Defining Arcs Interpreted by Mach3, Version 1.7

By R. G. Sparber

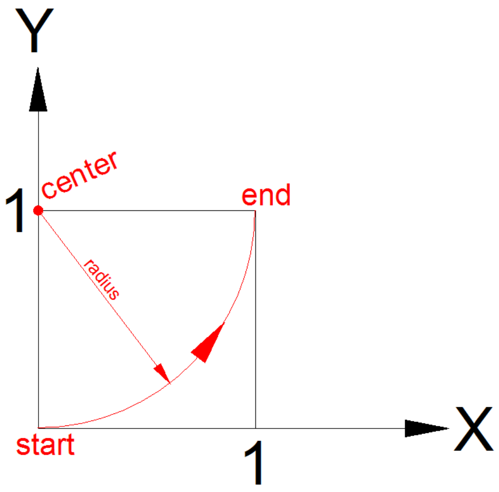
Abstract

An essential part of Computer Numerical Control (CNC) is the language used to control the cutting machine. This language, called G-code, can define straight lines and curves in at least two dimensions. I will present the arc as interpreted by CNC program Mach3, Version 3.

Table of Contents

[1] You are free to distribute this article but not to change it.

The Essential Basics



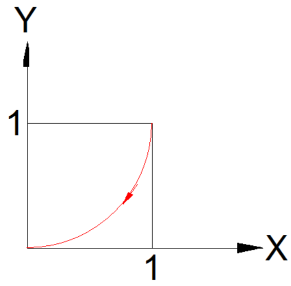
Here is an example of one possible arc. It is drawn on the XY plane with the center located at X=0, Y=1. It starts at X=0, Y=0 and ends at X=1, Y=1. This means it is moving in a counterclockwise direction. Given the defined center and both starting and end points, the arc must have a radius of 1.

This rambling description fully defines the path of this arc to both us and to Mach3. It is just a matter of placing this information into the "Center Format" as defined by Mach3.

If I specify the arc with the starting point, end, and radius, the "Radius Format" would be a good choice. I use CamBam to generate my G-code and it only seems to support Center Format so that is what I use.

When manually generating arc commands, you may find cases where Radius Format is easier to use. I will present both.

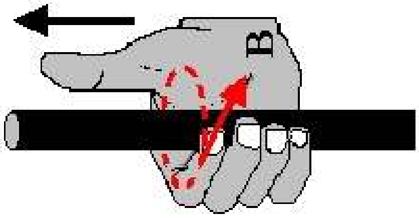
Clockwise and Planes



A word about the definition of clockwise and how it relates to the three planes. Consider the case of a clockwise rotation on the XY plane.

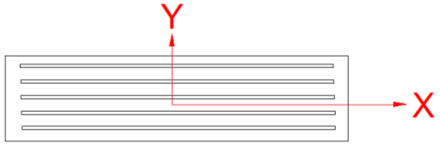
First of all, I call it the XY plane and not the YX plane because the first letter defines the horizontal axis. The second letter is the vertical axis.

I don't know about you, but with so many digital clocks around, I have to think a moment about the meaning of clockwise. Ah, that helps a bit.

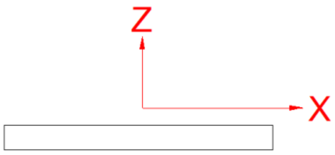


An alternate way to think about clockwise relative to a plane is the "right hand rule". Visualize the axis point out of the plane in the positive direction. Then point your thumb along this axis in the negative direction. Your fingers will curl in the clock wise direction.

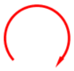
Now, consider the Z axis. It is pointing out of the screen. As you look at the red arc shown above, it is by looking along the Z axis in the negative Z direction.



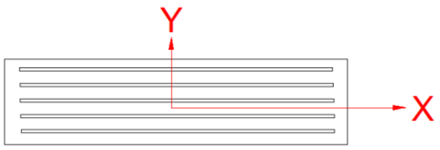
Thinking about my mill spindle relative to the table, a positive X movement causes the Center Of Rotation (COR) to move to the right relative to the table. A positive Y movement causes the COR to move from the front of the mill towards the column.

