REVIT RTC TECHNOLOGY CONFERENCE NORTH AMERICA 2014

Session 14

LAB: Throw Your Family a Curve- Part A

Paul F. Aubin: www.paulaubin.com

Class Description

Have you ever tried to control the shape of a curved form parametrically in the Family Editor? If so you have no doubt discovered that flexing them sometimes throws you a curve ball. In this session, we will explore several techniques to tame your unruly parametric curves. We will look at examples of circles, arcs, quarter round, half round, arches and even throw in some splines. We will look at both simple and compound curves. We will work primarily in the traditional Family Editor but will discuss how techniques apply to the massing Family Editor environment as well. We will explore curvature, rotation and throw in some trigonometry for good measure. After this session, I cannot guarantee that you will never have another misbehaving curve in your family content, but what I can promise is that you come away with several useful tools to help you tame them when curve mischief strikes!

About the Speaker:

Paul F. Aubin is the author of many Revit book titles including the widely acclaimed: *The Aubin Academy Mastering Series*, his all new *Renaissance Revit* and Revit video training at www.lynda.com/paulaubin. Paul is an independent architectural consultant providing Revit® Architecture implementation, training, and support services. Paul's involvement in the architectural profession spans over 20 years, with experience in design, production, CAD management, mentoring, coaching and training. He is an active member of the Autodesk user community, an Expert Elite and is a high-rated repeat speaker at Autodesk University, Revit Technology Conference and the BIM Workshops. His diverse experience in architectural firms, as a CAD manager, and as an educator gives his writing and his classroom instruction a fresh and credible focus. Paul is an associate member of the American Institute of Architects. He lives right here in Chicago with his wife and three children.

Contact Paul directly from the contact form at his website: **www.paulaubin.com**

Introduction

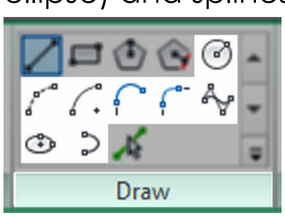
Families like straight lines. I don't have this on absolute fact, but rather on personal experience. Controlling the location of straight lines is easily accomplished in the Family Editor using simple reference planes or reference lines. When you introduce curves: arcs, circles and ellipses it becomes a little more difficult to control them parametrically in a reliable and stable way. Techniques and procedures designed to address this issue are what this lab is all about.

Datasets

The dataset files for this lab are provided on the local machines. All starting files for each exercise are provided and several "catch-up" files are also provided. If you get behind, please load the catch-up file to keep going with us. You can download a copy of the dataset here: https://app.box.com/s/ds4jkd92eno4vdnhwtmo

Curvature in the Traditional Family Editor

This lab focuses on techniques in the traditional family editor environment. I assume that you are familiar with the basics of the traditional family editor such as: establishing reference planes, creating labeled dimensions and parameters and building 3D forms and voids. This lab will *not* explore the massing family editor environment. As such, when we speak of curvature in the family editor in this paper, we are speaking of two-dimensional curves confined to the active work plane that include: circles, ellipses, arcs (portion of a circle), elliptical arcs (portion of an ellipse) and splines (Bezier splines with control handles).



Curves available in sketch mode for solid and void forms in the traditional family editor

These are the shapes you can use in the five solid and void forms available to us in the traditional family editor. We will explore several isolated scenarios in Part A of the lab and then combine techniques together to create more complex forms in Part B.

Constraint and Parameter Direct Attachment

The general rule-of-thumb in family creation is to create a clear and consistent hierarchy between the references, constraints and geometry. Typically this means that you would want to lay down your reference planes and reference lines first. You would next apply constraints and parameters to these references and flex them to be sure that they function properly. Finally, you would build your geometry and lock it to this properly flexing armature of references. This is the so-called "bones, muscle and skin" analogy. It is my experience that this is often the best approach in most situations. However, as with any rule or guideline there are always exceptions. In the next few lessons, we'll look at a few examples where the dimensions (be they constraints or parameters) will be applied *directly* to the geometry instead of a reference. The general rule should still be followed: if you can dimension your reference lines or reference planes first, and then attach geometry to them, it is generally preferred. But as we will see, this is not always possible or desirable, particularly when curves are involved.

Create a Parametric Circle

Two forms that will use the direct attachment method will be circles and ellipses. To parametrically flex and or reliably constrain these elements, we typically need to apply the dimensions directly the forms.

1. From the Application menu, choose **Open > Family**.
2. Select the file named: *Family Seed (Instance Based).rfa* and then click Open.
3. Save the file as: **Circle**.

All solid and void forms in the Family Editor can use circles. So you can perform the following procedure on any kind of form. To keep the exercise simple, we'll use an extrusion, but feel free to practice the steps on other forms as well later on. Since the seed already contains an extrusion we can simple edit it.

Work in the *Ref. Level* floor plan view

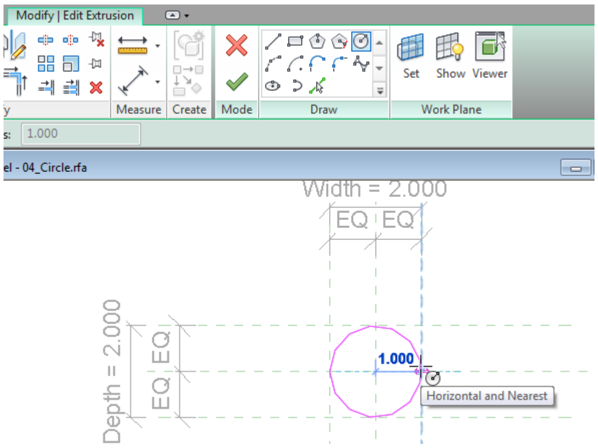
4. Select the extrusion onscreen.

⇨ On the Modify | Extrusion tab, click the Edit Extrusion button.

⇨ Select and delete the four existing sketch lines.

5. On the Draw panel, click the Circle icon.

⇨ Draw the circle centered on the reference planes in the file. Snap the size of the circle to the width and depth defined by the reference planes.

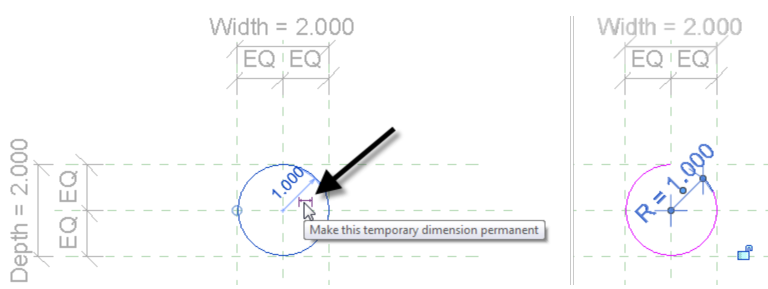


Draw a circle and snap it to the reference planes

Even though we snap the circle to the reference planes, it will not flex when the parameters flex. Try it out if you like. Instead we have to add another parameter to make our intentions known to Revit. We will add a radius dimension and parameter directly to the circle.

6. Select the circle, and then click the icon beneath the dimension to make the temporary dimension permanent. Click the Modify tool to cancel.

7. Label this dimension with a new instance based parameter and call it: **R**.



Create the dimension and a radius parameter

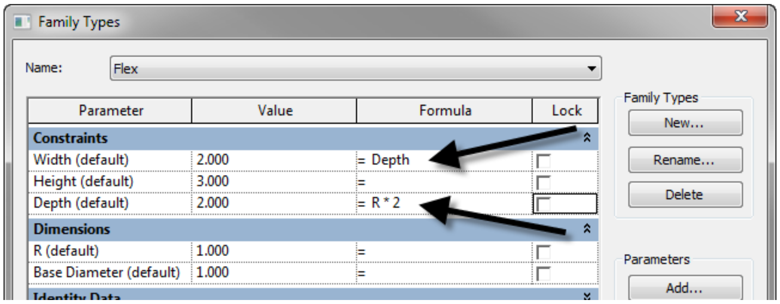
8. Finish the extrusion.

We could flex the extrusion now to be sure that it works, but since we already have the Width and Depth parameters that represent the overall extents of the circle, let's add a few simple formulas in the "Family Types" dialog to tie all three parameters together so they flex in a logical way.

We want Width and Depth to be equal, and R to be half of Width and Depth.

9. Open the "Family Types" dialog.

10. In the Formula column next to Width, Type: **Depth**. For the Formula next to Depth, type: **R * 2**.



Link the three parameters with formulas

> **Note:** We are ignoring the Base Diameter parameter for the time being. If you prefer, you can set the Width and Depth equal to Base Diameter times a multiplier.

11. Flex the family.

The circle should change size as you flex the radius. The width and depth should also flex with the radius.

> **Note:** If you prefer, you can use the Diameter dimension tool on the circle and thereby eliminate the need for the formula based on the R parameter. In this case, the diameter can be set equal to the either the Width or Depth parameter directly.

CATCH UP! You can open a file completed to this point named: *A_Circle.rfa*.

Create a Parametric Ellipse

Creating a parametric ellipse is very similar. For this one, we'll tie the width and depth separately to each ellipse axis. Otherwise, the procedure is nearly the same.

Continue in the previous file or open the catch-up file.

1. Save the file as: **Ellipse**.
2. Open the "Family Types" dialog.

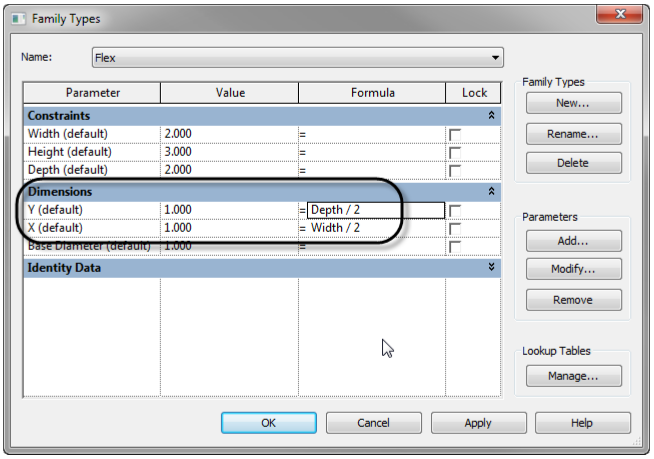
⇨ Clear the formulas.

3. Select the R parameter and then click Modify.

⇨ Rename it to: **X** and then click OK. For the Formula, type: **Width / 2**.

4. Add another instance based Length parameter named: **Y**.

⇨ For the Formula, type: **Depth / 2** and then click OK.



Edit the parameters to prepare them for the ellipse

The parameters are now ready, let's edit the extrusion next.

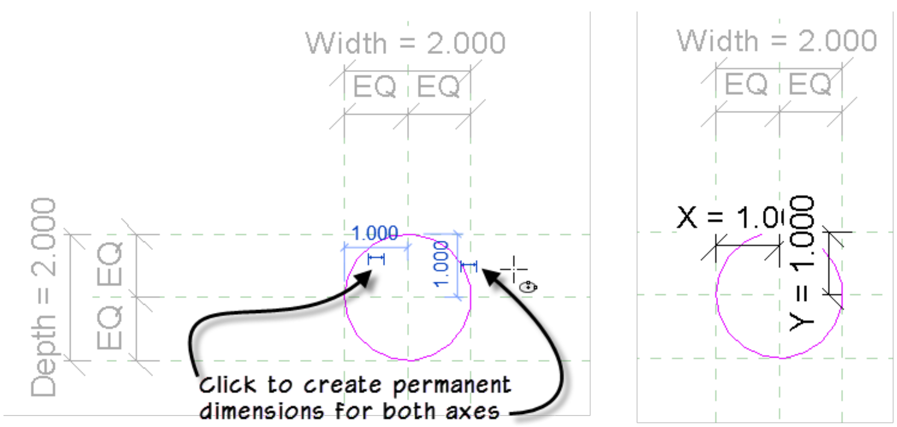
5. Select the circle extrusion and then on the ribbon, click the Edit Extrusion button.

⇨ Select the circle sketch and delete it.

6. On the Draw panel, click the Ellipse icon and then click the center point at the center of the square.

7. Snap the first ellipse axis to the intersection of the Center (Front/Back) and Right reference plane. Snap the other axis to the intersection of the Center (Left/Right) and Back reference planes.

8. Make both dimensions permanent (see the left side of the figure).



Draw an ellipse and make the dimensions for both axes permanent

At the moment, this ellipse looks like a circle. Mathematically, a circle is really just a special case of an ellipse. So you can actually use an ellipse all the time if you prefer and when you need a circle just flex it so that both axes are equal. Let's finish it and flex.

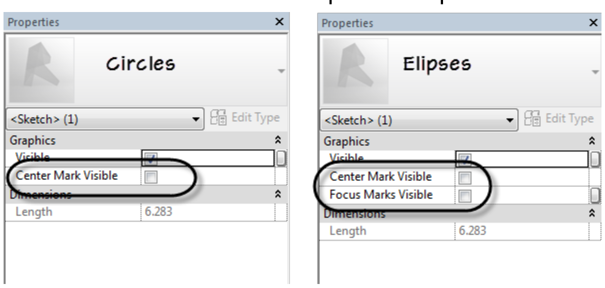
9. Label the vertical dimension with the Y parameter and the horizontal one with the X parameter (see the right side of the figure).
10. Finish the extrusion and then open the "Family Types" dialog and flex.

You should be able to input values for any of the four parameters. If you make the width and depth different, you will get an ellipse. Make them the same to get a circle.

CATCH UP! You can open a file completed to this point named: *A_Ellipse.rfa*.

Center Mark Visible

For curved objects like circles and ellipses you can display the Center Mark. This can be helpful to ensure that the element flexes in the way you expect and intend. Any curved object has this setting including arcs. (For ellipses, you can also display the foci). To display either, look for the checkbox on the Properties palette.



Enable the Center Mark or Focus Marks for Circles and Ellipses on the Properties palette

We'll look at some examples in the next topic.

Automatic Sketch Dimensions

You may have noticed that when you flex Revit families certain parts may be constrained automatically. Sometimes it behaves exactly as you want and expect, but not always. Revit does not always automatically constrain the points you expect. So how do you determine what automatic constraints are applied? The answer is "Automatic Sketch Dimensions." Automatic Sketch Dimensions can be made visible in the Visibility/Graphics Overrides dialog. Use the VG shortcut to open the dialog just like you would in a project. Once there, the Automatic Sketch Dimensions are a sub-component of Dimensions.

Enable Center Mark

1. Open the file: *A_Circle.rfa*.

2. Save the file as: **Circle (Sketch Dims)**.

3. Open the "Family Types" dialog. Clear the formulas and then click OK.

This means that the width and depth parameters are no longer linked together.

4. Select the extrusion onscreen and then on the ribbon, click the Edit Extrusion button.

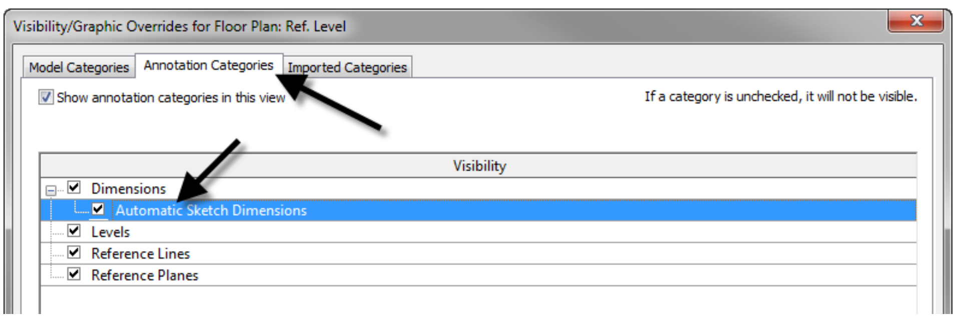
⇨ Select the circle onscreen. On the Properties palette, check Center Mark Visible (shown above in the previous figure).

⇨ Deselect the circle, but do not finish the extrusion yet.

Enable Automatic Sketch Dimensions

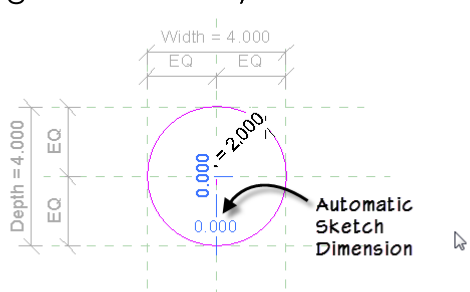
5. On the View tab, click the Visibility/Graphics button (or type VG) and then click the Annotation Categories tab.

⇨ Beneath the Dimensions category, check the Automatic Sketch Dimensions checkbox and then click OK.



Enable the display of Automatic Sketch Dimensions

Notice the two blue dimensions near the center of the circle. These are the Automatic Sketch Dimensions and by default have associated the center of the circle to the intersection of the two reference planes at the origin of this family.



Automatic Sketch Dimensions display in a blue color

6. Return to the "Family Types" dialog and flex the Width and Depth parameters.

   As noted above, without the formulas, these two flex independently.

7. Flex the radius (R) parameter.

   Notice that the circle stays centered on the reference planes.

8. Click OK to dismiss the “Family Types” dialog.

Revit is always looking to establish logical relationships in your families. It does this by placing Automatic Sketch Dimensions in logical locations. In this case, Revit assumed we wanted to keep the center of the circle aligned to the reference planes at the origin (maintain a zero distance in each direction). This may seem pretty logical. For a circle, there aren’t too many other logical assumptions to make, so constraining the center is a good bet. However, this simple example does not tell the whole story. Automatic Sketch Dimensions are not identical to constraints. They will adjust on-the-fly as you edit your family. Let’s try another example.

9. Drag the circle down about **0.500** units (it does not have to be exact, but if you drag more than .5, the Automatic Sketch Dimension will shift).

Notice that the vertical Automatic Sketch Dimension no longer reads zero, but now displays .5 or whatever amount you dragged it. If you flex now, it will keep the center of the circle offset from the reference planes by this amount. Again, this may be your intent and may seem logical, but in some cases it may not be what you wanted.

10. On the ribbon, click the Finish Edit Mode button.
11. Close the family without saving changes.

Let’s return to our Ellipse family and see another example.

12. Open the file named: *A_Ellipse (Sketch Dims).rfa* and tile the windows.
13. Open the “Family Types” dialog.

    Note that the formulas here have already been removed.

14. Flex the Width and Depth parameters.

Notice that this time, the object does not stay centered. Let’s edit the extrusion and take a look at the Automatic Sketch Dimensions. They have already been turned on in this file.

15. Select the extrusion onscreen and then on the ribbon, click the Edit Extrusion button.

Notice that the Automatic Sketch Dimensions are attached to the left side horizontally and to the center vertically. Logical? Perhaps; perhaps not. The point is, that if you do not like the assumptions that Revit makes with the Automatic Sketch Dimensions, you cannot edit them directly. Instead you have to add your own constraints and dimensions to make your intent known to Revit. As soon as you add a dimension or constraint of your own, the Automatic Sketch Dimension will be removed. You cannot simply delete them. You *must* add your own constraints or parameters to override (and therefore make unnecessary) the automatic behavior.

16. Select the ellipse onscreen. On the Properties palette, check Center Mark Visible (shown above in Figure 4.7).
17. Using the Align tool, align and lock the Center Mark to the Center (Left/Right) and Center (Front/Back) reference planes.

Notice that as soon as you lock the alignment, the Automatic Sketch Dimension disappears. The same is true if you add your own permanent dimension; even if you don't lock it. Any dimension, constraint or parameter will have a higher priority than the corresponding Automatic Sketch Dimension and as a consequence will disable it.

18. Flex the ellipse after aligning and locking and note that it now flexes around the intersection at the origin.
19. Finish the Edit Mode, and close the Ellipse file. You do not need to save.

Automatic Sketch Dimensions are an important factor in being successful in the Family Editor. If you are unaware of them, it can make your work in the Family Editor frustrating as you make guesses on how to force Revit to behave the way that you need. There are a few schools of thought on the use of Automatic Sketch Dimensions. The first is to have the goal of eliminating them in all families. To do this, you would need to add your own constraints and parameters at all locations where Automatic Sketch Dimensions appear. This can be easy to accomplish in simple families, but in more complex ones, it can become quite difficult to achieve. So an alternate approach is to replace only those Automatic Sketch Dimensions that run counter to your design intent. In other words, if the family is flexing properly with the Automatic Sketch Dimensions, you can leave them alone. It is really up to you.

TIP: In general, I prefer to eliminate the Automatic Sketch Dimensions where possible, but try not to become consumed by the task. If Revit insists on applying some Automatic Dimensions and they are not preventing my family from flexing properly, then I leave them alone. Your results may vary.

Quarter Round and Half Round (Astragal and Torus)

If you have successfully constrained, parameterized and flexed a circle or ellipse, you might next want to do the same for an arc. Arcs can be very similar to circles, but they do introduce an additional wrinkle; they have endpoints. If your center remains at a fixed location, arcs are pretty easy to control. If the center moves, you have the additional challenge of parameterizing the movement of the center. Let's consider the case where the center is fixed first.

Locking Endpoints

1. Open the file: *Family Seed (Sketch Dims).rfa*.
2. Save the file as: **Arc (Centered)**.

This file already has Automatic Sketch Dimensions enabled. There are no formulas in Family Types. To keep this example quick and easy, we'll build a model line. This way we can focus on just the arc instead of creating an entire form.

Work in the *Ref. Level* plan view.

3. On the Create tab, click the Model Line tool.

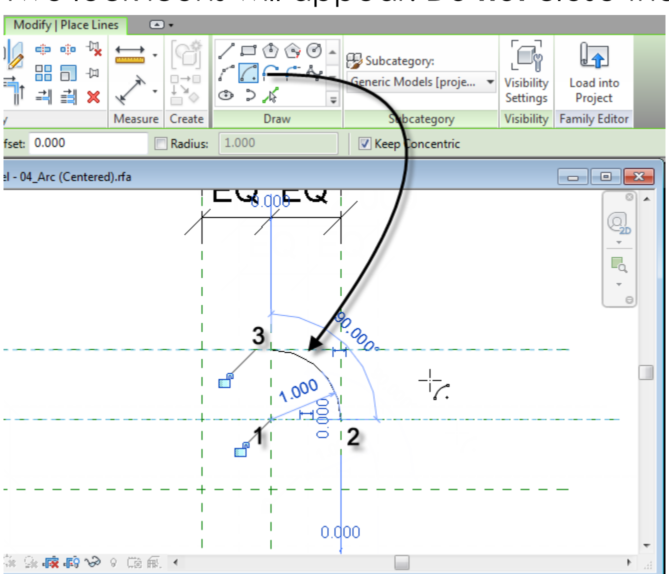
⇨ On the Draw panel, click the Center-ends Arc tool.

4. For the center, click at the intersection of the Center (Left/Right) and Center (Front/Back) Reference planes.

⇨ Snap the first endpoint to the intersection of the Center (Front/Back) and Right Reference planes.

⇨ Snap the other point to the intersection of the Center (Left/Right) and the Back Reference planes.

Two lock icons will appear. **Do not** close them.



