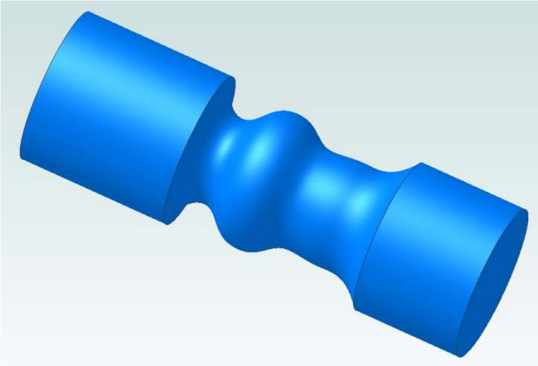
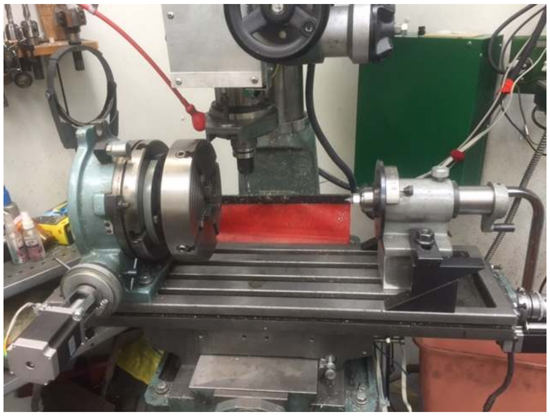
2 ¾ D Machining On a 4 Axis RF-30 Mill/Drill, version 1.1

By R. G. Sparber



It would not be so hard to make this part with a 5 axis screw machine and the related 3D software tools. The workpiece is fixtured so it can be turned. The cutter can be both spun and oriented. The result is an amazingly versatile (and expensive) machine.



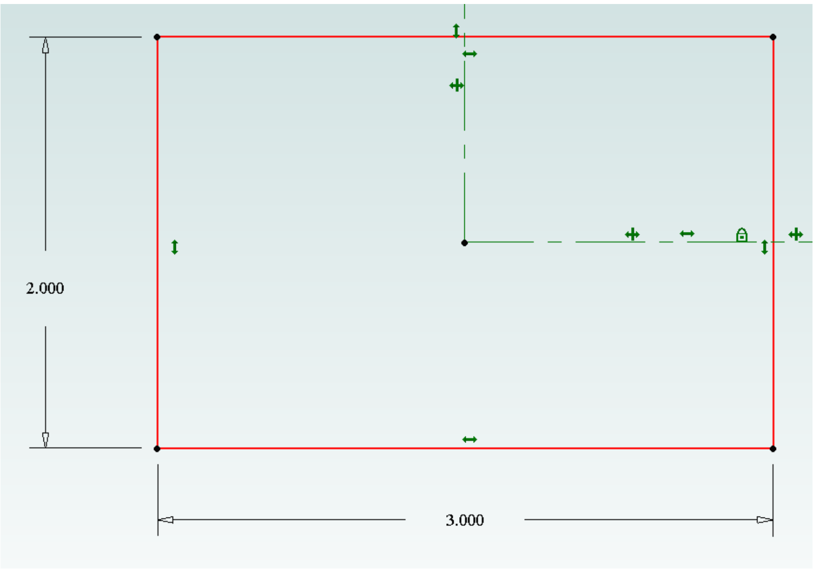
My challenge was to machine such parts with standard 2 ½ D software tools and a 4 axis Computer Numerical Control (CNC) driven mill/drill.

I have X, Y, Z, and A axis control. My A axis lets me rotate the workpiece. Unlike a 5 axis mill, I am unable to machine features like undercuts or angled holes without changing the fixturing.

