AUTODESK **UNIVERSITY** 2014

Parametric Classical Orders—A Journey with Revit Family Editor

Paul F. Aubin – www.paulaubin.com

AB5129: Years ago I began exploring the possibilities of creating the classical orders of architecture in Revit software. Parametric families are the cornerstones of Revit software, and this project presented an ideal way to push Family Editor to its limits. Challenges were many, but there were plenty of successes as well. This session is part case study (of the journey I took from the original idea all the way to the publication of a book) and part tutorial (to show the "nuts and bolts" of how we built the families), all demonstrated directly in Revit software. In this session I'll talk about scaling, content reuse, profiles, moldings, formulas, nested components, and complex forms in both the traditional and adaptive component family editors. I'll show the final successful versions and a few not-so-successful earlier versions as well. If you're interested in historic architecture and/or classical form, or you just like pushing family editor to extremes, then this session is sure to please.

Learning Objectives

At the end of this class, you will be able to:

- Use Revit software to emulate historic architecture
- Understand the process of how to create traditional forms in the family editor
- Componentize your families: reuse the parts and pieces
- Use formulas and other advanced techniques to build complex families

About the Speaker

Paul Aubin is the author of many Revit software books, including the widely acclaimed Aubin Academy Mastering Series, the all-new Renaissance Revit: Creating Classical Architecture with Modern Software, and the Revit training videos at www.lynda.com/paulaubin. Paul is an independent architectural consultant providing implementation, training, and support services for Revit Architecture software. Paul's involvement in the architectural profession spans 25 years, with experience in design, production, CAD management, mentoring, coaching, and training. He's an active member of the Autodesk, Inc., user community, as well as a top-rated repeat speaker at Autodesk University, Revit Technology Conference, Building Information Modeling (BIM) workshops, and the University of Minnesota. 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, and he lives in Chicago with his wife and 3 children.
Contact Paul at: www.paulaubin.com. Twitter: @paulfaubin

Don't wait for the movie…

Today I want to talk about the journey that led to the creation of my newest book:

Renaissance Revit: Creating Classical Architecture with Modern Software.

About the book:

This book tackles a subject never before approached in the world of Revit training manuals. In this exquisite hands-on guide, take a virtual tour through the history of architecture in a way like no other: hands-on! Paul's book is a tutorial of epic proportions that does not just talk about creating Doric, Ionic and Corinthian columns; it walks through the process step-by-step! This book brings together three of the author's favorite things: architecture, Revit and history. Following along with Paul as your guide, you'll learn the family editor in a completely unique way: building the classical orders!

Learn more at:

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

But there are some movies actually: http://youtu.be/GlWjK4DfBYs

And an video entire course at:
http://www.lynda.com/Revit-Architecture-tutorials/Revit-Family-Curves-Formulas/148423-2.html

More on that later... Let's get started.

Introduction

I am a long-time fan of Classical Architecture. I have always been fascinated by classical forms, history in general, Roman times in particular, and of course architecture. I am also pretty interested in technology in general and CAD and BIM software is particular. Therefore, the challenge of building examples from classical Architecture as fully parametric 3D forms proved too tempting for me to resist.

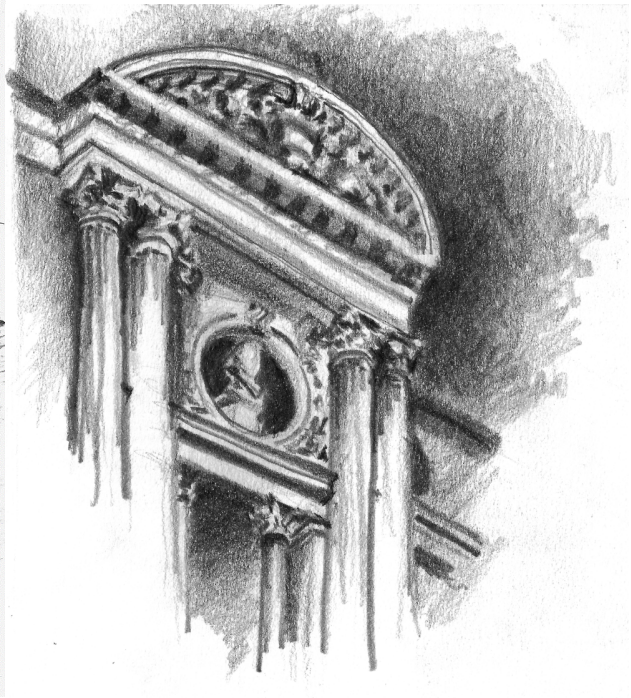
So in an attempt to marry my many interests with Revit's parametric qualities the result was a journey that spanned many years and culminated with the publication of Renaissance Revit. I ultimately called the book "Renaissance Revit" not only because it was a catchy title, but also because many of the Renaissance authorities looked to Roman classicism as their model to reinvent Europe's cityscapes. (I did include a few Greek examples in the book as well, but most of my forms are Roman and/or Renaissance style forms).

This class is about the process I went through. It is about the journey from an interest in architecture and classicism to a tangible result in the form of a physical book (and even some 3Dprinted models). If you don't have the book, don't worry. It is **not required** to appreciate the class. My goal is really to discuss a few of my interests (that perhaps you share as well) and talk about how I was able to use modern tools (mostly Revit) to help me explore, understand and create my own interpretations of the orders of Classical Architecture.

Sketches

The process for me began back in school with a semester of study abroad. We use sketches to help us understand our built environment, and in my semester in Europe, we did a lot of sketching! The

process of sketching allows you to understand what you are seeing much more than simply studying it in pictures from books or lecture slides. These are some examples from my college sketchbook. My sketch of Palladio's Villa Rotunda and a Roman aqueduct.



And one more: a sketch of the Paris Opera. These were drawn in the Summer of 1989.

Source Materials

In addition to first-hand observation and sketches, I have many source materials that I have referred to over the years for understanding classical forms, rules and proportions. My primary resource is a wonderful book by Robert Chitham. He has distilled the masters and their treatises into a very clever metric system of measurement.

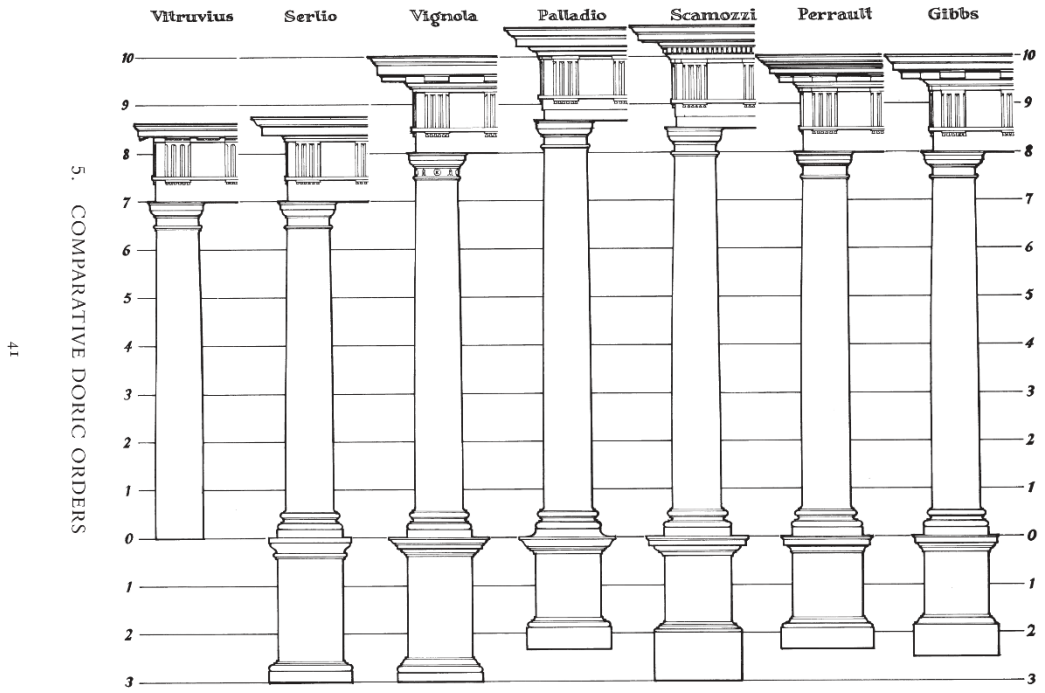


Figure 1 Sample plate from Chitham

Chitham used many sources to arrive at his own "idealized" form and proportion for each order. Figure 1 shows his comparison of the Doric order as presented by each of the masters. He then presents these idealized proportions in decimal measurements. This makes it very easy to transfer the measurements to formulas in the family editor (see Figure 2).

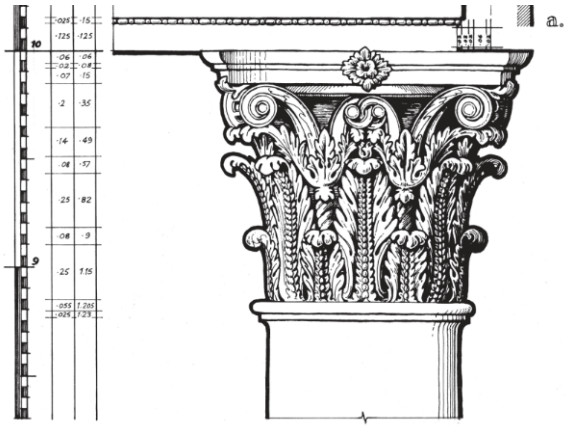


Figure 2 Chitham's Corinthian capital

I referred to several other books in my research. Another excellent one was: “The Architectural Orders of the Greeks and Romans: 100 Plates” Folio Edition by J.M. Mauch. This book contains some extraordinary drawings of many existing classical works. I used several other books as well. A complete bibliography is included in my book.

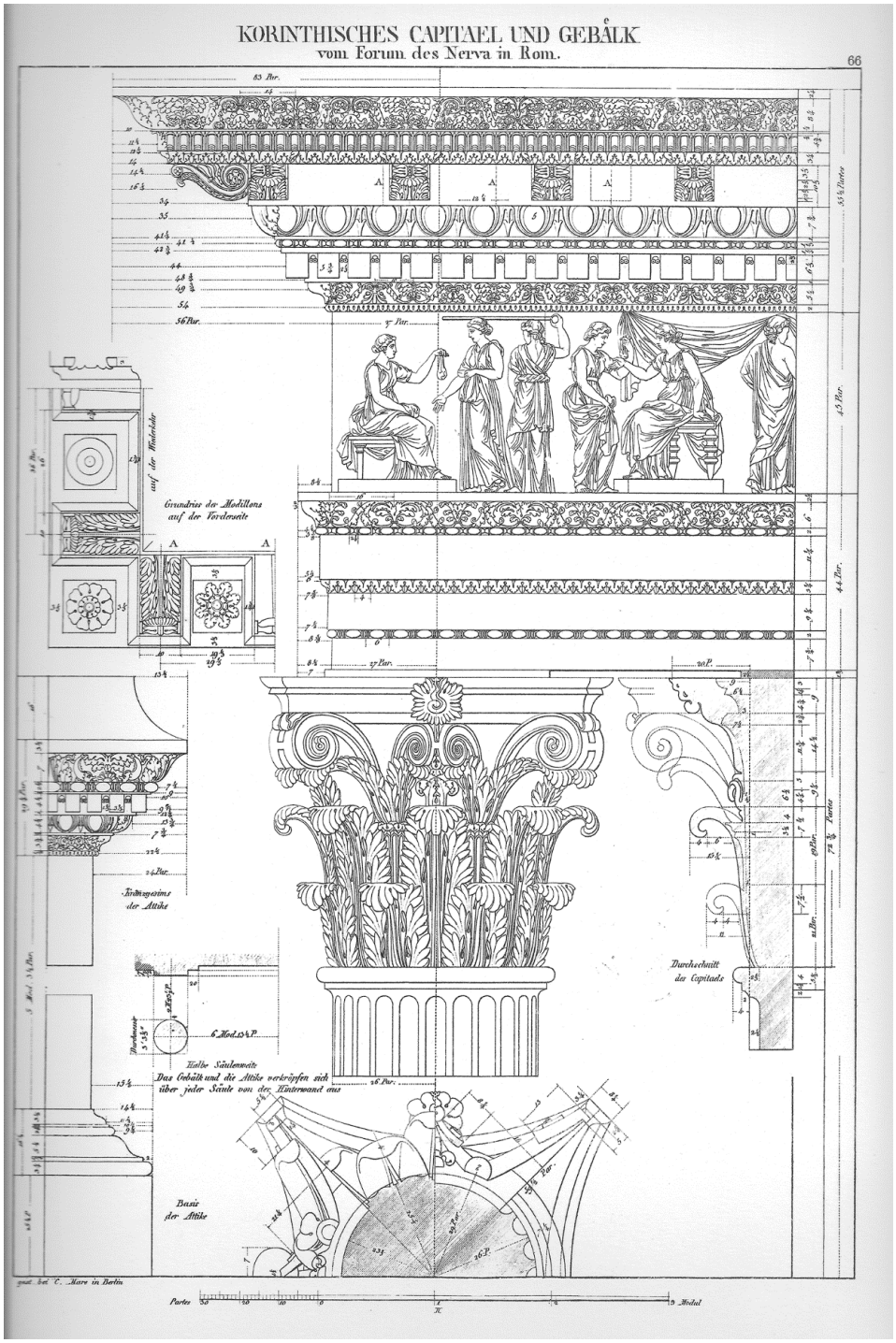


Figure 3 One of Mauch’s Corinthian capitals

I also used MANY online sources. These are just a few of my favorites:

- http://buildz.blogspot.com/ Zach Kron
- http://grevity.blogspot.com/ Andy Milburn
- http://revitswat.wordpress.com/ Kelvin Tam
- http://boostyourbim.wordpress.com/ Harry Mattison

Figure 4 Shades of Grey "Grevity" blog by Andy Milburn

There is a terrific blog (see Figure 5) called "**Shades of Grey**" and its author Andy Milburn has done some tremendous studies of architecture using Revit. So if classical is not your thing, I still think that process I have outlined here is well worth the effort. Andy's blog is a treasure trove of information both practical and philosophical. I highly recommend it. Andy talks about Revit being his "BIM pencil of choice" and I love that analogy. Andy also wrote the foreword to my book! So I owe many thanks to Andy for that as well.