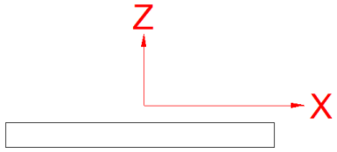


And a positive Z movement causes the spindle to rise away from the table.

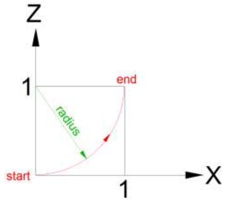
It should be no surprise that as we look down on the table, a clockwise rotation is  Try the right hand rule and you will get the same result.

This convention for the XY plane must hold for the other two planes.

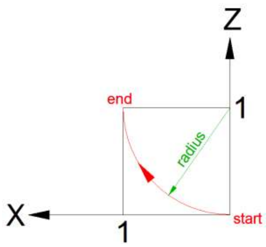




Next up is the XZ plane. Again position your eye so it is looking "down" on the plane. This orientation means looking in the negative Y direction.



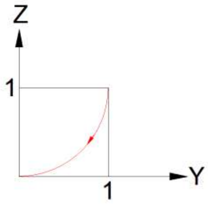
You were probably standing in front of your mill and saw this arc moving counterclockwise. Wrong viewpoint. You are looking at the XZ plane in the *positive* Y direction.



Walk around to the back of the mill. Then use your X-ray vision to look through the column. OK, you are now looking at the XZ plane in the negative Y direction. But wait! Isn't my arc now moving in a clockwise direction?

What I initially thought was a bug in Mach3 is actually correct. The trick is to always look along the perpendicular axis in its negative direction.

If you are using the right hand rule and are standing in front of the mill, point your thumb along the Y axis away from the column.



And finally, we have the YZ plane. I walk to the right end of the mill's table and look down the X axis in a negative X direction. To my relief, the arc is clockwise.

With the right hand rule, your thumb is along the X axis pointing to the left.

Center Format

The center of the arc is defined as an increment from the starting point.

The XY Plane Arc Command

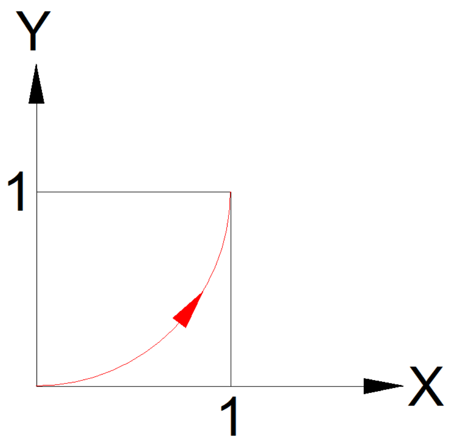
I have taken a single line of G-code and written it vertically so comments can be added:

G03	move counterclockwise
X1.000 Y1.000	end at X=1, Y=1
I0.000	X offset of center with respect to the start
J1.000	Y offset of center with respect to the start
F10	optional: move at 10 inches per minutes

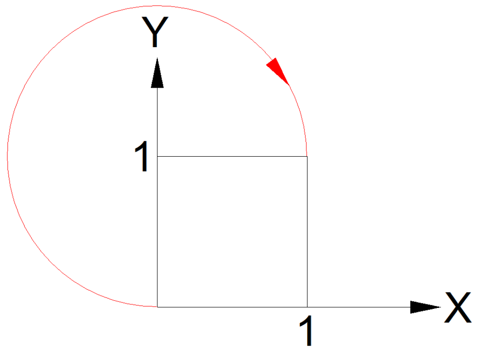
Write it as: G03 X1.000 Y1.000 I0.000 J1.000 F10
<direction> <end> <arc center offset from start> <feedrate>

Now, let's go into more detail.