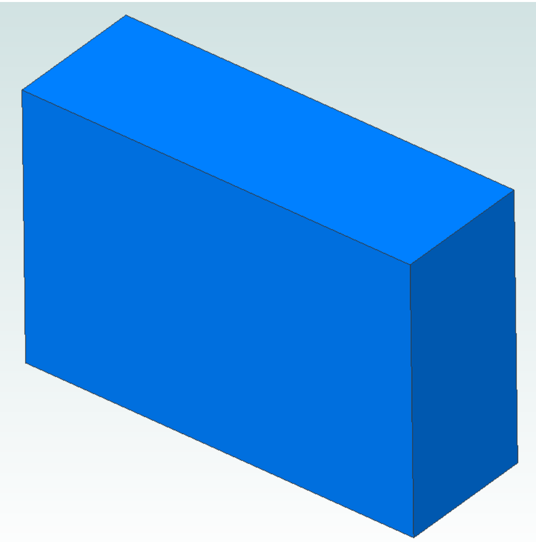
---

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

In CNC there is a concept called "2 ½ D". Let me explain it first.

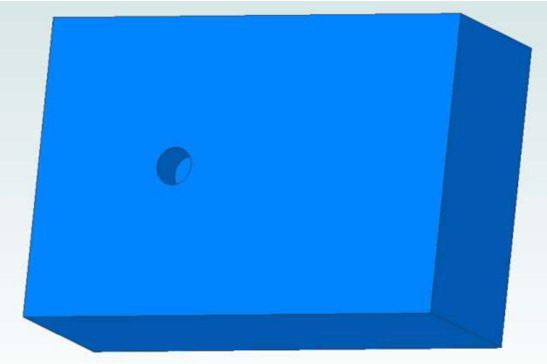


I have drawn a 2 by 3 rectangle on a XY plane. It is a two dimensional object: 2D.

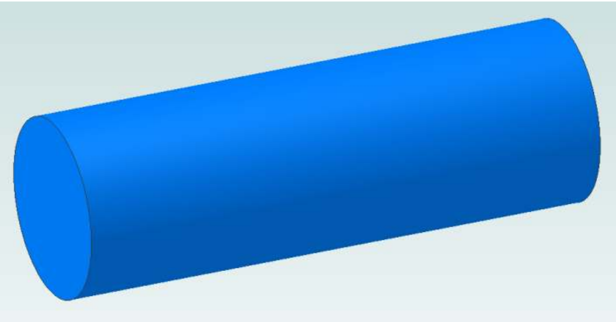


By "extruding" my rectangle along the Z axis, I have created a 1 by 2 by 3 rectangular solid. Although it really is a three dimensional object, notice that my Z dimension is uniform. I can define this object by first setting the thickness and then deal with a 2D object. For this reason, we call objects that can be extruded "2 ½ D".

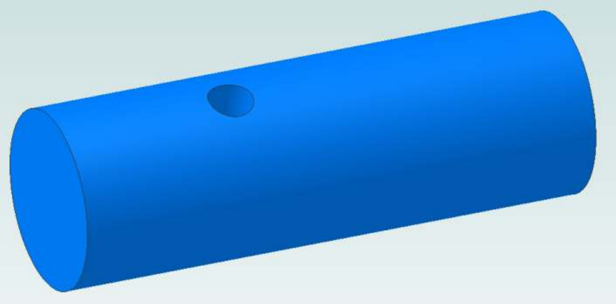
2 ½ D objects are easily machined on a mill using 3 axes. First I would set the 1 dimension using the Z axis. Then I could drive the X and Y axes to define my 2 by 3 surface.



After machining all 3 dimensions of the block, I can mill a hole. The hole is at a specified XY location and its depth is set by changing the Z axis. So this is also 2 ½ D.



Here I have drawn a circle of diameter 1 and extruded it into a cylinder of length 3. This too can be considered a 2 ½ D object.



The command needed to mill this hole is identical to what was used on my block.

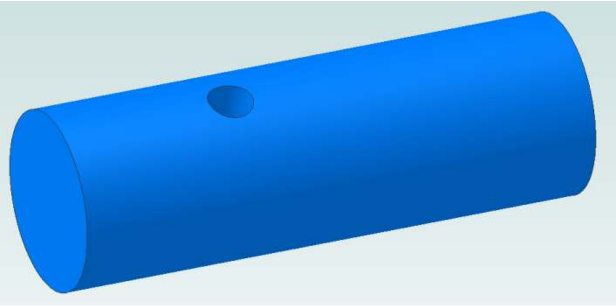
But these two cases are not identical. With my block, the hole could be milled after moving around the XY plane. With the cylinder, I would first machine the outside diameter and then lay it down. Repositioning the workpiece, in my humble opinion, takes us from 2 ½ D to a ***slightly*** higher dimension.