Zach Kron's Buildz blog is also must reading. Zach has done tremendous work with the Revit massing environment, rationalization of 3D form, fabrication and even an awesome annual virtual pumpkin carving contest. Definitely worth checking out. Recently Zach has shifted his considerable talents to dynamo. Dynamo is a visual programming language that works as a plug in to Revit. Powerful stuff.

Goals of my Project

As I noted previously, I have been working on this in one way or another for many years. In the previous topic, I discussed my sources. One of my sources: Andy Milburn (author of the Grevity blog) has on more than one occasion referred to his use of the computer and Revit as simply a "digital pencil." He talks about using this digital pencil to help him explore ideas and understand forms. This is very much like how we as Architects have been trained to use sketches. Using the computer this way is how I began the process of exploring Classical Architecture. I think originally, I thought that if I could draw the forms on the computer, it would provide similar benefits to what I experienced out sketching in the field across the great cities of Europe. At some point, the process took on a life of its own and several goals began to materialize. So as I began to solidify what I was attempting, a list of what I hoped to achieve began to become evident:

Accurate Classical Proportions and Forms

- Based on various books and research noted above

Fully scalable 3D models

- Adjusts to different sizes based on the size of the Base Diameter
- All pieces scale proportionately
- Optionally add other parametric behaviors like swapping nested components

Incorporate 3 Levels of Detail

- 3D models are heavy. I wanted to be able to use these components in projects.
- Add a medium (and low detail) model
- Add a coarse (sometimes 2D only) representation

Publish in Book and/or Video form

- As an educator, I am very interesting in sharing my findings
- This is why I published the book
- Video likely to follow the book, (already in conversations with lynda.com)

Using AutoCAD

In reality, the project began long before Revit. Many years ago (more than ten) I started it with AutoCAD Architecture, (then Architectural Desktop). I have been using Revit for this project for at least five years.

If I could just have the functions of both programs in one…

NOTE: THERE ARE NO DWG FILES IN MY REVIT FAMILIES. MY OWN PERSONAL "RULES" PREVENT ME FROM USING AUTOCAD FILES IN REVIT FOR THIS PROJECT.

Figure 5 This is a photo I took when I was there in 1989 for semester in college

One of my favorite buildings is the Piazza del Campidoglio by Michelangelo. I love the way that Michelangelo used classical forms in an entirely "modern" way. We could do an entire semester on the subject, and I have no doubt that there are courses out there that do just that, but for the purposes of this discussion, it was simply the subject of my first attempt. The Capitoline Hill seemed to me the perfect challenge.

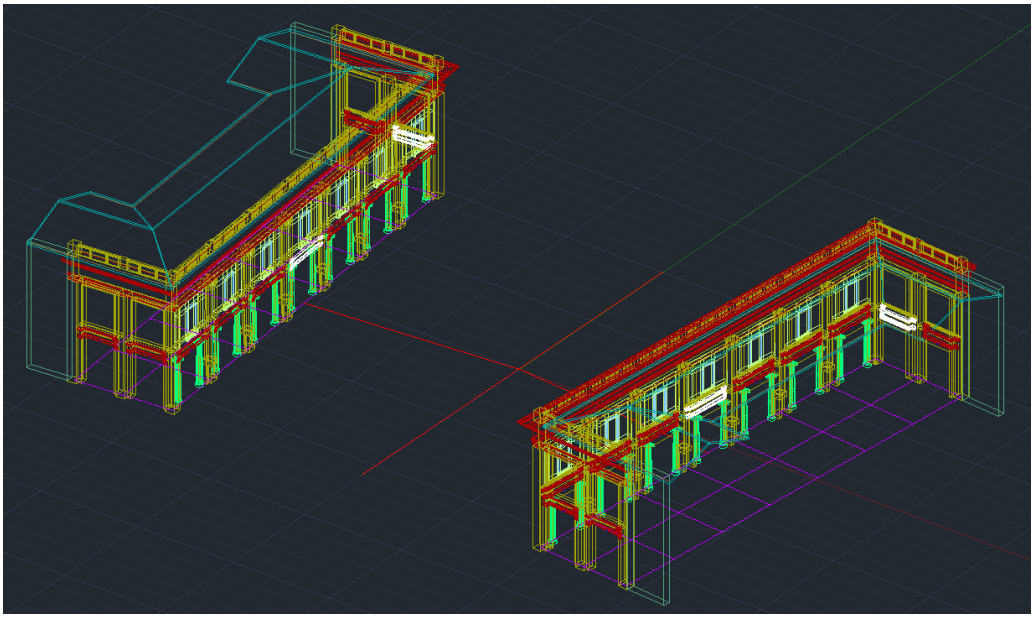


Figure 6 Rough form of the Campidoglio in AutoCAD Architecture

I did get around to roughing out the Campidoglio (see Figure 6). But life got busy after that... The Corinthian column for example would have to wait some time. In fact (spoiler alert) it would have to wait many years later for me to revisit the project in Revit. And even then, the Corinthian I developed in Revit is more generic and I have not yet tried to build a Revit version of the Campidoglio.

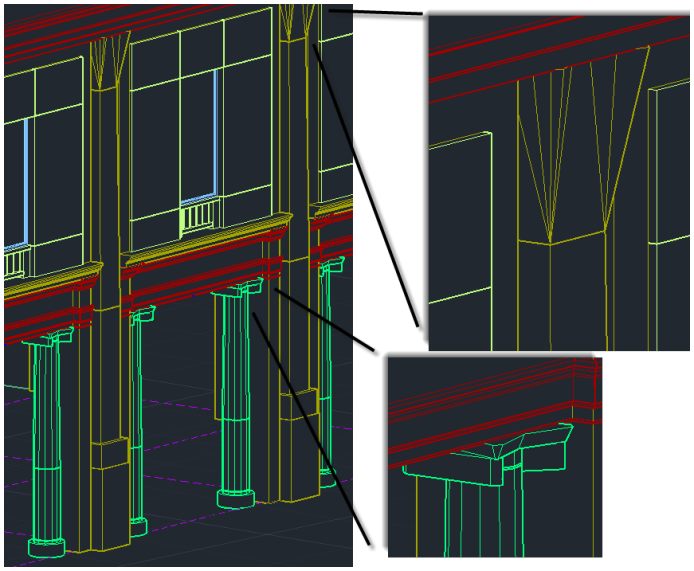


Figure 7 Low detail versions of Ionic and Corinthian

Here are some enlarged views of the low detail version. You can see that the Corinthian never got beyond the low detail stand-in in the AutoCAD model (see Figure 7). (Revit version to follow).

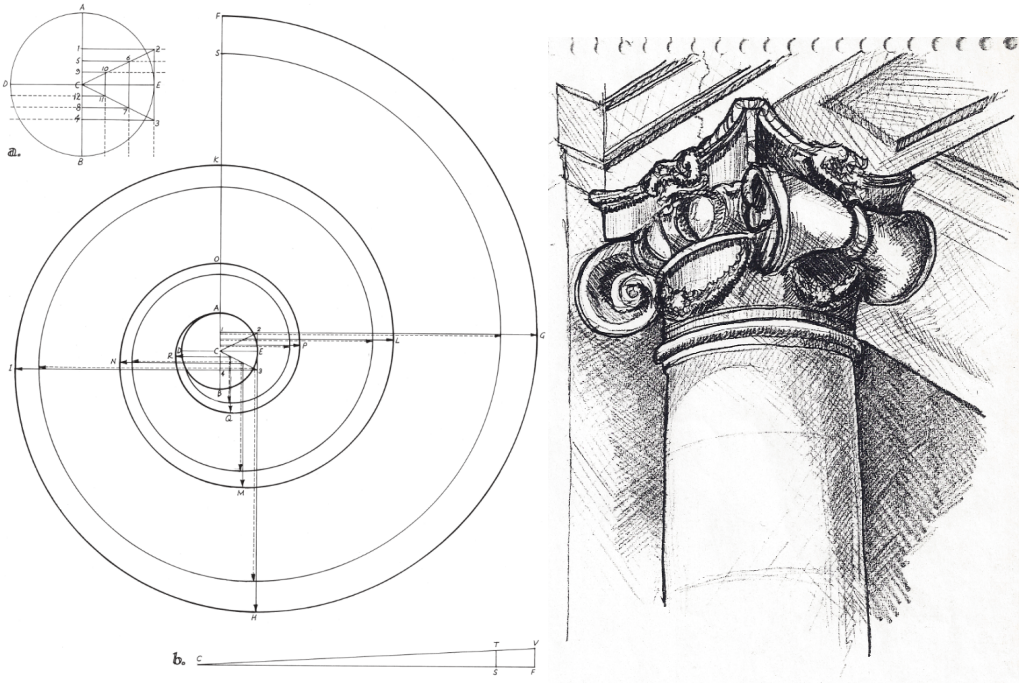


Figure 8 Chitham's Ionic volute construction and my sketch from the Campidoglio

I was also itching to dig into Chitham's book, so what better place to start than an Ionic volute? He presents a nice way to construct it which was easy enough to replicate in AutoCAD (see Figure 8). So off I went to build the volute and my first classical form in CAD. I even had some of my own sketches (right side) from that college visit to Rome that I noted above (now 25 years ago!) Where did the time go?

Working carefully from both Chitham and my own sketch, I constructed several profiles, which ultimately became "lofted" together to form the volute and horn.

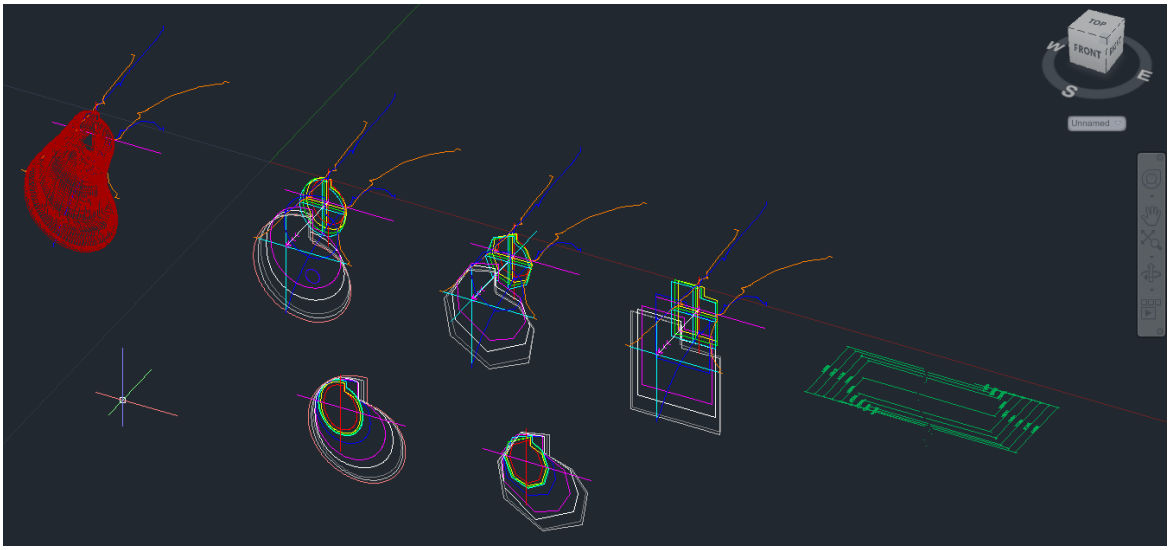


Figure 9 AutoCAD profiles used to create 3D lofted forms

Now I should note that even though I was using AutoCAD, it was actually AutoCAD Architecture, so even at that time (probably 10 or more years ago), I was trying to make the forms parametric. In other words, classical form being all about proportion, it was important to make it scale correctly. Now, scaling in AutoCAD is no problem. You can easily scale a block or an object. But I also was looking to take advantage of ADT's levels of detail, which, much like Revit allows the same object to have low, medium and high detail representations. To create the “lofted” form, I used a structural member which can have many profiles. You can see a little of that here in the three sets of profiles I devised (see Figure 9).

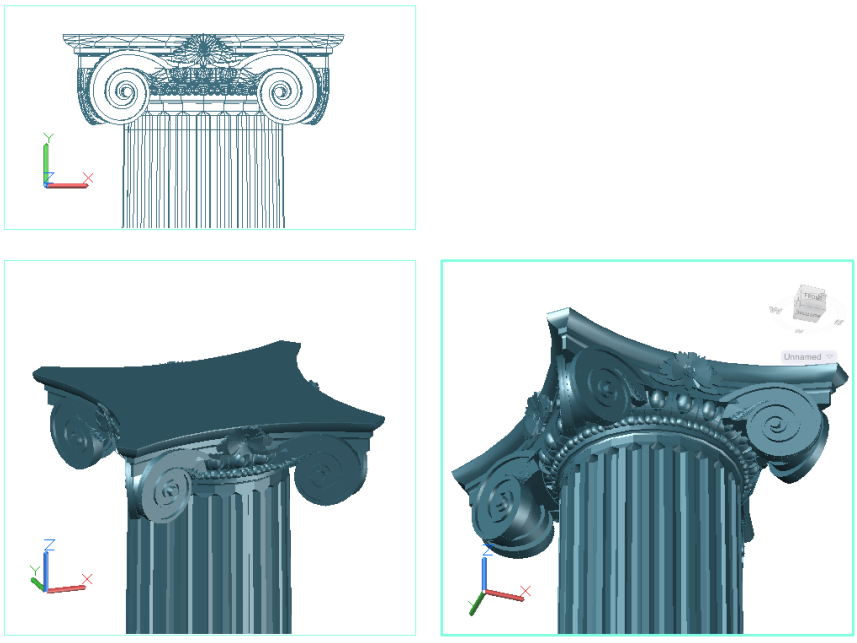


Figure 10 An AutoCAD Architecture Structural Member used to represent an Ionic Column

In Figure 10 is another version of my AutoCAD Architecture Ionic column showing fine detail (This is actually a variation of the one I used in the Campidoglio). As you can see it is a four sided version instead of a two-sided one. The final result was an ADT object (Structural Member) that incorporated AutoCAD display blocks and other settings native to the structural member object. The low detail version of this column can be seen above in Figure 7.

Figure 11 This is one of my earliest AutoCAD Architecture attempts brought to life on a 3D printer

Unfortunately, I have not had a chance to rebuild the Campidoglio in Revit yet. With the basic orders complete in Revit, I have many of the "raw" materials at this point. But Michelangelo stylized the orders quite a bit. So to build the model accurately will take more time and research. It's certainly on my list! But I need to go back to Rome.