Create a Model Line arc centered on the reference planes

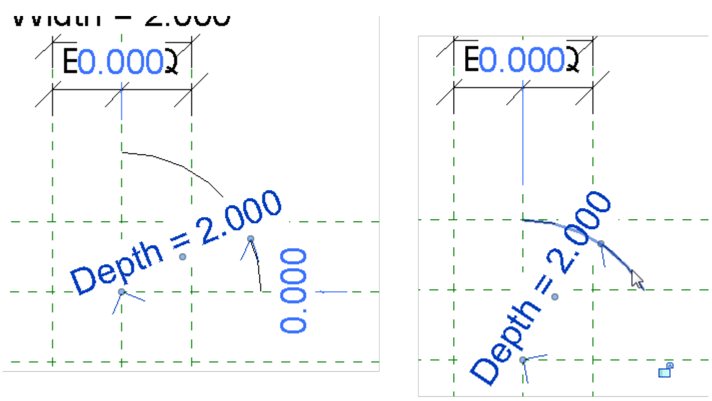
Notice all of the Automatic Sketch Dimension that appear. As noted above, the Automatic Sketch Dimensions try to anticipate how you intend to flex the object. They take into account the kind of geometry that you have. So in this case, since we have an arc, Revit is assuming we want to constrain the center point.

The more dimensions and constraints that you apply, the fewer automatic dimensions will be required. Of course there is a fine line we walk here. If you are not careful, it is easy to get the dreaded "this would over constrain the sketch" error.

5. Add a radius dimension to the arc.

This removes one of the Automatic Sketch Dimensions. Let's add a parameter to the radius so we can flex the size of the curve.

6. Select the radius dimension and label it with the existing Depth parameter (see the left side of the figure).



Relying on automatic dimensions vs. applying locks

Notice how the center remains fixed as the arc's radius increases. Revit tends to favor the center point location.

7. Delete the arc and then add it again. This time, **close the lock icons**.

8. Add the radius dimension and again label it with the Depth parameter (see the right side of the previous figure).

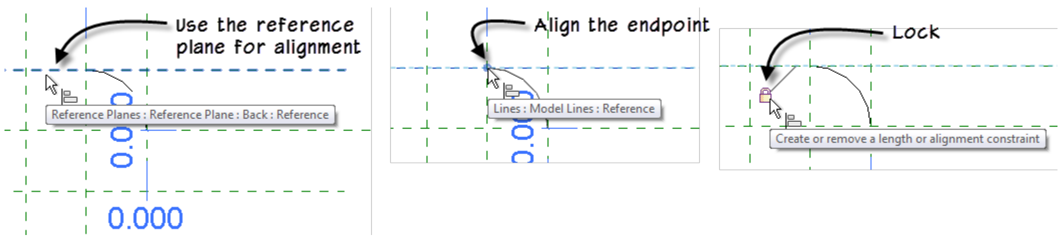
Notice that this time, since we applied the locks, the center ends up moving when you flex it. Chances are, the first behavior was a little more expected than the second one. But both had their issues.

I tend not to use the locks that Revit displays when drawing a shape. I use the Align tool instead to be more precise about where and what I am locking. With the Align tool, I see each lock being applied one at a time. I find this much more predictable then relying on the assumptions that Revit makes and offers.

9. Delete the arc and then add it one more time. Do not close the lock icons.
10. On the Modify tab, click the Align tool.

⇨ For the reference for alignment, click the back reference plane.

⇨ For the entity to align, click the endpoint of the arc. Lock it.



Align and lock endpoints

11. Repeat on the Center (Left/Right) reference plane and the same endpoint and again in both directions for the other endpoint. (This will be four alignments total).

Notice that as you finish aligning in all four directions, all of the Automatic Sketch Dimensions will disappear. The endpoints of the arc will now remain at the intersections of the reference planes.

12. Add the radius and label it with the Depth once again.

Were the results as you expected? This time, the center point moved. Considering that we just locked the endpoints to the intersections of the reference planes, there is really nothing else that could flex. I suspect however, that this may not be what you expected. It is likely that we would want the center to stay at the intersection of the two central reference planes and instead for the width and depth to flex. We can achieve this with a simple formula as we did in the last chapter.

13. Undo the application of the label to the radius, but keep the radius dimension.
14. Label it with a new instance parameter instead and call it: **R**.
15. Open the "Family Types" dialog and add a formula to both Width and Depth: **R * 2**.
16. Flex the family.

Notice that now when you flex the radius, it stays confined to the upper right quadrant defined by the reference planes. So depending on the specific behavior you require, the precise approach you use is a matter of preference.

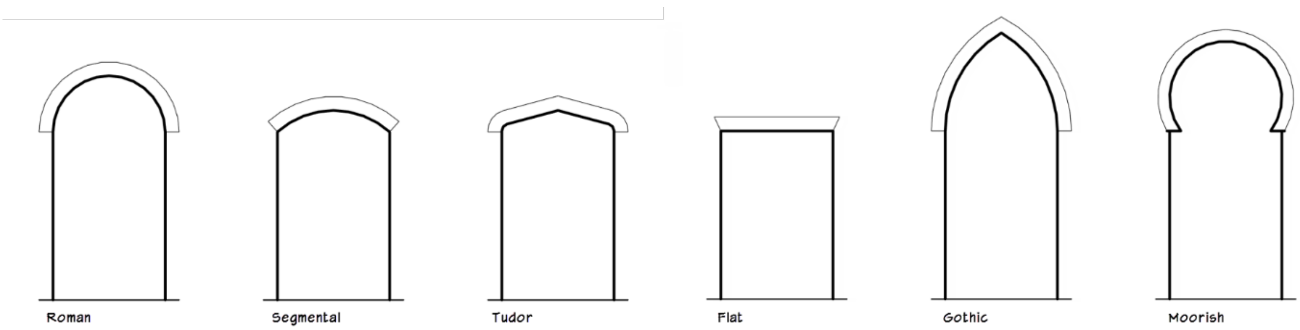
You can also try dragging the arc's endpoints to produce a different angle. 90° increments should be very stable. But even other angles should work well since the main controlling parameter here is the radius. To add even more stability, turn on the center mark (see above) and then align and lock the center in both directions to the reference planes. With both center and endpoints locked, you don't actually need the radius dimension. The width and depth and flex the arc and the alignments will be maintained. Feel free to experiment further before continuing.

You can use the previous techniques to create an astragal, a torus, quarter-round and half-round shapes. You can also use the half-round shape to create Roman arches.

CATCH UP! You can open a file completed to this point named: *A_Arc (Centered).rfa*.

Arches

The previous examples all had the center point located at and constrained to two reference planes. But what do you do when the center of your arc does not land on such a convenient location? This is the case for many types of arches. Look around the town you live and you are bound to find many different types of arches used in the various buildings.



A selection of arches

Create a Parametric Segmental Arch

We could do an entire lab or maybe several on the subject of arches alone. I won't have time and space to do all of them, but let's consider a few common shapes. Starting with a Roman arch. To build a Roman arch you can simple use the techniques covered above. A Roman arch is a semicircle, so the techniques discussed above will work well. Just turn on the center mark, align and lock it to reference planes and then lock the endpoints of your arc on both sides as well. You can flex its shape with standard width and depth parameters (although you might want to think of the depth as height in the case of an arch) and add a radius parameter if desired.

If you have one of the other forms, you might have a tougher time making the arch behave as it flexes. For example, let's consider a Segmental arch. This one has an eyebrow shape and as its name implies is a segment of a circle. The main issue with such an arch is that the center point moves as the arch flexes. So since the techniques above relied on locking the center point, they will not yield good results. Let's look at what must be done to make a segmental arch behave. Here we must break our rule about keeping constraints applied only to references again. The key to success is applying the labeled dimension directly to the geometry of the curve (like the circle and ellipse above) rather than the traditional approach of dimensioning the reference planes and then letting them flex the geometry.

> Note: it is possible to stay true to the traditional approach and dimension only references. It would require formulas to calculate where the references need to be to give the correct curve. We will see examples of this below.

One last preliminary point. The following techniques work with any family template, but I think that in many cases, you will find a face-based template a good choice for an arch. This makes it easy to place them on or in existing walls.

1. From the Application menu, choose **Open > Family**.
2. Select the file named: *Family Seed (Face Based).rfa* and then click Open.
3. Save the file as: **Segmental Arch**.

Our seed families all have the two default reference planes at the center. In this case, let's use the Center (Front/Back) reference plane (the horizontal one in the middle) as the spring line of the arch, so we'll need to delete the bottommost one.

4. Delete bottom reference plane. Also delete the extrusion.

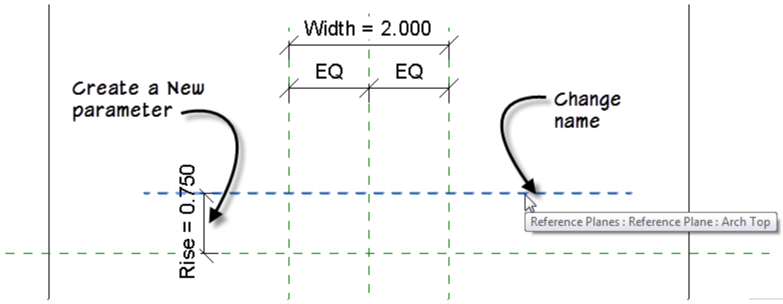
When you do this, you will be left with a single unlabeled vertical dimension.

5. Select the vertical dimension.

⇨ On the Options Bar, label it with a new instance parameter and call it: **Rise**.

6. Select the top reference plane and on the Properties palette, rename it to **Arch Top**.

7. Flex the family to ensure everything is working. Set the Rise to: **0.750**.



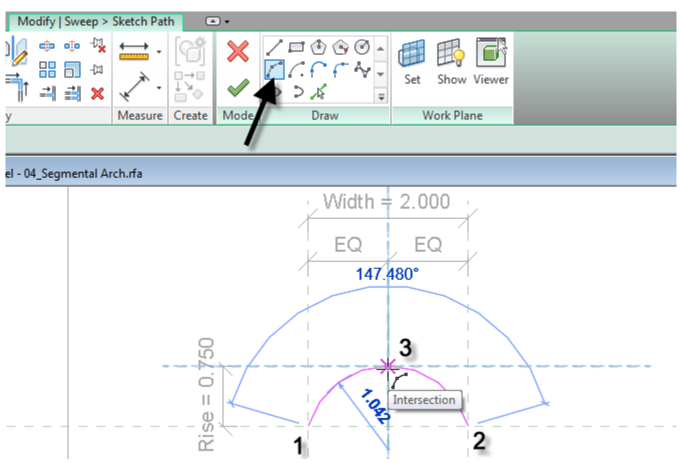
Configure the reference planes

With this framework in place, we are ready to build our geometry. While it is possible to use an extrusion here, it takes a little more effort to constrain it. This is because an extrusion would require two parallel curves. Instead, let's create a sweep. It is the path of the sweep that will form the arch shape.

8. On the Create tab, click the Sweep button. Choose the Sketch Path option on the ribbon.

9. From the Draw panel, choose the Start-End-Radius arc.

10. Snap the start and end points to the intersections where the left and right reference planes cross the Center (Front/Back) reference plane (see points 1 and 2).



Create the path of the sweep with a start-end-radius arc

⇨ Snap to the intersection of the Top and Center (Left/Right) reference planes for the radius (see point 3).

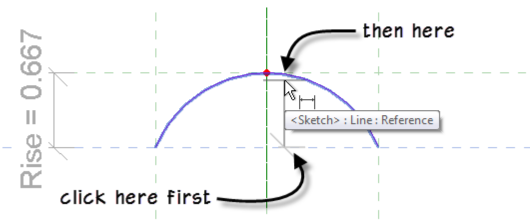
11. Using the Align tool, align and lock the start and end points of the arc to the reference planes in both directions.

Align and lock the endpoints of the arc to the reference planes in both directions

Aligning and locking works fine for the endpoints. Unfortunately we cannot align and lock the midpoint of the arc. But just the same we always want the midpoint of the arc to stay with the top reference plane. To do this, we'll dimension the sketch line directly.

12. Create an aligned dimension. For the first witness line, click the Center (Front/Back) reference plane.

⇨ Click directly on the arc next to add the second witness line.



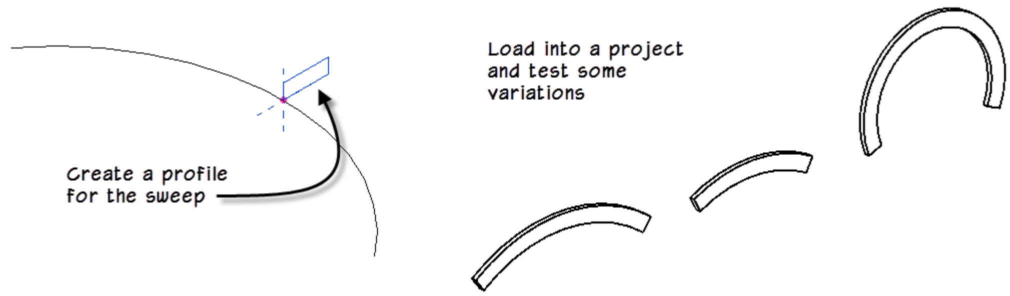
Dimension the arc directly

13. Label this dimension with the existing Rise parameter.

⇨ Flex the family.

That's it! The arch can now have nearly any rise value you like to create segmental arches of various shapes and proportions. If the value of Rise is equal to half of the Width parameter, you will get a Roman arch. If you go larger than this it will make an arch that has a Moorish shape. An arc must be curved, so you cannot use a value of zero. Therefore, to make a jack arch simply make a separate family with a rectangular form.

To complete the arch, just edit the sketch for the sweep profile and create a simple rectangle. If you prefer, you can use a more complex shape, but a rectangle will do the trick for now. We'll do a more complex arch later on.



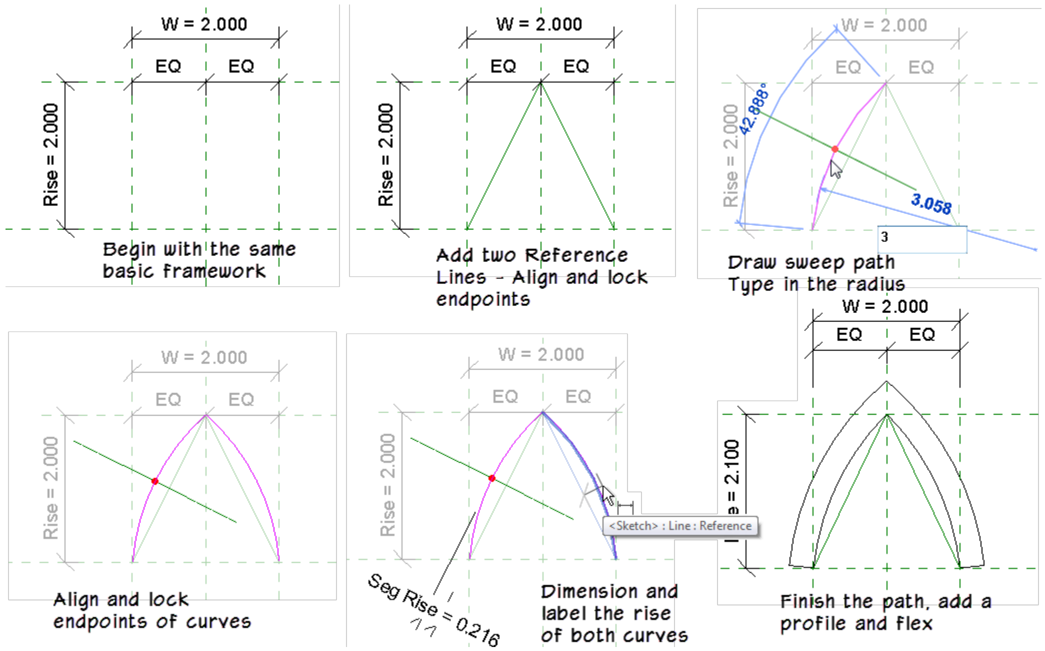
Complete the arch family and load it into a project

CATCH UP! You can open a file completed to this point named: *A_Segmental Arch.rfa*.

There are many related examples we could do next, but time does not permit us to do them all in the live lab. Therefore, the next few topics are provided here for you to work through on your own following the lab.

Gothic Arch

As noted, the previous family will yield many of the common arch forms, but not all. To create some of the others, we can leverage the same basic concepts. For example, a gothic shaped arch can be achieved by taking two segmental arch rigs and placing them in a triangular construct. I won't detail all of the steps here, but instead simply describe the overall process.



A gothic arch can be formed using essentially two segmental arches

Start with the same seed family and set it up the same way. You can even save the Segmental Arch family as a new name and just delete the sweep. Usually a gothic arch will have a higher rise, so flex the rise parameter to at least the same value as Width (shown as W in the figure). Draw two reference lines (*not* reference planes) forming a triangle. This will form the spring lines of each side of the arch. Use the Align tool to align and lock the endpoints of the reference lines to the intersections of the reference planes. Make sure you align and lock all four endpoints in both directions (eight total). It is a good idea to flex at this stage.

> **Note:** Sometimes when aligning and locking, Revit will complain that locking will over constrain the sketch. If you see this message, just click Cancel. No need to pursue it further as this message should be indicating that no further constraints are necessary at the location. Remember that having Automatic Sketch Dimensions displayed during this task can be very helpful in determining if you need to continue aligning and locking.

Begin your sweep and choose the Sketch Path option. Draw two Start-End-Radius arcs. The endpoints should snap to the ends of the reference lines. The radius can be anything, but they should be the same for both arcs. You can begin eyeballing the curve, and before clicking, type in a whole number value based on the temporary dimension displayed. This makes it easy to get the same radius on both sides. I used a value of **3** in the figure. Align and lock the endpoints of the curves to the reference planes.

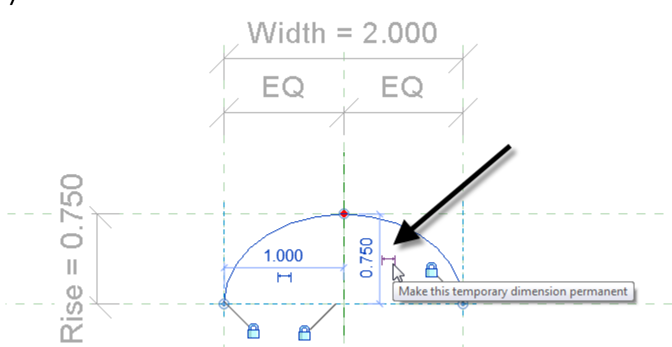
Create a new dimension between the *reference line* and the rise of each arch. Make sure to set the first witness line at the reference line, then click the curve. Label each of these with a new parameter called: **Seg Rise**. Flex and then Finish the path.

Finally, sketch the profile. You can also load in a profile family for the profile if you prefer. (We will explore profiles below). Flex the completed form. Be careful in flexing as certain combinations of Rise, Seg Rise and Width will cause it to fail, but it should work well for many combinations.

CATCH UP! You can open a file completed to this point named: *C_Gothic Arch.rfa*.

Elliptical Arches

If you would rather use an ellipse to form the path of your sweep, much of the process remains unchanged. You draw the ellipse using the partial ellipse shape on the Draw Panel, align and lock its endpoints just like the arc. However, when you try to dimension the elliptical arc, Revit will stubbornly refuse to highlight it like it did for the arc. So instead of placing the dimension with the Aligned Dimension tool, the trick is to first select the partial ellipse onscreen, then click the small "Make this temporary dimension permanent" icon. This will make the dimension for you and then you can label it. All else will work the same way.



Make the temporary dimension permanent to create a flexible elliptical arch

CATCH UP! You can open a file completed to this point named: *C_Elliptical Arch.rfa*.

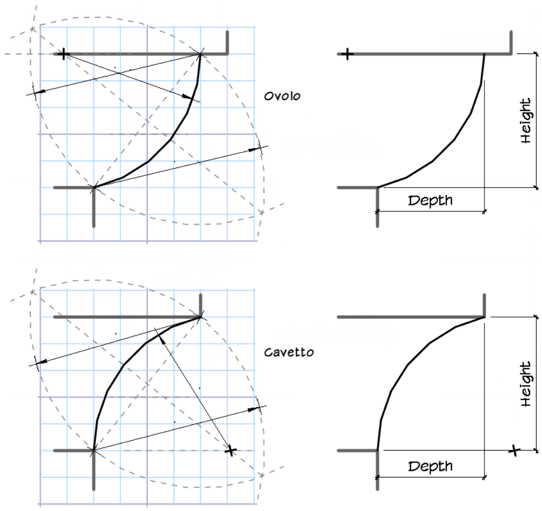
This concludes the bonus material. There are two lessons left in the paper. We will attempt to complete each of these in the live lab.

Profile Families

In this example, we will build a profile family. There are several advantages of profile families. Perhaps the most important benefit of profile families is that we can use them in several forms and families. For the task of creating complex flexible curved forms, it is much easier to build a profile family and get the shape flexing properly, and then apply it to any 3D forms required. The remaining examples will use profile families. Some challenges do exist. For example, profile families can contain reference planes, but *not* reference lines. Therefore if you tried to build something like the Gothic arch above you would not be able to use the same approach. The same would be true if you needed parametrically controlled rotation (which typically uses reference lines).
There are some acceptable solutions. For example, you can use trigonometry to derive X and Y coordinates from any angle. This is very effective and very stable, however, trig can be challenging and introducing many formulas can have a detrimental effect on overall performance. Another solution that we will explore is using a nested rig family. Nesting can also be detrimental to performance, so you will have to consider each use case carefully. We'll see several examples below.

Ovolo

Let's start with a practical example and build some common molding shapes. In his excellent book on classical architecture: *The Classical Orders of Architecture* Robert Chitham details the construction of many common architectural moldings. Below I have illustrated the first two moldings we will consider: the ovolo and cavetto. Both use a single curve, but constructing them introduces a challenge (an offset center point) that once solved, will help us create the more complex compound curves that follow.

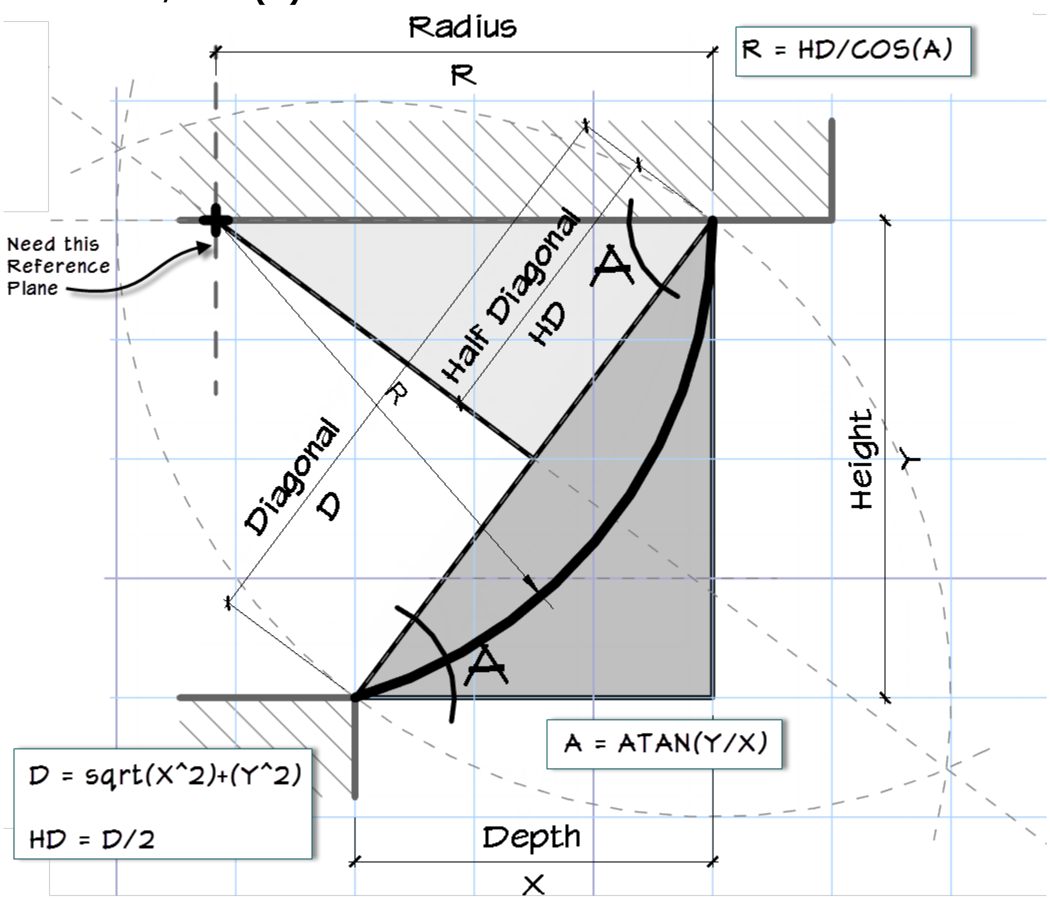


Constructing Ovolo and Cavetto profiles

Consider the shapes shown in the figure. The left side comes from Chitham and shows how to construct the curves using traditional drafting tools. The most likely way to describe the size of a molding like this and therefore the most likely way to flex this curve would be to ask for the depth and the height of the curved portion of the molding. This is shown on the right side. As you can see in the image, this width and depth of the curve portion are normally not equal, (A quick trip to your local lumber yard will allow you to verify this) meaning that the construction axes are along angled paths and as the dimensions flex. This also means that the center point is not always in the same location. Typically, reference lines are the best choice for controlling angles in families. However, as noted, reference lines are not available in profile families, so we could abandon our use of profiles and resort to building each molding with its own integral sketch. This would not be ideal since we would lose all benefits of using profile families. One of the most important being that they can be reused in multiple families. Therefore, if we can define them once in a good flexible profile family, we can then reuse them in almost unlimited ways in other content. So to build it as a profile family, we will have to resort to other techniques to control the shape of this family and ensure that it flexes properly.

