 X8.2664 Y6.2744 Z-0.2500
G1 X3.4479 Y6.2744 Z-0.2500
G1 X3.4479 Y1.5214 Z-0.2500
G40
G1 X2.8556 Y0.9132 Z-0.2500
G0 X2.8556 Y0.9132 Z0.1250
G0 X0 Y0
M5
M30
```

So I take it tool diameter compensation has to be done before Mach3?

The G-code needs the G41/G42/G40 in it, and must be coded correctly to allow for the lead in. But if you use G42 Dx, then all you'll need to do is change the tool diameter in the tool table to make adjustments for tool size.

Again, I explained it in my previous post, and offered an example for cutting around the outside. Here's one for the inside.

```
G40
M3
G0 Z0.1250
G0 X0.6391 Y0.7309 Z0.1250
G1 X0.6391 Y0.7309 Z-0.2500 F50
G42P0.25
G1 X0.0000 Y1.5000 Z-0.2500 F150
G1 X0.0000 Y3.0000 Z-0.2500
G1 X3.0000 Y3.0000 Z-0.2500
G1 X3.0000 Y0.0000 Z-0.2500
G1 X0.0000 Y0.0000 Z-0.2500
G1 X0.0000 Y1.9312 Z-0.2500
G40
G1 X0.5000 Y1.9312 Z-0.2500
```

```
G0 X0.5000 Y1.9312 Z0.1250
G0 X0 Y0
M5
M30
```

As Gerry has said one is for going down the left of the line (tool path) as you look in the direction of travel (g41) the other is for going down the right of the line as you look in the direction of travel (g42)
You can can also lie to it to alter the offset in relation to the size of the cutter to get a larger or smaller part (or hole)
G41 also climb cuts while G42 is conventional cutting.
So if your machine has a significant amount of backlash then always use G42 and go Anti clockwise to cut on the outside and clockwise to cut on the inside.

If I load Mach from scratch, and load this program, tool comp is NOT applied.

```
g90g20g61t1f1000
g01x-0.5y-0.5
g01g42x0.0y0.0
x2.0y0.0
x2.0y2.0
x0.0y2.0
x0.0y0.0
x2.0y0.0
g01g40x2.5y-0.5
m30
```

If I insert a D value, toll comp IS applied. For instance if I change the first line to....

```
g90g20g61t1d1f1000
or
g90g20g61d1f1000
```
(whichever one you prefer, pick your poison)

now....stay with me here...if you edit the code AGAIN, back to the original way it was, (without the D value, and only a T call) tool comp magically works, and I can modify the value in the tool table, and MACH will indeed account for those changes.

So, in short..... If you use code without a D value, a tool call alone will not work. If you ADD a D value, and then remove it, a tool call alone WILL work. Strange indeed.

Just reading through this post on cutter comp, Graham says g41 or g42 on a lead in (g1 or g2-g3)-(Correct me if I'm wrong,) but I was always taught
Not to use cc on a lead in arc. Always g41 or g42 on a linar (g1) move first, than the arc. example-g41 x0y0 than g2 or g3 for your blend. Then when you arc out for blend-g40 with no g1. I always make my lead in g1 no less than my cutter di.
Thats the way I always do it.
Ed V,

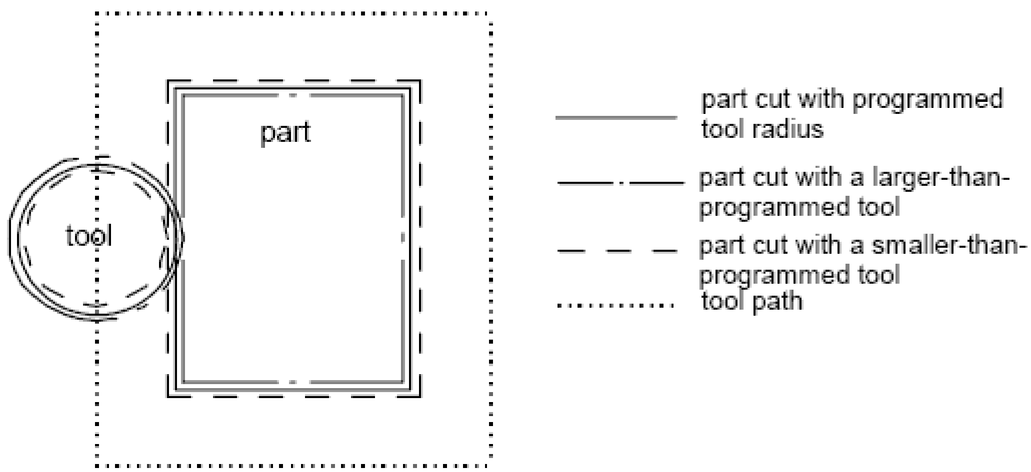
Hi Ed,

your comments are true, but it also depends on the control some need 90 degree only approach others can work out vectoring some will apply the cc during the rapid to position. If you read through the G-code examples you will find that the samples are code as you were taught. CC can also be applied on a single arc lead in.

Graham.

Understanding Cutter Compensation

Cutter compensation is one of the most useful things to know when CNC machining. Cutter compensation allows you to program the geometry and not worry about the toolpath. It also allows you to adjust the size of your part, based on the tool radius used to cut your part. This is useful when you can't find a cutter of the proper diameter. This is best explained in the graphic below.

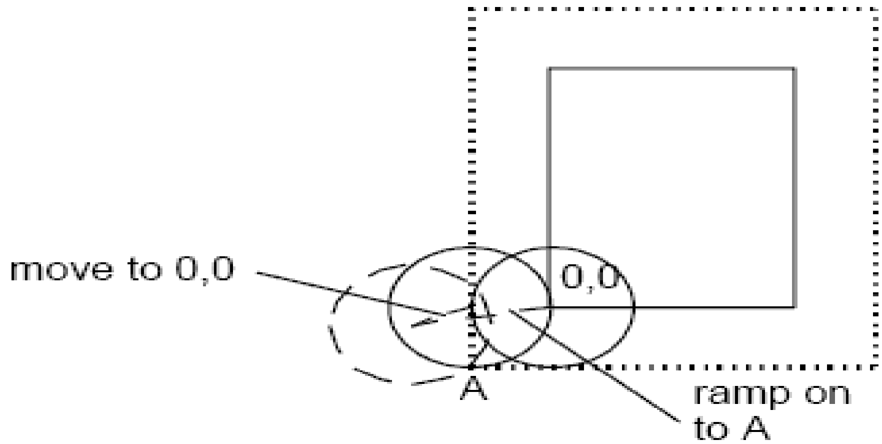


The solid circle is the nominal sized tool. The dashed circle is an undersized tool, and the dash-dot circle is the oversized tool. With a little imagination, you can see all the possibilities for tweaking your part, or getting your part made with any size endmill.

Turning Cutter Compensation On and Off

It is important to note that cutter compensation becomes active after the next line move or rapid that is at least the length of the tool radius. Failure to account for this will give a funny part. A good method around this is to zero your part and program a move away from the part in the X and Y direction equal to the tool radius. Then move back to 0,0, and then continue cutting your profile. See the graphic below. Note the tool center is now perpendicular

And to the left of point A.