If I was running a 5 Axis Screw Machine, then the part would not need to be repositioned. My cutter would be moved within its 5 axes to mill this hole.

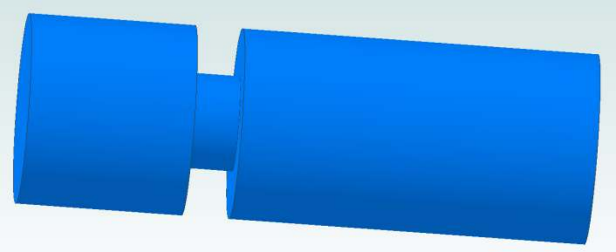
So what do we call it when a workpiece is machined 2 ½ D, repositioned, and again machined 2 ½ D? At the risk of becoming a human lightning rod, I will call it "**2 ¾ D**" at least within this article.

In the above example, I manually repositioned the workpiece. My question also applies when a 4th axis is moving the part.

Manually repositioning the cylinder is one example of 2 ¾ D. But the broader case is setting either a new position or a new ***velocity***.



Say I leave the spinning cutter down in this hole and start to rotate the part. My rotational velocity went from zero to a finite number.



I get a reduced diameter section as if the part was on my lathe and a parting tool was used. However, I'm using the side and bottom of my end mill here and running on the centerline of the part.

The standard way to do this operation is to specify the depth of the cut and the number of degrees of rotation. This causes the end of the cutter to spiral down to its final diameter. Then we go another full revolution to insure uniform diameter. For example,

```
G1 F1 A360 Z0.25
G0 A720
```

My A DRO ends up showing a movement of 720 degrees.

An alternate approach is to hack a line of 2 ½ D code:

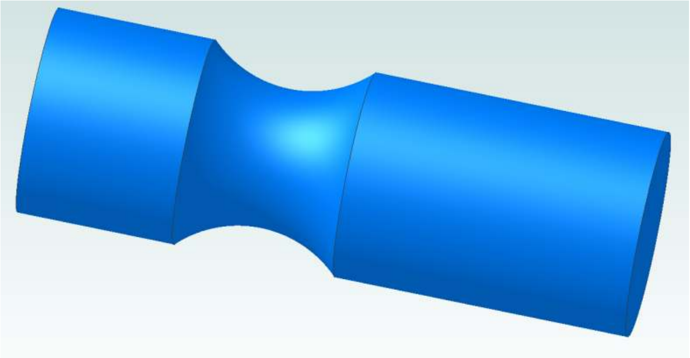
```
(start rotation)
G1 F1 Z0.25
(pause to allow one full rotation)
```

This says to start the cylinder rotating and keep it rotating. Then feed the cutter down to its final depth. When done, pause to insure the cutter visits the full perimeter of the part. My A DRO does not increment.

In both approaches we have the ability to leave the cutter at Z0.25 and move along the X axis to widen the gap.

In other words, I can use my CAM software tools to define a profile and then add a few lines of code manually to shape cylinders. I do need to pause at the final diameter to insure that the cutter visits the full 360°.

Clearly, for this example, the alternate approach is more complex than the standard one.



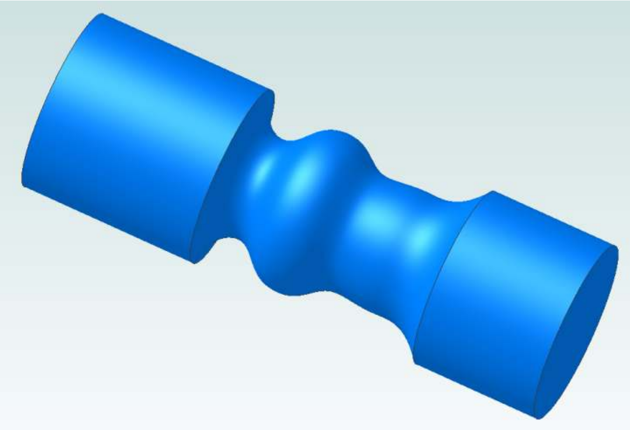
Look what I get when I move the cutter along the Y axis so it is side milling. The cutter is lowered so its bottom is just below the center of rotation of the part.