Renaissance Revit

Fast forward a few years and into the Revit era. I had not been using Revit for very long when the thought of returning to my classical project occurred to me. One of Revit's big strengths is parametric modeling. So, it seemed like a natural for this project. Little did I know what I was getting into…

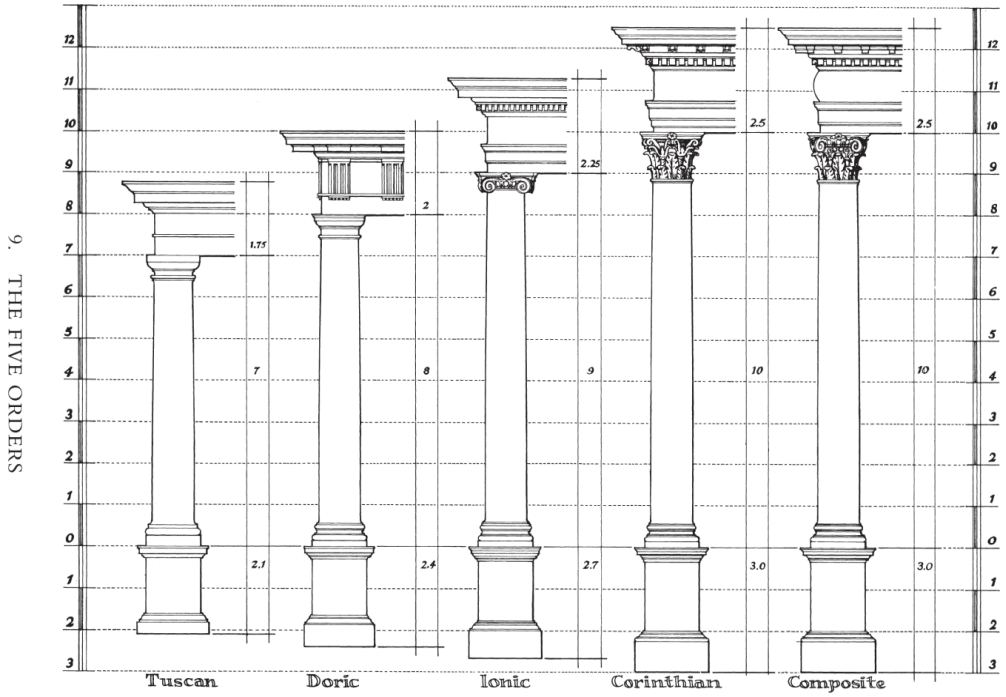


Figure 12 Chitham's five orders

I certainly would like to revisit the Campidoglio and build it completely in native Revit geometry. It is still on my bucket list. But I thought I would start simpler this time. Returning to Chitham, he has a very nice complete roundup of the entire classical vocabulary. His Plate 9 in particular shows each of his "idealized" orders presented together for comparison (see Figure 12). I decided to take a similar approach.

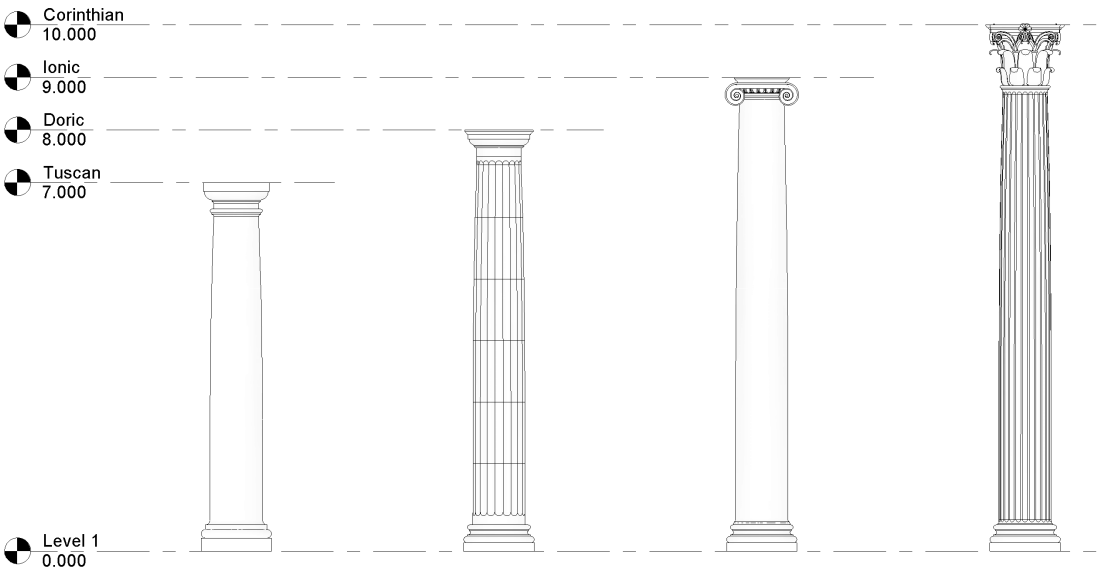


Figure 13 My four Revit orders

Figure 13 gives a look at the four orders that I create; step-by-step in the book Renaissance Revit. (I have omitted Composite in my book).

Revit is often regarded as "better" than AutoCAD, but better can be a relative term. In general the approach to things in Revit is quite different than in AutoCAD. But “different” does not always mean “better.” Overall, Revit is certainly better tool to produce buildings for most projects, but it is not better than AutoCAD in *all* ways.

One very simple assumption permeates ALL procedures in Revit. In general, you are expected (and encouraged) to build things in your Revit model AS they will be built in real life. As a consequence, something as simple as being able to use the scale command to change the size of basic model elements is NOT available in Revit!!! (see Figure 14)

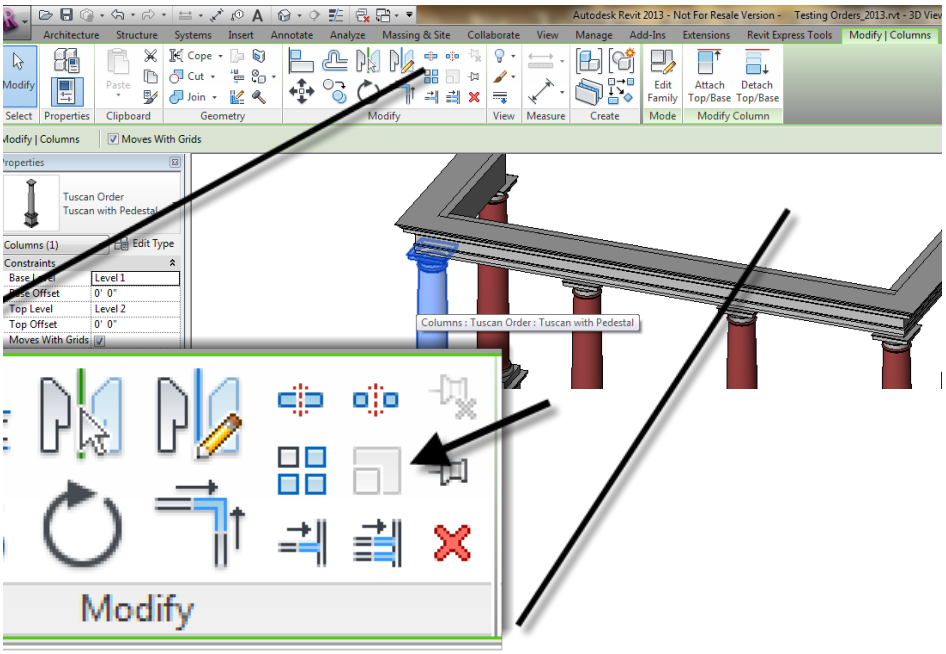


Figure 14 You cannot use the scale command on model elements in Revit

This job just got a whole lot more complicated…

For most of the standard building components that we use in buildings today, the reasoning behind this Revit "limitation" or "prohibition" is quite sound. Rather than scale something, we change its dimensions to the "correct" or “desired” size. However, when the "correct" size is NOT fixed, and in fact is proportional to other key dimensions in the model, then the inability to easily scale items becomes a rather frustrating limitation.

However, undeterred, I pressed on. I do love a challenge…

You can break down most physical elements into a series of simpler forms. These can even often be simple boxes. It is very easy to control the length, width and height of a box in Revit: simply add parameters for each dimension.

To control all three together, use a third parameter and a formula! (see Figure 15)

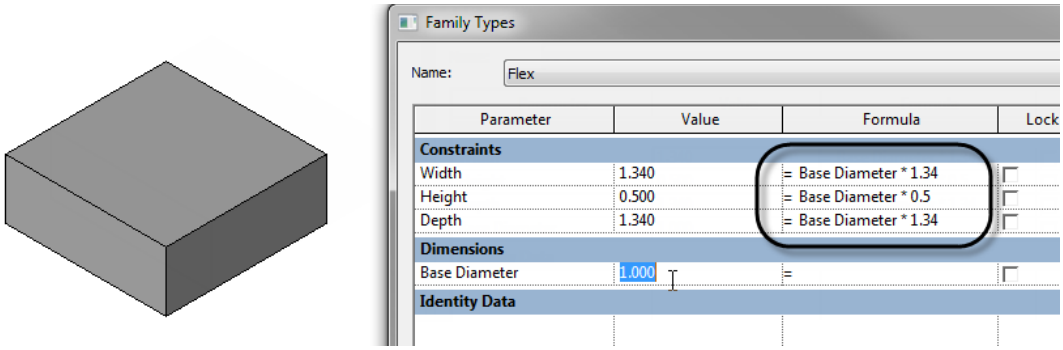


Figure 15 Using formulas to control the proportion of parameter values

So the easiest solution to the inability to scale most component families is to simply tie all the dimensions together and drive them from a common proportion parameter. Simple right? We'll see…

If you combine a few boxes you have a schematic of your column. Each of these boxes is in turn controlled by similar formulas driven by the same controlling parameter (see Figure 16).

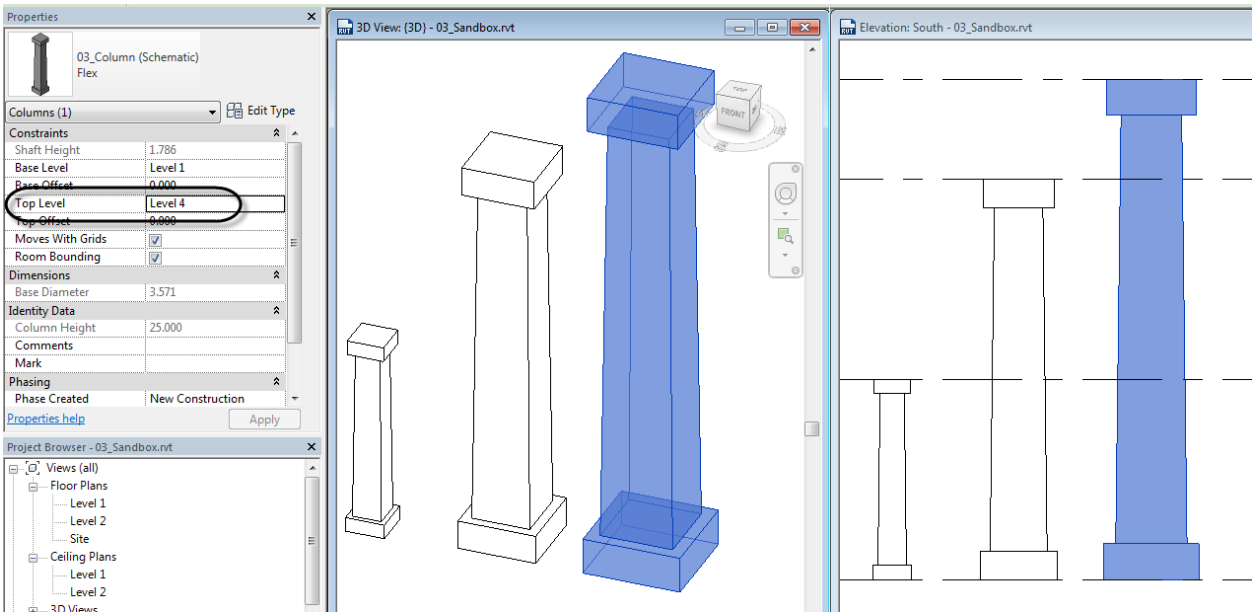


Figure 16 Creating a schematic column from a series of parametrically driven boxes

I took the schematic column a bit further and use the overall height of the column to drive the size and proportions of the nested components. This is accomplished by measuring the height between the two levels (base and top level) and then using this value in the formula to drive the other dimensions. For example, in a Tuscan column, we divide the height by 7 while for a Corinthian we divide by 10.

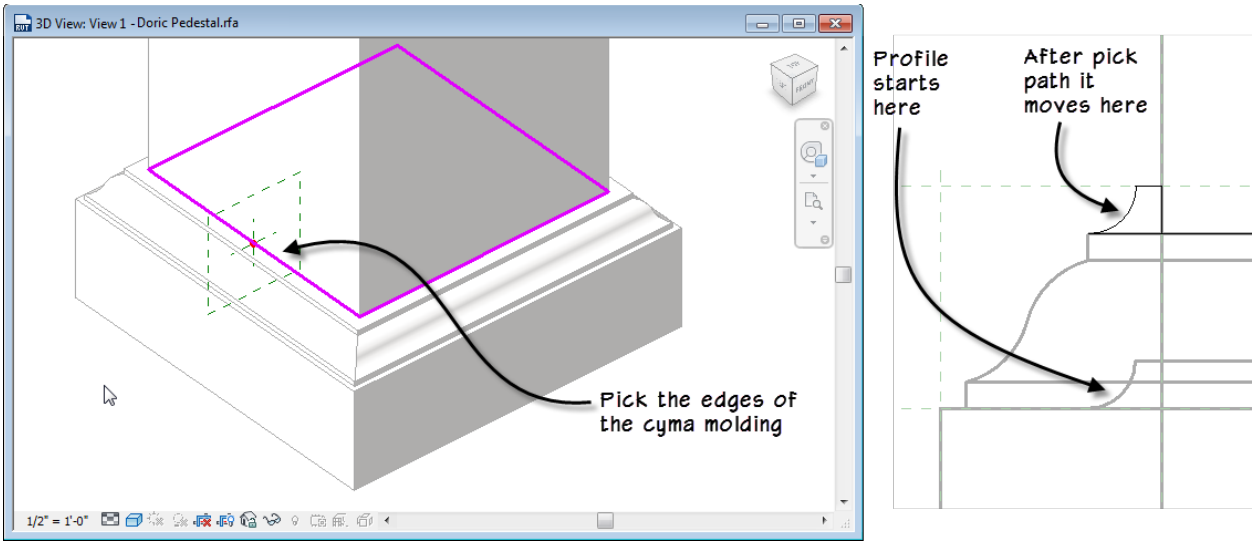


Figure 17 Adding moldings and other details for medium and fine versions

Supplement this with some parametric moldings (Figure 17 uses a sweep) and you are on your way!