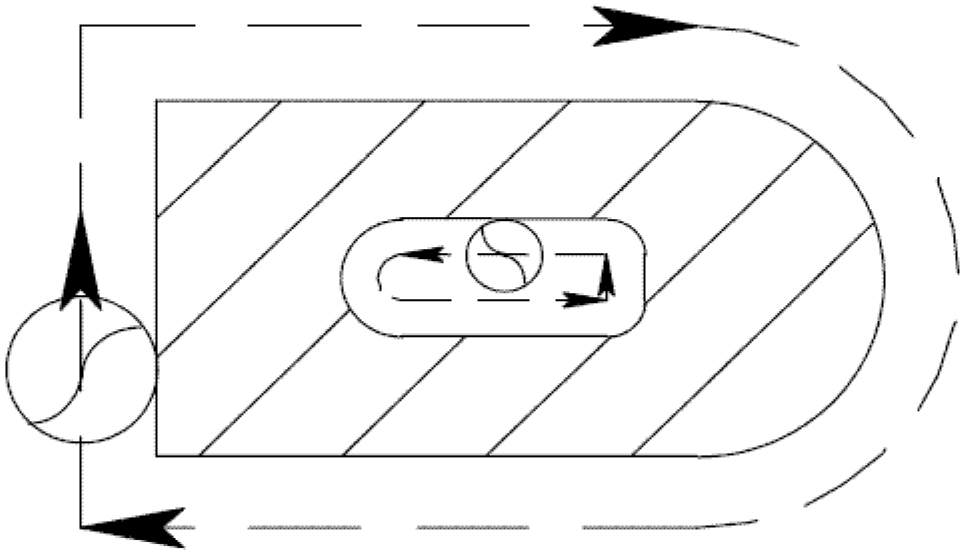
.

To turn cutter compensation off, you must do a ramp off move similar to the ramp on move. Again, send the tool off in the X and Y direction a distance equal to the tool radius. For the graphic above, after reaching 0,0 turn off cutter compensation and ramp off to A. Depending on the shape, you may have to go beyond 0,0 to eliminate any ’nurkies’(a nurkie is an unintentional over or under cut left by the tool). This terminates cutter compensation, and you can go on to something else.

There are three G-Codes involved in using cutter comp : G41 initiates cutter comp to the left of the path; G42 initiates cutter comp to the right of the path; and G40 cancels cutter compensation.

Climb & Conventional Cutting with Cutter Compensation

This refers to the direction of cut relative to the direction the spindle is turning. In CNC milling it is common practice to climb cut. With the spindle turning clockwise, as it usually does, cut a contour in a clockwise direction. If cutting a pocket or hole cut counter clockwise. Your parts will come out okay if you go the wrong way, but surface finish and accuracy is usually better when you climb cut.



This part is shown with two tools climb cutting
The big tool is set cutter comp left
The small tool is also set cutter comp left

Summary

Cutter Compensation:

- allows you to program the geometry not the tool path
- is useful when you don't have the right endmill
- is helpful in tweaking your part size
- allows you to compensate for tool wear
- is generally a neat and powerful thing to know about

NIST Appendix B. Cutter Radius Compensation

This appendix discusses cutter radius compensation. It is intended for NC programmers and machine operators. Researchers and developers may find it useful. SAI system installers will probably not find it useful.

See Section 3.5.10 for additional information on cutter radius compensation.

B.1 Introduction

The cutter radius compensation[1] capabilities of the Interpreter enable the programmer to specify that a cutter should travel to the right or left of an open or closed contour in the XY-plane composed of arcs of circles and straight line segments.

Cutter radius compensation is performed only with the XY-plane active. All the figures in this appendix, therefore, show projections on the XY-plane.

Where the adjacent sides of remaining material meet at a corner, there are two common ways to handle the tool path. The tool may pass in an arc around the corner, or the tool path may continue straight in the direction it was going along the first side until it reaches a point where it changes direction to go straight along the second side. Figure 2 shows these two types of path. On Figure 2:

- Uncut material is shaded in the figures. Note that the inner triangles have the same shape with both tool paths.
- The white areas are the areas cleared by the tool.
- The lines in the center of the white areas represent the path of the tip of a cutting tool.
- The tool is the cross-hatched circles.

Both paths will clear away material near the shaded triangle and leave the shaded triangle uncut. When the Interpreter performs cutter radius compensation, the tool path is rounded at the corners, as shown on the left in Figure 2. In the method on the right (the one not used), the tool does not stay in contact with the shaded triangle at sharp corners, and more material than necessary is removed.

There are also two alternatives for the path that is programmed in NC code during cutter radius compensation. The programmed path may be either (1) the edge of the material to remain uncut (for example, the edge of the inner triangle on the left of Figure 2), or (2) the nominal tool path (for example, the tool path on the left side of Figure 2). The nominal tool path is the path that would be used if the tool were exactly the intended size. The Interpreter will handle both cases without being told which one it is. The two cases are very similar, but different enough that they are described in separate sections of this manual. To use the material edge method, read Appendix B.3. To use the nominal path method, read Appendix B.4.

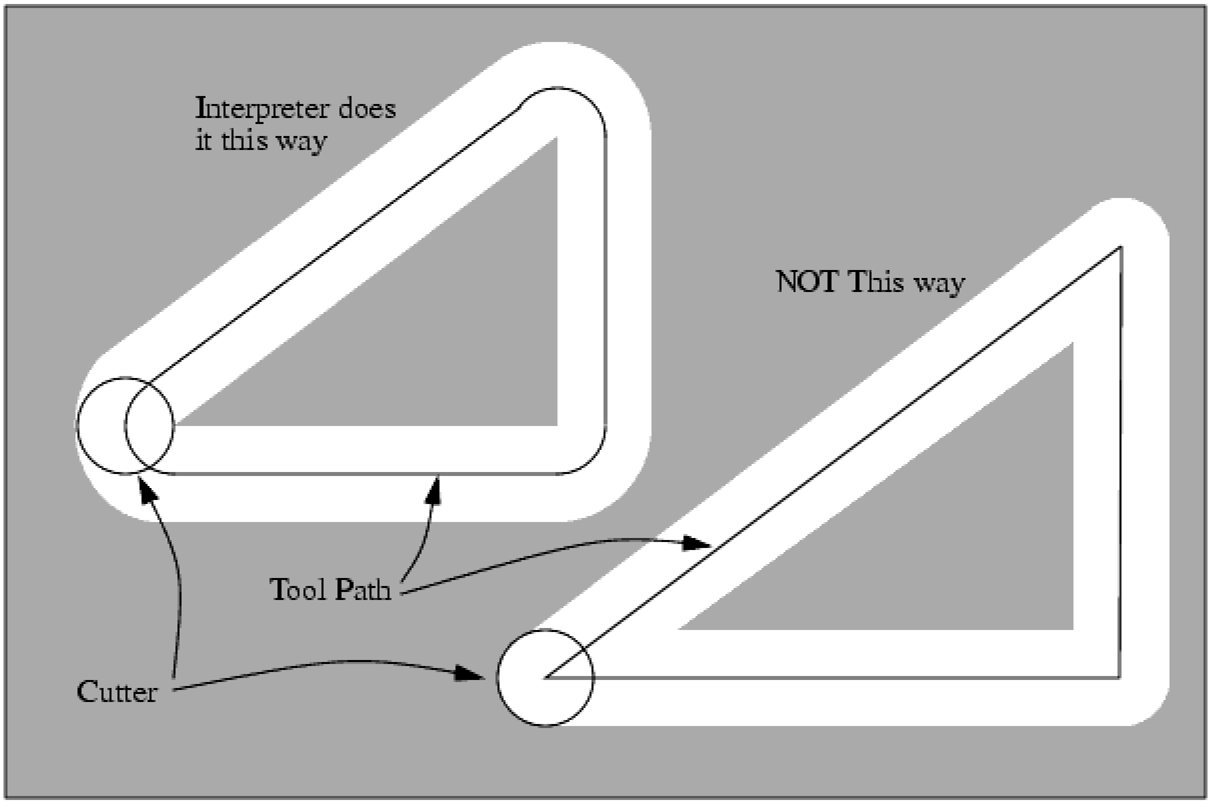


Figure 2. Two Cutter Radius Compensation Methods

Z-axis motion may take place while the contour is being followed in the XY-plane. Portions of the contour may be skipped by retracting the Z-axis above the part, following the contour to the next point at which machining should be done, and re-extending the Z-axis. These skip motions may be performed at feed rate (G1) or at traverse rate (G0). The Z motion will not interfere with the XY path following. The sample NC code in this appendix does not include moving the Z-axis. In actual programs, include Z-axis motion wherever you want it.