G03 X1.000 Y1.000 I0.000 J1.000 F10

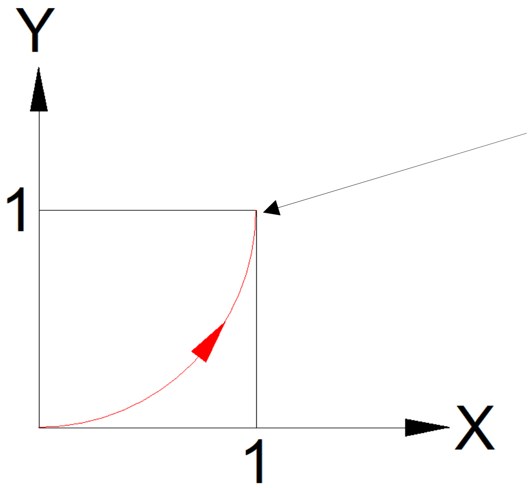


G03 says to move counterclockwise.



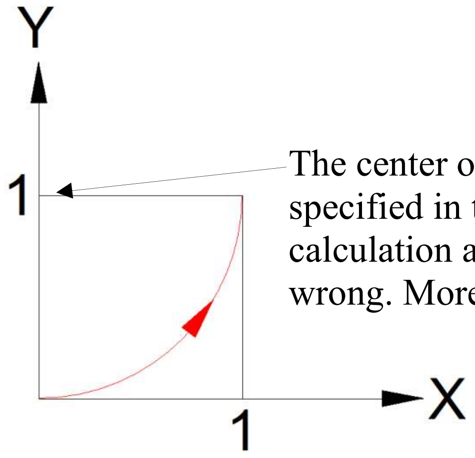
If G03 is replaced with G02, I move clockwise. Given the same start and end points, here is the arc.

G03 X1.000 Y1.000 I0.000 J1.000 F10

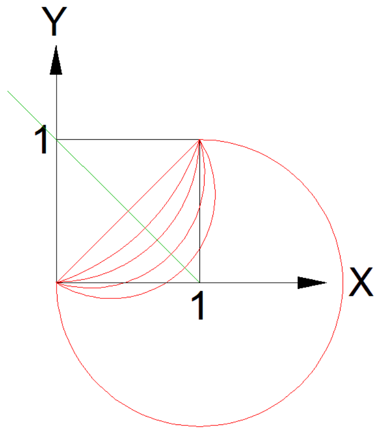


X1.000 Y1.000 is the location of the end of the arc.

So far, we know the starting point of the arc because it was defined by a previous G-code instruction. The end point of the arc is specified within the arc instruction. We also know the direction the cutter following this arc will move. What is not yet specified is the center of the arc.



The center of this arc is at X=0, Y=1. This is not explicitly specified in the g-code but calculated instead. Mach3 does this calculation and will stop executing the G-code if the answer is wrong. More on this later.

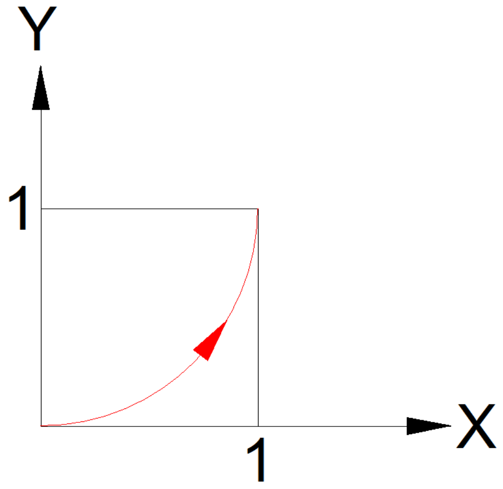


Here I have held the start and end points of various arcs constant and changed the center of the arcs along the green diagonal line. Notice that I generated a family of arcs. The straight red line passing through the center of the square is for an infinately large radius. The longest arc is with its center on the X axis.

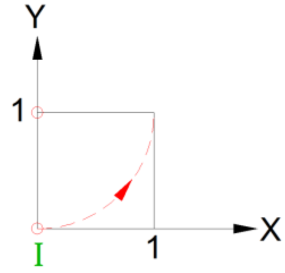
Notice that each arc has moved through a different number of degrees. Furthermore, I can adjust the start and end point to place the arc at any location.

In general, by specifying the start, end, and center of an arc, you have completely defined its path. So how is this done?