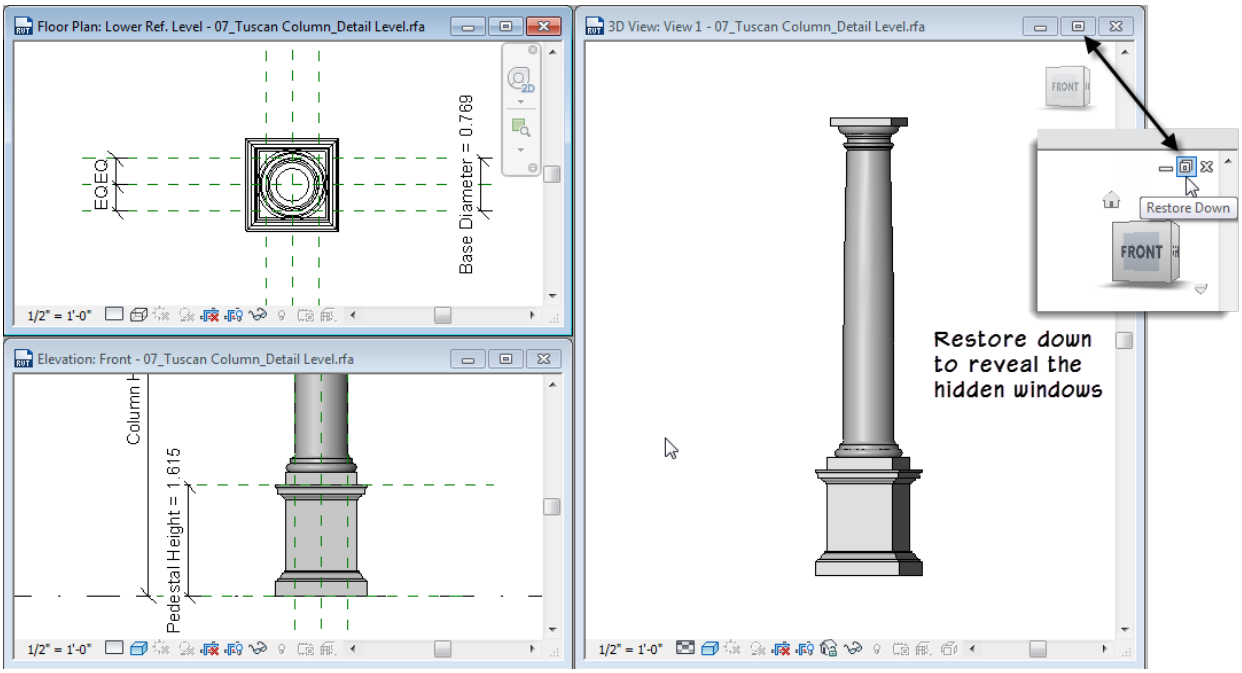


Figure 18 A completed Tuscan order built up from smaller parametrically driven pieces

The final result (see Figure 18) is simply a collection of the smaller parametric pieces: all of them tied back to a single Base Diameter parameter!

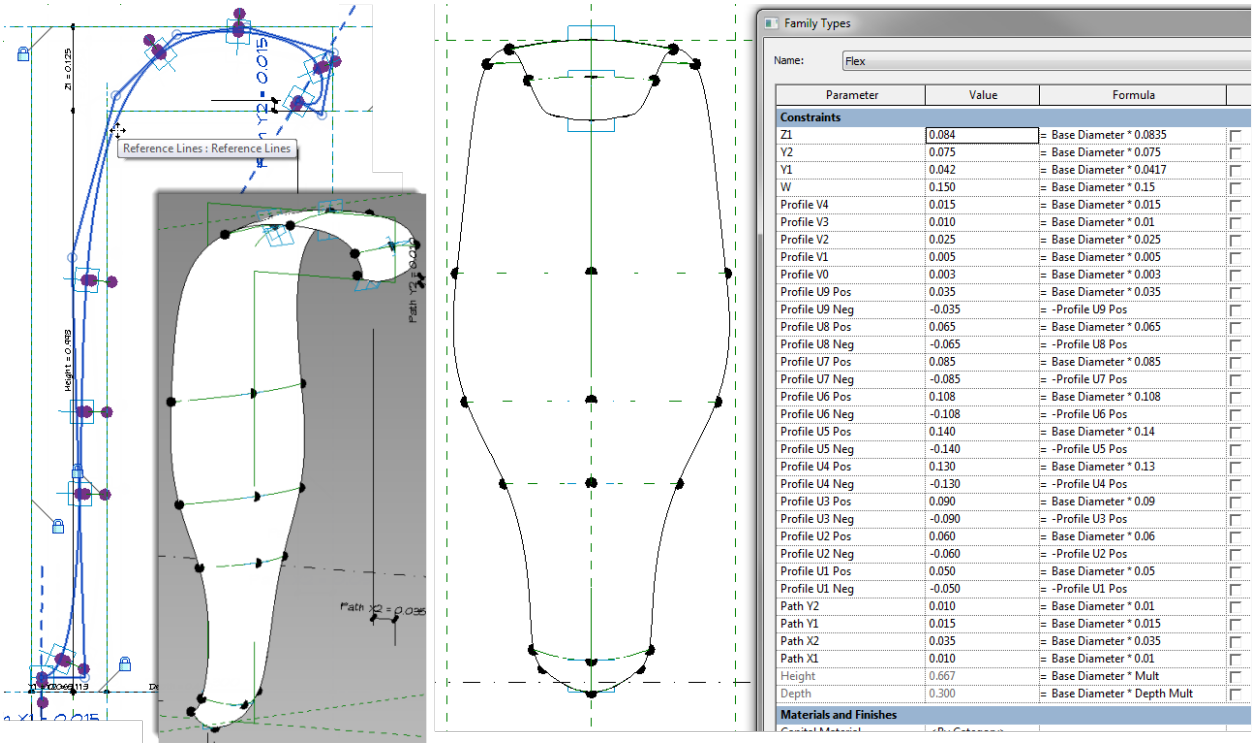


Figure 19 The pieces of the higher orders like the Corinthian use considerably more parameters

The basic orders are easily accomplished in the traditional family editor. However, some of the more ornate and complex forms (like the Corinthian) require adaptive components in the massing environment to adequately represent some of the more organic forms. This can also mean that we often end up with more complex constructs and many more parameters (see Figure 19).

Summarizing the Challenges:
Several challenges had to be addressed when beginning the project in Revit.

- ⇨ Scaling Geometry is Tricky
- ⇨ Parameters offer a solution, but they can get complex
- ⇨ You have to be comfortable in both the traditional and massing family environments

Taming Curves

Backing up a bit, let me elaborate a bit more on why I decided to write a book on creating Classical Architecture with modern software. I am not trying to revive classicism or suggesting that you should be interesting in doing so. I am very interested in classicism to be sure, but my reasons are more academic in nature. There is a great deal of complexity in classical forms. And building such forms in today's 3D software is actually quite challenging. In an environment where most of the education materials available to learn our software mostly use simple (and sometimes simplistic) datasets, I was really very interested in creating not only a more complex dataset, but one that would have immediate and universal recognition and appeal. Whether you like Classical Architecture or not, you cannot deny its ubiquitousness, nor the complexity of its forms. Even the simple orders like Tuscan and Doric are not that simple when you really dig into them. Recall that one of my early goals for the project is education

and combine this with these other points and you can hopefully begin to see why tackling the creation of parametric classical orders in the Revit family editor proved so compelling a project for me. Everything I have attempted to do, and everything I have documented can be applied to any kind of Revit family. There are no strategies or techniques here that are in any way limited to Classical Architecture. Please keep this in mind as we continue the explorations that follow.

With that bit of rationale out of the way, consider the topic at hand. Most classical forms use curves. They use common moldings described by curves both simple and complex. There are arches, curved colonnades, balustrades, etc. Therefore, making curves flex in the family editor consistently and reliably, regardless of the family you are creating, is an absolutely essential family editor skill that you must master if you hope to build anything beyond the basic box.

There are several strategies that we can use to make our curves flex properly in the family editor. The trick is to ensure that they flex proportionately and that the curves do not become malformed in the process. If you have ever tried to do this in the family editor before, you know that this is not always as easy as it seems.

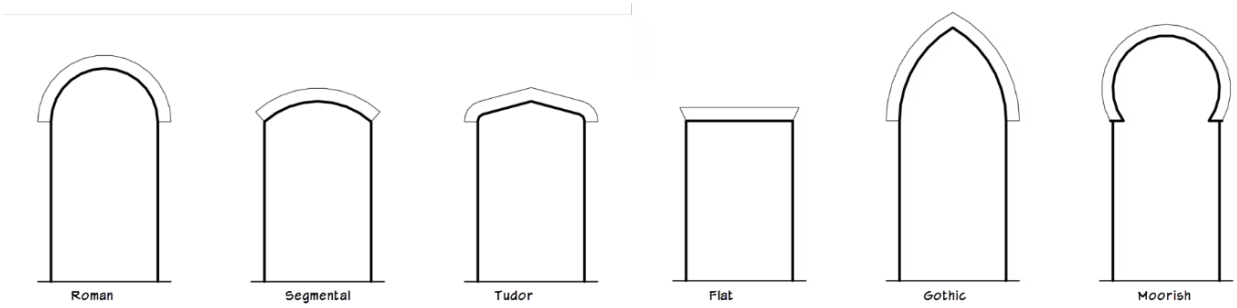


Figure 20 A very common need for parametric curves occurs when building arches

For an example, let's consider some very common arch shapes (see Figure 20). Many buildings use arches so it is a good example for us to consider. But it does not really matter what the form is; to "tame" a curve in Revit, you need to understand what drives the curve and make sure that whatever this is, that it is properly constrained. Consider a simple arch like a Roman, Segmental or even Moorish shape. In all three of these you have a simple curve that has two endpoints and a radius. If you can adequately constrain each of these items, you will always be able to reliably flex the arch without its breaking.

The trick to controlling an arch form or other similar curve is to lock both endpoints to both a horizontal and vertical reference plane and then dimension from the spring line *directly* to the arc itself. Normally you would avoid dimensioning directly to the geometry, but in the case of curves it can be quite effective (see Figure 21).

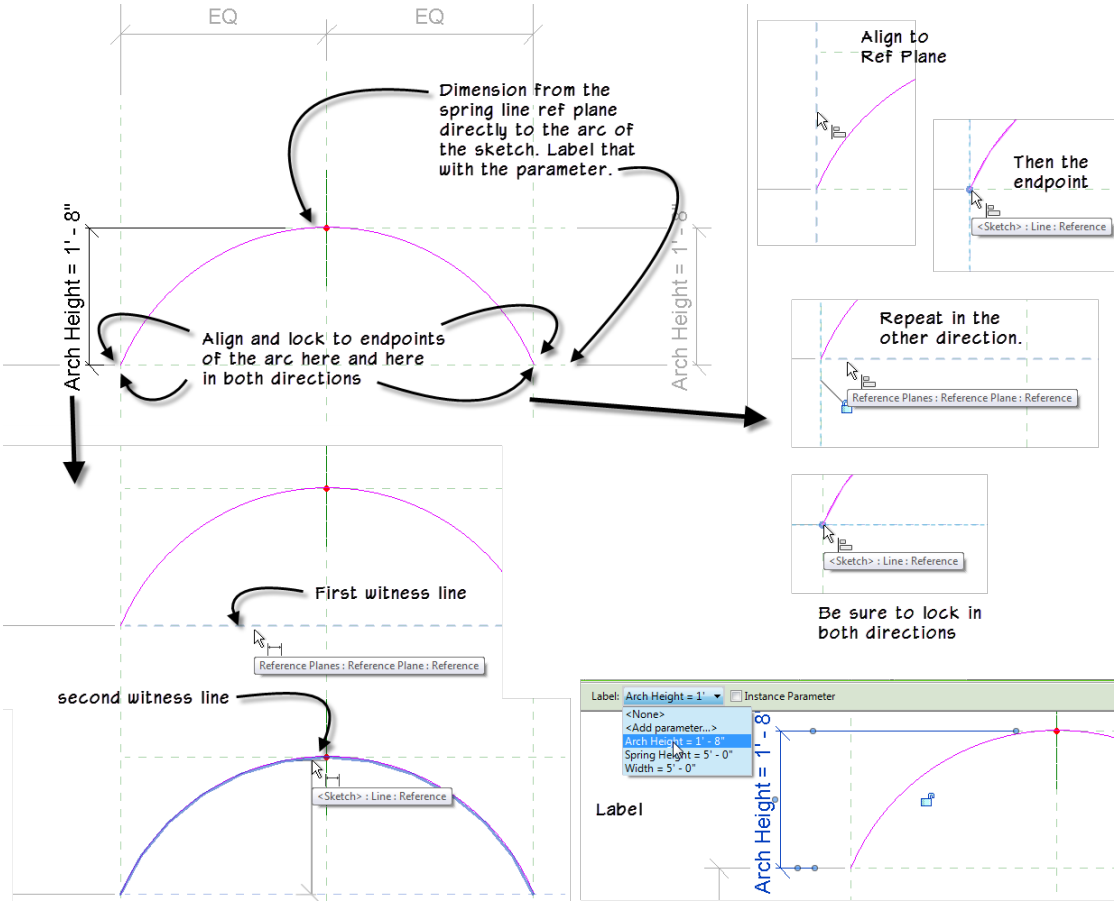


Figure 21 Constraining a basic arch form requires that you constrain the endpoints and the radius

This technique can be applied to any geometry in the family editor: model lines, extrusions, sweeps, etc. (see Figure 22).