Rotational axis motions (A, B, and C axes) are allowed with cutter radius compensation, but using them would be very unusual.

Inverse time feed rate (G93) or units per minute feed rate (G94) may be used with cutter radius compensation. Under G94, the feed rate will apply to the actual path of the cutter tip, not to the programmed contour.

B.1.1 Data for Cutter Radius Compensation

The Interpreter world model keeps three data items for cutter radius compensation: the setting itself (right, left, or off), program_x, and program_y. The last two represent the X and Y positions which are given in the NC code while compensation is on. When compensation is off, these both are set to a very small number (10^{-20}) whose symbolic value is "unknown". The Interpreter world model uses the data items current_x and current_y to represent the position of the center of the tool tip (in the currently active coordinate system) at all times.

B.2 Programming Instructions

B.2.1 Turning Cutter Radius Compensation On

To start cutter radius compensation keeping the tool to the left of the contour, program G41 D- . The D word is optional (see "Use of D Number", just below).

To start cutter radius compensation keeping the tool to the right of the contour, program G42 D-

In Figure 2, for example, if G41 were programmed, the tool would move clockwise around the triangle, so that the tool is always to the left of the triangle when facing in the direction of travel. If G42 were programmed, the tool would stay right of the triangle and move counterclockwise around the triangle.

B.2.2 Turning Cutter Radius Compensation Off

To stop cutter radius compensation, program G40. It is OK to turn compensation off when it is already off.

B.2.3 Sequencing

If G40, G41, or G42 is programmed on the same line as tool motion, cutter compensation will be turned on or off before the motion is made. To make the motion come first, the motion must be programmed on a separate, previous line of code.

B.2.4 Use of D Number

Programming a D word with G41 or G42, is optional.

If a D number is programmed, it must be a non-negative integer. It represents the slot number of the tool whose radius (half the diameter given in the tool table) will be used, or it may be zero (which is not a slot number). If it is zero, the value of the radius will also be zero. Any slot in the tool table may be selected. The D number does not have to be the same as the slot number of the tool in the spindle, although it is rarely useful for it not to be.

If a D number is not programmed, the slot number of the tool in the spindle will be used as the D number.

B.3 Material Edge Contour

When the contour is the edge of the material, the outline of the edge is described in the NC program.

For a material edge contour, the value for the diameter in the tool table is the actual value of the diameter of the tool. The value in the table must be positive. The NC code for a material edge contour is the same regardless of the (actual or intended) diameter of the tool.

B.3.1 Programming Entry Moves

In general, two pre-entry moves and one entry move are needed to begin compensation correctly. However, if there is a convex corner on the contour, a simpler method is available using zero or one pre-entry move and one entry move. The general method, which will work in all situations, is described first. We assume here that the programmer knows what the contour is already and has the job of adding entry moves.

B.3.1.1 General Method

The general method includes programming two pre-entry moves and one entry move. See Figure 3. The shaded area is the remaining material. It has no corners, so the simple method cannot be used. The dotted line is the programmed path. The solid line is the actual path of the tool tip. Both paths go clockwise around the remaining material. A cutter one unit in diameter is shown part way around the path. The black dots mark points at the beginning or end of programmed or actual moves. The figure shows the second pre-entry move but not the first, since the beginning point of the first pre-entry move could be anywhere.

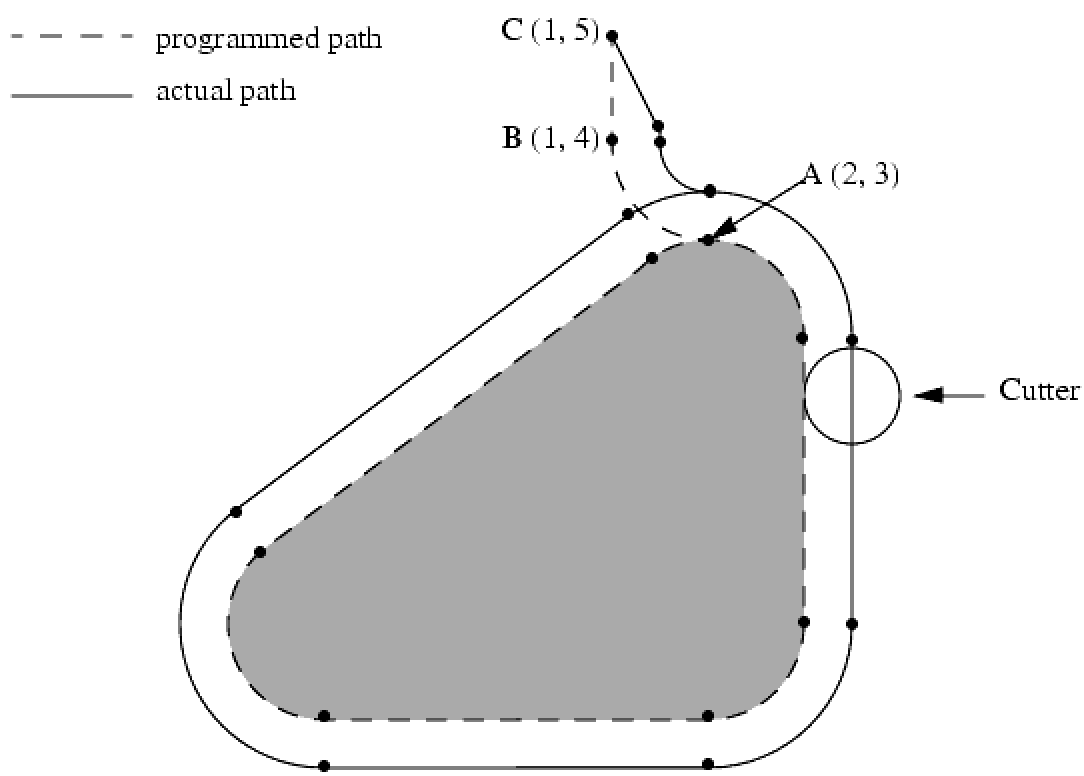


Figure 3. Cutter Radius Compensation Entry Moves

for material edge contour

First, pick a point A on the contour where it is convenient to attach an entry arc. Specify an arc outside the contour which begins at a point B and ends at A tangent to the contour (and going in the same direction as it is planned to go around the contour). The radius of the arc should be larger than half the diameter given in the tool table. Then extend a line tangent to the arc from B to some point C, located so that the line BC is more than one tool radius long. After the construction is finished, the code is written in the reverse order from the construction. The NC code is shown in Table 12; the first three lines are the entry moves just described.