Using trigonometry to model traditional molding forms

As noted at the start of this topic, there are two viable options here. In the examples that follow, we'll look at both. Let's start with trigonometry. Depending on the complexity, the trig might be simple or complex. In this case, we have a simple case of similar triangles. You will be asking your user to input the desired height and depth of the molding. We will call these X and Y in the profile family. When you make a triangle from X and Y, you can easily derive the hypotenuse which we will call **D** (for Diagonal). Using trig, and these values, we can easily derive one of the angles which we will call **A**. Since we know all three sides, we have lots of options to choose from. For this example I went with the ATAN function performed on X and Y: **ATAN(Y/X)** to arrive at A. This new angle, coupled with half of D (we'll call this **HD**) can be applied to the second triangle. Finally with this information, the COS function will give us the length of the radius: **HD / COS(A)**.



Applying trigonometry to locate the required reference planes

Provided with the dataset is a seed file for a profile family. It has the two default center reference planes positioned at the lower left corner. Two additional reference planes named: X Mid and X Max, and Y Mid and Y Max are placed above and to the right of these. They are dimensioned and constrained already. To this framework, we need to add a vertical reference plane to left for the radius location indicated in the figure. Our formulas will help us locate it.

1. From the Application menu, choose **Open > Family**, select the file named: *Family Seed (Profile).rfa* and then click Open.

⇨ Save the file as: **Ovolo Profile**.

2. To the left of the Center (Left/Right) reference plane, draw a vertical reference plane. Name it **Radius**.

⇨ Add a dimension between the X Max reference plane and Radius.

⇨ Dimension and Label it with a new Type parameter called **R**.

3. Open the "Family Types" dialog.

4. Using the Add button on the right, create two new Type-based Length parameters: **D** and **HD**. Also create one Type-based Angle parameter called **A**.

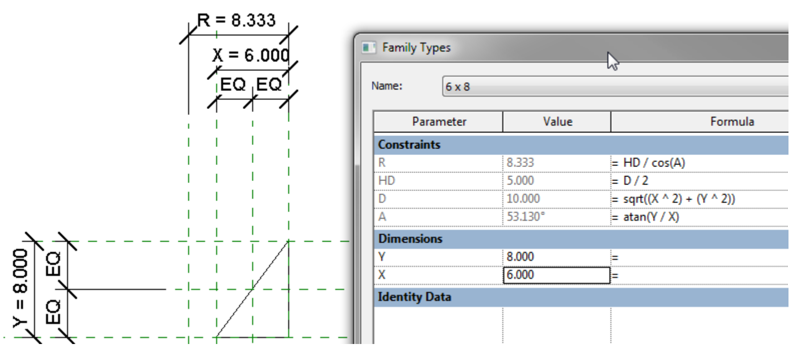
   Group all of these under Constraints.

5. Input the formulas as follows:

TABLE 1

Parameter	Formula
D	sqrt((X ^ 2) + (Y ^ 2))
HD	D/2
A	atan(Y / X)
R	HD / cos(A)

6. Flex the family by creating a few Family Types.



Create the parameters and Family Types

There is a simple triangle in this family already just to facilitate flexing. We can now delete this and draw the actual profile shape that we need.

7. Delete the triangle onscreen.

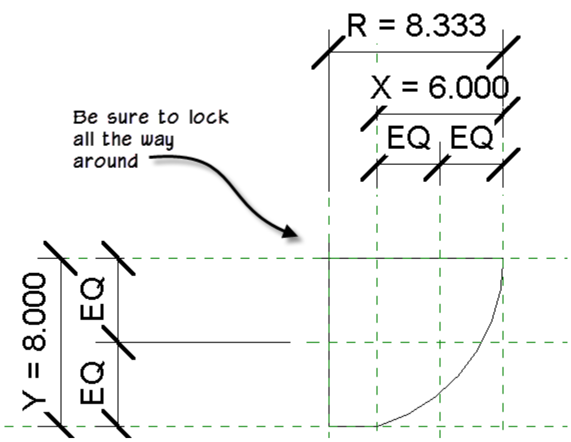
CATCH UP! You can open a file completed to this point named: *A_Ovolo Profile (Ref Planes).rfa*.

8. On the Create tab, click the Line tool.

9. Draw a Center-ends arc with the center at the intersection of the Y Max and Radius reference planes.

   Be sure to turn on the center mark, align and lock the center in both directions as well as the endpoints.

10. Draw vertical and horizontal lines locked to the reference planes for the leftmost edge and the top and bottom.



Create the profile lines

CATCH UP! You can open a file completed to this point named: *A_Ovolo Profile.rfa*.

11. Flex the completed version when done.

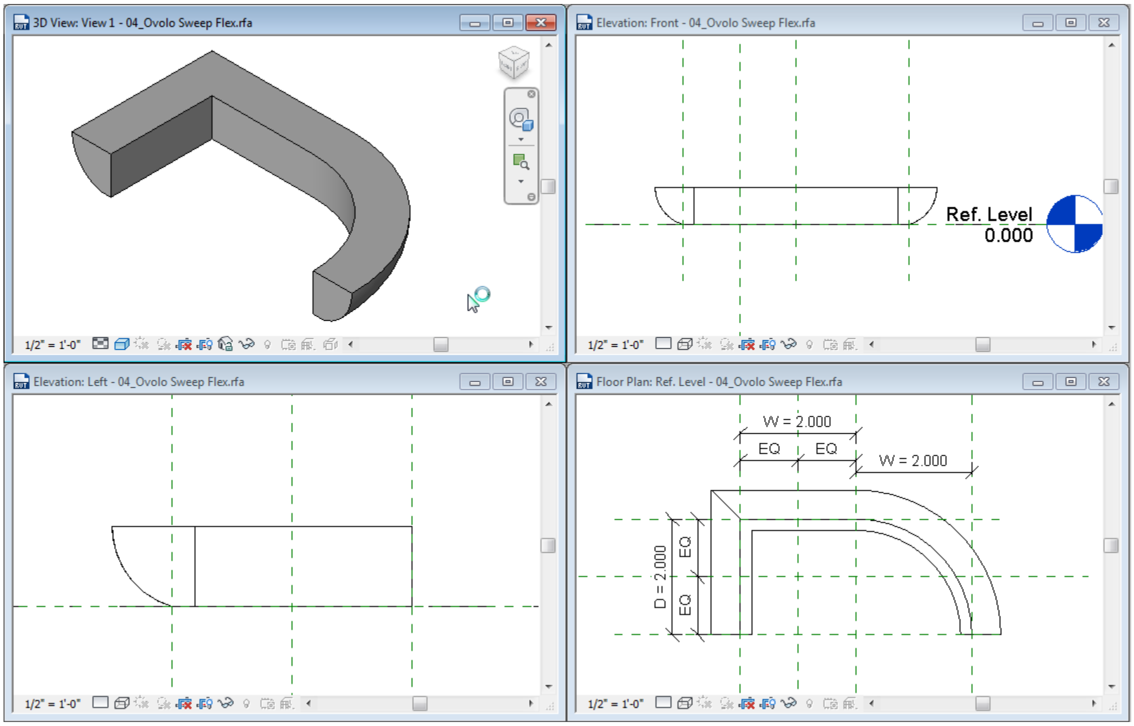
You can load this profile into any project or family now and use it shape anything from sweeps, to wall profiles to railings. A simple family file is provided here to test it out.

12. Open *A_Ovolo Sweep Flex.rfa*.

A version of the profile family is already loaded, but if you prefer, you can load your version instead.

13. Select the sweep and edit it. On the ribbon, click the Select Profile button and then from the drop-down list, choose one of the types you previously defined.

14. Finish the sweep and flex the family.



Apply the profile to the sweep form

This "flex" file that we have open is like a sandbox file. We are using it just to test out the profile and ensure that it works as expected. The family has two simple parameters: W and D. If you flex them, you will see the shape of the sweep path adjust. You can see by doing this how your profile follows along both straight and curved path edges. If you want to change the profile, edit the sweep again and pick a different type from the list. We can even edit the type of the nested profile family and link up the X and Y parameters with driving parameters here in the host family. I will not go into the steps at this time, but feel free to experiment with this if you like.

15. Close both family files.

Using a nested rig in a Profile family

For various reasons, you may wish to avoid formulas. Some folks find them cumbersome and complex. (Some folks just prefer not to revisit High School math…) They also can impact performance if there are many of them in use. An alternative is available to formulas. We can build a separate "rig" family and then nest it into our profile family. This will help us overcome the limitation of not having reference lines to control rotation.

> Note: The following technique works in the traditional family editor only. The conceptual massing enviornment (CME) cannot use any 2D families. So profile and detail item families are not supported by the CME. However, you can perform the same basic procedure by building a generic model family in place of the profile family and within this generic model, using model and reference lines instead of detail items for the rig. The remaining concepts would apply.

Cavetto

In this example, we will look at the cavetto curve. It is exactly the same construction as the ovolo with the curve reversed. So the radius reference plane needs to be on the right with the center of the curve at the intersection of the radius and Center (Front/Back) reference planes. In all other ways, we could use the same strategy and formulas (from Part A). But in this example, we'll look at an alternative: using a "**rig**" on which to build the curve form instead.

The starting family in this case uses the same seed we used for the ovolo profile above. But it has one extra reference plane to control the depth of the form.

16. Open the file: *A_Cavetto Profile (Rig)_Start.rfa*.
17. Save the file as: **Cavetto Profile**.

We'll also need our rig family. The rig is a detail item family and will define an adjustable angle. Normally you would use reference lines to control angles parametrically, but we cannot use reference lines in profile families. And while we can use reference planes, they do not work well in controlling angles. So instead of reference lines, we will simply draw lines instead. However, if we draw the lines directly in the profile family, they will be seen by Revit as part of the profile. If we instead draw our guide lines in a detail Item family they can be used for our framework or "rig" and not be seen as part of the profile.

18. Open the file named: *Family Seed (Detail).rfa*.
19. Save the file as: **Single Curve Rig**.

This seed was created from the *Detail Item.rte* template. Reference planes were added and a few dimensions and parameters to save time.

20. On the Create tab, click the Line tool. Snap the first point to the intersection of the Left and Front reference planes and the second point to the Right and Back reference planes.

⇨ Align and lock the endpoints to the reference planes in both directions on each end.

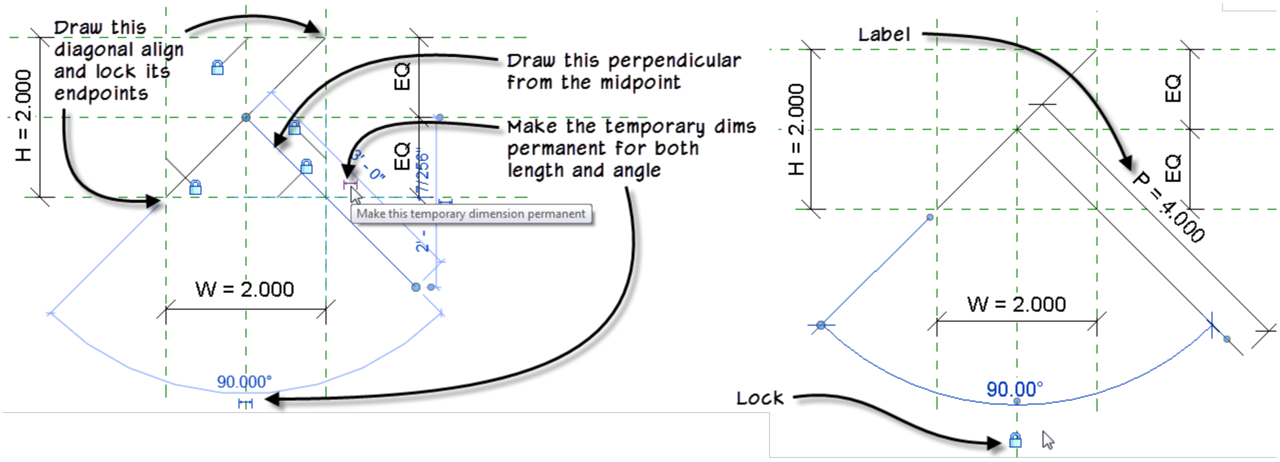
⇨ Flex to be sure it adjusts with the width and depth.

21. Draw a second line starting at the midpoint of and perpendicular to the first line. Make it approximately the same length as the other one.

⇨ Select the line and then click the small "Make this temporary dimension permanent" icon for both the length and angle dimensions (see the left side of the next figure).

22. Select the new linear dimension and label it with the parameter called **P** (already in the file).

⇨ Lock the angle dimension (see the right side of the figure).



Build the rig in a Detail Item family

A subcategory called Guide Lines set to a light blue color is already provided in this file. This will help the geometry stand out.

23. Select both lines and on the Properties palette, change the Subcategory to **Guide Lines**.

24. Save the family.

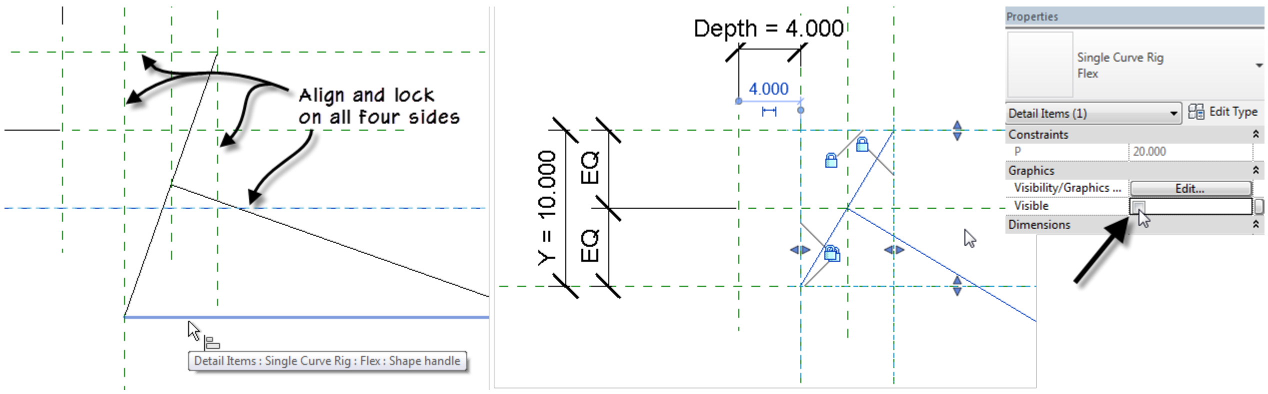
CATCH UP! You can open a file completed to this point named: *B_Single Curve Rig.rfa*.

25. Click the Load into Project button.

⇨ Click to place it onscreen. Don't snap to anything.

⇨ Align and lock it on all four sides.

For each alignment, first click one of the reference planes in the host family, and then click the nested shape handle edge in the detail item family. Use TAB if necessary to get **shape handle** each time.



Nest in the rig, align and lock it on all four sides

26. Flex the family.

Notice that the nested detail item family changes shape with the host family. The diagonal line stays aligned with the flexing reference planes and the perpendicular line remains perpendicular. The intersection of the perpendicular line and the Center (Front/Back) reference plane is the center of our curve for this profile.

27. Select detail item and on the Properties palette, uncheck the Visible checkbox (see the right side of the previous figure).

This makes the rig invisible in all families that use the profile; we will only see it here were it is needed.

CATCH UP! You can open a file completed to this point named: *B_Cavetto Profile (Rig)_A.rfa*.

With the rig in place, we draw the lines that make the profile shape next.

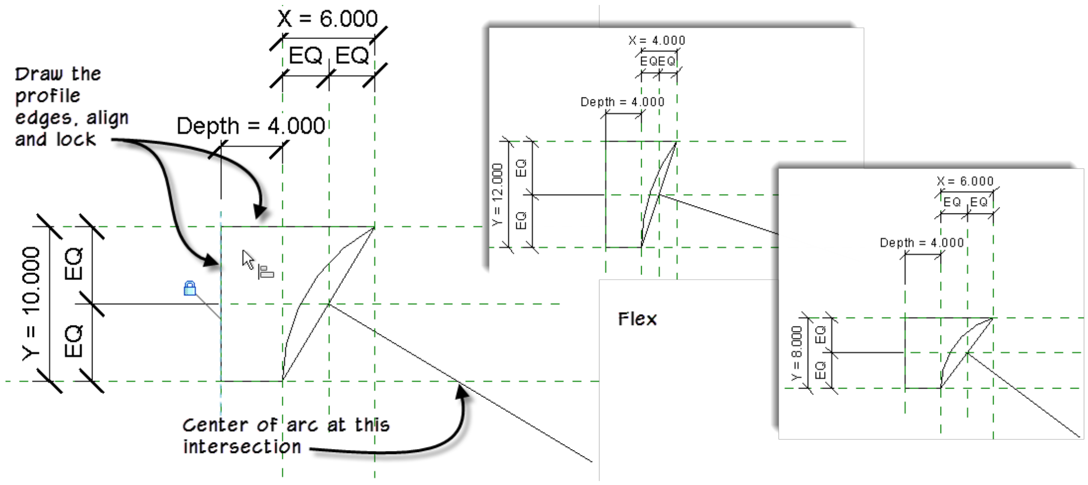
28. Draw the curve as we did above for the ovolo. Use the intersection of the perpendicular line and the Center (Front/Back) reference plane as the center.

29. Align and lock the endpoints of the curve.

> **Important:** Pay attention to the Status line as you align. You want to align to the reference planes in the profile family, *not* the references or shape handles in the nested family. Use TAB as necessary.

Notice that after you align and lock the endpoints in both directions, that all of the Automatic Sketch Dimensions will disappear. As such, you do not have to turn on the center mark or align and lock the center. However, it is a nice extra precaution. Furthermore, even though in most previous examples we aligned in the X and Y direction, you can actually enable the center mark and align it horizontally to the Center (Front/Back) reference plane and then align it again to the diagonal line in the nested family.

30. On the left, top and bottom, draw straight lines. Align and lock anywhere an Automatic Sketch Dimension appears.



Create the profile shape and align and lock as necessary

CATCH UP! You can open a file completed to this point named: *B_Cavetto Profile (Rig).rfa*.

31. Flex the family and then load it into the *B_Cavetto Sweep Flex.rfa* file to test it out in a sweep.

There are pros and cons to this approach and the formula approach that we explored at the end of Part A. If you get the formulas right, using formulas and trigonometry is very stable. But as noted, it can impede performance. Using the rig is a clever work around to some of the built-in limitations. It can be quite stable as well, but you have to be careful about which points you align and lock and make sure you do not inadvertently create bad references that prevent the families from flexing. You are encouraged to try both approaches and compare and contrast your results. With these techniques in hand, let's move on to more complex curves: the Cyma and Cyma Reversa.

To be continued…

See you back in here for **Part B**. Thank You!

Want to learn more?

I have a video course related to the material in this lab, visit: www.lynda.com/paulaubin and check out: **Revit: Family Curves and Formulas**

And of course there is my book: ***Renaissance Revit: Creating Classical Architecture with Modern Software*** as well.

Session 15

LAB: Throw Your Family a Curve- Part B

Paul F. Aubin: www.paulaubin.com

Class Description

Have you ever tried to control the shape of a curved form parametrically in the Family Editor? If so you have no doubt discovered that flexing them sometimes throws you a curve ball. In this session, we will explore several techniques to tame your unruly parametric curves. We will look at examples of circles, arcs, quarter round, half round, arches and even throw in some splines. We will look at both simple and compound curves. We will work primarily in the traditional Family Editor but will discuss how techniques apply to the massing Family Editor environment as well. We will explore curvature, rotation and throw in some trigonometry for good measure. After this session, I cannot guarantee that you will never have another misbehaving curve in your family content, but what I can promise is that you come away with several useful tools to help you tame them when curve mischief strikes!

About the Speaker:

Paul F. Aubin is the author of many Revit book titles including the widely acclaimed: *The Aubin Academy Mastering Series*, his all new *Renaissance Revit* and Revit video training at www.lynda.com/paulaubin. Paul is an independent architectural consultant providing Revit® Architecture implementation, training, and support services. Paul's involvement in the architectural profession spans over 20 years, with experience in design, production, CAD management, mentoring, coaching and training. He is an active member of the Autodesk user community, an Expert Elite and is a high-rated repeat speaker at Autodesk University, Revit Technology Conference and the BIM Workshops. His diverse experience in architectural firms, as a CAD manager, and as an educator gives his writing and his classroom instruction a fresh and credible focus. Paul is an associate member of the American Institute of Architects. He lives right here in Chicago with his wife and three children.

Contact Paul directly from the contact form at his website: **www.paulaubin.com**

Introduction

This is the second part of a two-part lab. Hopefully you joined us for Part A because this one picks up right where the other left off. The subject of this lab is parametrically controlled curves in the Revit family editor. We'll stay in the traditional family editor, but nearly all techniques transfer over.

Datasets

The dataset files for this lab are provided on the local machines. All starting files for each exercise are provided and several "catch-up" files are also provided. If you get behind, please load the catch-up file to keep going with us. You can download a copy of the dataset here:

https://app.box.com/s/ds4jkd92eno4vdnhwtmo

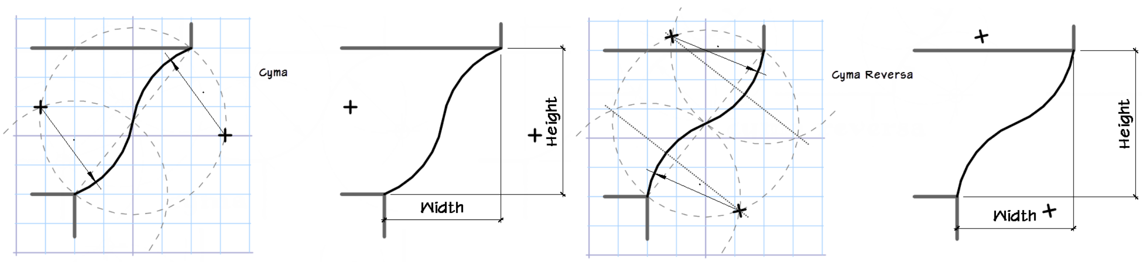
Complex Curves and Compound Curves

All of the examples covered in Part A were single curves; in other words, there has only been one curve that we were trying to flex. In such cases, if you ensure that you constrain and/or parameterize the key geometric aspects of the curve, you will usually get good results. For example, with a circle or arc, if you constrain the center and radius, it will usually flex properly. With an ellipse, the center and axes usually give good results. However, as the forms that you wish to flex become more complex in shape, sometimes even constraining the center and radius will not be enough.