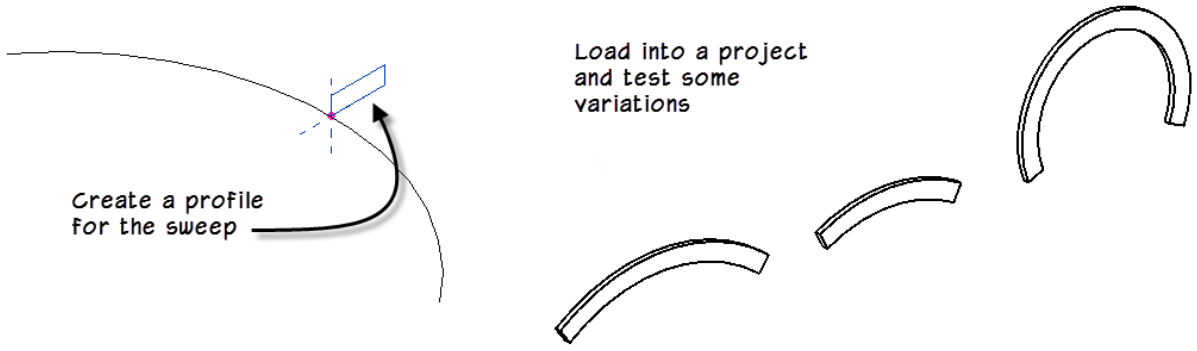


Figure 22 This technique can be applied to sweeps or any other form in the family editor

Common molding shapes are slightly more complex, but similar strategies apply. In general, mathematically speaking we need three bits of information to describe a curve. Therefore, if you find a reliable way to constrain three aspects that describe the curve, you will always get it to flex properly.

Chitham gives very simple guidelines on how to lay out most common moldings (see Figure 23). I have taken these and devised ways to replicate each of these parametrically in the family editor. In the arch example above, the curves were aligned and locked to the reference planes.

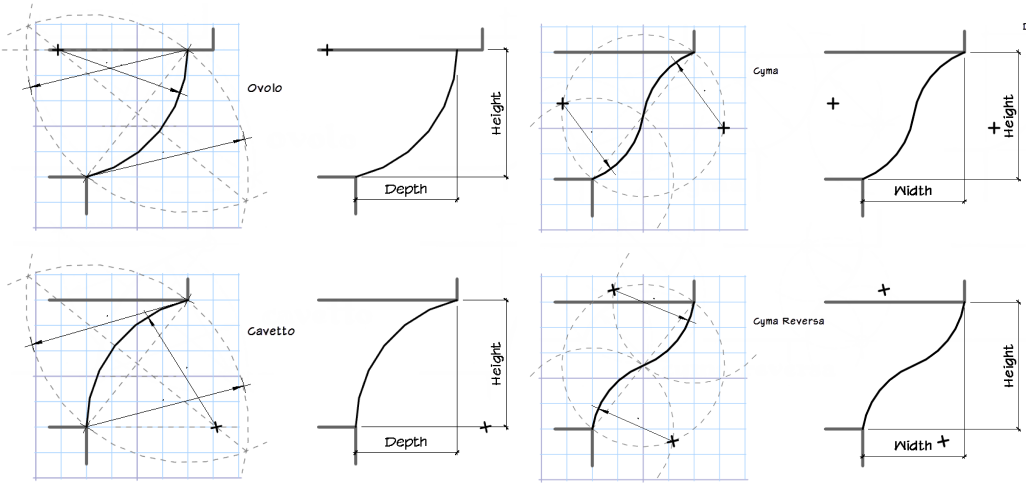


Figure 23 Common molding shapes showing Chitham's construction method and the variables established in Revit

In this example, trigonometry will be introduced to formulas that drive the parameters to ensure that the reference planes are not allowed to flex to values that make the curves fail (see Figure 24).

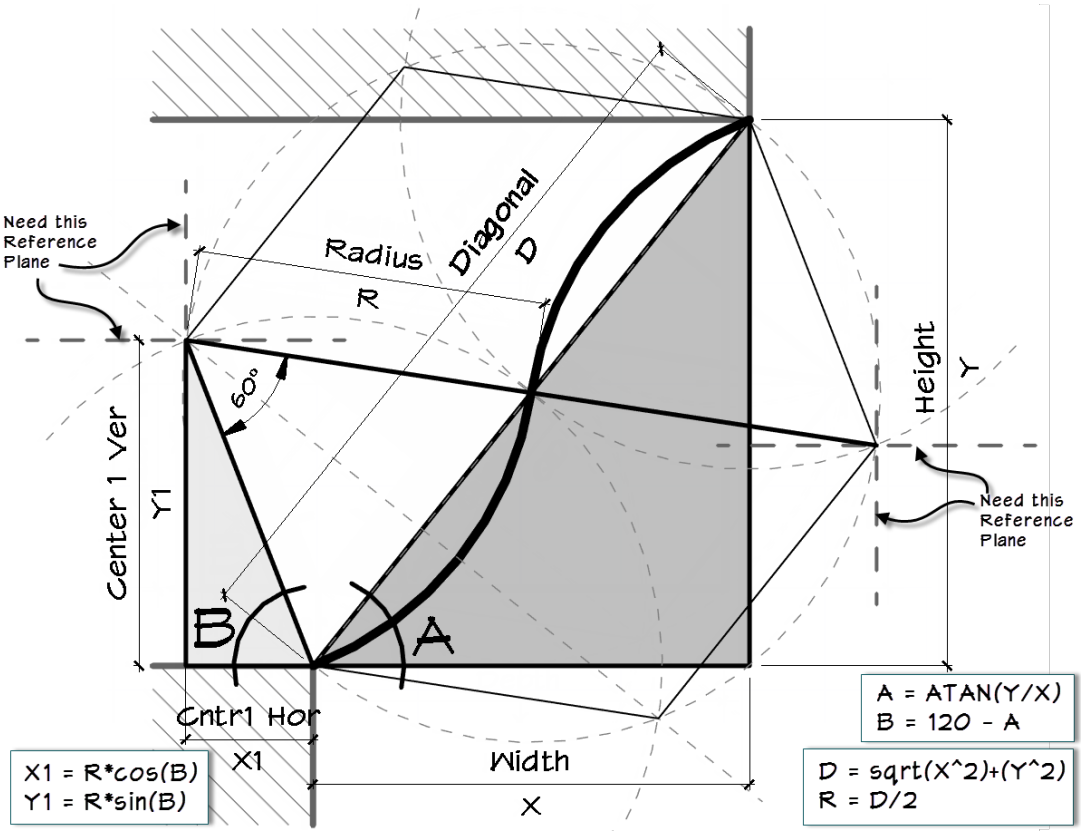


Figure 24 Diagram showing the parameters and the trig formulas required to properly flex the family

Moldings are best approached by creating a Profile family. This profile can then be reused in other families to create 3D form. To control the curve parametrically, in addition to trigonometry (which I typically like the best as it is very stable), you can also build a "rig." A rig is a construct that you build and parameterize and then you "hang" geometry off of it. There are different possible rigs depending on the kind of family you are building, category, etc. If you are working in a profile family, you can nest in a detail item rig. This allows you to build a nice stable form using simple detail lines and then once it is nested into the profile family, you can align and lock geometry to the rig. When the rig flexes, it carries the associated geometry with it. (There is a good example of using detail item rigs in the Ionic column below).

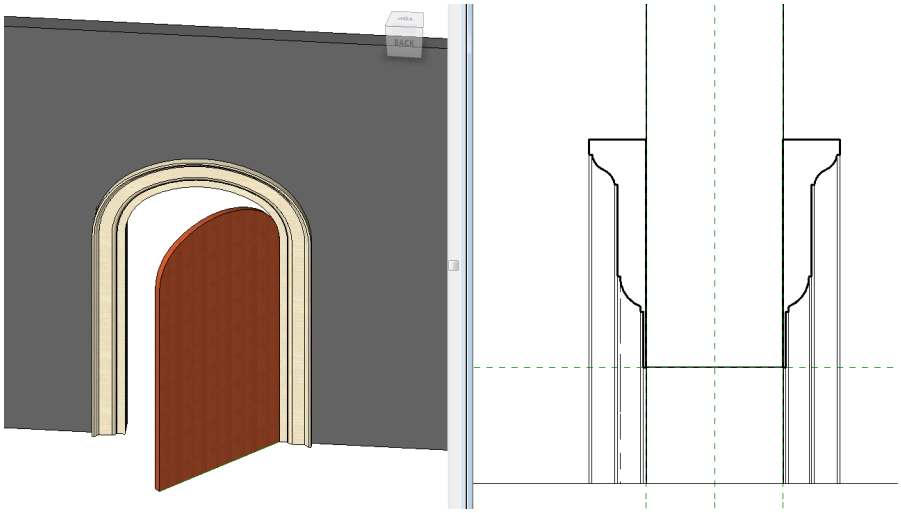


Figure 25 Combining curves, arches, moldings and parametric rotation in a single family

A profile defines the arch shape. These curves flex parametrically. The same profile is used for both the door opening and the door panel. A complex profile shape is defined using trigonometry or rigs and then loaded into a sweep around the opening shape for the molding.

Taming Curves Summary

- ⇨ Dimension Directly to the curve
- ⇨ Build Profile Families
- ⇨ Nest Detail Component Rigs into Profiles
- ⇨ Use Trigonometry

The Tuscan Order

Let's take a quick look at each order, starting with Tuscan. For most of my orders, there were many versions before I arrived at the final one. Here are a few of my Tuscan orders.

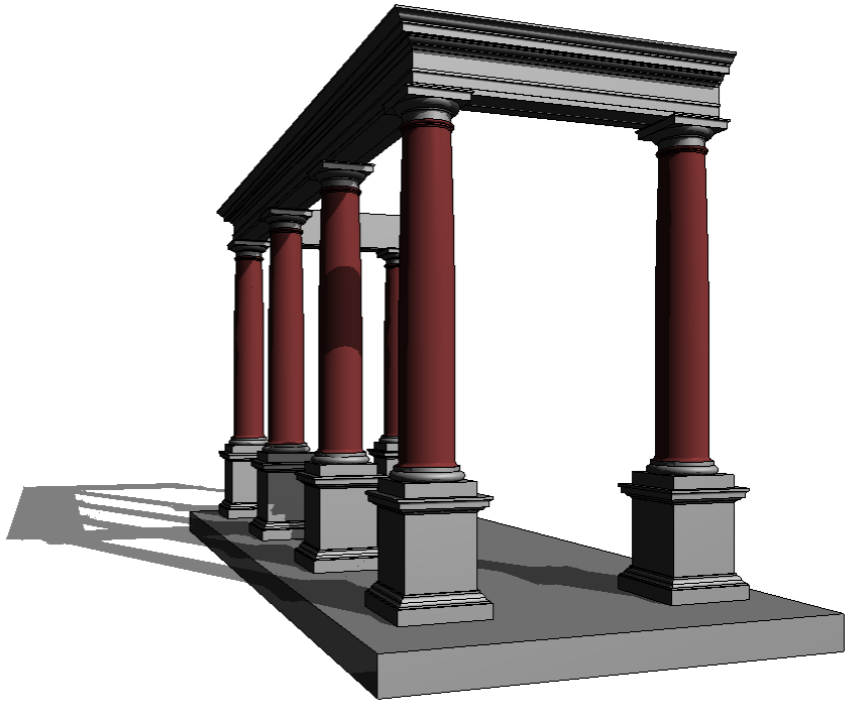


Figure 26 My earliest Revit classical family was the Tuscan order complete with pedestal

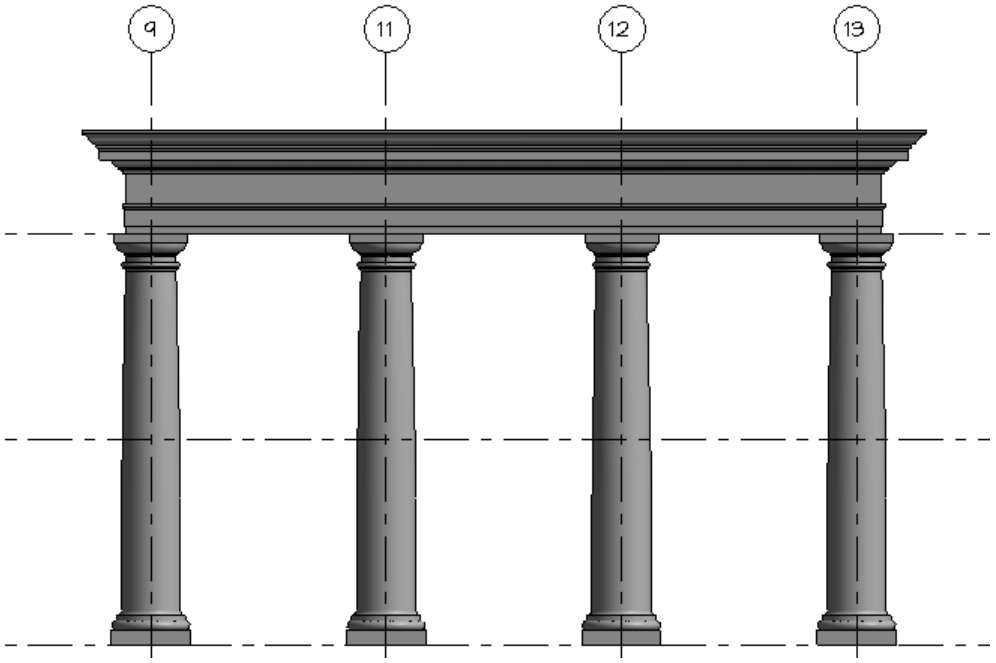


Figure 27 My current version of the Tuscan order

You may not notice much difference in these two, but the main difference is in how the moldings and other curved portions of the base and capital are devised. The original version was created before I had mastered the curve taming techniques discussed in the previous topic. While I did reuse some of the strategies from previous versions, I ended up rebuilding all of the orders from scratch for the book. This is the latest version of the Tuscan. It contains separate nested families for each of the following (see Figure 28):