- N0010 G1 X1 Y5 (make first pre-entry move to C)
- N0020 G41 G1 Y4 (turn compensation on and make second pre-entry move to point B)
- N0030 G3 X2 Y3 I1 (make entry move to point A)
- N0040 G2 X3 Y2 J-1 (cut along arc at top)
- N0050 G1 Y-1 (cut along right side)
- N0060 G2 X2 Y-2 I-1 (cut along arc at bottom right)
- N0070 G1 X-2 (cut along bottom side)

• N0080 G2 X-2.6 Y-0.2 J1 (cut along arc at bottom left) • N0090 G1 X1.4 Y2.8 (cut along third side) • N0100 G2 X2 Y3 I0.6 J-0.8 (cut along arc at top of tool path) • N0110 G40 (turn compensation off)
Table 12. NC Program for Figure 3

Cutter radius compensation is turned on after the first pre-entry move and before the second pre-entry move (including G41 on the same line as the second pre-entry move turns compensation on before the move is made). In the code above, line N0010 is the first pre-entry move, line N0020 turns compensation on and makes the second pre-entry move, and line N0030 makes the entry move.

B.3.1.2 Simple Method

If there is a convex (sticking out, not in) corner somewhere on the contour, a simpler method of making an entry is available. See Figure 4.

First, pick a convex corner. There is only one corner in Figure 4. It is at A, and it is convex. Decide which way you want to go along the contour from A. In our example we are keeping the tool to the left of the remaining material and going clockwise. Extend the side to be cut (DA in the figure) to divide the area outside the material near A into two regions; DA extended is the dotted line AC on the figure. Make a pre-entry move to anywhere in the region on the same side of DC as the remaining material (point B on the figure) and not so close to the remaining material that the tool is cutting into it. Anywhere in the diagonally shaded area of the figure (or above or to the left of that area) is OK. If the tool is already in region, no pre-entry move is needed. Write a line of NC code to move to B, if necessary. Then write a line of NC code for a straight entry move that turns compensation on and goes to point A. If B is at (1.5, 4), the two lines of code for the pre-entry and entry moves would be:

- N0010 G1 X1.5 Y4 (move to B)
- N0020 G41 G1 X3 Y3 (turn compensation on and make entry move to A)

These two lines would be followed by four lines identical to lines N0050 to N0080 from Table 12, but the end of the program would be different since the shape of remaining material is different.

It would be OK for B to be on line AC. In fact, B could be placed on the extension outside the part of any straight side of the part. B could be placed on EF extended to the right (but not to the left, for going clockwise), for example.

If DA were an arc, not a straight line, the two lines of code above would still be suitable. In this case, the dotted line extending DA should be tangent to DA at A.

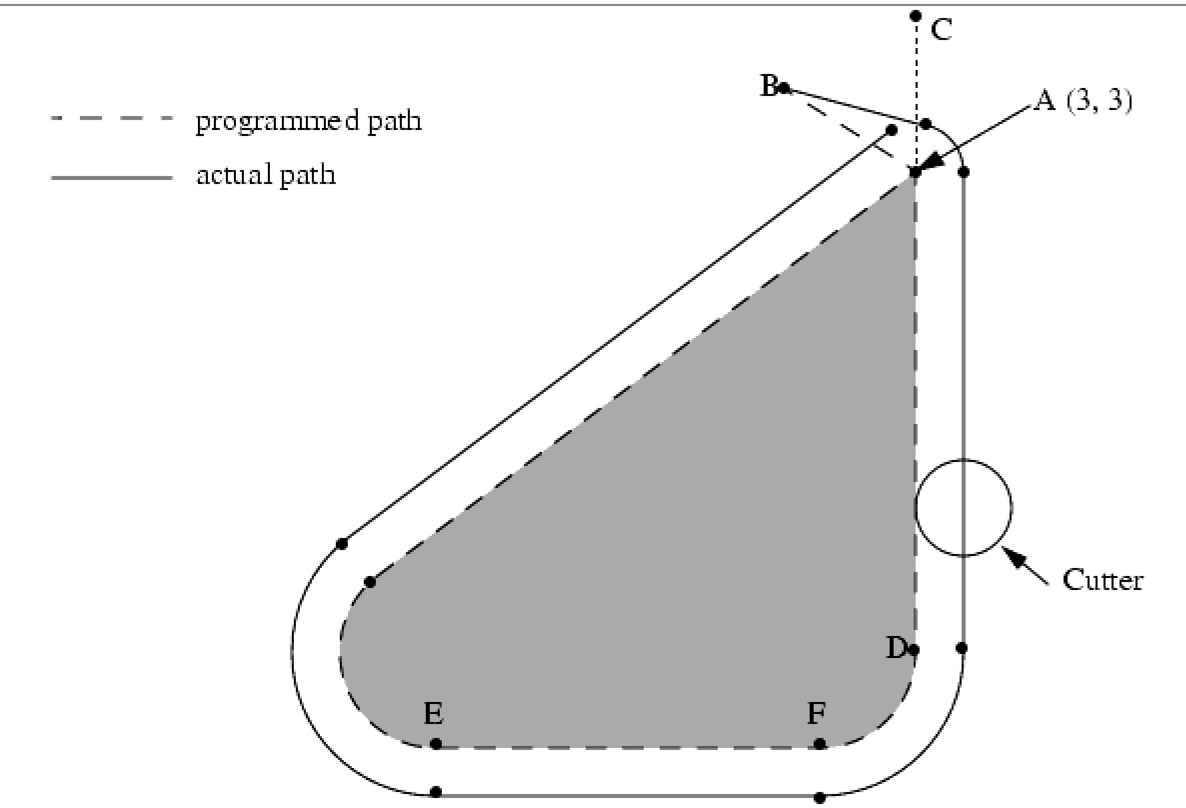


Figure 4. Simpler Cutter Radius Compensation Entry Move

for material edge contour

B.4 Nominal Path Contour

When the contour is a nominal path contour (the path a tool with exactly the intended diameter would take), the tool path is described in the NC program. It is expected that (except for during the entry moves) the path is intended to create some part geometry. The path may be generated manually or by a post-processor, considering the part geometry which is intended to be made. For the Interpreter to work, the tool path must be such that the tool stays in contact with the edge of the part geometry, as shown on the left side of Figure 2. If a path of the sort shown on the right of Figure 2 is used, in which the tool does not stay in contact with the part geometry all the time, the Interpreter will not be able to compensate properly when undersized tools are used. A nominal path contour has no corners, so the simple method just described will not work.

For a nominal path contour, the value for the cutter diameter in the tool table will be a small positive number if the selected tool is slightly oversized and will be a small negative number if the tool is slightly undersized. If a cutter diameter value is negative, the Interpreter compensates on the other side of the contour from the one programmed and uses the absolute value of the given diameter. If the actual tool is the correct size, the value in the table should be zero. Suppose, for example, the diameter of the cutter currently in the spindle is 0.97, and the diameter assumed in generating the tool path was 1.0. Then the value in the tool table for the diameter for this tool should be -0.03.