My XY movement defines a radius. Since the part is turning continuously, I get this saddle.

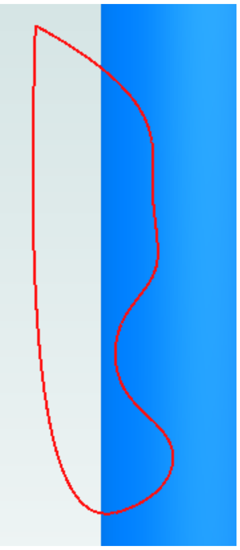
This too can be done by adding a rotation command to the line of code that defines the arc.

Using my alternate approach, I start by drawing a 2D radius and ended up with a saddle that I call 2 ¾ D.

OK, so much for boring shapes.



Drawing in 2D provides limitless flexibility. No need to just draw a radius, I can draw any shape line as long as it does not have undercuts[2].



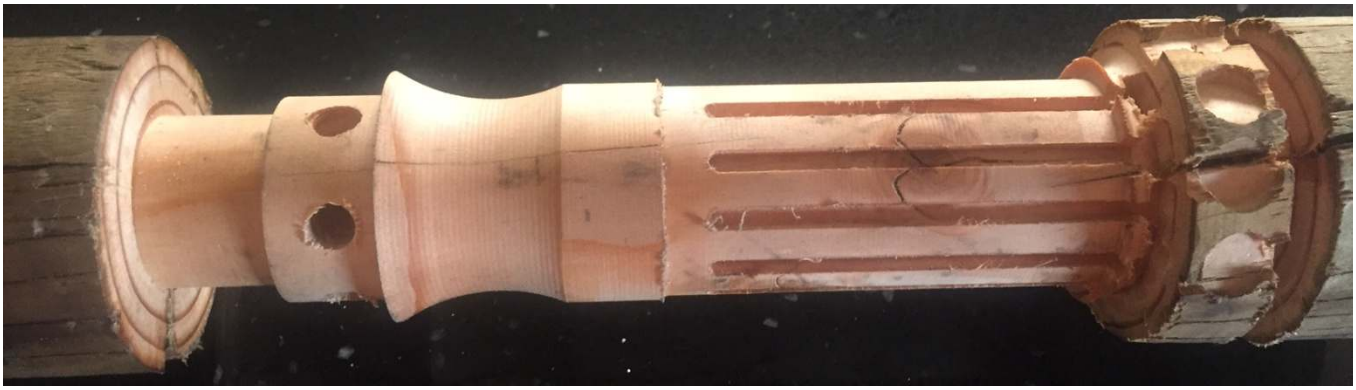
First I drew a freehand 2D figure. In my drawing program I used Revolve Cut to create the rendering above. But in order to generate the complex g-code, I simply treat this as a 2D object cut into a rectangular block. The longitudinal axis of the block is the center of rotation of my cylinder.

I add a continuous rotate command in front of the g-code and bracket it with pause command to insure the ends are square.

If I used the conventional rotate format, I would need to add an index command to each line of code. Given the complex shape of this figure, that would be a tall order.

---

[2] An undercut cannot be side milled.