- ⇨ The Capital
- ⇨ The Shaft
- ⇨ The Base
- ⇨ The Pedestal
- ⇨ The Entablature

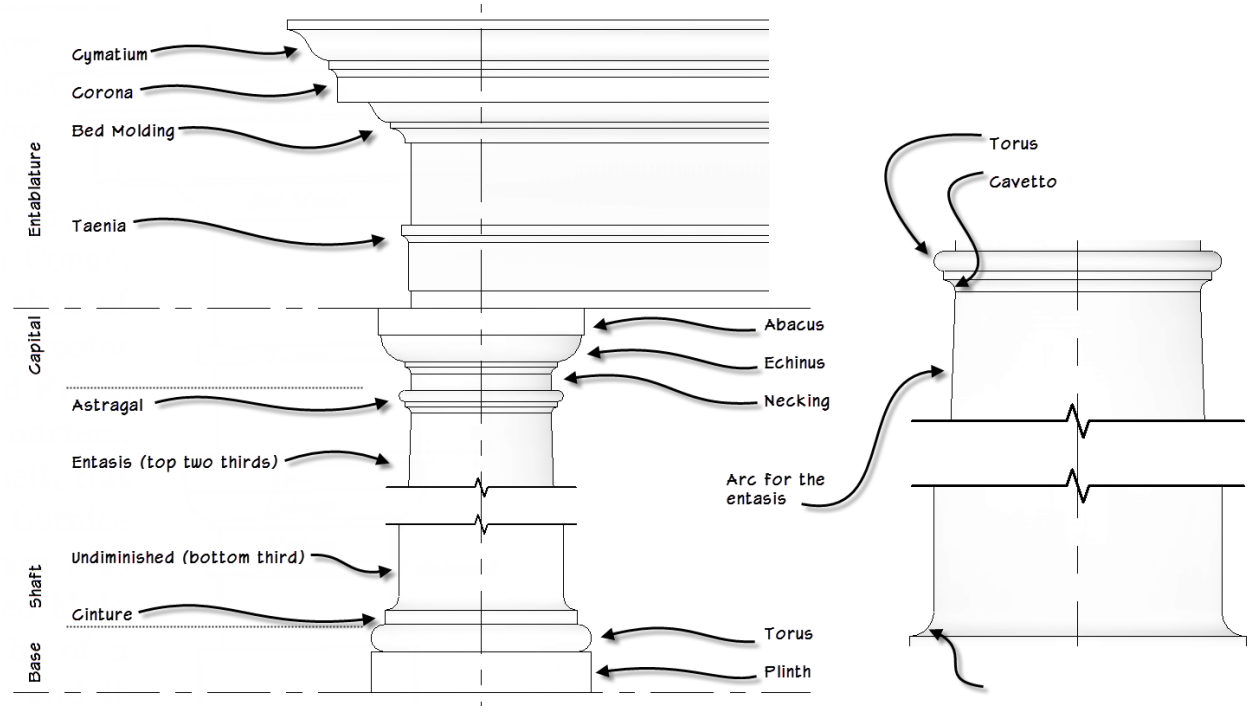


Figure 28 Here's a detailed look at each piece (pedestal not shown)

Tuscan Column and Beam

Here is a simple colonnade made from the Tuscan order. The order is built up from simple boxes. Each major component is a simple family containing a box driven by length, width and height parameters. Each of these parameters uses formulas (as shown earlier) to tie them together in the correct proportions. Once nested into the main family, the parameters are linked up in the host family so that a single parameter called: Base Diameter drives the entire family. Most moldings use sweeps around circular paths. These sweeps use nested profile families that take advantage of the curve constraining techniques covered previously. Once again, these components are controlled by the parameters in the host family. Note also that both coarse and fine versions are included (see Figure 29).

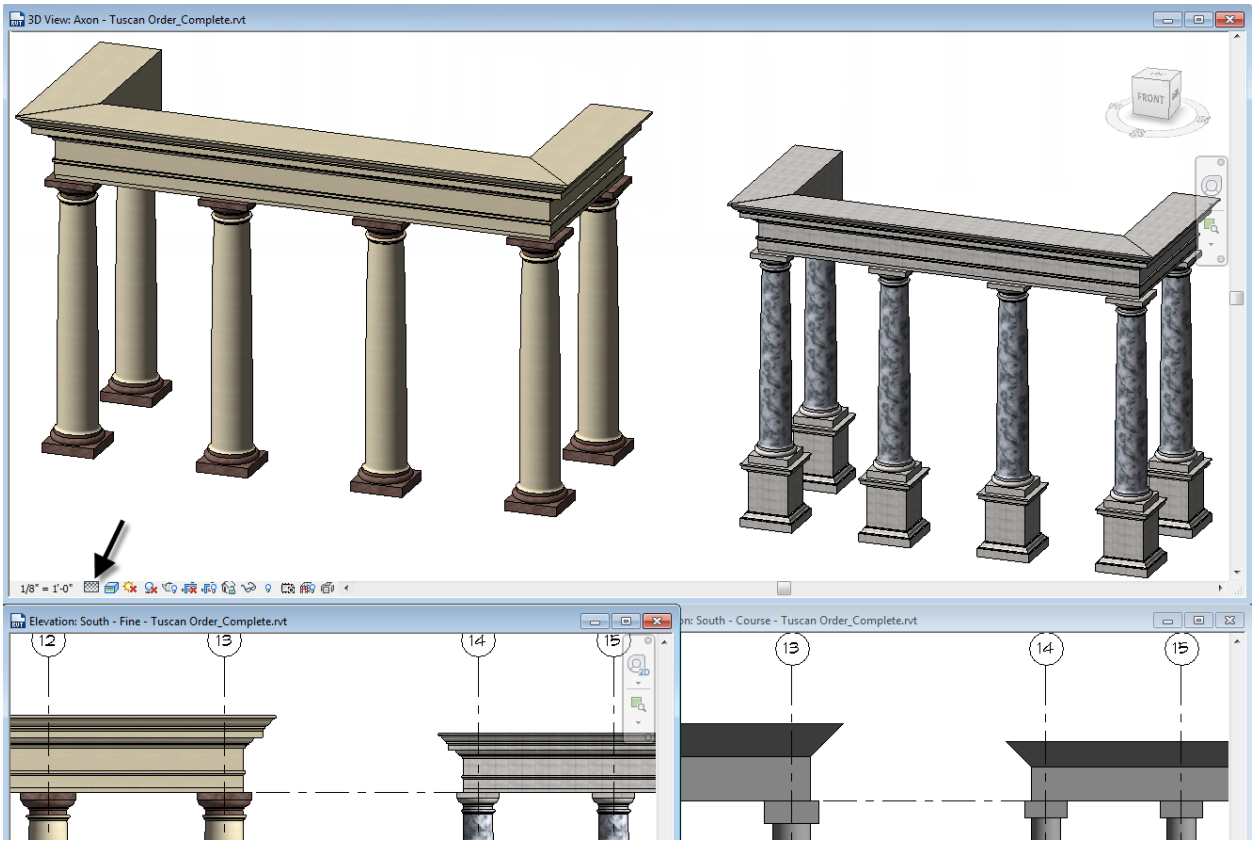


Figure 29 Tuscan Order with coarse and fine versions

The Tuscan columns are stand-alone Column families. The family contains four nested families: Capital, Shaft, Base and Pedestal. Each one relies primarily on Extrusions for any elements that are square in plan. Elements that are round in plan use sweeps. (It is possible to use revolves, but revolves cannot use profiles. So I typically use sweeps).

- Material parameters are linked through each nested family to allow for control from the parent family.
- The pedestal can be turned on or off using a visibility parameter and this in-turn adjusts the overall height of the family.
- The entablature in the Tuscan order is a beam (structural framing) family. You have to right-click to miter the corners.

The Doric Order

Doric uses similar strategies. Again, the main family contains the same four nested families (pedestal, base, shaft and capital), it uses materials and has the visibility control for optionally showing the pedestal. More reference planes are used this time to constrain the dimensions and parameters.

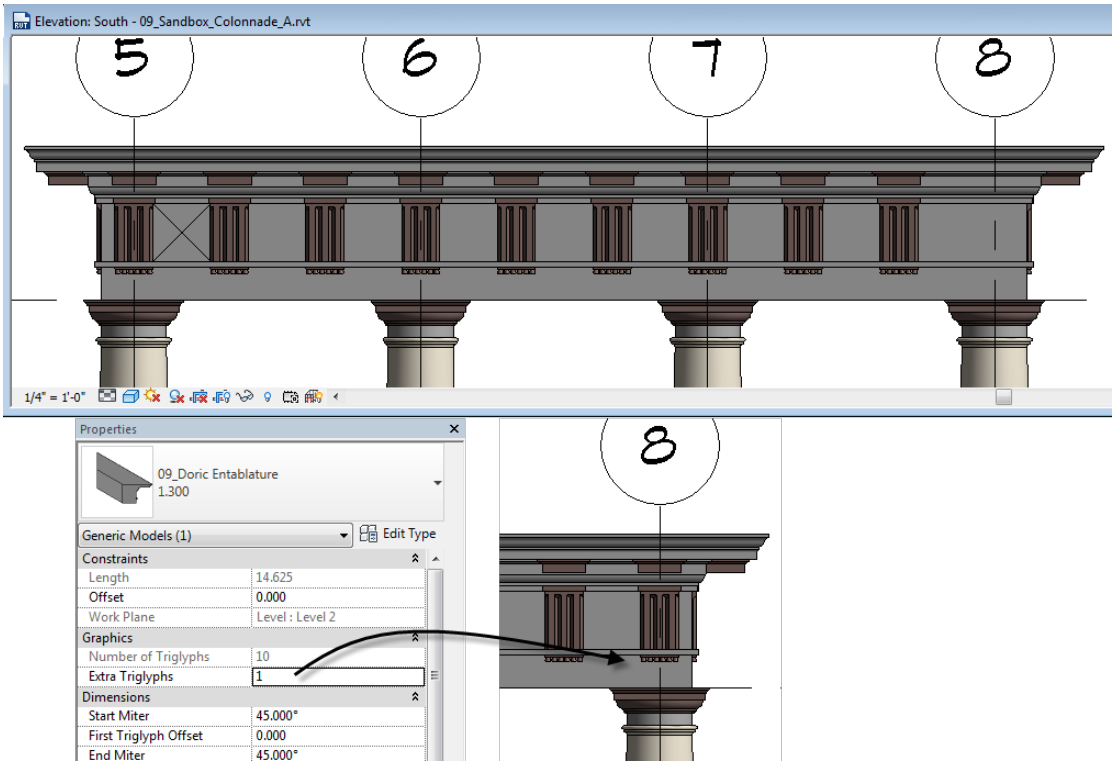


Figure 30 Laying out the Doric Entablature

The entablature is what really drives the Doric. There are very specific rules for triglyphs and metopes. The proportions of the triglyphs are fixed and metopes must be square. This means that Doric intercolumniation only allows certain spacings for proper layout of the triglyphs and metopes (see Figure 30).

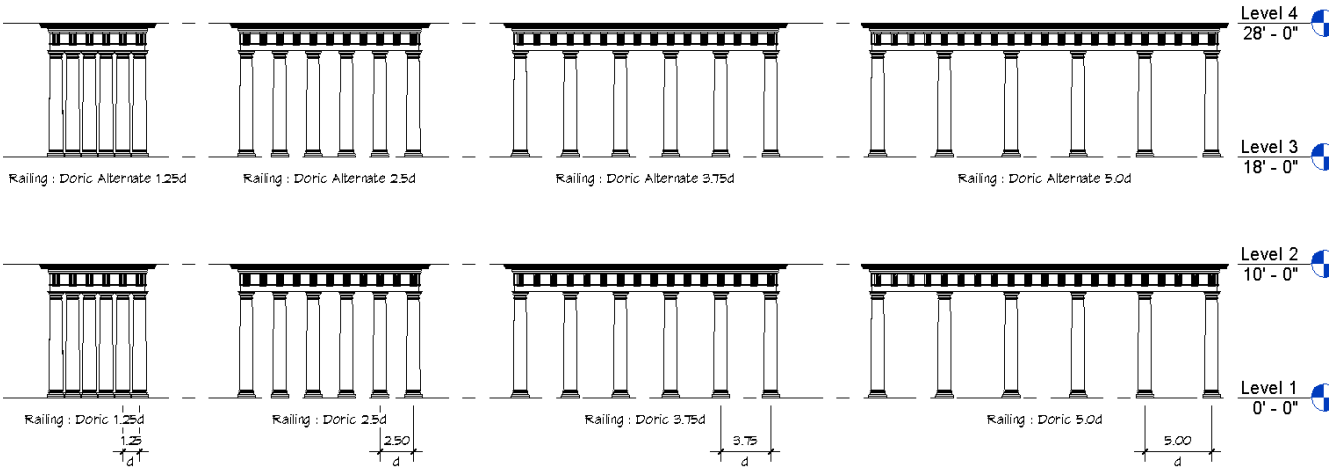


Figure 31 Railing types for Doric intercolumniation

The example shown in Figure 30 uses a generic model family with nested components for the triglyphs. This is ultimately the approach I settled on, but before doing so, I experimented with using a Railing Type to lay these out (see Figure 31). This is not an ideal solution since there is no way to change the

category of the railing, but there are some nice benefits to doing it this way such as being able to define types for each intercolumniation and then simple sketching the “railing” at a properly calculated length.

Doric using a Railing Family

This is mapping of the family nesting. The railing type uses both profiles for the rails and baluster families for the columns and triglyphs. The baluster family then has the nested column family, which in-turn has the other four nested families within it (see Figure 32).

- ⇨ Railing Type
  - ⇨ Rails – (Profile Families) for Entablature
  - ⇨ Balusters – (contain Column Families) for Columns
    - ⇘ Doric Baluster & Doric Baluster with Triglyph
      - ⇘ Doric Order
        - ⇧ Capital
        - ⇧ Shaft
          - ⇧ Profile
        - ⇧ Base
        - ⇧ Pedestal
  - ⇨ Balusters – for Triglyphs
  - ⇨ Baluster Posts – for corner Columns

Figure 32 Doric order as a railing type

Ultimately, I like the solution of a custom beam (generic model in this case) family with repeating array families for the triglyphs and mutules better (see Figure 33).

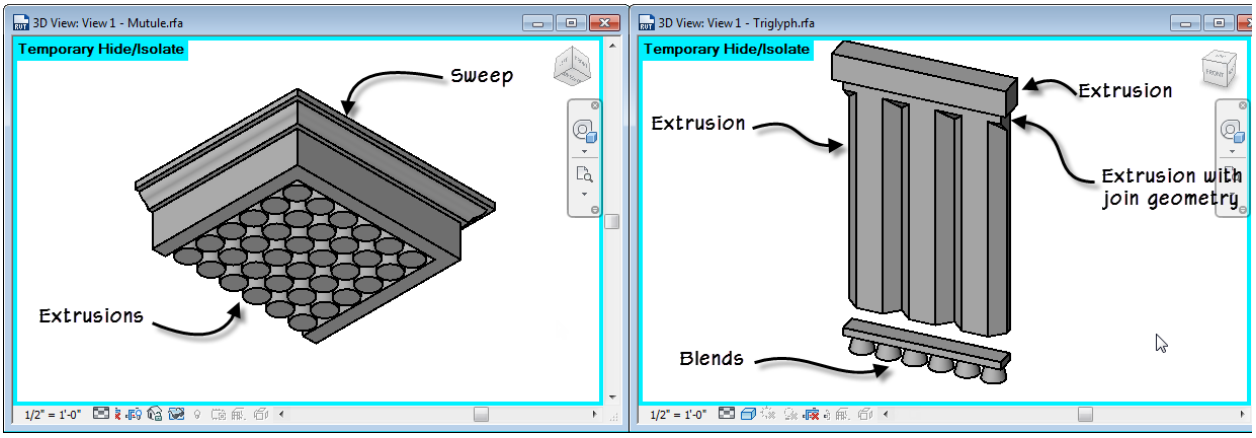


Figure 33 The Mutule on the left and the Triglyph on the right

Il Tempietto

So after having built many Doric pieces, I wanted to put them to the test. What better building that the Tempietto.