The nominal tool path needs to be programmed so that it will work with the largest and smallest tools expected to be actually used. We will call the difference between the radius of the largest expected tool and the intended radius of the tool the "maximum radius difference." This is usually a small number.

The method includes programming two pre-entry moves and one entry moves. See Figure 5. The shaded area is the remaining material. The dashed line is the programmed tool path. The solid line is the actual path of the tool tip. Both paths go clockwise around the remaining material. The actual path is to the right of the programmed path even though G41 was programmed, because the diameter value is negative. On the figure, the distance between the two paths is larger than would normally be expected. The 1-inch diameter tool is shown part way around the path. The black dots mark points at the beginning or end of programmed moves. The corresponding points on the actual path have not been marked. The actual path will have a very small additional arc near point B unless the tool diameter is exactly the size intended. The figure shows the second pre-entry move but not the first, since the beginning point of the first pre-entry move could be anywhere.

First, pick a point A on the contour where it is convenient to attach an entry arc. Specify an arc outside the contour which begins at a point B and ends at A tangent to the contour (and going in the same direction as it is planned to go around the contour). The radius of the arc should be larger than the maximum radius difference. Then extend a line tangent to the arc from B to some point C, located so that the length of line BC is more than the maximum radius difference. After the construction is finished, the code is written in the reverse order from the construction. The NC code is shown in Table 13; the first three lines are the entry moves just described.

- N0010 G1 X1.5 Y5 (make first pre-entry move to C)
- N0020 G41 G1 Y4 (turn compensation on and make second pre-entry move to point B)
- N0030 G3 X2 Y3.5 I0.5 (make entry move to point A)
- N0040 G2 X3.5 Y2 J-1.5 (cut along arc at top)
- N0050 G1 Y-1 (cut along right side)
- N0060 G2 X2 Y-2.5 I-1.5 (cut along arc at bottom right)
- N0070 G1 X-2 (cut along bottom side)
- N0080 G2 X-2.9 Y0.2 J1.5 (cut along arc at bottom left)

- N0090 G1 X1.1 Y3.2 (cut along third side)
- N0100 G2 X2 Y3.5 I0.9 J-1.2 (cut along arc at top of tool path)
- N0110 G40 (turn compensation off)

Table 13. NC Program for Figure 5

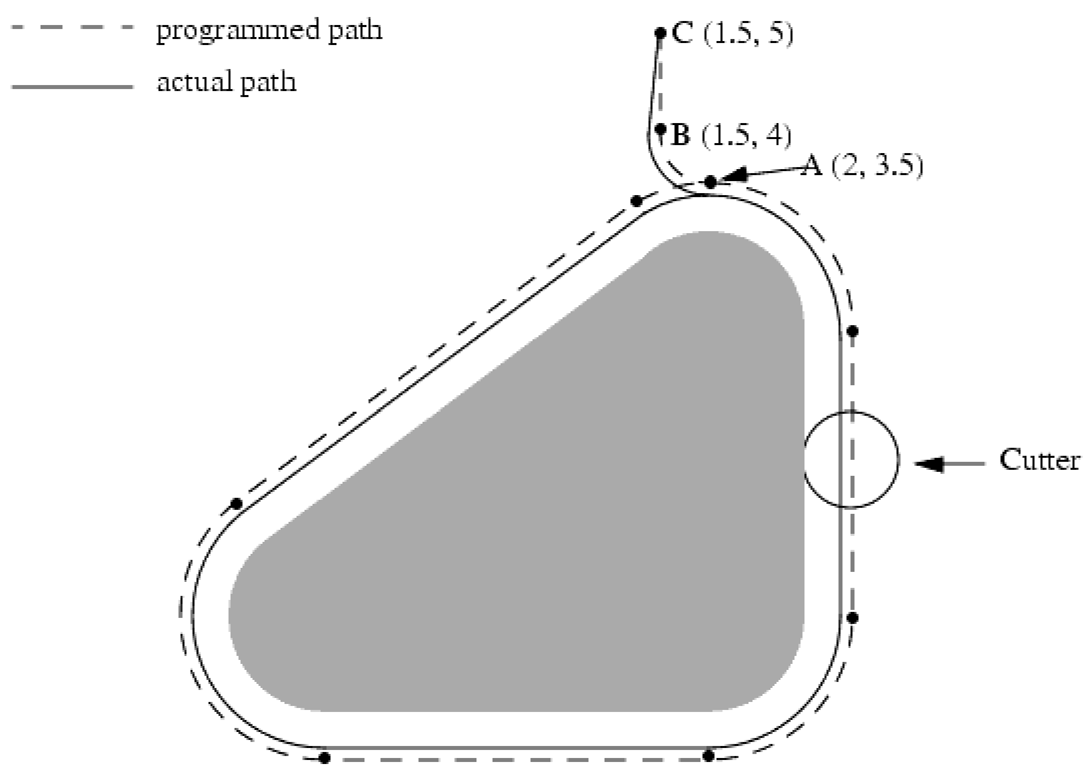


Figure 5. Cutter Radius Compensation Entry Moves

for nominal path contour

Cutter radius compensation is turned on after the first pre-entry move and before the second pre-entry move (including G41 on the same line as the second pre-entry move turns compensation on before the move is made). In the code above, line N0010 is the first pre-entry move, line N0020 turns compensation on and makes the second pre-entry move, and line N0030 makes the entry move.

B.5 Programming Errors and Limitations

The Interpreter will issue the following error messages involving cutter radius compensation. In addition to these, there are several bug messages related to cutter compensation, but they should never occur.

- 1. Cannot change axis offsets with cutter radius comp
- 2. Cannot change units with cutter radius comp
- 3. Cannot probe with cutter radius comp on
- 4. Cannot turn cutter radius comp on out of xy-plane
- 5. Cannot turn cutter radius comp on when on
- 6. Cannot use g28 or g30 with cutter radius comp
- 7. Cannot use g53 with cutter radius comp
- 8. Cannot use xz-plane with cutter radius comp
- 9. Cannot use yz-plane with cutter radius comp
- 10. Concave corner with cutter radius comp
- 11. Cutter gouging with cutter radius comp
- 12. D word with no g41 or g42
- 13. Multiple d words on one line
- 14. Negative d word tool radius index used
- 15. Tool radius index too big
- 16. Tool radius not less than arc radius with comp
- 17. Two g codes used from same modal group.

Most of these are self-explanatory. For those that require explanation, an explanation is given below.

Changing a tool while cutter radius compensation is on is not treated as an error, although it is unlikely this would be done intentionally. The radius used when cutter radius compensation was first turned on will continue to be used until compensation is turned off, even though a new tool is actually being used.

B.5.1 Concave Corner and Tool Radius Too Big (10 and 16)

When cutter radius compensation is on, it must be physically possible for a circle whose radius is the half the diameter given in the tool table to be tangent to the contour at all points of the contour. In particular, the Interpreter treats concave corners and concave arcs into which the circle will not fit as errors, since the circle cannot be kept tangent to the contour in these situations. See Figure 6. This error detection does not limit the shapes which can be cut, but it does require that the programmer specify the actual shape to be cut (or path to be followed), not an approximation. In this respect, the NIST RS274/NGC Interpreter differs from interpreters used with many other controllers, which often allow these errors silently and either gouge the part or round the corner.