Consider situations where there is more than one arc segment making up a compound curve. Or even situations with a curve meeting a straight line at a tangent. There are endless possible examples. In this topic we'll consider a few of the more common examples. In similar fashion to the examples above, the key is going to be carefully constraining the curves so that you remove any ambiguity. **You want to make it very clear to Revit what your intentions are.** If you do this, everything *should* flex properly.

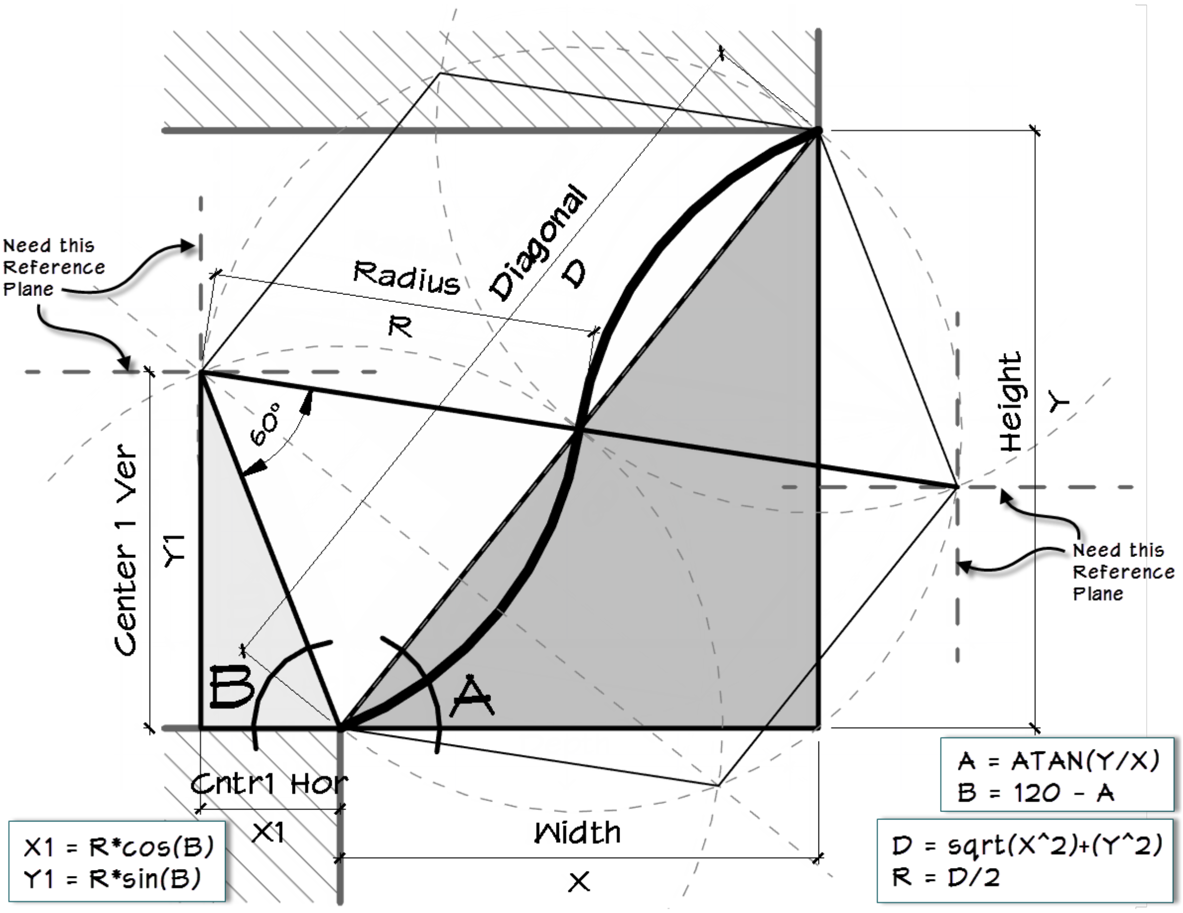
Cyma

Now that we have the basic techniques we need in hand, let's try our first compound curve. The next figure shows two very typical molding profiles: the Cyma and Cyma Reversa. Like we saw in Part A, the left comes from Chitham and shows how the profiles are constructed traditionally, a circle is created whose diameter matches the diagonal between the X and Y (Height and Depth). This circle is intersected with two arcs of the same radius which, when intersecting perpendiculars are drawn, creates four equal segments along the diagonal. The points where the arcs intersect the circle are the centers for the arcs used to create the Cyma. Once again, our user inputs will be Height and Width of the molding.



Constructing Cyma and Cyma Reversa profiles

It just so happens that this construct also creates a regular hexagon whose vertices intersect the circle at the same locations. Compare the gray dashed construction arcs with the superimposed hexagon.



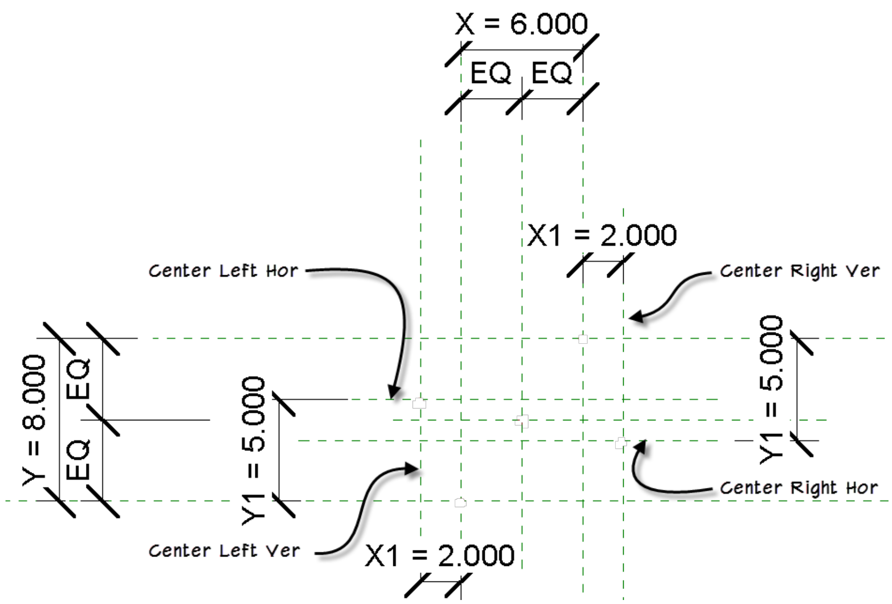
Applying trigonometry to locate the required Cyma reference planes

A regular hexagon can be divided into six equilateral triangles. The sides of these triangles are each equal to half the length of the diagonal (between the Height and Width). This distance (the side of the equilateral triangle) is the radius of the arcs used for the cyma and cyma reversa profiles. We can use trigonometry or nested detail components to construct this profile family. The trigonometric formulas are shown in the figure and in the table below.

The basic idea is that the user input is the Height (Y) and Width (X). This is used to determine the angle of the diagonal (D) which is in turn used to locate the center point of the two arcs of the compound curve and their radii. We'll start with a file based on the ovolo example above.

1. Open the file: *B_Cyma Profile_Start.rfa*.
2. Save the file as: **Cyma Profile**.

Some of the work has been done here already. To create this file, a copy of the Ovolo file (created in Part A of this lab) was saved. This means that some of the reference planes and some of the formulas were already in place. This includes the insertion point at the lower left corner, the X Mid, X Max, Y Mid and Y Max reference planes, and the parameters X, Y, A, R and D. HD was not needed and has been removed. Additional reference planes have been added: Center Right Ver, Center Right Hor, Center Left Ver and Center Left Hor, the parameters X1 and Y1 are applied to the reference planes already. These additional parameters listed in the table below.



Starting family contains the reference planes and parameters

The formulas have not yet been added except those that came from the ovolo family. We will do this task now.

3. Open the "Family Types" dialog. Using the following table, input the formulas shown for each parameter.

TABLE 1

Parameter	Formula
Y1	R*sin(B)
X1	R*cos(B)
R	D/2
D	sqrt((X ^ 2) + (Y ^ 2))
B	120 – A
A	atan(Y / X)

4. Click Apply to test the values.

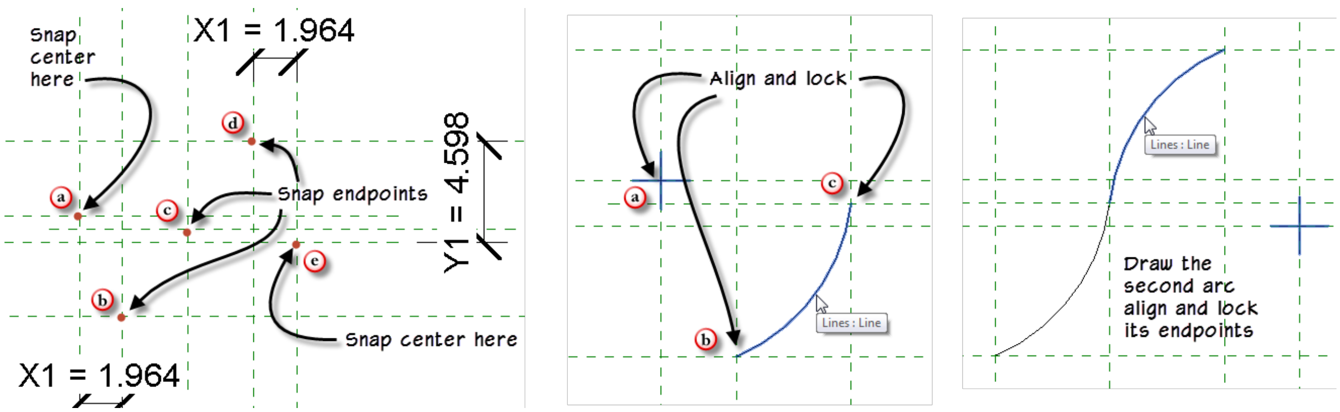
The reference planes should adjust slightly.

5. Try flexing with each of the types in the family already. Click OK when finished.

6. On the Create tab click the Line tool and then click the Center-Ends arc icon.

⇨ Snap the center of the arc to the intersection of the Center Left Hor and Center Left Ver reference planes.

7. Snap the one endpoint to the intersection of the Center (Front/Back) and Center (Left/Right) reference planes and the other to the intersection at X Mid and Y Mid (see the small dots at locations **a**, **b** and **c** in the figure).



Draw the curves

⇨ Select the arc and on the Properties palette, check the Center Mark Visible checkbox.

⇨ Align and lock the center point and each arc endpoint to the reference planes in both directions (6 alignments total) (see the middle of the figure).

8. Repeat the process by drawing a second arc with center at point **e** (the intersection of the Center Right Hor and Center Right Ver reference planes) and endpoints at locations **c** and **d**.

⇨ Turn on the center mark and align and lock all points (see the right side of the figure).

9. Open "Family Types" and flex the curve.

The curves should flex properly and remain constrained to the reference planes.

10. Draw vertical and horizontal lines locked to the reference planes for the leftmost edge and the top and bottom.

CATCH UP! You can open a file completed to this point named: *B_Cyma Profile.rfa*.

11. Flex the completed version when done.

You can load this profile into any project or family now and use it shape anything from sweeps, to wall profiles to railings. A simple family file is provided here to test it out.

12. Open: *B_Cyma Sweep Flex.rfa*.

A version of the profile family is already loaded, but if you prefer, you can load your version instead.

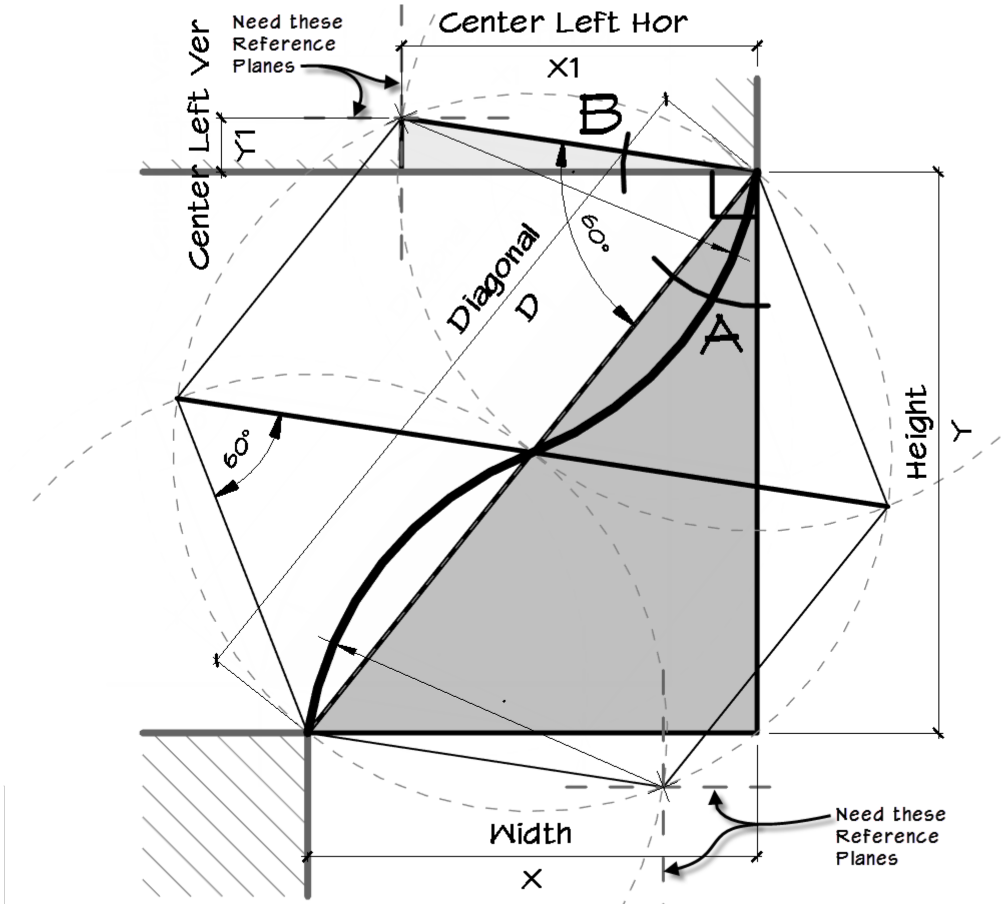
13. Select the sweep and edit it. On the ribbon, click the Select Profile button and then from the drop-down list, choose one of the types you previously defined.
14. Finish the sweep and flex the family. Close all files.

There are many related examples we could do next, but time does not permit us to do them all in the live lab. Therefore, the next few topics are provided here for you to work through on your own following the lab.

Cyma Reversa

The Cyma Reversa is essentially the same shape just with the arcs reversed. So all we need to do to create one is save the Cyma family with a new name and redraw the arcs facing the opposite direction. However, the formulas do need adjustment due to the changed locations of the arc center points. The overall strategy and form is largely similar. You can try your hand at one if you like, or you can simply open the example provided here. There are two versions of the profile that you can edit from the Families branch of the Project Browser.

CATCH UP! You can open a file completed to this point named: *C_Cyma Reversa Sweep Flex.rfa*.



Cyma Reversa Construction

The critical angle is angle B. There are two known angles in its vicinity, the right angle between the reference planes X Max and Y Max and the angle between the diagonal and the top edge of the implied hexagon. The diagonal, as we saw above, is derived from the Height (Y) and Width (X) parameter inputs and the Pythagorean Theorem. As the diagrams for the cyma (above) and this figure illustrate, the diagonal's endpoints form two vertices and becomes the bisector of an implied hexagon. The hexagon's other four vertices determine the locations of the center points of the arcs of Cyma and Cyma Reversa curves. Since a regular hexagon is easily divided into six equilateral triangles, we also know that the angle between the diagonal (bisector) and the hexagon's top edge is 60°. These known angles make it easy to calculate angle B, which in turn gives us the values X1 and Y1 and the locations of the required reference planes.

Cyma and Cyma Reversa Rigs

If you look on the *Families* branch in the two testing files: *B_Cyma Sweep Flex.rfa* and *C_Cyma Reversa Sweep Flex.rfa* you will find two versions of the nested profile families. The ones with the (Trig) suffix use the trigonometry formulas and the ones with (Rig) suffix use a 2D detail item family rig nested in them. Feel free to experiment with each one and open them to explore if you wish. This will be left to you as an exercise to do after the conclusion of the lab.

Proportional Scaling Strategies

Configuring curved forms so that they can be reshaped parametrically in a predictable and stable way has been the focus of all the examples in this lab. But so far all of the examples allow you to flex the X and Y values separately. In many cases this will be perfectly acceptable, but in some cases, you may wish to introduce further constraints to lock in a certain proportion.

Naturally, you could just take great care to always make sure your inputs to X and Y match the desired proportions, but of course, this approach is hardly foolproof. With the framework we already have in place for most of the examples created so far, it is very easy to apply an additional constraint to the parameters to force them to flex proportionally. For example, if you like the proportion achieved when X = 4 and Y = 5, simply add a formula in "Family Types" for X that reads **Y * .8**. Other formats work as well, such as **Y * 5/4**, or a formula for Y instead reading: **X * 1.25**. It doesn't really matter which one you use, as they will all yield the same results. This is because in Revit simple arithmetic formulas are bidirectional. So you can edit the value of either X or Y and the other will update accordingly automatically.

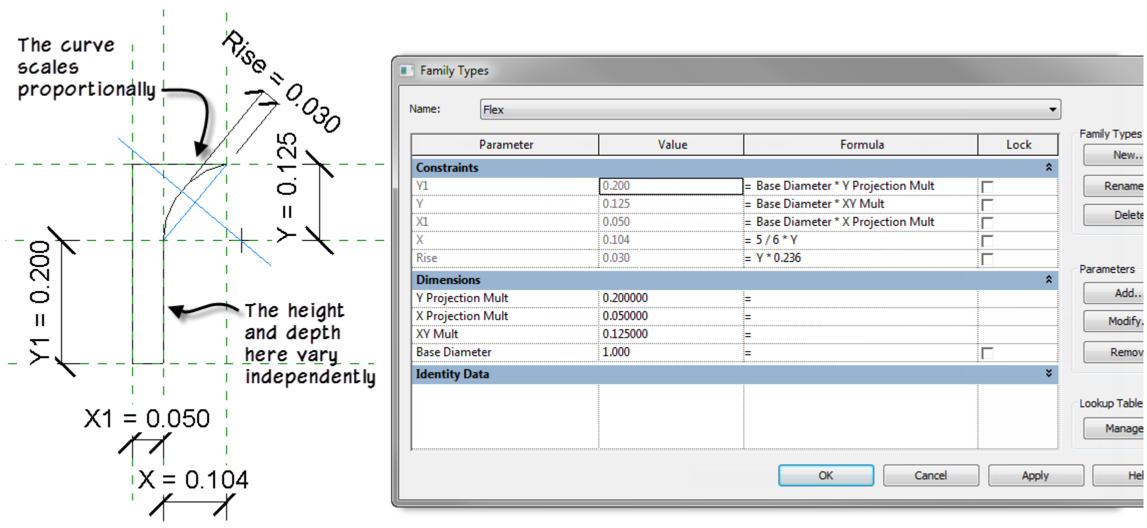
To add another level of flexibility, albeit with a touch more complexity, you can introduce a multiplier. This will enable you to establish the proportion of two or more parameters to one another, but also scale the entire family based on a single multiplier value. I have explored a few ways to approach this and have included a few additional profile families in the dataset that utilize some various scaling and proportional strategies. Let's have a look at them now.

Corona

A Corona is really just a variation of the Cavetto that we considered above. It has a similar curve with a long fillet projection beneath it. I have made the projection flexible and variable. However, I have made the curve portion proportional. In addition, I have tied everything together so that a single **Base Diameter** parameter can scale the entire shape. So when flexing this family, you can choose different heights and depths for the rectangular portion, but when you scale, the curve always maintains its proportion (5/6 of Y in this case).

SAMPLE! You can open a sample file named: *C_Corona Profile (Fixed Proportion).rfa*.

Feel free to open the file, and consider this figure.



Construction of a Corona Profile

Y1 and X1 drive the size of the lower rectangular portion of the profile. Y1 and X1 are derived formulaically from user inputs to **Y Projection Mult** and **X Projection Mult**. These two parameters are formatted as Number parameters (not Length). This means that they cannot drive lengths directly. To make them drive the length parameters Y1 and X1, they are multiplied by a length parameter. For this I used a parameter called: **Base Diameter**. Depending on the use of any moldings created from this profile, you can input appropriate values for the multipliers to yield a molding of the required size. The third numeric parameter called: **XY Mult** is used to control the size of both Y and X. It is applied directly to Y in its formula and indirectly to X since X is derived from Y.

The additional innovation in this family is the use of a Line-based Detail Item rig. We looked at examples of Detail Item rigs above. The rigs above were designed to allow disproportional scaling. In other words, the rig can flex differently for X and Y. These used a standard Detail Item template. The example here uses a Line Based template. A Line-based family is handy because you place it by clicking two points instead of one. This means that you can build it to scale proportionally based on the length between the two clicks. However, once the rotation is established with the two clicks, flexing it to a different proportion will often break it. So such rigs are best used in proportional scaling families like this one. Feel free to open the nested Line-based family and explore. This rig gives us a stable chord for our arc, so we can apply a parameter to the arc rise in much the same way that we did in the "Create a Parametric Segmental Arch" topic in Part A of the lab. Here I am deriving the **Rise** parameter as a set proportion of the Y parameter. I am doing it that way to ensure that the "height" of the rise remains a constant proportion, but still scales with the Base Diameter (which is built into the formula for Y).

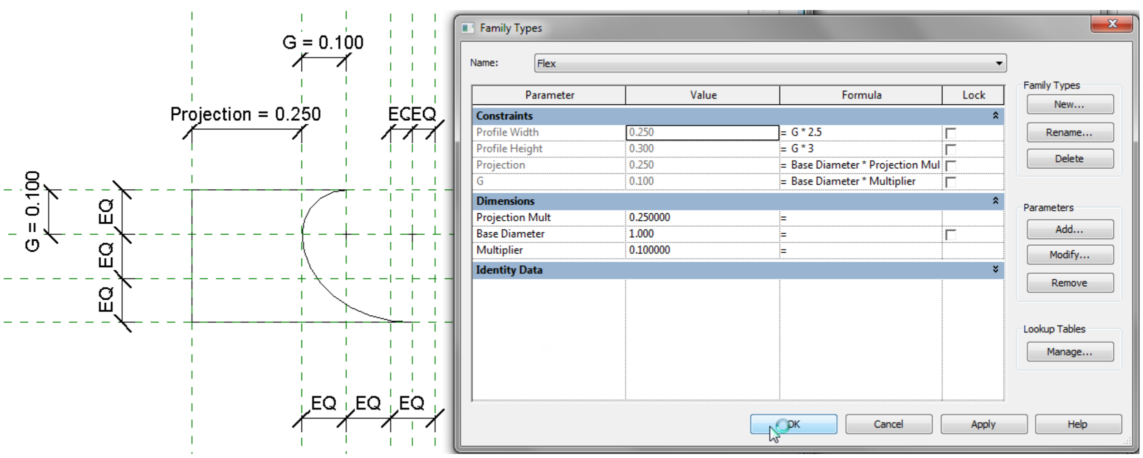
The Corona profile family introduces some complexity, but compensates for this by also introducing some additional flexibility. However, to make it truly flexible, it needs a few more enhancements. We'll explore a detailed example below, but let's look at one more proportional example first.