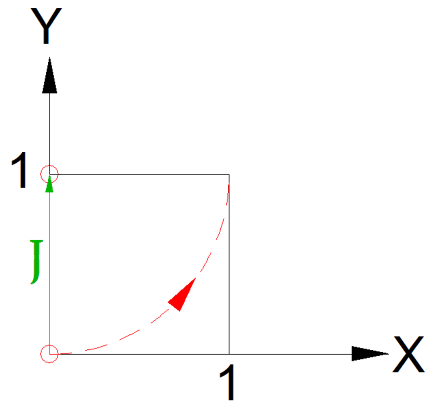
G03 X1.000 Y1.000 I0.000 J1.000 F10



Here is where things start to look a bit strange. But stick with me because this approach actually makes a lot of sense. I0.000 J1.000 specifies the center of the arc with respect to the *starting* point, X=0,Y=0. In other words, I0.000 and J1.000 are incremental rather than absolute. They tell Mach3 the increment between the starting point and the center of the arc.



An I0.000 tells Mach3 that the center's X axis position is zero distance along the X axis from the start point. Since we started at X=0 and move a distance of zero, the X location of the center of the arc is also at zero.

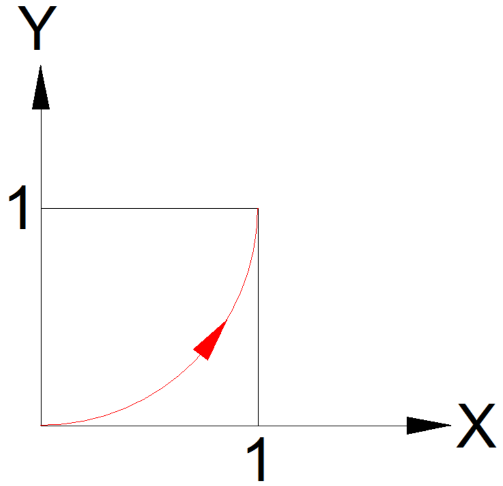


Similarly, J1.000 tells Mach3 that the arc center's Y axis position is 1 inch along the Y axis from the starting point. Since we started at Y=0 and move a distance of 1 inch, the Y location of the center of the arc is at 1 inch.

Therefore, the calculated center of the arc is at X=0, Y=1.

Ideally, the distance from starting point to the center of the arc must be the same as the distance from end point to center since the arc only has a single radius. But the numbers that define these points may have been generated by a computer so could contain round off error. If Mach3 insisted on the distance from the center to the starting point to be exactly equal to the distance from the center to the end point, the command would often fail. Instead, it tolerates a difference in calculated radius of 0.0002 inches or 0.002 mm. If the variation in radius is more than this, you have a bug.

G03 X1.000 Y1.000 I0.000 J1.000 F10



F10 is the feed rate. Else ware[2] I have specified that I want to use inches and not Metric so the "10" means *inches* per minute.

The arc command does not live by itself. Let's look at the bigger picture.

[2] I will present this later.

I will again present a complex command line of code (called a block) vertically and then assemble it into a single line.

G20	specified that I am using inches rather than Metric
G91.1	I am running in incremental IJ mode
G64	Constant Velocity[3]
G40	Cutter compensation is off[4]

So we get the first block of the program:

G20 G91.1 G64 G40

This is followed by a command that tells Mach3 to run in absolute distance mode for the axes.

G90 I am running in absolute distance mode.

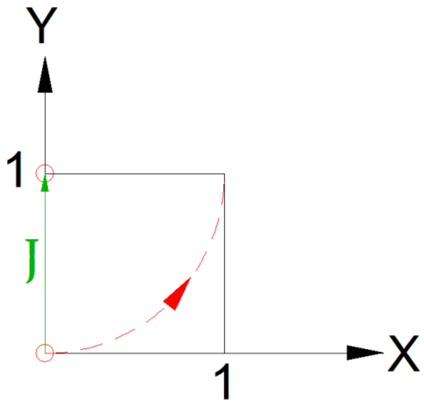
A word about CamBam related to the above two blocks. When I generate G-code with CamBam, I used to see: G20 G90 G91.1 G64 G40. Notice that G90 is in the same block as G91.1. Additionally, I have Mach3 defaults set up for Distance Mode: Absolute and IJ Mode: Incremental. This was done via the Config|General Config… screen. It works as expected but for the wrong reason.