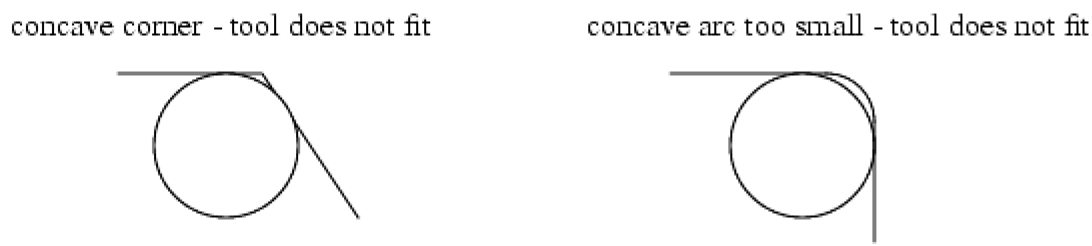


Figure 6. Two Cutter Radius Compensation Errors

In both examples, the line represents a contour, and the circle represents the cross section of a tool following the contour using cutter radius compensation (tangent to one side of the path.)

B.5.2 Cannot Turn Cutter Radius Comp on When On (5)

If cutter radius compensation has already been turned on, it cannot be turned on again. It must be turned off first; then it can be turned on again. It is not necessary to move the cutter between turning compensation off and back on, but the move after turning it back on will be treated as a first move, as described below.

It is not possible to change from one cutter radius index to another while compensation is on because of the combined effect of rules 5 and 12. It is also not possible to switch compensation from one side to another while compensation is on.

B.5.3 Cutter Gouging (11)

If the tool is already covering up the next XY destination point when cutter radius compensation is turned on, the gouging message is given when the line of NC code which gives the point is reached. In this situation, the tool is already cutting into material it should not cut. More details are given in Section B.6.

B.5.4 Tool Radius Index Too Big (15)

If a D word is programmed that is larger than the number of tool carousel slots, this error message is given. In the SAI, the number of slots is 68.

B.5.5 Two G Codes Used from Same Modal Group (17)

This is a generic message used for many sets of G codes. As applied to cutter radius compensation, it means that more than one of G40, G41, and G42 appears on a line of NC code. This is not allowed.

B.6 First Move into Cutter Compensation

The algorithm used for the first move after cutter radius compensation is turned on, when the first move is a straight line, is to draw a straight line from the programmed destination point which is tangent to a circle whose center is at the current point and whose radius is the radius of the tool. The destination point of the tool tip is then found as the center of a circle of the same radius tangent to the tangent line at the destination point. If the programmed point is inside the initial cross section of the tool (the circle on the left), an error is signaled as described in Section B.5.3. The concept of the algorithm is shown in Figure 7.

The function that locates the destination point actually takes a computational shortcut based on the fact that the line (not drawn on the figure) from the current point to the programmed point is the hypotenuse of a right triangle having the destination point at the corner with the right angle.

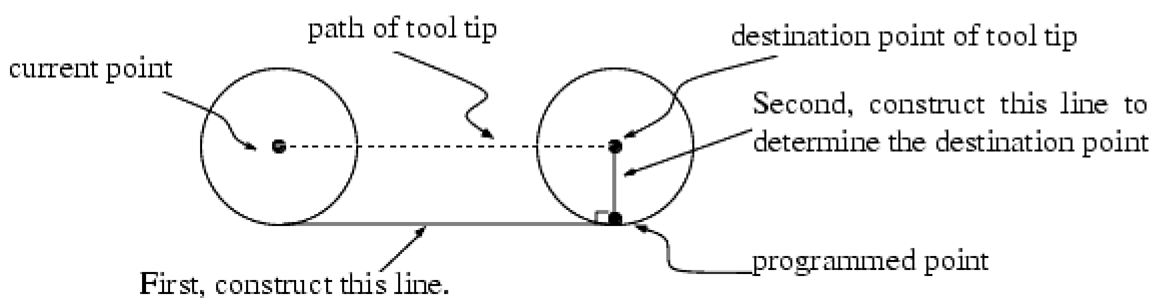


Figure 7. First Cutter Radius Compensation Move - Straight

If the first move after cutter radius compensation has been turned on is an arc, the arc which is generated is derived from an auxiliary arc which has its center at the programmed center point, passes through the programmed end point, and is tangent to the cutter at its current location. If the auxiliary arc cannot be constructed, an error is signalled. The generated arc moves the tool so that it stays tangent to the auxiliary arc throughout the move. This is shown in Figure 8.

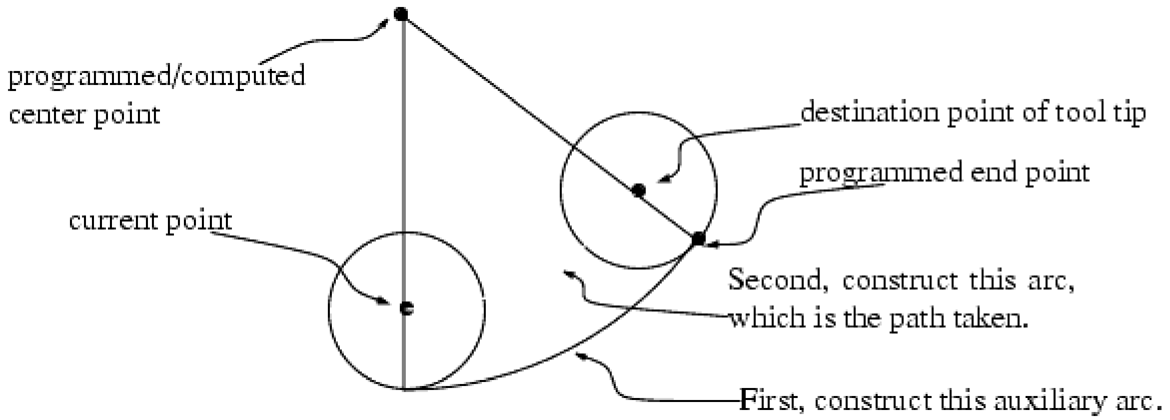


Figure 8. First Cutter Radius Compensation Move - Arc

Figure 8 shows the conceptual approach for finding the arc. The actual computations differ between the center format arc and the radius format arc (see Section 3.5.3).

After the entry moves of cutter radius compensation, the Interpreter keeps the tool tangent to the programmed path on the appropriate side. If a convex corner is on the path, an arc is inserted to go around the corner. The radius of the arc is half the diameter given in the tool table.

When cutter radius compensation is turned off, no special exit move takes place. The next move is what it would have been if cutter radius compensation had never been turned on and the previous move had placed the tool at its current position.

The Interpreter signals when cutter radius compensation is turned on or off by calling the COMMENT canonical function with a message to that effect.

1

The term "cutter diameter compensation" is often used to mean the same thing.

Cutter Compensation

Cutter compensation is used typically to compensate for the difference in the dimensions of the actual cutter used in machining and the cutter used for programming in VisualMill. For example, if the cutter used in programming in VisualMill is 0.25 inches and due to tool wear the actual cutter is only 0.24 inches in size, the user can compensate for this in the controller rather than having to re-program the operation in VisualMill again.