Here we see some photographs (see Figure 34) and sketches of the Tempietto (see Figure 35) and some images of the model I built of it in Revit (see Figure 36).

Figure 34 The Tempietto

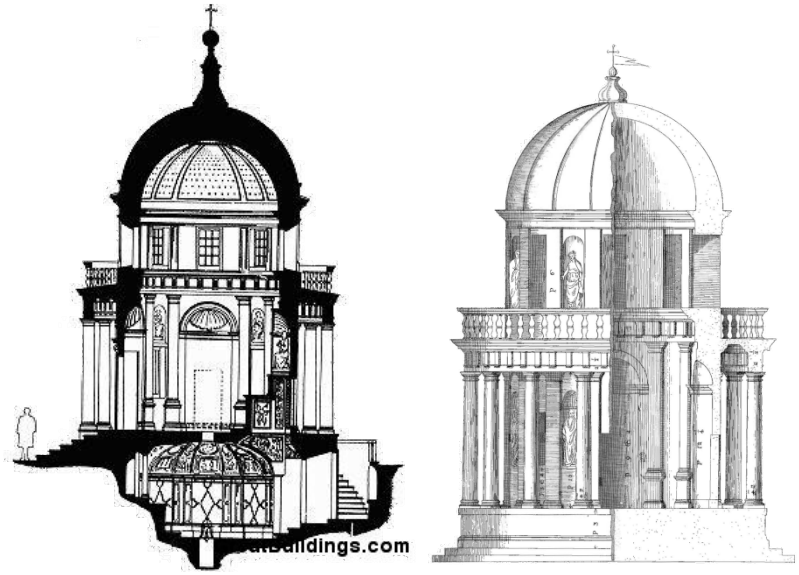


Figure 35 Cross section and elevation

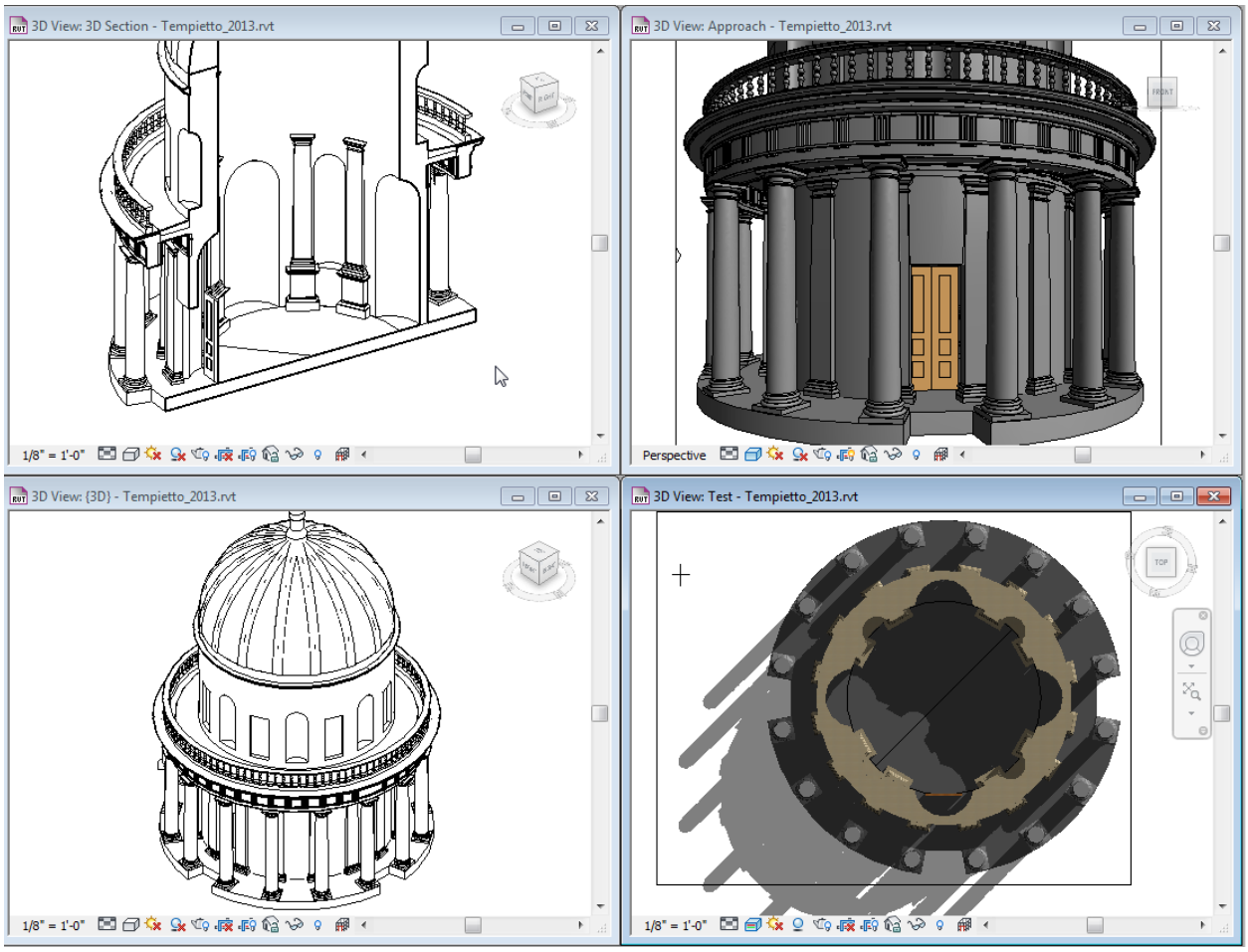


Figure 36 My Revit model of the Tempietto

My Revit model is not complete, but I have built the overall form in Revit and used many of the Doric elements as well as creating additional elements that were required like niches and coffers.

(I have not rebuilt this with my latest version of the Doric. So the model you see here is using the older version—the railing type).

I had to calculate the correct circumference from photographs and make sure that it worked with the intercolumniation. Maybe Bramante himself performed similar calculations.

I made a pilaster version of the Doric for the interior edges of the colonnade (see Figure 37) and the inside of the building. For niches and coffers, simple face based generic model families with voids did the trick.

The balustrade required the creation of another family for its balusters. Chitham covers balustrades for each order, so I was able to create a family that matches what is required.

The roof is an in-place family for the dome. I made component families for the ribs.

Figure 37 Exterior colonnade of the Tempietto

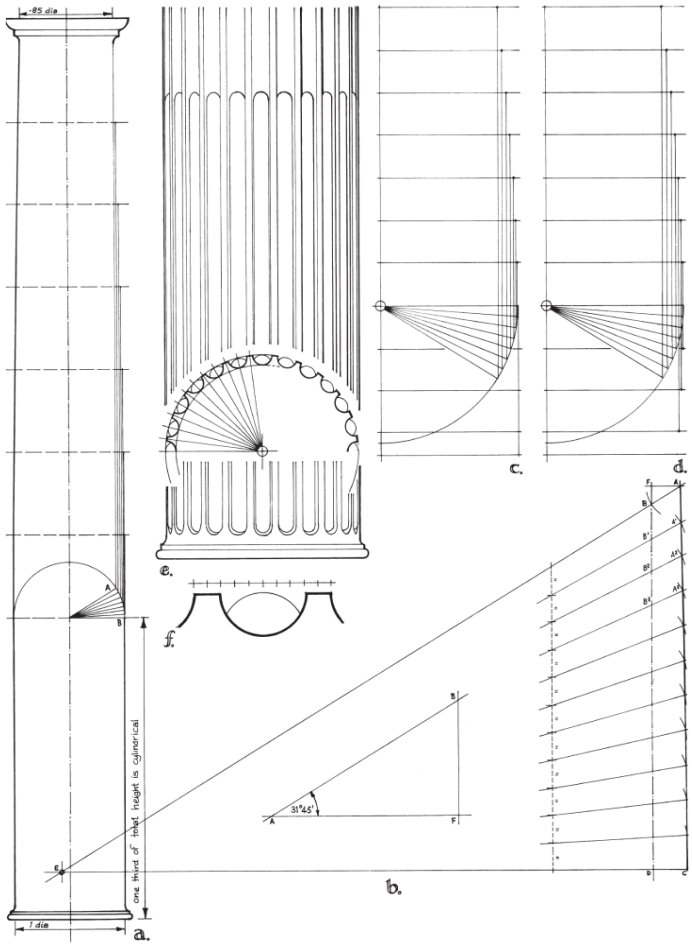
Doric Summary

To summarize the approach on the Doric order, it builds on the strategies employed for the Tuscan and begins looking at other important issues like intercolumniation and some non-traditional approaches to repeating columns and adding the entablature.

- The Pieces
    - Triglyphs
    - Mutules
- Intercolumniation
- Doric Colonnade
    - Beams
    - Railings
    - Other

Column Shafts

All of the orders have similar shafts. Only the height varies for most (the Tuscan is slightly different at the bottom). A single shaft family can therefore be made parametric and accommodate all of the orders. This is true for smooth shafts. There are some variations on fluted shafts as well.



61. THE COLUMN: DIMINUTION AND FLUTING

Figure 38 Plate 61 from Chitham

Turning again to Chitham, the Diminution can be established to create the tapered effect on the column (see Figure 38). Chitham discusses the difference between diminution and entasis and why he chose to refer to his as diminution. The points he makes are interesting and valid, but for our purposes, it is fine to use the terms interchangeably. I tend to use entasis more, since it is a more widely used term for describing this phenomenon. Fluting is also set out in the plate. We'll discuss fluting a bit later.

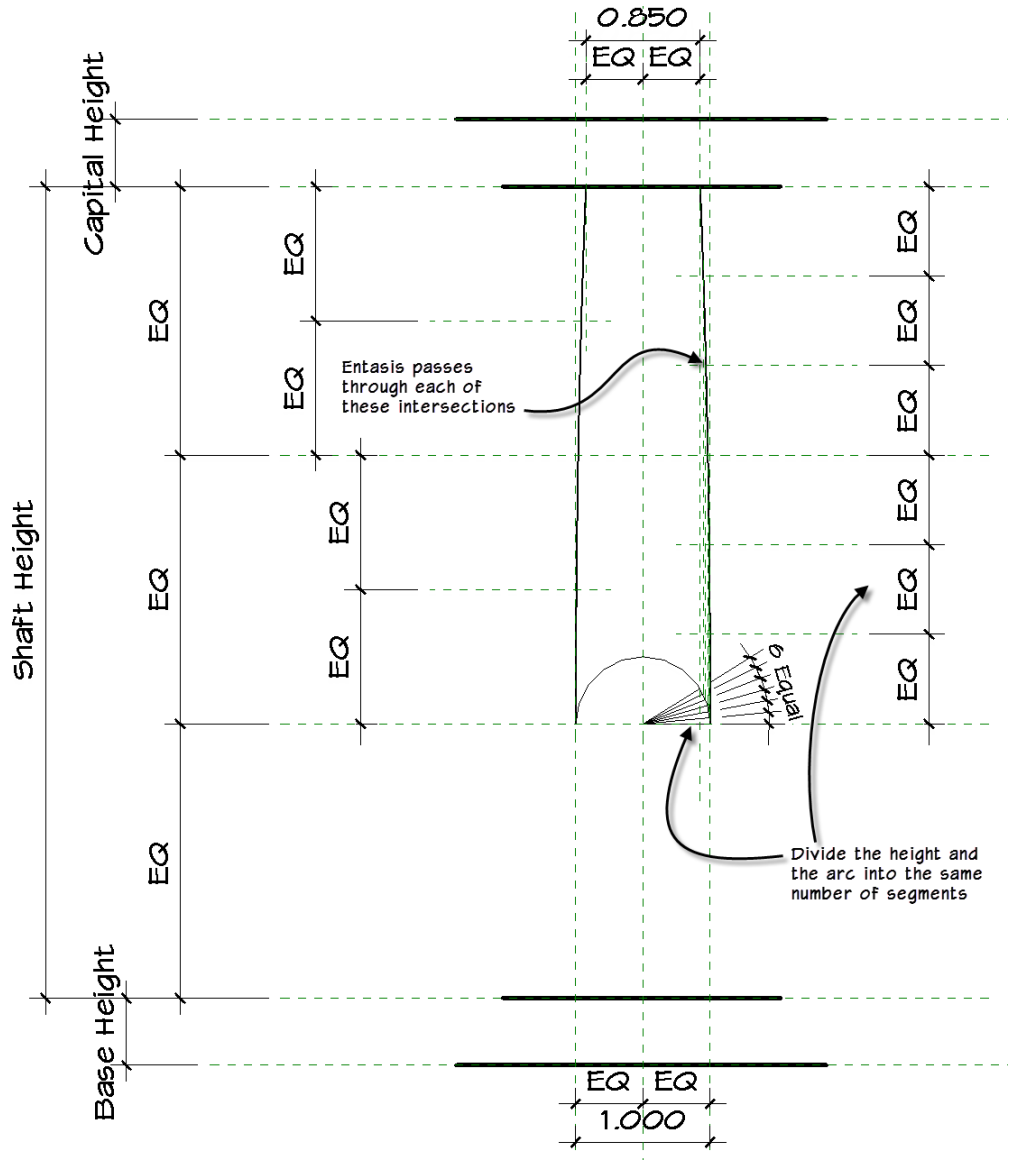


Figure 39 Using equality dimensions helps with layout and reducing formulas and parameters

Using the method described by Chitham, I began to lay out the entasis in Revit. Using equality dimensions limits the quantity of parameters required (see Figure 39).