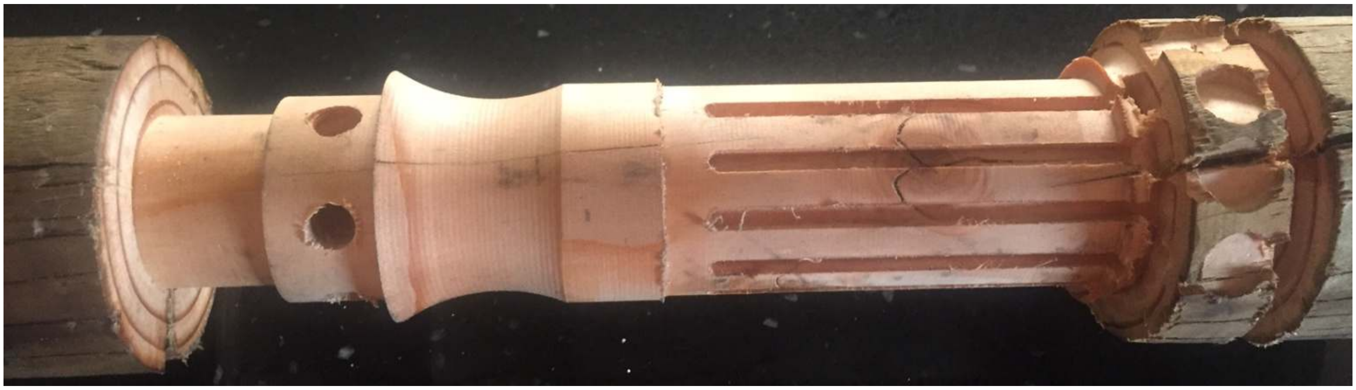


And finally, here is the result of a test run that employs both side milling and end milling. I am machining an old wooden stake in order to find my programming errors. Plenty to be seen here but you can also see some sharp corners and a nice contour.

Starting on the left, I run my ¼ inch end mill along the centerline. A pause at the start and finish gives me the uniform shoulders. Then I raise the cutter up a little and cut the slightly larger diameter. Next is the saddle. This requires me to move the cutter off of the centerline and start side milling. With the bottom of the cutter slightly lower than the center or rotation, I simply follow a 2D curve.

A third and forth uniform diameter comes next so I move my cutter back to the centerline and start above the part. At the end of the forth uniform section is a fillet. The end mill is again moved off center so I am side milling as with the saddle. It is supposed to transition from the forth uniform diameter to the side wall of the largest diameter. An error in both X and Y caused me to miss that mark.

The final cut with the part continuously rotating is on the far right. It defines the right face of the largest diameter. It was cut with the end mill back on the centerline.



After executing an M5 command, I changed to an 1/8 inch end mill. Then I executed a M3 S0 (that is S<zero>) command to stop A axis continuous rotation but keep the cutter turning. G0 A0 moves me back to zero degrees.

With the cutter at the centerline, I cut a hole on the left and the slot in the middle. At the right end of the slot, I move forward and backwards along the Y axis. There was a bug here and I went too far so ran into the adjacent feature.

The cutter is then raised enough to get on top of the large diameter on the right. I ran out of wood so you can't see that it cut a flat. Then it milled two pockets that are symmetric around the centerline. These operations are repeated every 60 degrees[3].

With the cutter out of the way, I execute an A30 which indexes the part to 30 degrees. Then I repeat the slot cutting operations. These operations are also repeated every 60 degrees[4].

In all cases, I first deal with the A axis and then machine as if I was doing 2 ½ D.

The file is 44K bytes but most of it is copy and paste.

---

[3] 0, 60, 120, 180, 240, and 300 degrees.

[4] 30, 90, 150, 210, 270, and 330 degrees.

You may be wondering about this continuous rotate command. I have a Variable Frequency Drive (VFD) on my spindle motor. RPM is set with a dial within easy reach. I never felt the need to set the RPM using g-code. I just use the code to turn the spindle on and off. This is done with the M3 and M5 commands.

Mach3 permits me to define a spindle motor as using a Pulse/Direction interface. In other words, the spindle motor would be controlled just like my axis stepper motors. I then wire the spindle's Pulse output to my A axis Pulse input[5]. I also have a connection from my A axis output to the Pulse input. This unorthodox arrangement gives me the ability to free rotate or index the A axis.

To free rotate the A axis at 1 RPM, I execute

```
M3 S2300
```

M3 says to turn on the spindle. It now also starts pulses to be sent to my A axis driver. The S2300 defines the rate at which pulses are sent. Empirically I found that 2300 pulses per minute translates to about 1 RPM.

For the full story, see http://rick.sparber.org/CNCHW.pdf starting at page 18.

Acknowledgments

Thanks to Martin from Leasingham for opening my eyes to this approach and reviewing the article.

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

---

[5] I wonder if this could be done by assigning the same output pin to spindle Pulse and A axis Pulse. That would save a Break Out Board output pin.