In order to do this the user needs to do the following:

1) Turn cutter compensation on in the Operation
2) Specify the cutter compensation value and the compensation register in the controller (the controller needs to be capable of doing this)

A few things to watch out for:

1) Cutter compensation makes sense only in 2-1/2 axis operations. If you are using roughing (pocketing & facing) the compensation will be turned on only in the final passes.
2) Make sure you are not using Zig-Zag cut traversal in any of the methods that you want to turn compensation on.
3) Make sure you have a linear motion for the controller to turn on the compensation value on. If your first motion is an arc the controller will not be able to turn on the compensation. Thus, in 2-1/2 axis profiling, make sure there is a linear entry motion for the controller to be able to turn compensation on.

Determining G41 or G42

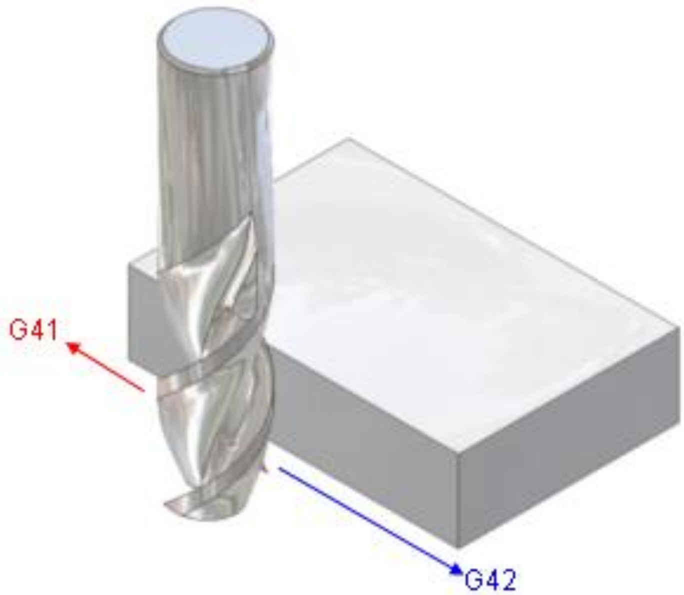
G41: the cutter is to the left of the part when looking in the direction of the cut.

G42: the cutter is to the right of the part when looking in the direction of the cut.

Climb milling features: use G41.

Conventional milling features: use G42.

Since we normally climb mill, we will generally use G41 on a machining center.



Turning Cutter Diameter

Compensation On

- **Prior to compensation, you must take into account the cutter ’s radius and locate the cutter offset by its radius.**

- **In the move turning compensation on, try to make the move as perpendicular to the following move as possible, and try to turn compensation on with the cutter off the part. Note: the D code should be in the same block as the G41 or G42.**

- **The move to turn compensation on should be equal or greater than the cutter’s radius.**

- **Once compensation is on, ignore the cutter ’s size and program as if the cutter has 0 diameter. This usually simplifies programming, especially arc programming.**

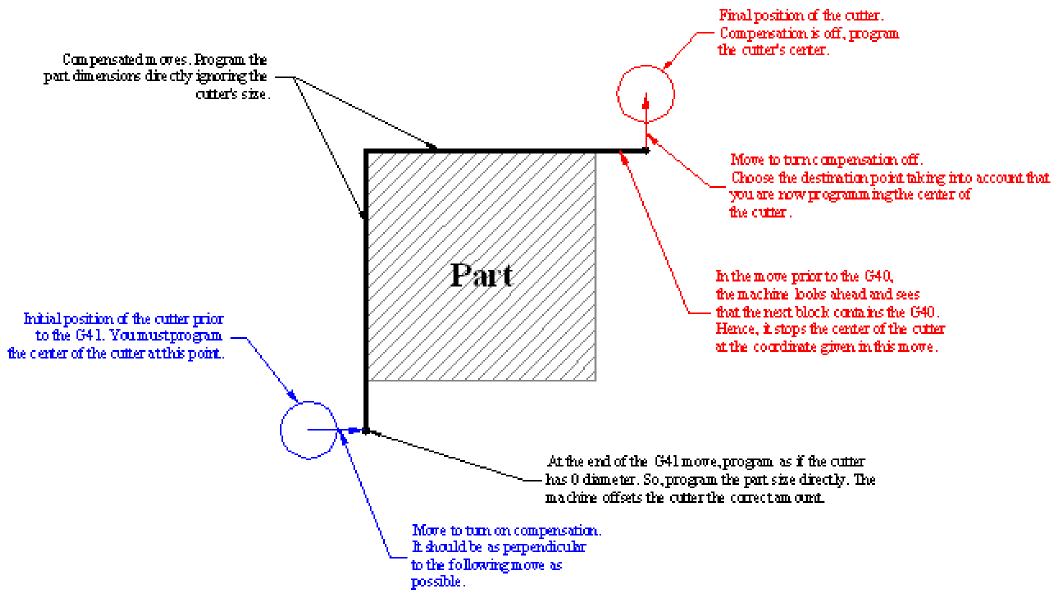
Turning Cutter Diameter Compensation Off

- When performing cutter compensation, the machine looks ahead in your program several blocks so it can calculate tangent points.

- In the move prior to turning compensation off, the machine moves the cutter to the center point in the direction of this move, and then to the cutter's center point in the direction of the G40 move.

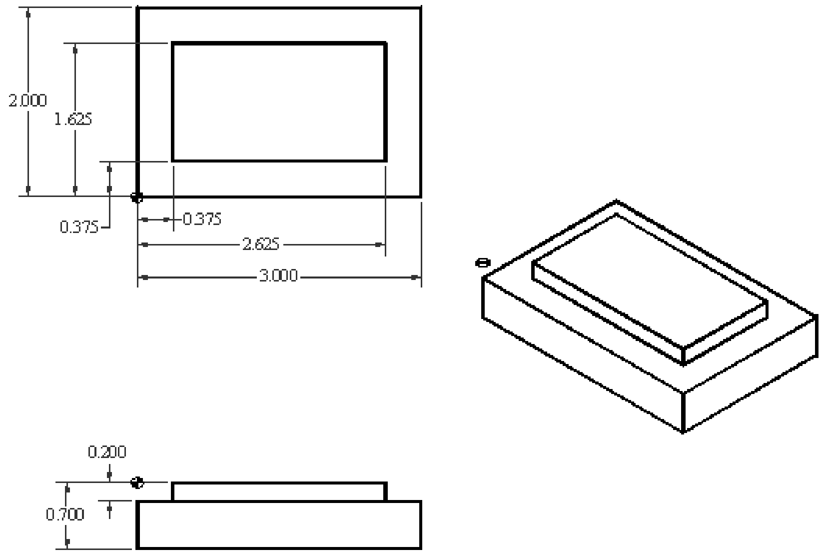
- If possible, turn compensation off with the cutter off the part. If that is not possible, try to select the end point in a clear area of the part such as the center of a pocket.

A Graphical Look at Cutter Diameter Compensation



A Compensation Example

We'll program this part with a ¾" endmill, T1. We'll assume the length and thickness are already machined. So, we will take one roughing pass leaving 0.010" on the profile, and then we will finish the profile with tool diameter compensation.