G90 and G91.1 are members of the same "Modal" Group. This means that only one member can be active in the same block.

Mach3 sees G90 and then G91.1 in the same block. It skips all but the last member, G91.1, and sets IJ Mode to Incremental. We end up with Distance Mode set to Absolute because it is the default while having IJ Mode Incremental because of G91.1 and not because it is the default! How's that for confusing? The correct way to solve this problem is to put G90 and G91.1 in separate blocks as done here. Fortunately, it is easy to change CamBam so it always does this action. See http://rick.sparber.org/ABCM.pdf for details.

[3] This parameter is beyond the scope of this article.

[4] This parameter is beyond the scope of this article.



The next line of the program tells Mach3 that we are working in the XY plane. This is done with the **G17** command. If I were to specify a plane different from the one defined in the arc command, Mach3 will stop interpreting the arc command and flag an error.

It is legal to draw an arc on one plane and then change planes so another arc can be drawn on the new plane.

Before I execute the arc command, I must define the starting point. That could be as simple as saying **G00 X0.000 Y0.000** which tells Mach3 to go to X=0 Y=0 at top speed.

Putting this all together we get:

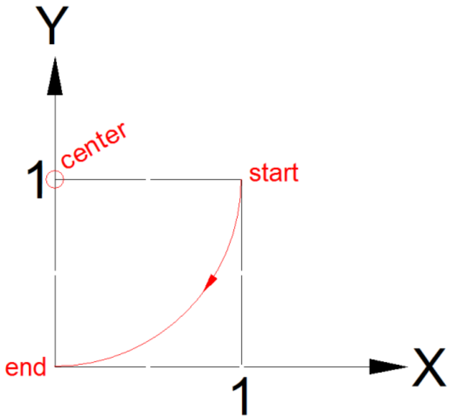
G20 G91.1 G64 G40 (set up the G-code's environment)
G90 (I am running in absolute distance mode)
G17 (I am operating in the XY plane)
G00 X0.000 Y0.000 (here is my arc starting point)
G03 X1.000 Y1.000 I0.000 J1.000 F10 (ah, there is my arc command)
And at the end of the program you should find **M30** which says this is the end and it is time to rewind to the start.

Negative Offsets

There will be cases where the offset increments will be negative.

Say we have

G00 X1.000Y1.000 (here is my arc starting point)
G02 X0.000 Y0.000 I-1.000 J0.000 F10 (arc is clockwise)

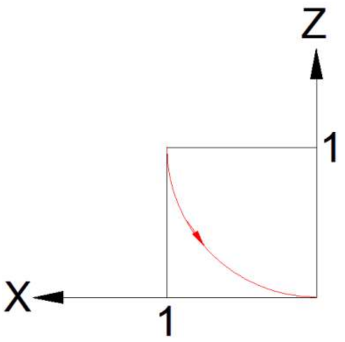


That **I-1.000** is telling Mach3 that the center of the arc along the X axis is to the left of the start. Moving to the left along the X axis is a negative move.

The XZ Plane Arc Command

We are not limited to the XY plane. If I replace G17 with G18, I select the XZ plane.

Recall that with the XY plane selected, the arc command used incremental parameters **I** and **J** where **I** was the increment along the X axis and **J** was the increment along the Y axis. If you choose a plane involving the Z axis, you would use **K** as the associated increment. For example,



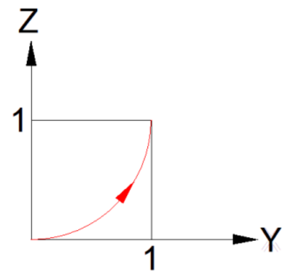
```
G20 G91.1 G64 G40
G90
G18 (selects the XZ axis)
…

G00 X1.000 Z1.000  (starting point)
G03 X0.000 Z0.000  I-1.000 K0.000  F10 (draw the arc CCW on the XZ plane)
M30 (end and rewind program)
```

The start of the arc is at X=1, Z=1. The end of the arc is at X=0, Z=0. The center of the arc is at a -1 offset from the start along the X axis and at zero offset along the Z axis. If the flip in the direction of the X axis confuses you, please refer back to page 4.