Scotia

The Scotia profile shown below is also fully parametric and scales proportionally. It uses some of the same techniques, but the approach is a little different and a little simpler.

SAMPLE! You can open a sample file named: *C_Scotia Form Flex.rfa*. Then on the Project Browser, beneath *Families*, right-click *Scotia Profile* and choose **Edit** to open the profile.

A scotia is made up of a circular arc and elliptical arc. (As we noted in Part A of the lab, a circle is really just a special case of an ellipse, so we could argue that the scotia is two elliptical arcs, one circular in form, but this is a semantic distinction). The approach shown in the provided file and figure here uses an overall fixed proportion of 3 to 2 ½. To avoid having lots of unnecessary parameters, a grid of reference planes is used with equality dimensions to flex them. There are four reference planes set equally in both horizontal and vertical directions. A parameter called **G** is used to size one bay of the grid in each direction, which in turn sizes the entire gird. To get the ½ bay in the X direction, the last bay is subdivided again with an equality dimension and an extra reference plane. Once again the **Base Diameter** parameter is used to scale the overall proportions. Like the Corona, in order to make the profile flexible enough to use in varying scenarios and families, a **Multiplier** parameter is introduced which is multiplied by Base Diameter to give us **G**. G drives the size of the grid and family overall. A separate multiplier (**Projection Mult**) is used to scale the size of the Projection only. The arc and ellipse are simply aligned and locked to the reference planes (both the centers and endpoints) using techniques covered previously.



Construction of a Scotia Profile

No rigs or trigonometry are necessary in this family. If you keep the goals of the family fairly limited you can often avoid the more complex approaches. This makes the families easier to construct, maintain and troubleshoot. The downside is that this family only scales in the proportion built into it. If you want it to scale disproportionally, you would need to plan for Grid X and Grid Y. This might require more formulas, trigonometry or rigs. All are possible of course, but sometimes it is easier to just save the family as a new name and build two or more, each with a different proportion. Advantages and disadvantages can be argued for each.

Create a Variable Corona Profile

If you look again at the corona above, you can see that the shape of the Corona profile includes the curved portion at the top and a flat extension below. In applications where this molding is to be used directly beneath another molding or feature, this will work OK. But for many common applications it will be useful to have a small fillet (flat molding or band) above the curved portion of the corona. We could build the fillet as a separate solid, but I think that in most cases, we could benefit from having the fillet as an integral part of the Corona Profile. So let's make that modification now in a copy of this family. Furthermore, at the start of this topic, we also discussed that this particular profile has the proportion of 5/6 built in. Let's also add another multiplier and make it possible to vary the X and Y relationship of the curve.

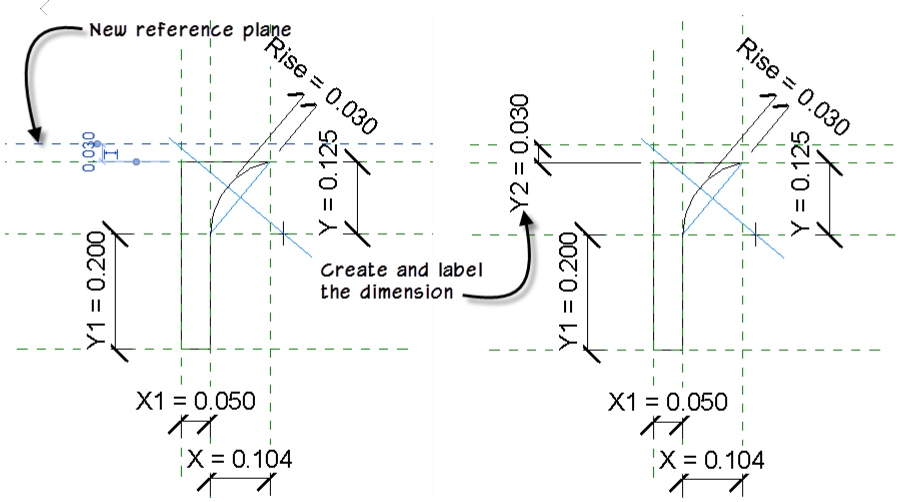
1. Open the *C_Corona Profile (Fixed Proportion)* file.
2. From the Application menu, choose **Save As > Family**.

⇨ For the name input: **Corona Profile (Variable Proportion)**.

3. Copy the upper horizontal reference plane up **0.030** units.

⇨ Add a dimension between the new reference plane and the one below it. (Be sure to dimension the reference planes, *not* the profile lines).

⇨ Label the dimension with a new type-based parameter named: **Y2**. Group it under Constraints.



Copy the upper reference plane and label it with a new parameter

Recall from above, that the lower portion of this family is flexible based on multiplier parameters. Let's do the same for this new parameter. Instead of tying it directly to Base Diameter, we'll add another multiplier parameter.

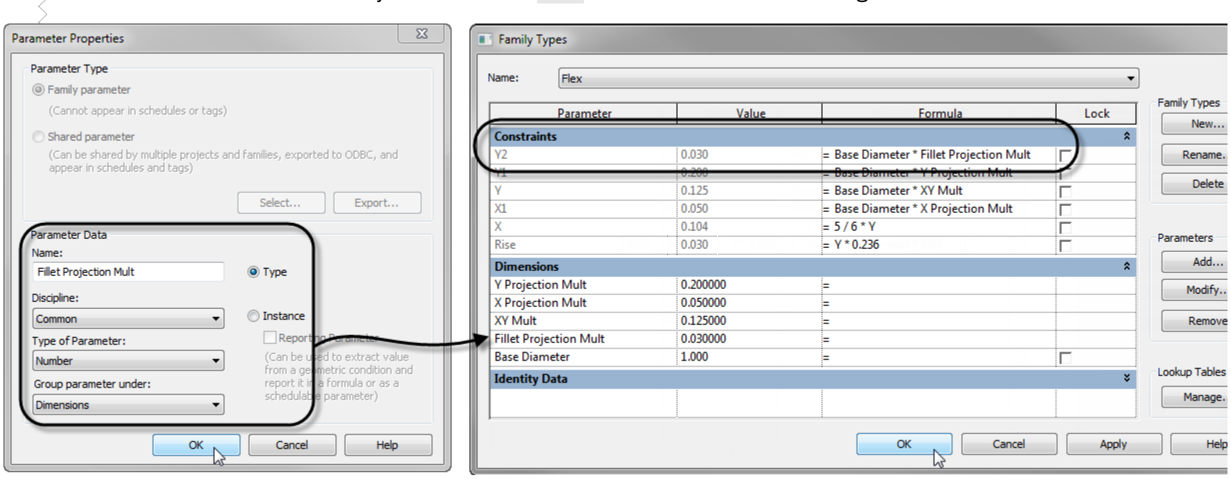
4. Open the "Family Types" dialog.

5. Click the Add button and create a new **Number** parameter.

⇨ Name it: **Fillet Projection Mult**.

⇨ Make sure it is a Type parameter and group it under **Dimensions**.

⇨ Set the value of Fillet Projection Mult to **.030** (see the left side of the figure).



Create a new multiplier parameter for the fillet

6. In the Formula field next to Y2, input: **Base Diameter * Fillet Projection Mult** and then click OK (see the right side of the figure).

⇨ Flex the Fillet Projection Mult parameter.

Only the top reference plane should adjust at this point. If you flex the Base Diameter however, the entire family will scale proportionally including the location of the new reference plane.

As noted above, this family uses a line based detail item rig. In my experiments, this is a novel approach, but unfortunately seems to behave badly when you scale it disproportionally. So as long as you keep the ratio between X and Y at 5/6, this family will perform nicely. However, to make this profile more flexible and useful to all the places we might need it, I think we should swap out the detail item family with the *Single Curve Rig* we used above. This also means we will need another multiplier and to adjust the formulas. Since we are still in "Family Types" let's start there.

7. Select XY Mult and on the right, click the Modify button.

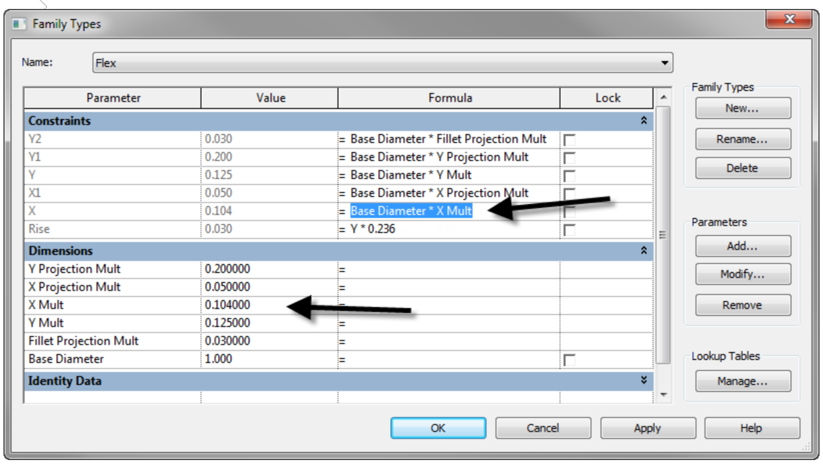
⇨ Change the name to: **Y Mult** and then click OK.

Notice that upon renaming, it also renames this parameter in all the formulas as well.

8. Create a new **Number** parameter called **X Mult**. Group it under **Dimensions**.

⇨ Set the value to match X, currently **.104**.

⇨ Change the Formula for X to: **Base Diameter * X Mult**.



Create a new multiplier parameter for X

It is important that you use the same values that are here initially. If you use a different value, when you apply the formula, Revit will try to flex the curve. This does not always go so well. Even with the same values, there may be some round off in the decimals, so when we try to close the dialog, we will get a warning. This is why we need to replace the line based rig with the more stable one we built above.

9. Click OK to dismiss the dialog. Do *not* flex before closing.

⇨ In the warning that appears, click the Remove Constraints button.

10. Delete the top horizontal profile line, the arc and the detail item rig (see the left side of the next figure).

11. On the Project Browser, expand *Families > Detail Items*.

⇨ Right-click the *Guide Lines* family and choose **Reload**.

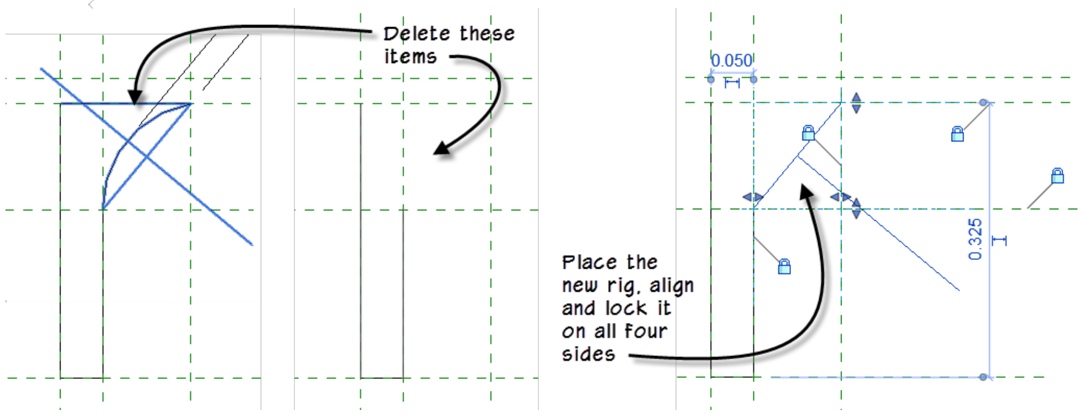
⇨ Browse to the folder containing your dataset files and choose the *C_Single Curve Rig.rfa* file. (You can also use your version created above if you prefer).

⇨ When prompted that the family already exists, choose the first option: **Overwrite the existing version**.

12. Expand the family name on Project Browser, drag Flex and drop it in the view window to place it onscreen. Align and lock it on all four sides

TIP: This family comes in very large initially. You can set the X and Y size on the Properties palette to about .25 before aligning. This will reduce the size and prevent your having to zoom out very far. Once aligned and locked, the sizes will match the context.

For each alignment, first click one of the reference planes in the host family, and then click the nested shape handle edge in the detail item family. Use TAB if necessary to get shape handle each time (see the right side of the figure).



Delete the top portion and add the new rig

⇨ Select the detail item, on the Properties palette, uncheck the Visible checkbox.

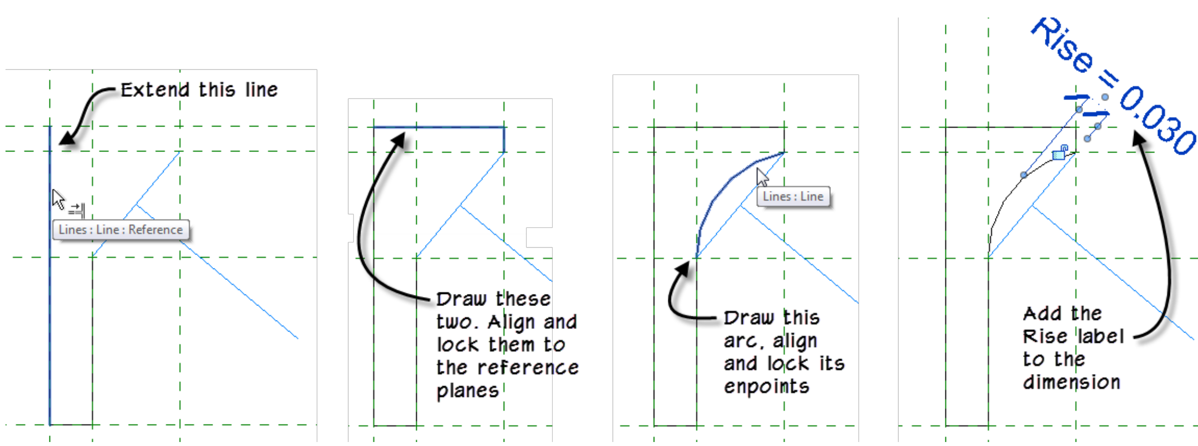
This is the same process performed above in the "Cavetto" topic.

13. Extend the vertical line on the left to the new topmost reference plane.

⇨ On the Create tab, click the Line tool and draw two new lines, one horizontal and one short vertical back down to the rig.

⇨ Align and lock the two new lines to the reference planes (see the left two images in the next figure).

> Important: Make sure that for each alignment, you first click on a reference plane, not any other geometry or the detail item, then click the line.



Adjust the shape of the profile

14. On the Create tab, click the Line tool and choose the Start-End-Radius arc.

⇨ Snap the endpoints to the ends of the detail rig.

⇨ Snap the radius when the arc show tangent to the other lines.

⇨ Align and lock the endpoints of the arc to the reference planes (see the third image in the figure).

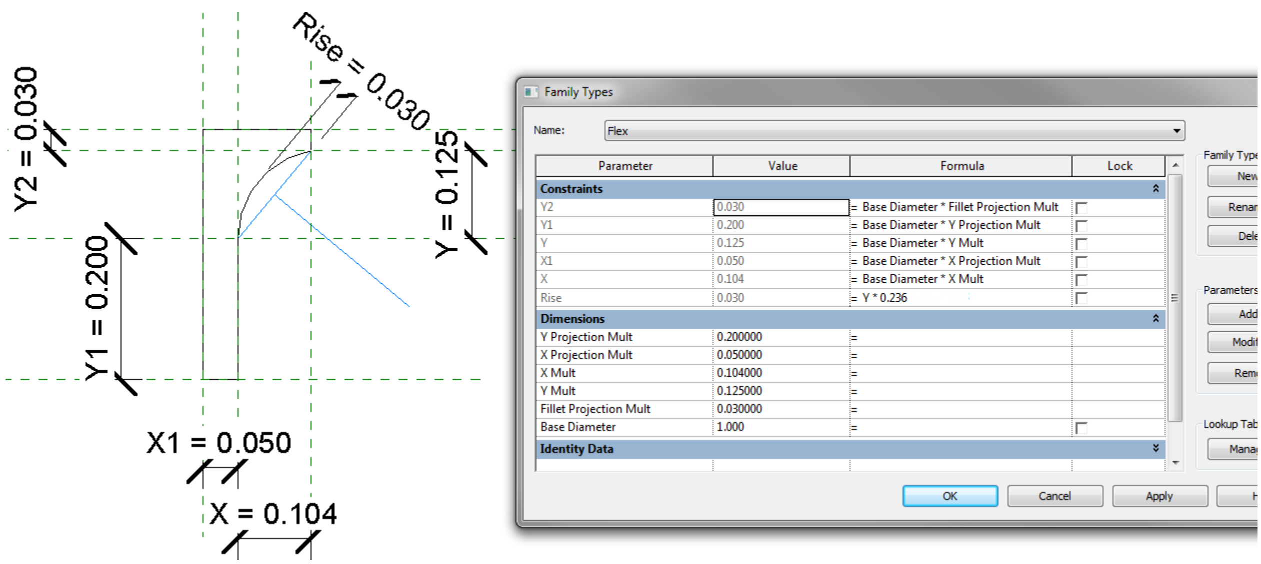
> Important: Make sure that for each alignment, you first click on a reference plane, not any other geometry or the detail item, then click the endpoint of the arc. Use tab if necessary.

15. Finally, add a dimension between the diagonal line in the detail item rig and the arc.

⇨ Label the dimension with the **Rise** parameter (see the right image in the figure).

> Note: This procedure is the same one we followed above in the "Create a Parametric Segmental Arch" topic; in Part A of the lab.

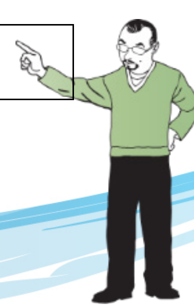
When you deleted the lines above, you most likely lost some dimensions too. So be sure to replace any missing dimensions. For example, the Y and Y2 dimensions. The next figure shows the completed Corona Profile with its "Family Types" showing the parameters and formulas.



The completed Corona Profile and its parameters

16. Save and close the file.

> **CATCH UP!** You can open a file completed to this point named: *C_Corona Profile (variable Proportion).rfa*.



Controlling a Spline

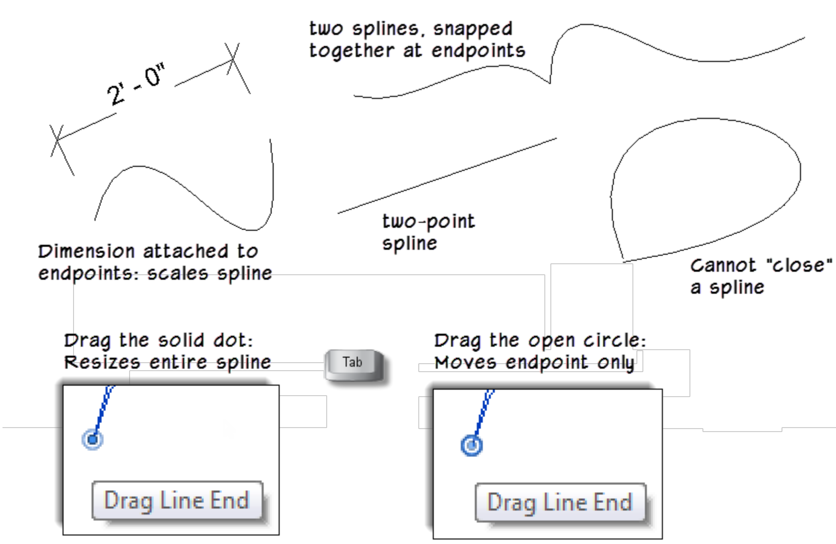
All of the examples so far have used some combination of arcs, circles and ellipses. To wrap up our inventory of curves and techniques to parametrically control curves, we'll take a brief look at splines. In the traditional family editor, the spline is the last type of curve that we have. The spline in the traditional family editor is a Bezier spline.

According to Wikipedia:

> A Bézier curve is a parametric curve frequently used in computer graphics and related fields… In vector graphics, Bézier curves are used to model smooth curves that can be scaled indefinitely.

SAMPLE! You can open a sample file named: *C_Traditional Spline.rfa*.

In Revit, a spline is drawn as a series two or more control points. Splines are open curves in Revit; there is no "close" option. You can have as many intermediate points between the start and end point as required to shape the curve. With a little practice, you can create fairly complex curves from splines with little effort. When creating 3D forms from splines, the surfaces will remain smooth; no facets or edges. This can be a big advantage for certain types of forms. If you need a hard edge, consider other types of lines or create more than one spline. The top of the figure shows some possibilities.



Working with Splines

Another interesting and useful characteristic of splines is that if you drag either endpoint, the entire spline will stretch and scale proportionally at the same time. This means that as you stretch the implied line that connects the start and end points, your spline will maintain its shape and scale proportionally as it grows larger or smaller. You do not need to do anything special to achieve this behavior. This is the built-in behavior of splines (and as the Wikipedia definition above noted, of Bezier curves in general).

If you wish to actually reshape your spline at one of its ends, make sure you press TAB to cycle the selection into the open circle at the grip point. The open circle is the control handle, while the solid dot is the endpoint. Endpoints will scale the spline. The open circles reshape it (see the bottom of the previous figure).

Note also in the figure that you can attach a dimension to the endpoints. This allows you to scale your spline using parameters and labeled dimensions. You can attach the dimension directly to the endpoints (TAB as necessary) or you can align and lock the endpoints to reference planes and flex them that way instead.

Given the built-in behavior of splines to scale proportionally when stretching the endpoints, it is possible to use them to shape your profiles and potentially avoid some of the techniques covered above. You can achieve a close approximation of many of the forms shown above, but they will not be completely precise. In other words, most of the moldings previously discussed use arcs: segments of circles. Splines can approximate these curves very closely, but will not be perfectly circular. If you build your splines carefully would anyone notice the difference? Perhaps not, but I much prefer to limit the use of splines to forms that do not use circles, arcs, and ellipses. For example, when more organic or freeform shapes are required, splines are a powerful alternative. In later chapters I will show some of these examples. Feel free to experiment with splines a bit before you continue.

This concludes the bonus material. There are two lessons left in the paper. We will attempt to complete each of these in the live lab.