Follow Planning and Programming Steps (1 - 5)

1. **Examine drawing.**
2. **How will we hold the raw material – in a vise up on parallels.**
3. **Decide what cutters to use – given a ¾ " HSS endmill. We previously calculated 3056RPM and 24IPM.**
4. **Write down the exact sequence of operations:**

   A. Rapid position the cutter clear in – Y.

   B. Rough machine the profile leaving 0.010"material.

   C. Position the cutter clear of the part.

   D. Turn on compensation, finish machine the profile.

   E. Turn off compensation.
   F. Program end.

5. **Convert the sequence of operations to a program:**
   Program Start
   Rough Machining
   Finish Machining with Compensation
   Program End

The Program

Program Codes	Action
%	Program Start
O999	
G20 G40 G49 G80 G99	
G91 G28 Z0	
G90	
T1M6	Load tool 1, ¾" HSS endmill.
S3056 M3	Set the spindle RPM and direction.
G0 G90 G54 X-0.01 Y-0.4	Go to initial position in the WCS using fixture offset G54.
G43 H1 Z-0.2 M8	Rapid to depth with length compensation, coolant on.
G1 Y2.01 F24.	Feed machining the left step.
X3.01	Machine top step.
Y-0.01	Machine right step.
X-0.4	Machine lower step and feed off the part.
G0 Y-0.4	Position move.
G41 X0.375 D1	Move to turn compensation on. Note the D code.
G1 Y1.625	Left edge.
X2.625	Top edge.
Y0.375	Right edge.
X-0.4	Lower edge. Since G40 is in the next block, center of the cutter ends at X-0.4
G0 G40 Y-0.4	Turn compensation off. Center of cutter now at X-0.4 Y-0.4
Z0.1	Lift above the part.
M9	Program end.
M5	
G49	
G91 G28 Z0	
G28 Y0	
G90	
M30	
%	

```
%
O999
G20 G40 G49 G80 G99
G91 G28 Z0
G90

T1M6
S3056 M3
G0 G90 G54 X    -0.01 Y  -0.4
G43 H1 Z   0.2 M8
G1 Y2.01 F24.

X3.01
Y-0.01
X-0.4
G0 Y  0.4
G41 X0.375 D1
G1 Y1.625
```

```
X2.625
Y0.375
X-0.4
G0 G40 Y   -0.4

Z0.1
M9
M5
G49
G91 G28 Z0

G28 Y0
G90
M30
%
```

http://technology.calumet.purdue.edu/met/mfet/275/lecture18/index_files/frame.htm

---

Use of G12/G13 Counter Bore Cycle

When circle milling, we typically use the G2 or G3 (directional) command along with a G41 or G42 code (for the cutter compensation) and the I code to dictate the radius or circle size. You'll also have to program the ramp-in and ramp-out move for the end mill.

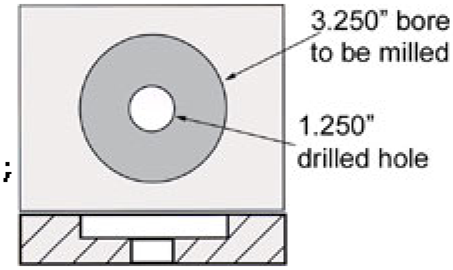
This sounds complex. The G12/G13 codes can simplify programming for circle milling. This cycle is a one-line command that can easily produce a quality bore with a minimum of programming stress.

The counterbore is machined with an end mill. The G12 command machines the circle in a clockwise direction and the G13 in a counter clockwise direction.

Climb milling is preferable to conventional milling. As climb milling is safer and machining forces are directed into the machine table, we will focus on the counter clockwise milling with G13.

To machine a 3.250 size counter bore, let's use a 0.750 size end mill. The bore is started with a 1.250 diameter drilled hole at the center of the bore. Take a look at the program code to machine the circle.

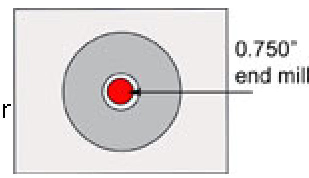
```
T1 M6; (0.75 end mill)
G17 G54 X0 Y0 B0;
G43 H1 Z.1 M3 S1018 M8;
G1 Z-.5 F50;
G13 D1 I0.225 K1.25 Q0.25 F4.07;
G0 Z1.0 M9;
G40;
```



The line of program code that mills the circle is:
G13 D1 I0.225 K1.25 Q0.25 F4.07;
G13 = the counter clockwise spiral bore command
D1 = enable tool radius compensation
I0.225 = start position diameter
K1.25 = ending diameter (part size)
Q0.25 = width of cut per spiral

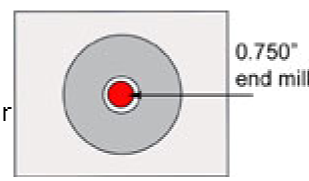
Programming considerations

This example programs to the center of the end mill. This means that the program values need to allow for the radius of the cutter. We can then start with the tool cutter radius value with a zero setting in the offset file.



Programming considerations

This example programs to the center of the end mill. This means that the program values need to allow for the radius of the cutter. We can then start with the tool cutter radius value with a zero setting in the offset file.



To mill this counterbore we are using a 0.750 diameter 4-flute end mill. The part has a pre-drilled 1.250 diameter hole. The G13 code programs a spiral interpolation to create a 3.250 diameter hole.

Our cutter SFPM is 200 with a 0.001 chip load.

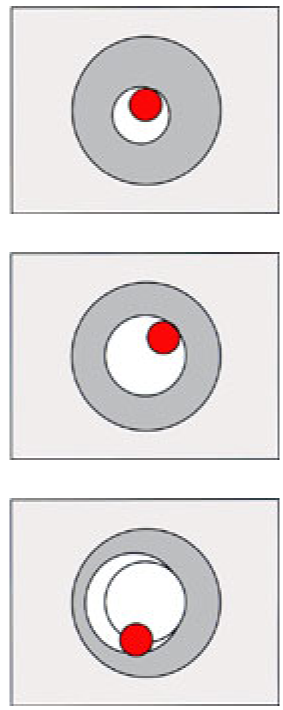
The I value is calculated based on a diameter of 1.200. This is the pre-drilled hole diameter of 1.250 less a 0.050 clearance move (1.250 - 0.050 = 1.200). We will feed the endmill down in the Z axis at the 1.200 diameter position to clear the pre-drilled hole.

Approach diameter = 1.200"
I value = (part diameter - cutter diameter) ÷ 2
I value = (1.200 - 0.750) ÷ 2 = I0.225

Ending diameter = 3.250 (bore size)
K value = (part diameter - cutter diameter) ÷ 2
K value = (3.250 - 0.750) ÷ 2 = K1.25

The Q value should not exceed 2/3 of the cutter diameter in general. The actual width of the spiral depends on the cutting conditions.

Note that the G12/G13 commands enable cutter compensation without the need of the G41/G42 codes. You still must use the G40 command to cancel the cutter comp.



The end mill spirals to open the drilled hole into the required counter bore.

Use of G12/G13 Counter Bore Cycle

Advantages

The main advantage is the ease of programming. With other circle milling programming methods, the coding is more complex. This is a great method to machine chuck jaws for gripping round parts.

With G12/G13 you can mill the same bore in a number of spots with the same program codes. Just change the hole's absolute start point and away you go!

This method has been around for a long time, give it a try. It can be a real handy programming method for many uses.