The YZ Plane Arc Command

G19 selects the YZ plane.



```
G20 G91.1 G64 G40
G90
G19 (selects the YZ axis)
…

G00 Y0.000 Z0.000  (starting point)
G03 Y1.000 Z1.000  J0.000 K1.000  F10 (draws the arc counterclockwise)
M30 (end and rewind program)
```

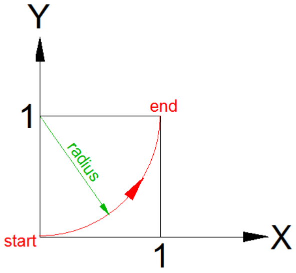
The start of the arc is at Y=0, Z=0. The end of the arc is at Y=1, Z=1. The center of the arc is at a zero offset from the start along the Y axis and at a 1 inch offset along the Z axis.

Radius Format

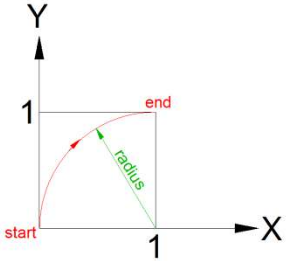
The center of the arc is defined by a radius, starting and ending point.

The XY Plane Arc Command

In Center Format we defined the arc with the starting point, ending point, and center point with respect to the starting point. In Radius Format we define the arc with the starting point, ending point, and radius. Since we are not using IJK, no need to set the environment to incremental mode with **G91.1**. There are some limitations on the number of degrees these arcs can traverse. My best advice is to watch the motion carefully on your first dry run.



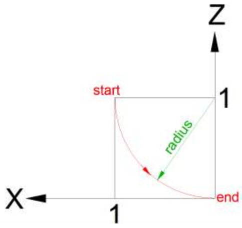
```
G20 G90 G64 G40 (set up the G-code's environment)
G17 (I am operating in the XY plane)
G00 X0.000 Y0.000  (here is my arc starting point)
G03 X1.000 Y1.000  R1.000  F10 (CCW arc command)
M30 (end and rewind program)
```



Notice what happens when we just change from counterclockwise to clockwise. The center of the arc moved from X=0, Y=1 to X=1, Y=0.

```
G02 X1.000 Y1.000  R1.000 F10 (CW arc command)
```

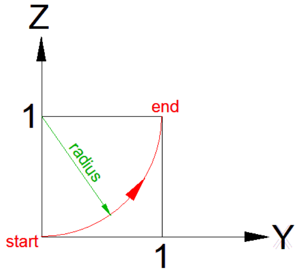
The XZ Plane Arc Command



G20 G90 G64 G40 (at start of program; set up the G-code's environment)
G18 (I am operating in the XZ plane)
G00 X1.000 Z1.000 (here is my arc starting point)
G03 X0.000 Z0.000 R1.000 F10 (CCW arc command)
M30 (end and rewind program)

Note that we are looking along the Y axis in the negative direction so the X axis appears flipped as described on page 4.

The YZ Plane Arc Command



G20 G90 G64 G40 (at start of program; set up the G-code's environment)
G19 (I am operating in the YZ plane)
G00 Y0.000 Z0.000 (CCW arc command)
G03 Y1.000 Z1.000 R1.000 F10 (arc command)
M30 (end and rewind program)

Acknowledgement

Thanks to Paul Thompson for getting me to think more about Radius Mode. Thanks to Gerry (Ger21) and BR549 on the machsupport.com web site. They helped me understand the interaction between G90, G91.1, and the default values for Distance Mode and IJ Mode. BR549 also helped me with the standard G-code format. Thanks to Martin Connelly (from the UK) for helping me understand clockwise on the XZ plane.

I welcome your comments and questions.

If you wish to be contacted each time I publish an article, email me with just "Article Alias" in the subject line.

Rick Sparber
Rgsparber.ha@gmail.com
Rick.Sparber.org