Let's Get Serious

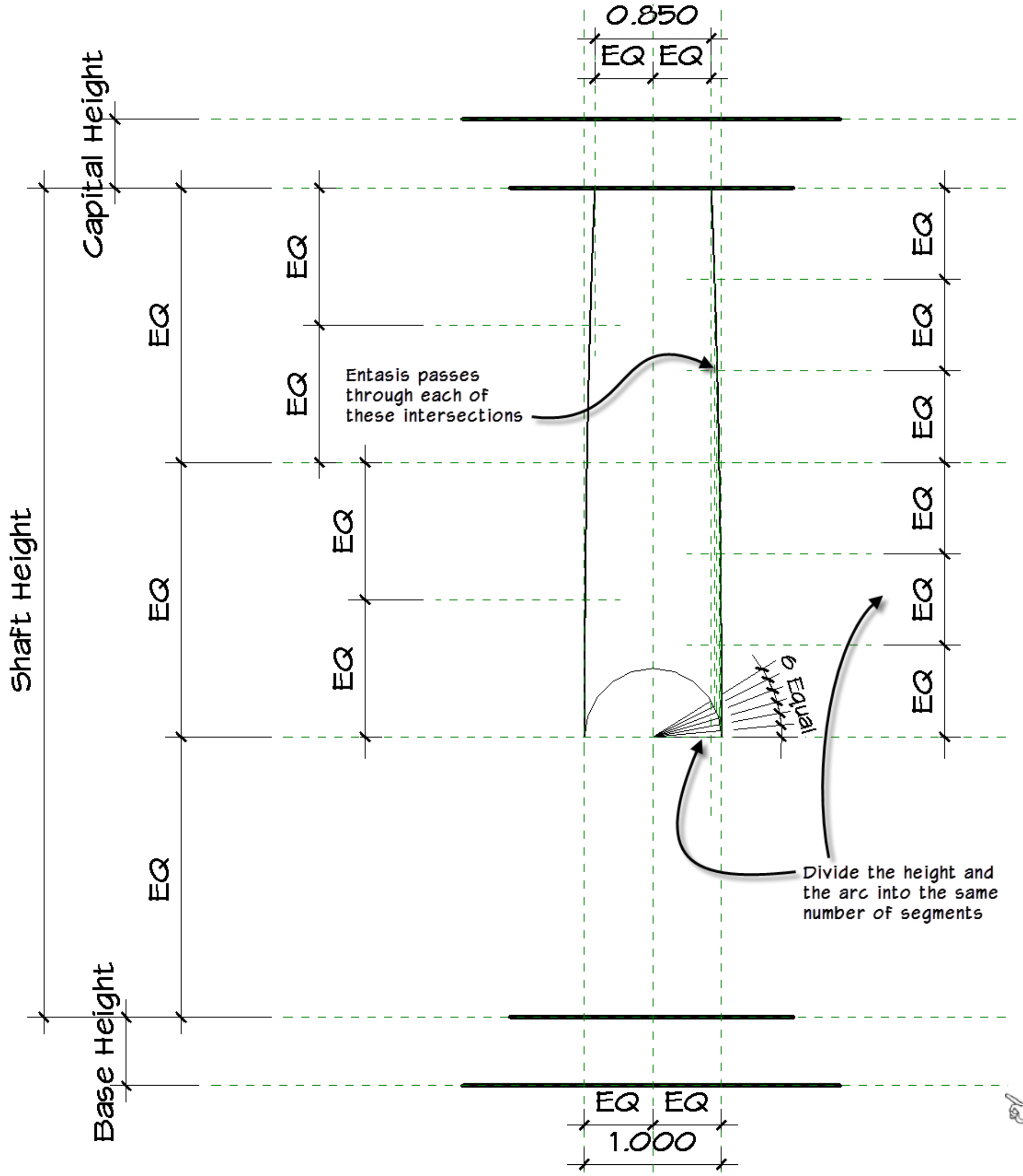
We have covered a lot of ground in these labs so far. To finish things up, we are going to look at two more complex examples. The first will use formulas to control the shape of an elliptical arc. The second will create a complex spiral form of progressively smaller arcs using a nested detail item rig. The first example is the entasis of a classical column shaft. Entasis is the phenomenon of slightly diminishing the diameter of the column shaft towards its top. The spiral form comes from the volutes of an Ionic column capital. So both of these examples are drawn from classical architecture, but the techniques used can be used in any application.

Build a Smooth Shaft with Elliptical Entasis

In classical columns, the shaft diminishes in diameter as it reaches the top. This diminution often called "entasis" does not follow a straight path (as you would get with a blend in Revit), but rather is slightly curved as it diminishes. There are a few ways that you can achieve the appropriate shape in the family editor. If you are in the traditional family editor, you need to sketch the profile of the column vertically along its length; as if it were being turned on a lath and rotate this around 360°. This can be done with a revolve or a sweep. For all the reasons I have already noted, I will use a profile in this example.

As you might expect, if you review the various masters and treatises you will find a few ways to plot the proper entasis. I like to use the method outlined by Robert Chitham in his book: "The Classical Orders of Architecture" (also used by many other Renaissance authorities). First you divide the top two thirds of the shaft into equal parts; you can choose any number, I have typically used 4 or sometimes 6. Next, you draw a semicircle at the base and project a line down from the top of the shaft at its narrowest point. To calculate this point, Chitham notes that most authorities diminish the column

1/6th of its diameter. This would be about 0.840. Chitham goes with 0.850 and I have adopted this value as well. Where this vertical line meets the circle, we get an arc. Divide this arc into the same number of parts. Finally, you project lines from these points until they intersect the horizontal lines divided above. This gives you the points through which to pass your curve.



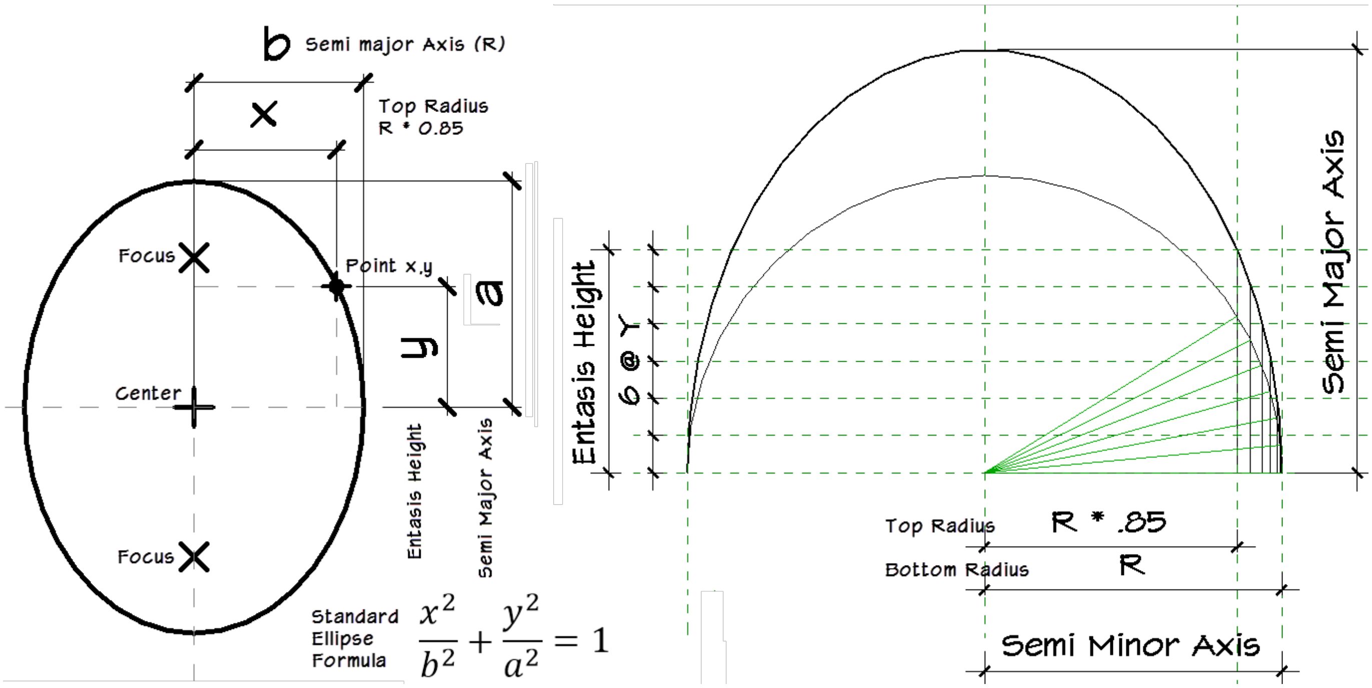
Constructing the shaft entasis profile curve

If you draw an arc with a very large radius, it would come very close to passing through these points; almost imperceptibly close. So if you want to stop reading now and just draw an arc, that's OK. But if you want it to be more accurate, continue reading!

Another option is to use a spline. A spline is a nice option because you can make as many control points as you need to make the shape of the spline intersect at the required points. We also saw that locking the two open ends of the spline will maintain the shape of the curve when you flex. So this can be quite effective. However, as I was building the layout described above to draw the shape traditionally, it occurred to me that if you take the divided circle and imagine stretching it up vertically, you would be transforming the circle into an ellipse and the point divisions on the arc portion would continue to line up with your vertical divisions. If you stretch it far enough, you end up with a very tall and thin ellipse. This basically means that the curve profile of the entasis is actually an elliptical arc. So in this topic I am going to present the steps required to build a parametrically driven elliptical profile here. You can also try the spline approach as an exercise on your own if you prefer.

Create a Parametric Elliptical Arc

To build an ellipse in Revit, you need to know its center and the length of each axis. For an elliptical arc, you basically need just the axis lengths. Revit determines the center automatically. So the challenge is to take the construction technique recommended by the renaissance authorities (for hand drafting the entasis curve) and convert this to the inputs that Revit requires to create an accurate ellipse. The next figure shows a compressed diagram (I shortened it vertically) to illustrate the concepts and what inputs are needed.



Applying the standard formula for an ellipse to the entasis and our known variables

I spent a lot of time trying to figure out the formula. (Maybe a little too much in retrospect). I owe thanks to many resources online and especially to Desirée Mackey who helped me personally with the formula. The basic concept is that there is a standard mathematical formula for an ellipse. It has four inputs typically labeled a, b, x and y. The ratio of the squares of these is always constant and equal to one:

$$\frac{x^2}{b^2}+\frac{y^2}{a^2}=1$$

On the left side of the figure I have drawn a standard ellipse diagram and labeled each part. The two axes typically get assigned the variables **a** and **b**. Some point on the ellipse is labeled as point **x,y**. On the right side, I have taken the diagram from the previous figure and compressed it vertically to make it easier to comprehend. In our case, we have three of the required values in the formula, so we can input them into the standard formula and determine the missing value.

The total of the height of the six divisions (I chose 6, but it can be any value you choose) is the height of the top two thirds of the shaft minus any upper moldings. I called this **Entasis Height**. This is the y value in the standard formula. Each division is shown with a horizontal reference plane. The bottom radius: **R** is half of our column base (Base Diameter / 2). The top radius is 0.850 times this number (the radius at the top of the shaft minus any moldings: 0.85 adopted from Chitham). This gives us the b and x in our ellipse formula. You can see the diagonal lines dividing the arc equally and how these project up to intersect the ellipse at the horizontal divisions.

So our missing variable is a; the Semi Major Axis. This is required to draw the ellipse in Revit, so we must figure out what this value is in order to draw the proper ellipse. Simply input what we know into the formula and then solve for: a.

$$a = \sqrt{\frac{y^2}{1 - \frac{x^2}{b^2}}}$$

Or to put that in Revit format:

sqrt((Entasis Total ^ 2) / (1 - (Top Radius ^ 2) / (Base Radius ^ 2)))

We will need to define the parameters within the formula as well and add a reference plane for the semi major axis.

1. Open the file named: *B_Column Shaft_Start.rfa*.

   The *Front* elevation view will open. Minimize the view.

2. On the Project Browser, expand *Families > Profiles* and then right-click on *Column Shaft Profile_Start* and choose **Edit**.

⇨ Type: WT to tile the window.

   Since there is only one window, it makes this window full screen, while leaving the original elevation window minimized.

This profile is using the arc for entasis. We will delete this, create some parameters with the appropriate formulas for the ellipse and then draw the elliptical arc.

3. Delete the large arc at the top two thirds of the profile.

4. Open the “Family Types” dialog and add the parameters shown in table below. (To make it simpler, the formulas are saved in a TXT file called: *Ellipse Formulas.txt*. You can copy and paste from there if you prefer).

   They are all **Length** parameters, they should all be **Type** parameters and group them under **Constraints**.

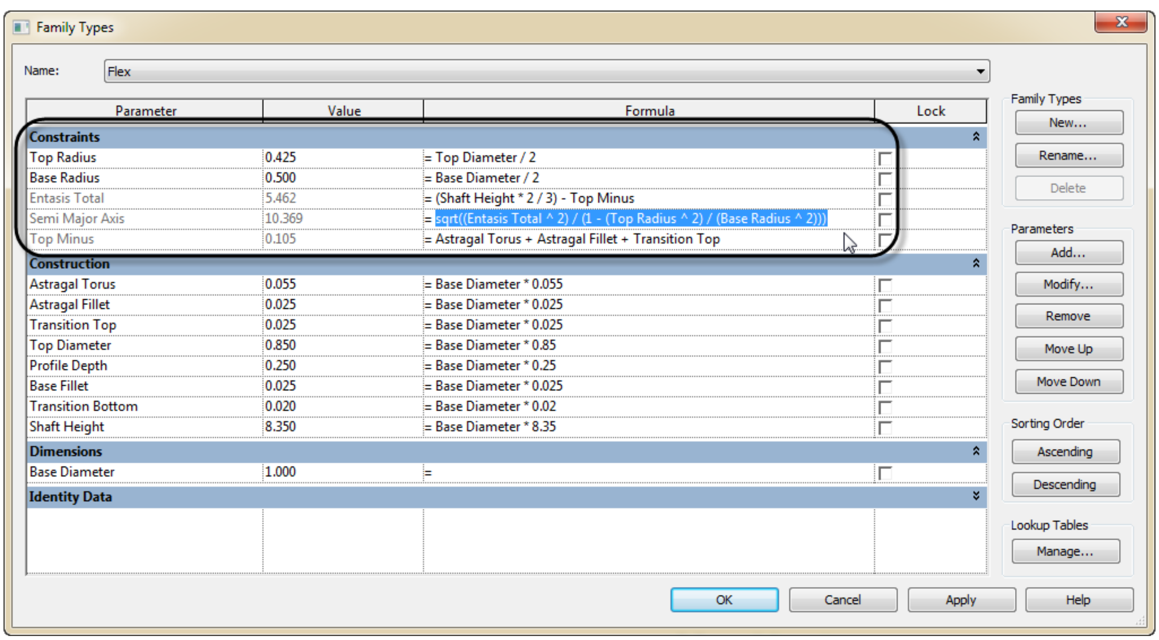
NEW in 2015: Use the **Move Up** and **Move Down** buttons in the "Family Types" dialog to rearrange the parameters in the same order as the table! You can optionally add tooltips too. ***Woohoo!***

> **Note:** Input Semi Major Axis last. Be sure all of the other parameters are in place with their formulas working before you complete Semi Major Axis. The numbers in the Value column are for your information—when you fill in the formulas, these values should automatically appear.

TABLE 3

Parameter	Value	Type of Parameter
Top Radius	0.425	Top Diameter / 2
Base Radius	0.500	Base Diameter / 2
Entasis Total	5.462	(Shaft Height * 2 / 3) - Top Minus
Semi Major Axis	10.368	sqrt((Entasis Total ^ 2) / (1 - (Top Radius ^ 2) / (Base Radius ^ 2)))

There is already a Top Diameter parameter in this family. To get the Top Radius we simply half that. Likewise for the Base Radius, we half the Base Diameter. This gives us our **x** and **b** values. To calculate the Entasis Height (**y** in the ellipse formula above), I created two separate parameters, but it is possible to merge all of this into a single formula if you prefer. Top Minus totals all of the moldings at the top of the shaft. This is subtracted from 2/3 of the shaft height.



Create the parameters and their formulas

After inputting the others, input Semi Major Axis. It is a long and complex formula, so type it carefully (or paste from the TXT file). To square a value in

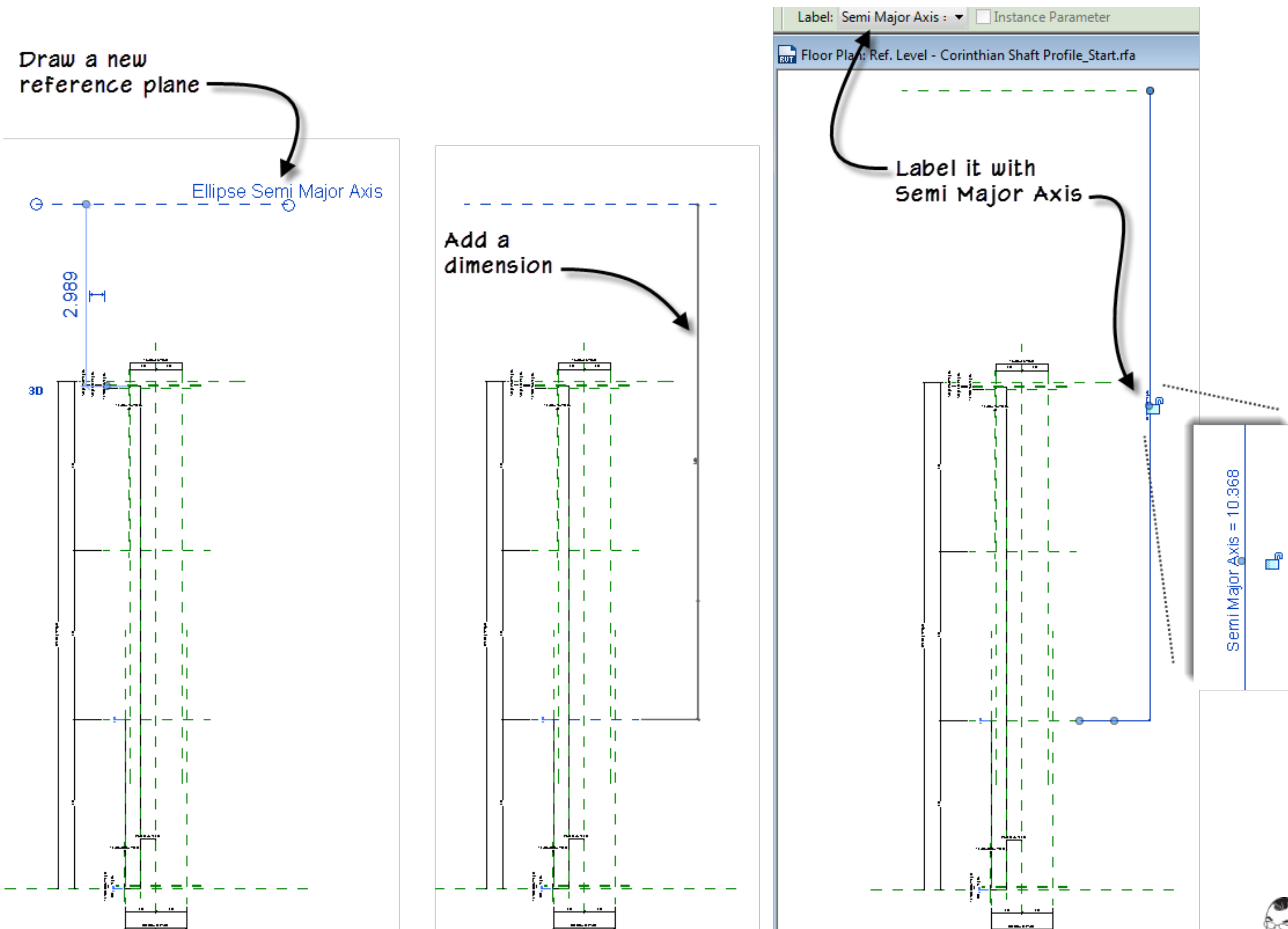
Revit, you raise it to the power of 2. This is done with the carrot (^) symbol. You can use this symbol to raise to any power. So, 4^2 is four squared. 4^5 is four to the fifth power. You can substitute either number for a parameter so that M^N power is also easy to achieve (where M and N are parameter names). To take the square root, type: **sqrt()** and input the value in the parenthesis.

After creating the parameters and formulas, the values should match the table.

5. Click OK to dismiss "Family Types."
6. Zoom out and add a horizontal reference plane above the profile a few units away.

⇨ Add a dimension between this new reference plane and the first third (where the entasis begins).

⇨ Label it with the Semi Major Axis parameter.



Add a reference plane and label it with the Semi Major Axis parameter

We are now ready to draw the ellipse.

7. On the Create tab, click the Line button. Choose the Partial Ellipse icon.

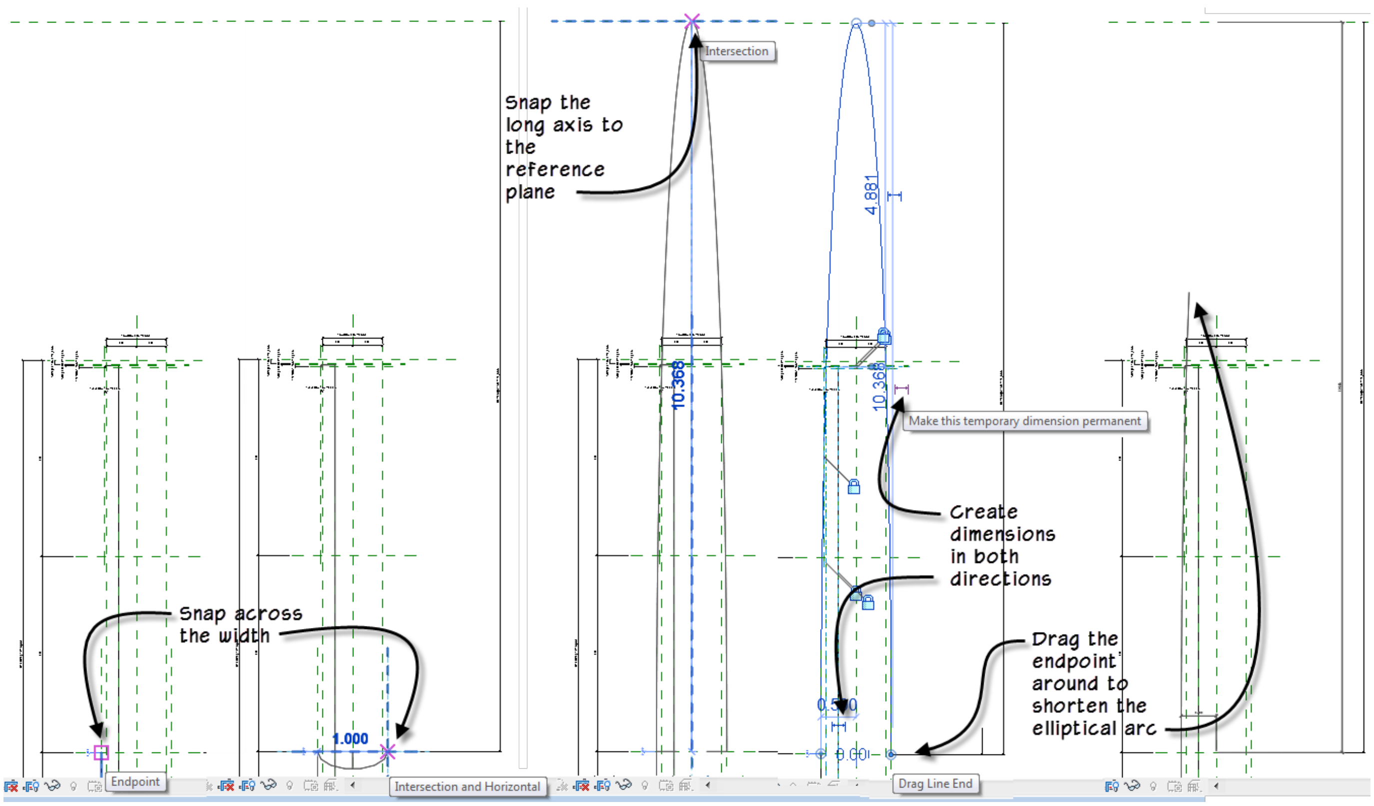
⇨ Snap the first two points across the width of the shaft profile snapping to the intersections of the reference planes (see the left side of the next figure).

⇨ Snap the long axis to the horizontal reference plane (the one you just drew) at the top. Cancel the command.

⇨ Select the new ellipse, and then click the “Make this temporary dimension permanent” icon in both directions (see the middle of the figure).

8. Using the shape handle at the end on the right side of the elliptical arc, drag it around (over the top) and release close to the top of the shaft (see the right side of Figure 12.6).

⇨ Zoom in and stretch the grip handle again to snap it precisiely to the intersection of the reference planes named: H2 and Top Left (it should snap right to the endpoint fo the small arc at the transition top)



Draw the elliptical arc, dimension it and then adjust the length

9. Label the two dimensions created in the previous step.

⇨ Label the long vertical dimension on the ellipse with the **Semi Major Axis** parameter.

⇨ Label the short horizontal dimension with the **Base Radius** parameter.

10. Align and lock the enpoints at both ends of the ellipse to the reference planes in both directions.

CATCH UP! You can open a file completed to this point named: *Column Shaft Profile.rvt*.

11. Click Load into Project and overwirte the existing.

You will see a very slight change in the profile of the shaft. If you like, you can rename the profile on Project Browser to remove the "_Start" suffix.

12. Flex the family and then save and close the file.

CATCH UP! You can open a file completed to this point named: *B_Column Shaft.rfa*.

You can apply this technique to any ellipse or partial ellipse that you need to control parametrically in your family content. Just remember that we are using the standard formula for an ellipse. If you know or can determine three of the four variables, you can use a formula to find the last one. This means that you can exert great control over any elliptical form you may use. Extending the concept a little further, if you can determine the mathematical description of your 3D form, you can apply that knowledge to any family you create. You simply have to convert the standard mathematical formulas into Revit syntax.

A great resource to help you with this is found at a post over on Revit Forum:

http://www.revitforum.org/tutorials-tips-tricks/1046-revit-formulas-everyday-usage.html

With the permission of the post's original author: Klaus Munkholm, I have reproduced the trigonometry portion of the post at the end of this paper.

The Ionic Capital Volutes (Scrolls)

The final exercise in this lab is perhaps the most ambitious. I have cherry-picked it out of my book: *Renaissance Revit* and included it here for two reasons. First, for those of you craving something complex to really put the techniques of this lab to the test, the Ionic volute delivers! Second, it uses nearly all of the techniques we have covered so far, including being a robust example of using a nested detail item rig. Unfortunatly, to fully give you the background on this family would require a lab of its own and several more pages of documentation (like maybe a whole book's worth ☺). So I will just give the necessary biots here and focus on the turoial. I hope that when complete, the form will speak for itself. However, if you would like to dig deeper into the theory and background, please consider getting a copy of *Renaissance Revit*.

http://paulaubin.com/books/renaissance-revit/

To properly construct the volute of the Ionic Order, it takes no less than 24 separate arcs. Talk about putting our curve constraining strategies to the test! The volute is a spiral form made from progressively smaller arc segments. Each is 90° and reduces in radius until they converge on the "eye" (a round protrusion) in the middle region (but not exactly in the center) of the volute. If you want to learn how to construct this form, there are dozens of authorities on the subject. So you choice of source material is vast. Robert Chitham in his book: *The Classical Orders of Architecture* had this to say about the Ionic volute:

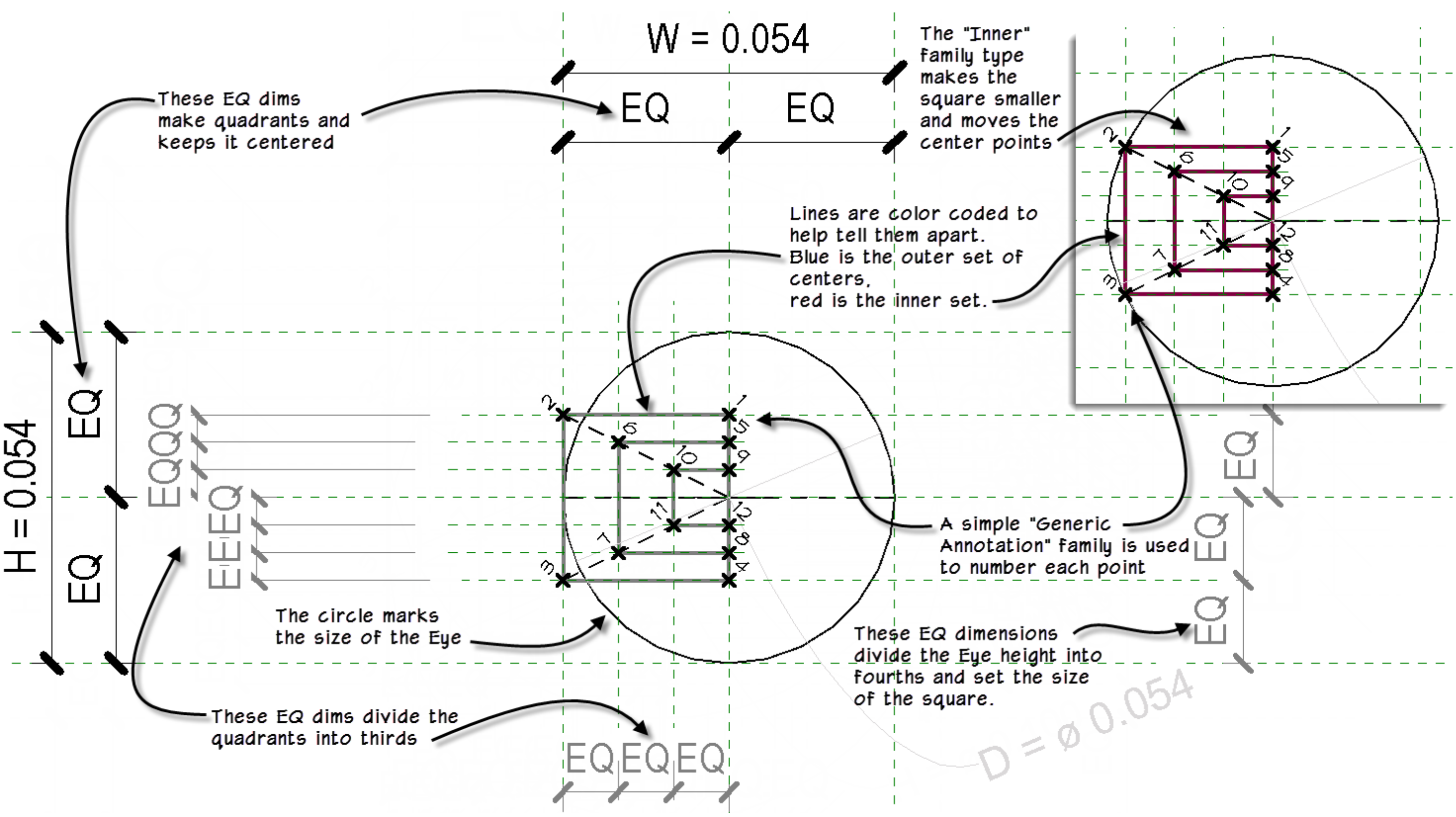
> "The volute is the most difficult single element in the orders to draw, because the dimensional tolerances are so fine."

I would be hard pressed to disagree with this statement. I have had many failed attempts before arriving at a success for this tutorial. Actually, if we were only concerned with drawing a single static representation of the volute, then it is fairly easy to acomplish using a computer program like Revit. There are many steps to be sure, but the form creation is mostly repeditive; not difficult. However, to not only draw the form, but make it fully parametric is another matter entirely. This has proved quite challenging. If you completed the earlier lessons like the "Ovolo," "Cyma" and "Cyma Reversa" topics above, then I am sure you can imagine the challenge that we are up against. As before, I have provided progress files along the way, so if you get stuck feel free to open a catch-up file to continue.

Understanding the Approach

At the start of the lab, we looked at several examples to define parametric curves. Typically, we would want to create intersecting reference planes for each center point and each endpoint. However, the sheer quantity of arcs in this example presents a practical limitation. With 24 arcs required, that would be quite a few reference planes indeed! And most of them in a very tight space. For this reason, I am going to use the nested detail component rig approach coupled with equality dimensions instead. We will still have several reference planes to deal with; we will just hide many of them inside the nested family to make it easier to work with. The good news on this appraoch is that we will have virtually no parameters to create! Instead, we will leverage the fact that everything is based off of proportions and rely on equality dimensions to subdivide each part into smaller and smaller parts. There will still be plenty of detailed and meticulous work requiring patience to complete, so ***consider yourself forewarned***.

Let's start with a look at the rig. I have provided this as a detail component family with the dataset. I have an illustration of the family here. It is a bit intense, but remember, we have to locate 24 center points! You can open the file from our starting file below. It is nested in on the Families branch already and is called: *Volute Eye Rig*.



A Detail Component family will be used as a rig to locate the centers in the Ionic volute

The layout technique starts with a circle for the eye of the volute. A square with the side equal to the radius of this circle is constructed to one side of the circle and centered vertically.The outermost reference planes describe a square shape. These have parameters H and W applied to control the width and height. In reality I could have used the same parameter for both, but just left these two in here from the seed family. The square has two diagonals from the center of the circle to the outside corners. The vertical edge of the square passing through the center is divided into six parts. I did this with reference planes and equality dimensions. Finally, the intersections of each diagonal and the dividing reference planes gives us the 12 points we need for the outside edge of the spiraling fillet. I created some subcategories in the "Object Styles" dialog (Manage tab). I assigned them to colors to make it easier to read everything. I drew lines connecting the points and used a blue colored line to help identify the outer edge.

For the inner edge of the fillet, I have opted to calculate offset with a simple formula. You can open "Family Types" to see the formulas and how they are applied if you like. Basically when you choose between the "Inner" and "Outer" types, it adjusts the scale of the family and turns on and off the correct set of colored lines.

To keep track of the center points, I created a simple annotation family to number each point. To create this family, I used the *Generic Annotation.rft* template and created a small "X" and a label.

Create the Volute Profile Family

The file we will use to start the exercise has some basic reference planes already in place as well as some of the parameters we will need. The detail component rig discussed here is already in the file. We'll finish a few setup items and then begin drawing the volute spiral. So let's get started.

1. Open the file named: *Ionic Volute Profile_Start.rfa*.

⇨ Save the family as and name the new file: **Ionic Volute Profile**.

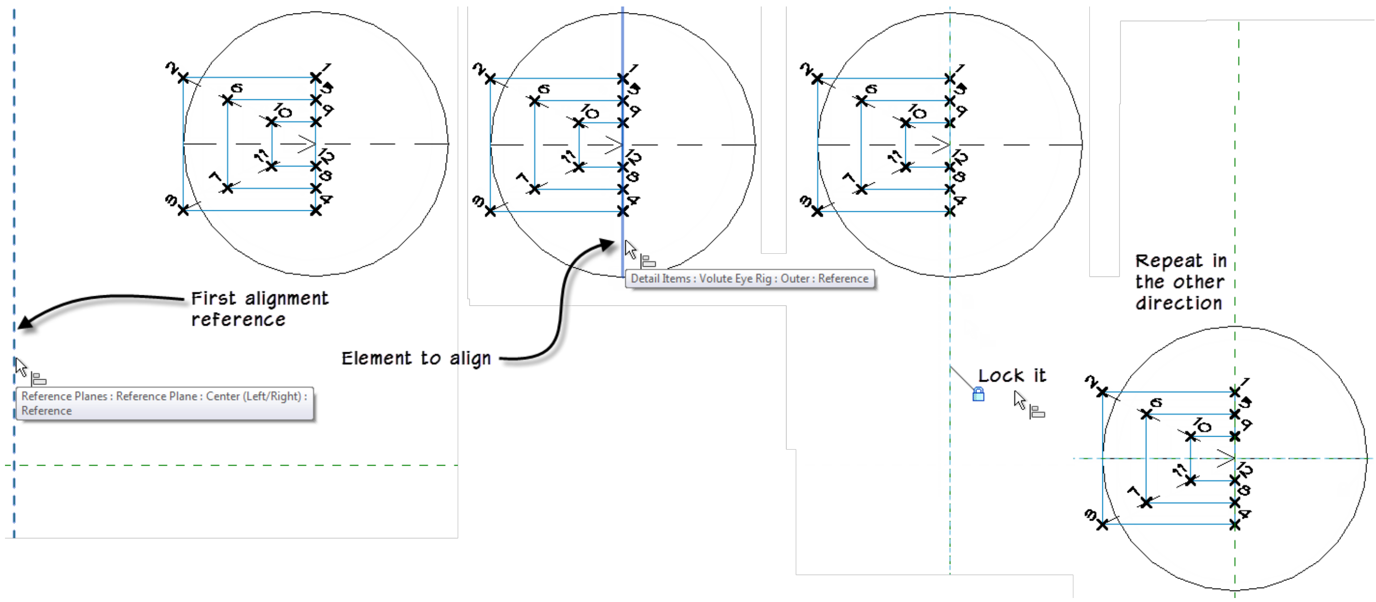
Everything will appear very small. I have set the scale to full scale given the small size of the volute eye rig. If you set the scale much larger, the numbers become illegible. So we'll have to rely on zooming for the time being.

2. Zoom in on the Volute Eye Rig detail component (already onscreen).
3. Select the detail rig onscreen. On the Properties palette, click the Edit Type button.

⇨ In the "Type Properties" dialog, click the Associate Family Parameter button next to Base Diameter, choose Base Diameter and then click OK twice.

4. Align the detail component rig onscreen to the intersection of the Center (Left/Right) and Center (Vertical) reference planes. Lock in both directions.

   Align so the circle is centered—the square will appear off to the left.



Align and lock the Detail Component to the centers

Zoom in closely on the detail rig.

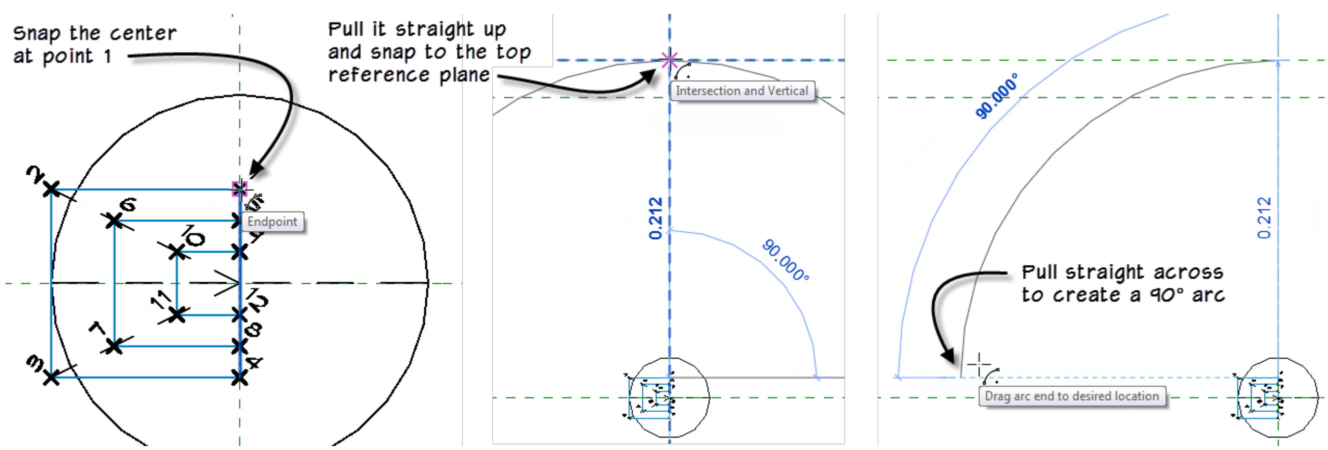
5. On the Create tab, click the Line button and then click the Center-ends Arc.

⇨ Snap the center point to the intersection at point 1 on the rig.

⇨ Pull straight up 90° and snap to the topmost reference plane. (Make sure it is exactly 90° before you click).

⇨ Pull straight to the left at 90° again and then click.

You should have a quarter circle centered on point 1.



Draw the first arc centered on point 1 and 90°

6. Cancel the command, select the arc and on the Properties palette, turn on the center mark.

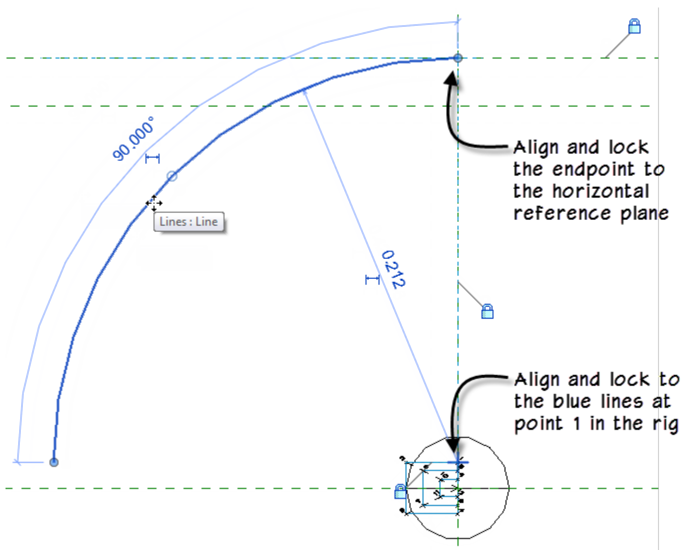
7. Zoom in on the detail rig. Align and lock the center mark to the **detail rig** (*not* reference planes) in two directions.

Use the blue vertical line in the center of the rig for the first reference. Lock it to the center mark.

Use the blue horizontal line at the top of the square for the second reference. Lock it to the center mark.

Notice that the Automatic Sketch Dimensions disappear when you lock. Be very careful about which edges you select. For the first twelve points, we only want to align and lock to the **blue** lines in the rig. (TAB as required).

8. Zoom back out. Align and lock the endpoint at the top of the arc to the horizontal reference plane. (*One direction*; horizontal reference plane only this time).



Align and lock the arc at the center point and the first endpoint

9. Open "Family Types" and flex the Base Diameter to **2.000** and then click OK.

Notice that the arc adjusts as expected. The center is constrained to the detail rig. The detail rig has its internal Base Diameter parameter linked to the Base Diameter in this file so it is flexing as well. And with the endpoint locked to the reference plane, the radius is established as well.

10. Reset the Base Diameter back to **1.000** to continue.

11. Save the file.

CATCH UP! You can open a file completed to this point named: *Ionic Volute Profile_A.rvt*.

12. Return to the Line command, select the Center-ends Arc again and snap the center point to point 2 this time.

- Move to the left and snap to the endpoint of the existing arc.
- Move straight down at 90° and then click. (NOT the intersection, it will *only* touch the horizontal reference plane).

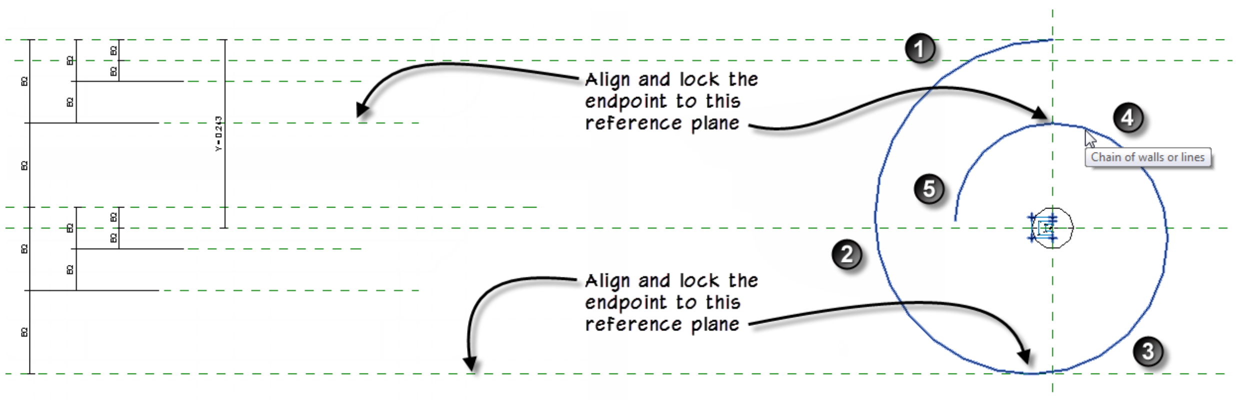
13. Once again, cancel, turn on the center mark, align and lock (both directions to the blue lines of the rig, and *just* the horizontal reference plane down below).

⇨ Flex again.

Now here's the really tedious part: we have to repeat this process 10 more times for the outer edge of the fillet and then 12 more times for the inner edge. Yes, 22 more arcs, alignments and locks. So these two were some of the easy ones. **Unfortunately there is no shortcut**. So just be patient and methodical and be sure not to miss any steps along the way or things might misbehave when you flex. I recommend **frequent** flexing as well. This way if one of the arcs is misbehaving, you will find out which one right away.

14. Repeat the process to add arcs 3 through 5.

Add each arc one at a time and stop and turn on the center mark and align and lock as you go. Align and lock any time you see an Automatic Sketch Dimension appear. Arcs 2 and 3 can be aligned to the horizontal reference plane at the bottom of the screen. Arcs 4 and 5 will sense a reference plane off screen to the left. You can align and lock to this reference plane even though it is off to the side. This helps a lot to keep the onscreen clutter in check.



Create the first five arcs of the outer shape of the volute fillet

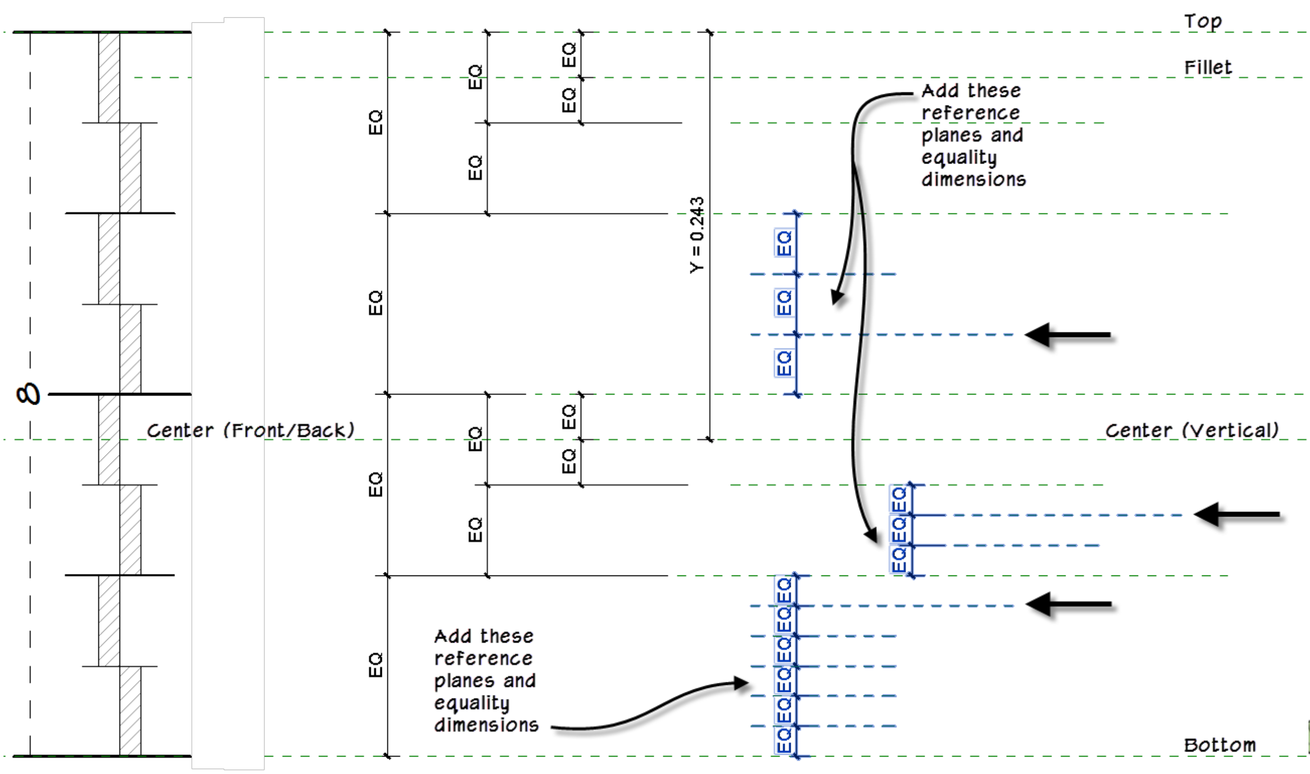
15. Save the file.

CATCH UP! You can open a file completed to this point named: *Ionic Volute Profile_B.rvt*.

Add Reference Planes with Equality Dimensions

The collection of reference planes off to the left side at first looks somewhat chaotic. However, there is a strategy to how I have laid them out. Zoom the screen so you can see all of the reference plans at the left. In the previous figure I have added a graphic scale to the left to help illustrate the relationships of the reference planes. Note first that everything is slightly off center. The overall distance between Top and Bottom reference planes is the height of the volute. The Y parameter controls the distance from the Center (Vertical) reference plane (which is the origin of this family) and the Top reference plane. The Center (Vertical) reference plane is also the location of the volute eye. Everything else is driven by a series of nested equality dimensions.

As you study the groupings of reference planes, notice that I have varied the lengths of each set to help make them easier to understand. I also stagger the dimensions. So first we divide the total height into four. Then we subdivide the top quarter in half to make each an eighth of the total. Then subdivide it in half again to give us the size of the fillet at the top. I carried this reference plane all the way across since we start our arcs at the top. A similar subdivision occurs to locate the eye at the Center (Vertical) reference plane.



Analyze the reference planes and equality dimensions existing in the file and add additional ones

As you saw when creating arcs 4 and 5, even if the reference plane is off screen, you can still snap to it and lock to it. So, let's create a few more groups of equally spaced reference planes. We really only need three for the purposes of snapping geometry. These are the ones indicated with the bold arrows in the figure. However, in order to use the equality dimensions you have to add additional ones as well. I like to vary the length make the important ones stand out. So when creating the reference planes indicated in the figure, make the ones with the arrows a little longer.

16. Using the previous figure as a guide, create three groups of reference planes and equality dimensions as indicated.

⇨ At the bottom, create five total (for six equal spaces) with the top one longer. Add the equality dimension.

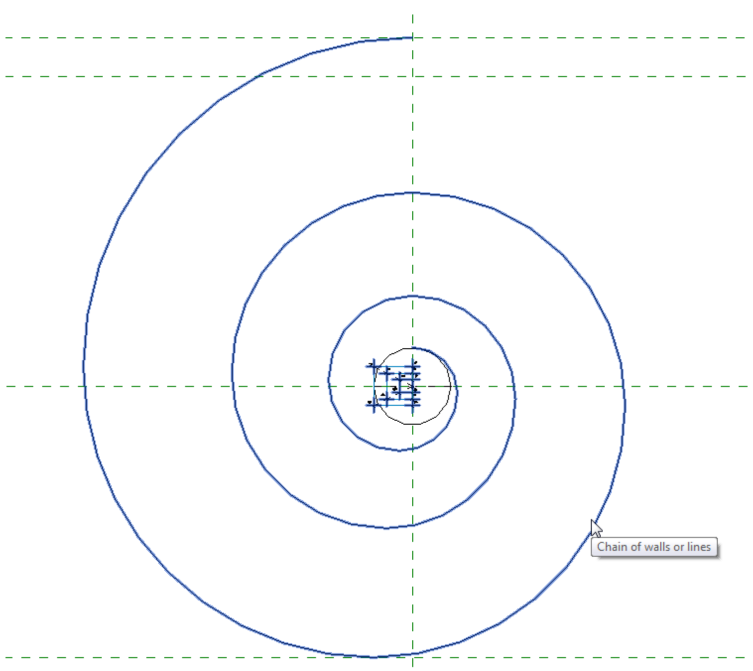
⇨ In the other two areas, add two reference planes (for three equal spaces).

CATCH UP! You can open a file completed to this point named: *Ionic Volute Profile_C.rvt*.

17. Continue the procedure outlined above to add arcs 6 through 12.

Remember, it is best to add each arc one at a time and stop and turn on the center mark and align and lock as you go. Align and lock any time you see an Automatic Sketch Dimension appear.

Be sure to flex often as you work.



Add the remainder of the 12 arcs and flex to ensure everything is working

18. When you complete arc 12, flex again trying several different values like: **0.500**, **0.750**, **1.250**, **1.500** and **2.000**. Return to **1.000** before continuing.

19. Save the file.

CATCH UP! You can open a file completed to this point named: *Ionic Volute Profile_D.rvt*.

Create the Inner Fillet Edge

We are half way there! The twelve arcs we have here are the outside edge of the volute fillet. We need to create a second spiral inside of the first one that will be the inside edge of the fillet. The process is nearly the same. But this time, I'll save you some effort. I built another rig to help us along.

Continue in the same file or open the catch-up file.

1. Select the Volute Eye Rig onscreen.

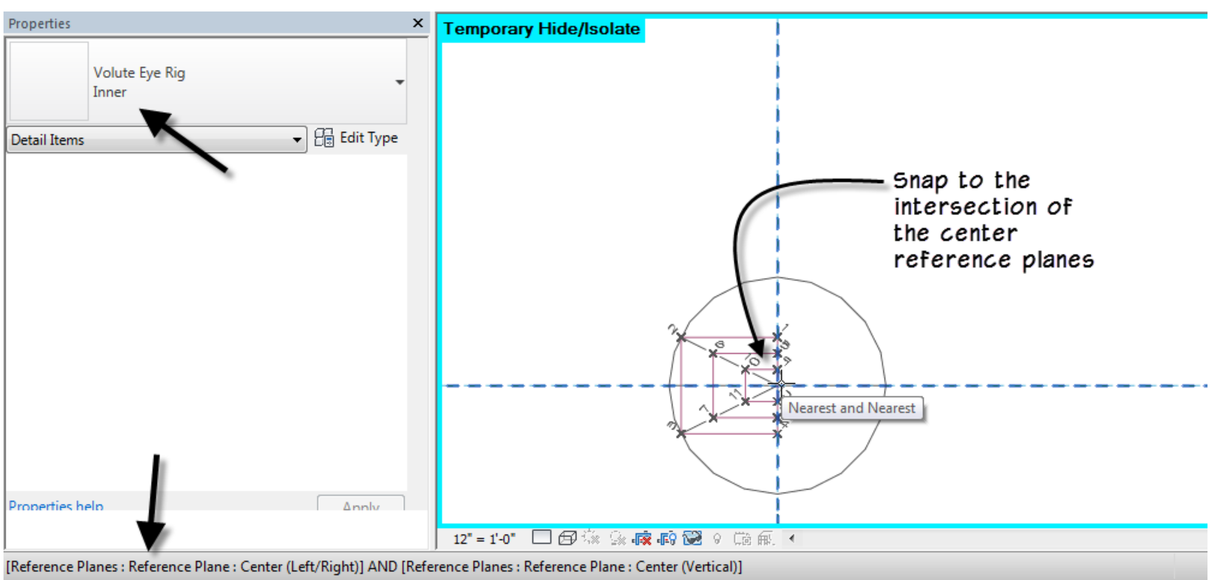
⇨ On the View Control Bar, click the Temporary Hide/Isolate pop-up (sunglasses) and choose **Hide Element**.

2. Highlight one of the arcs, press TAB (they will all highlight) and then click to select the chain.

⇨ From the Temporary Hide/Isolate pop-up choose **Hide Element** again.

3. On the Create tab, click the Detail Component button.

4. From the Type Selector, choose **Volute Eye Rig:Inner** and then click at the intersection of the two center reference planes to place it.



Place another rig for the inner points

5. Align and lock this rig in both directions.

⇨ Select the element onscreen, on the Properties palette, click the Edit Type button and link up the Base Diameter parameter.

We'll also need another collection of reference planes like the ones on the left in the same proportions but slightly closer together. We could simply layout more reference planes, carefully place them in groups, add dimensions, toggle on equality, etc. But I have created another detail item family for this instead. The family contains a copy of all the reference planes and equality dimensions on the left. In fact, I have set up two types in this family like the Volute Eye Rig, so if you wanted to, you could use it for both the inner and outer fillet arcs. But since we already built the outer ones I would suggest leaving them as is. You can use this rig for any future volutes you might need to create however.

6. On the Create tab, click the Detail Component button again.
7. On the Type Selector, choose **Reference Rig:Inner** and then place an instance at the intersection of the two center reference planes. (Same location as the other item).

⇨ Align and lock in both directions. (Pay close attention to the Status Bar to make sure you are selecting the correct alignment edges).

⇨ Select the element onscreen, and then on the Properties palette, click the Edit Type button and link up the Base Diameter parameter.

8. Flex the family. Reset to **1.000** before continuing.

Insert the Reference Rig to control the radii of the inner fillet arcs

CATCH UP! You can open a file completed to this point named: *Ionic Volute Profile_E.rvt*.

Following the procedure above, create all of the arcs. (If you opened the catch-up file, be sure to hide the arcs and Outer Volute Eye Rig again).

Zoom in closely on the Volute Eye detail rig.

9. On the Create tab, click the Line button and then click the Center-ends Arc.

⇨ Snap the center point to the intersection at point 1 on the rig.

⇨ Pull straight up 90° and snap to the topmost reference plane. (Make sure it is exactly 90° before you click).

⇨ Pull straight to the left at 90° again and then click (similar to one created above).

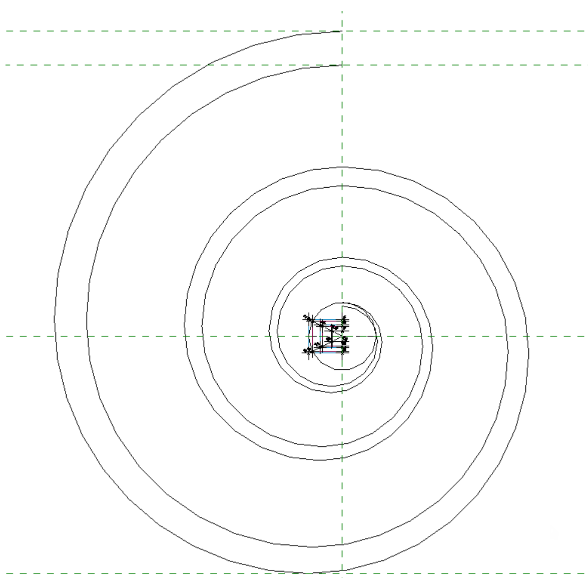
⇨ Enable the center mark and then align and lock the endpoints at each end in both X and Y directions.

You should have a quarter circle centered on point 1.

10. Repeat the process to add arcs 2 through 12.

Remember, take your time. Align everything carefully. Try to eliminate all Automatic Sketch Dimensions as you go. It is possible to draw all the arcs and then align and lock everything, but if you are not extra diligent in doing so, you can easily miss some. So even though it is more tedious, I recommend doing one arc at a time. Save and flex regularly.

11. On the View Control Bar, from the Temporary Hide/Isolate pop-up, choose **Reset**.



All of the arcs complete

12. Save the file.

CATCH UP! You can open a file completed to this point named: *Ionic Volute Profile_F.rvt*.

Complete the Profile

We've completed the hard part. All that remains is to add a few additional line segments to close off the profile. Off to the right in this file is a vertical reference plane named: "Centerline of Column." There is a labeled dimension (X1) connecting this to the Center (Left/Right) reference plane.

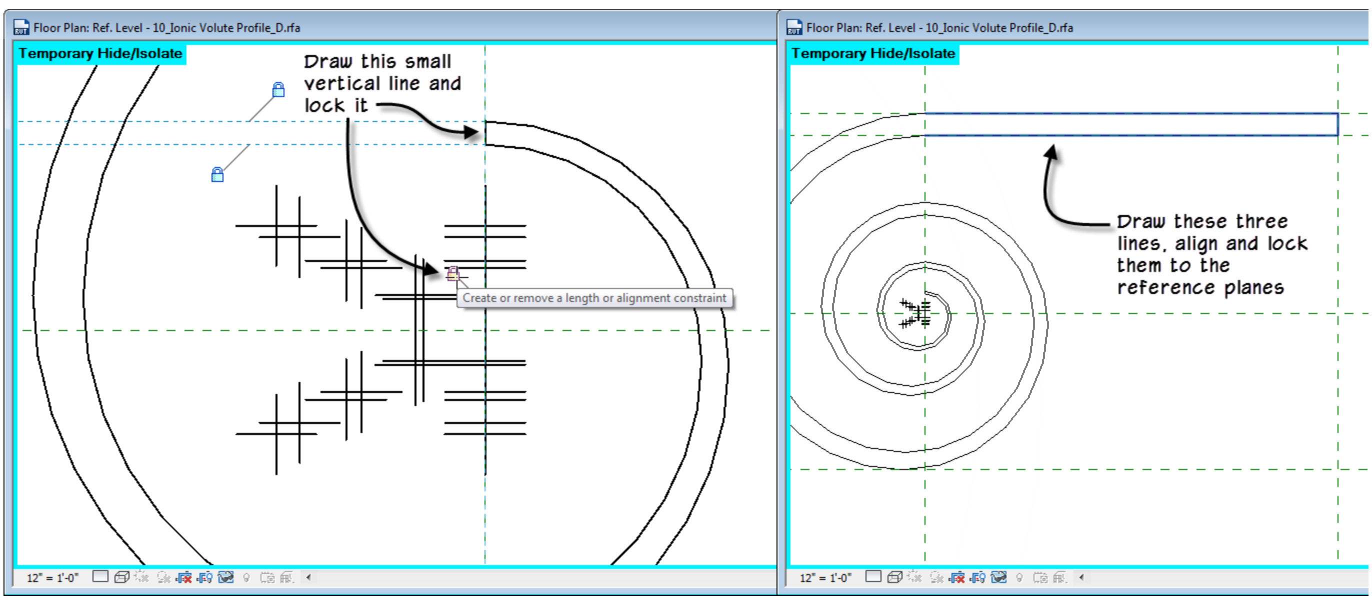
Let's add the remaining lines needed to enclose this profile.

Continue in the Ionic Volute Profile file or open the catch-up file.

13. Zoom in close on the eye.
14. Draw a small straight line to connect the endpoints of the inner and outer arc 12.

⇨ Align and lock it to the vertical reference plane.

TIP: If you have trouble selecting the vertical reference plane to align, use Temporary Hide/Isolate to hide the rig detail families.



Add a vertical line to close the ends of inner and outer arc 12. Complete the horizontal portion at the right

15. Add three straight lines at the right:

⇨ One aligned and locked with the Top reference plane.

⇨ One aligned and locked to the Centerline of Column reference plane.

⇨ The last one aligned and locked to the Fillet reference plane.

16. Open "Family Types" and flex the Distance Center to Eye parameter. Reset to **1.000** before continuing.

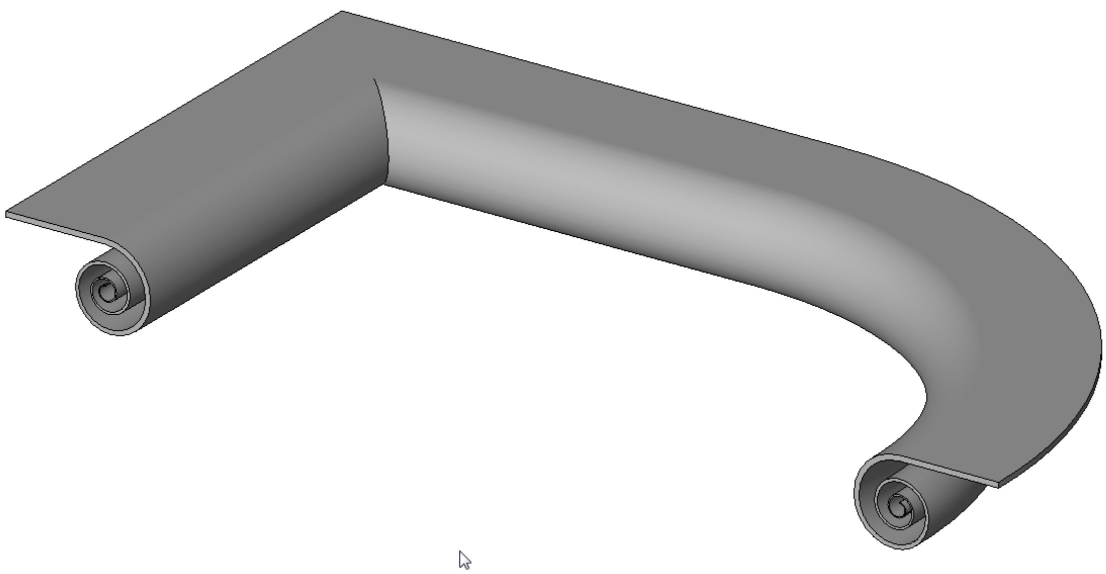
17. Reset the Temporary Hide/Isolate and then save the file.

Congratulations! That was the toughest profile to create in the whole lab!

CATCH UP! You can open a file completed to this point named: *10_Ionic Volute Profile.rvt*.

I have provided a file to flex this in: *Volute Sweep Flex.rfa*.

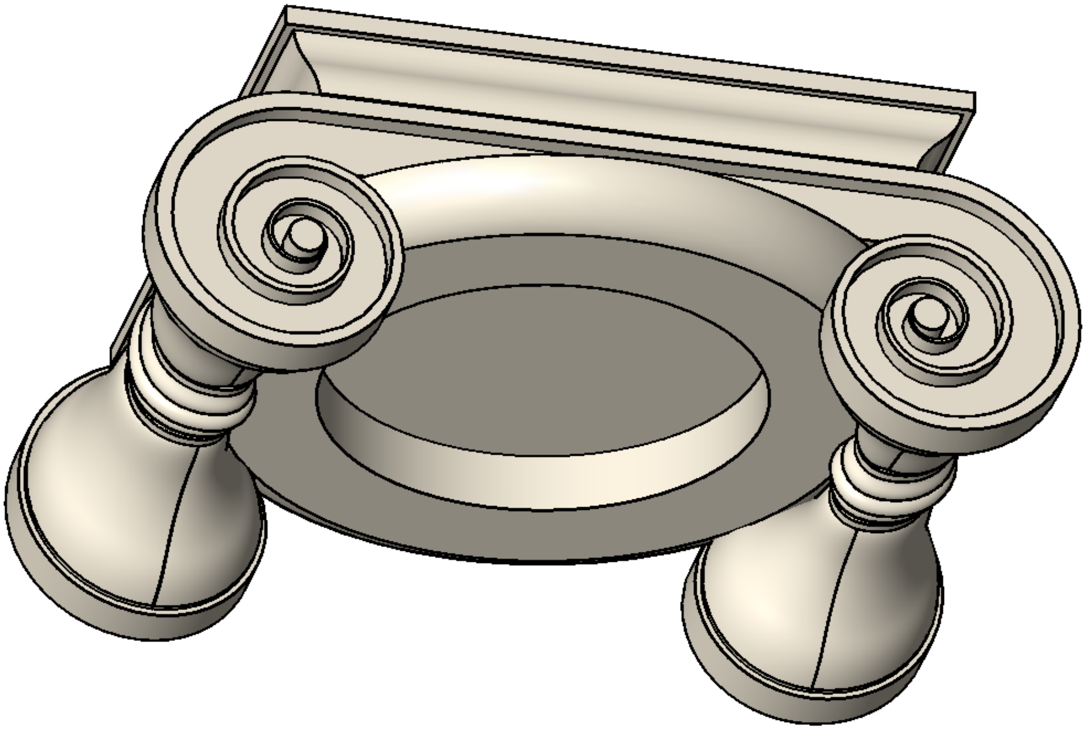
Note exactly the correct application, but it does certainly prove the viability of the profile.



Test out the completed profile in a sandbox file

If you prefer to see it in the proper context, I have provided a completed Ionic capital file for you to play with. Unfortuantely time and space in this already over-packed lab do not allow me to elaborate on how the rest of it was created, but all of the details are in Renaissance Revit!

The file is called: *Finished Ionic Capital.rfa*.



The profile used in a more proper context

Conclusion

If you have hung with me this far you now have all the tools you need to tame even the most unruly of parametric curves in the family editor. Even though I focused on the traditional environment, nearly all of the techniques (save for using generic models in place of profiles) work equally well in the conceptual massing environment. Enjoy!

Want to learn more?

I have a video course related to the material in this lab, visit: www.lynda.com/paulaubin and check out: **Revit: Family Curves and Formulas**

And of course there is my book: ***Renaissance Revit: Creating Classical Architecture with Modern Software*** as well.

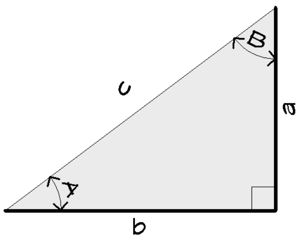
Thank you!

Trigonometry Cheat Sheet for Revit

Which parts are known?

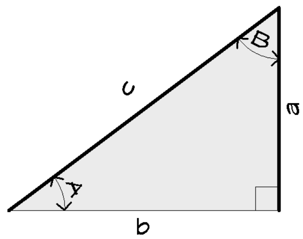
Two Sides

Known: a & b



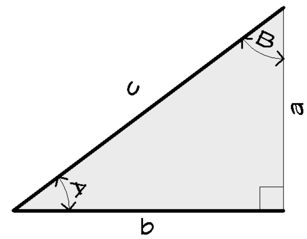
c = sqrt(a^2 + b^2)
A = atan(a / b)
B = atan(b / a)

Known: a & c



b = sqrt(c^2 - a^2)
A = asin(a / c)
B = acos(a / c)

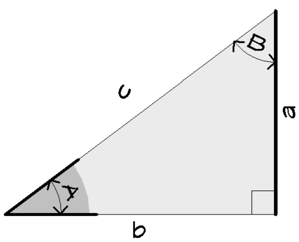
Known: b & c



a = sqrt(c^2 - b^2)
A = acos(b / c)
B = asin(b / c)

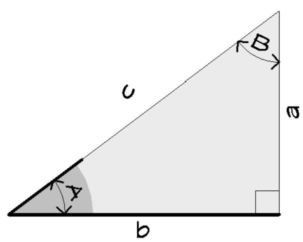
One Side & One Angle

Known: a & A



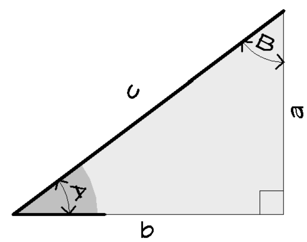
b = a / tan(A)
c = a / sin(A)
B = 90° - A

Known: b & A



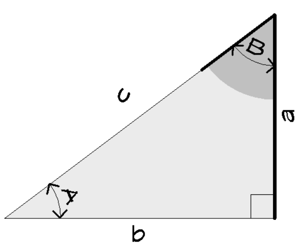
a = b * tan(A)
c = b / cos(A)
B = 90° - A

Known: c & A



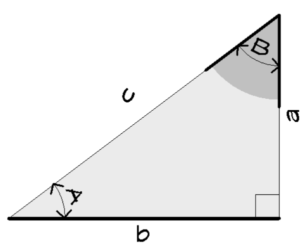
a = c * sin(A)
b = c * cos(A)
B = 90° - A

Known: a & B



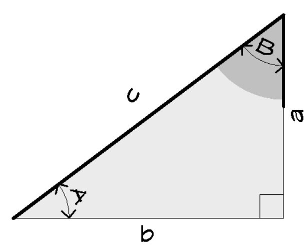
b = a * tan(B)
c = a / cos(B)
A = 90° - B

Known: b & B



a = b / tan(B)
c = b / sin(B)
A = 90° - B

Known: c & B



a = c * cos(B)
b = c * sin(B)
A = 90° - B