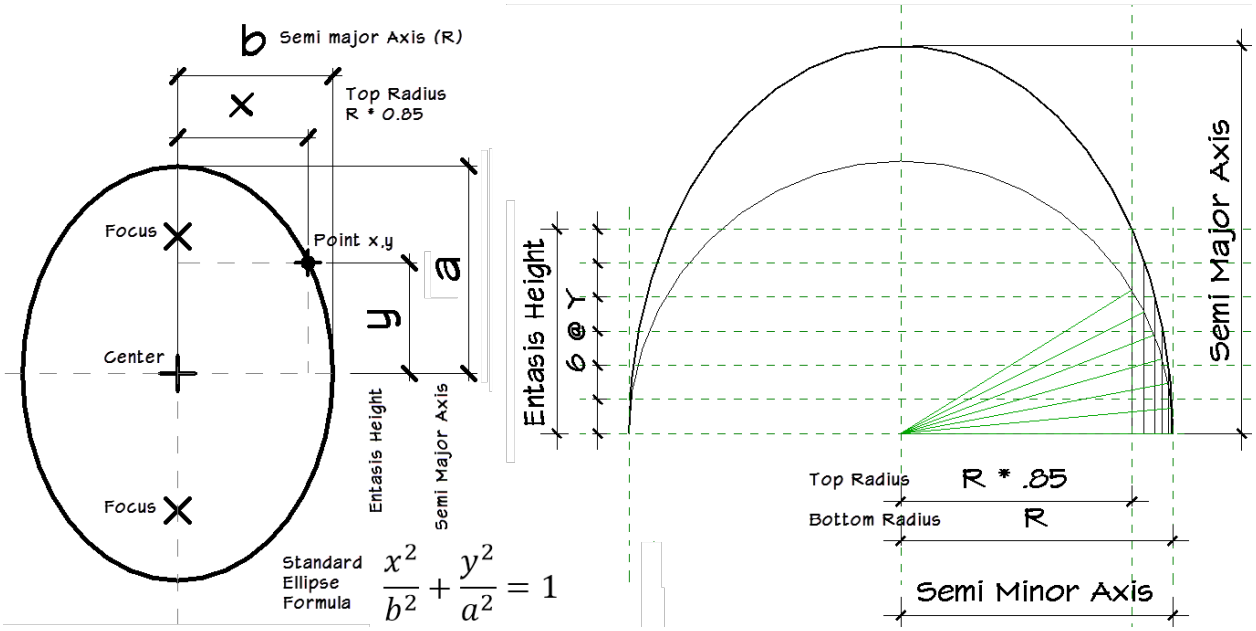


Figure 40 Entasis is actually elliptical

In building the shaft, I discovered that the entasis as recommended by Chitham and others is actually elliptical (see Figure 40). This means that you can use the formula for an ellipse in the family and build the entasis extremely accurately. However, the formula is quite complex and I had to enlist the help of my good friend Dezi Mackey to build the formula (see Figure 41).

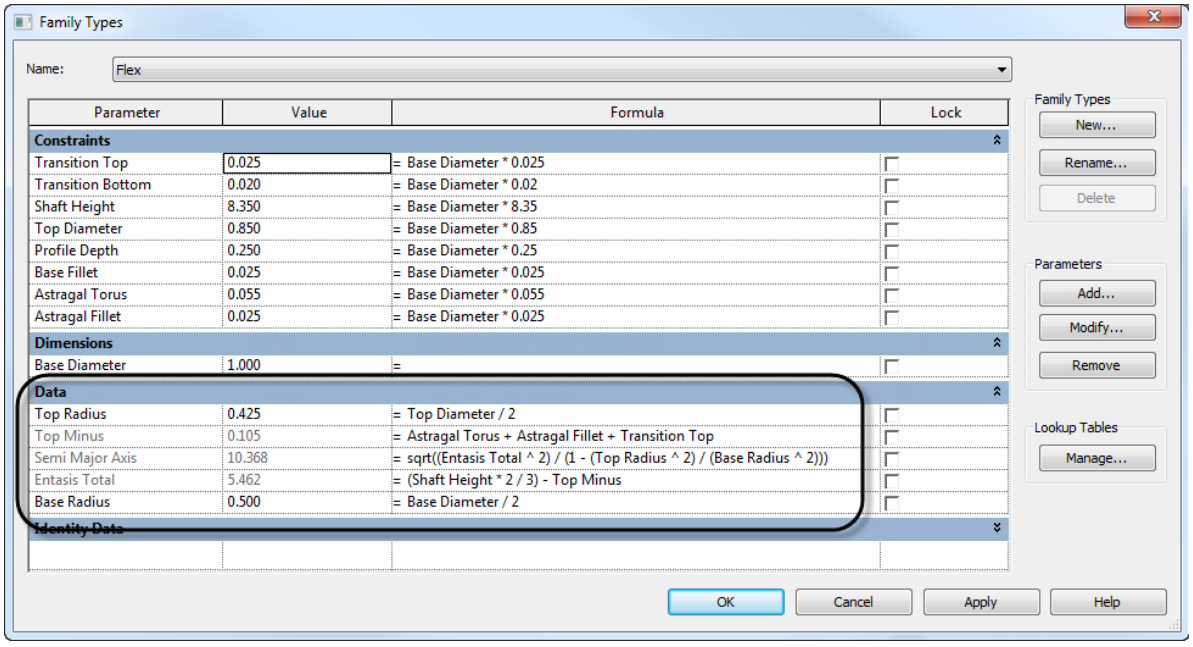


Figure 41 Here is the formula in Revit format

If you would like to create a simpler profile, whose shape is not exact, but very close, you can draw a circular arc with a very large radius instead. The difference in curvature is nearly imperceptible. I used this approach for my coarse detail version.

When creating a shaft family that uses entasis, fluting has proven quite challenging. This is because you are creating a 3D form that has curvature in all three directions. My preference would be to create the fluting with voids. But getting the voids to follow the entasis and also repeat 20 or 24 times around the circumference has proven quite difficult. Instead I have two different approaches depending on the order (see Figure 42).

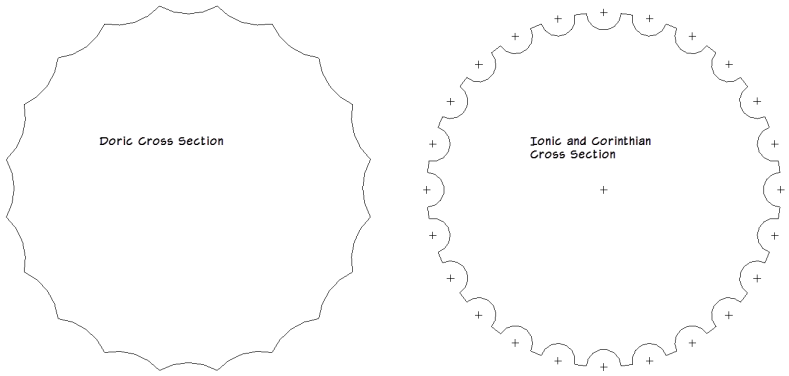


Figure 42 Doric uses differently shaped channels than the other orders

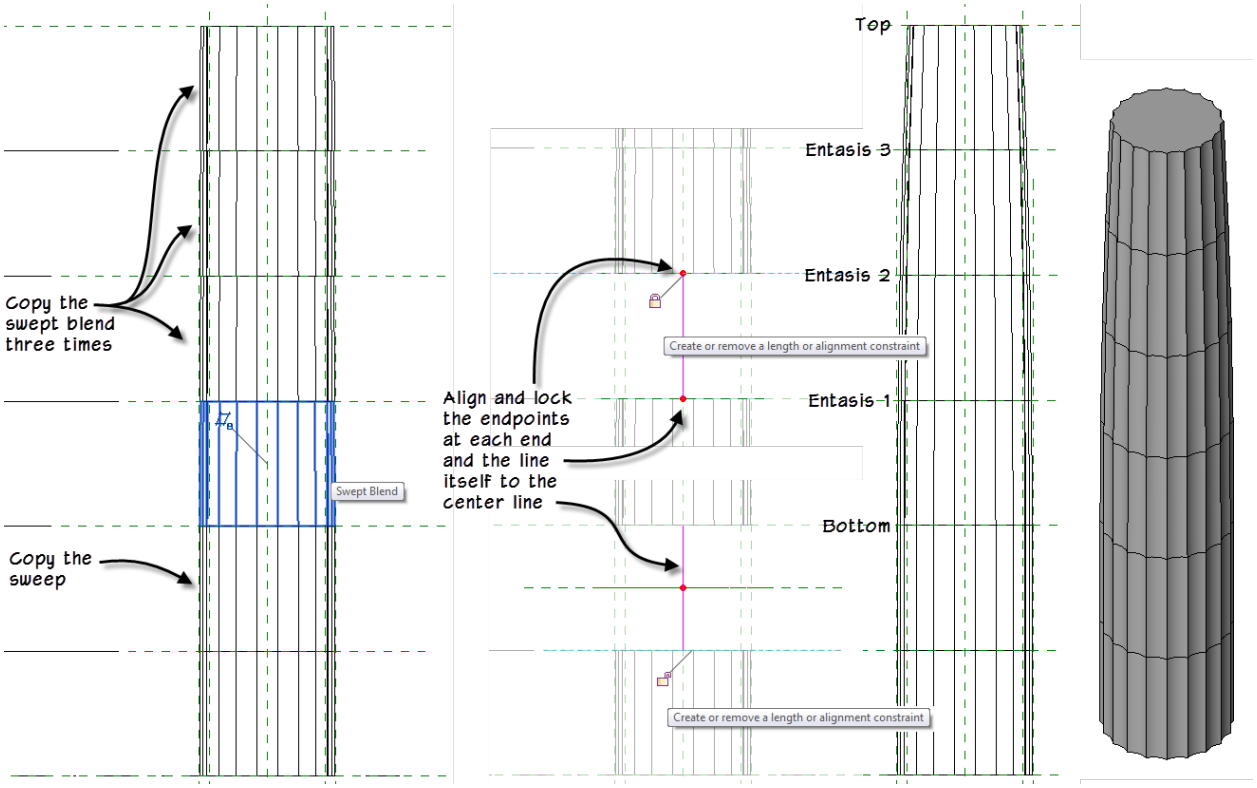


Figure 43 Stack blends to create a Doric fluted shaft

For the Doric shaft the channels are shaped differently and there are fewer of them. The approach I used there was to stack up several blends that slowly taper. This creates a Greek-style Doric using this approach (see Figure 43).

The other approach is to do an adaptive family that lofts the profiles as they diminish up the shaft height. I did this for the Corinthian. I stacked a series of profiles (each diminishing in size) and built a lofted form in the massing environment for the fluted shaft. It is a bit heavier than I would like, but perhaps not a whole lot more so than would be the case with 24 voids (see Figure 44).

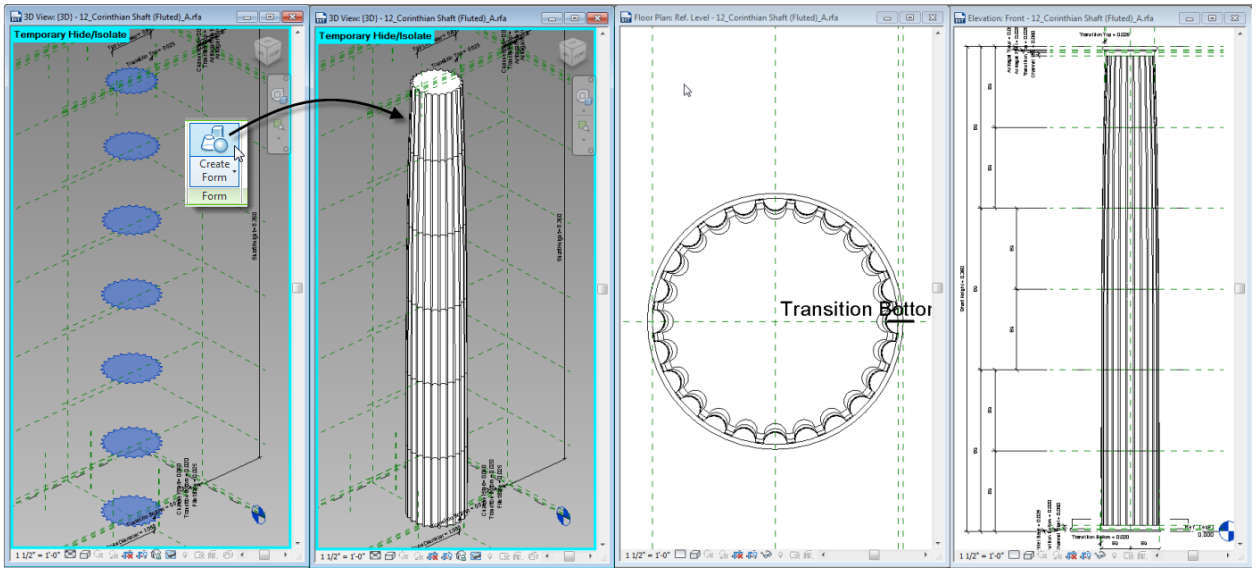


Figure 44 Using an adaptive component by lofting a series of stacked profiles

At the top and bottom a separate form using voids for the routed ends was required. I went through several iterations of this form before finding one that worked well (see Figure 45).

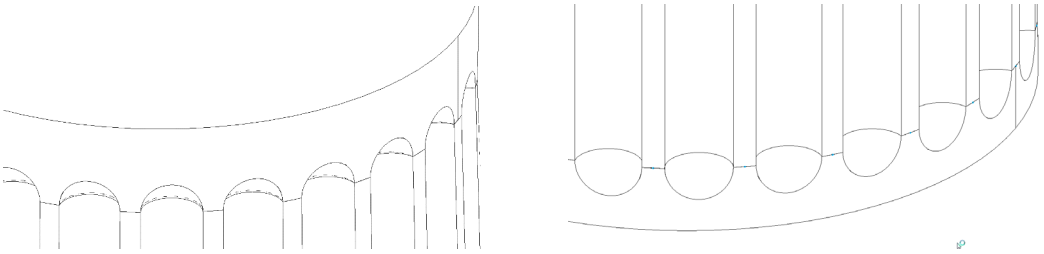


Figure 45 Voids at the top and bottom of the channels

A few seams do show, but overall it is not bad.

Column Shaft Summary

- ⇨ Revolve vs. Sweep
- ⇨ Diminution/Entasis
  - ⇨ Arc vs. Spline vs. Ellipse
- ⇨ Fluting
  - ⇨ Stacking Blends
  - ⇨ Adaptive Component
- ⇨ Parametric Control

The Ionic Order

I wanted to push the traditional family editor as far as possible. This Ionic capital is modeling completely in the traditional family editor! It is not an adaptive component (see Figure 46 Ionic Capital in the family editorFigure 46).

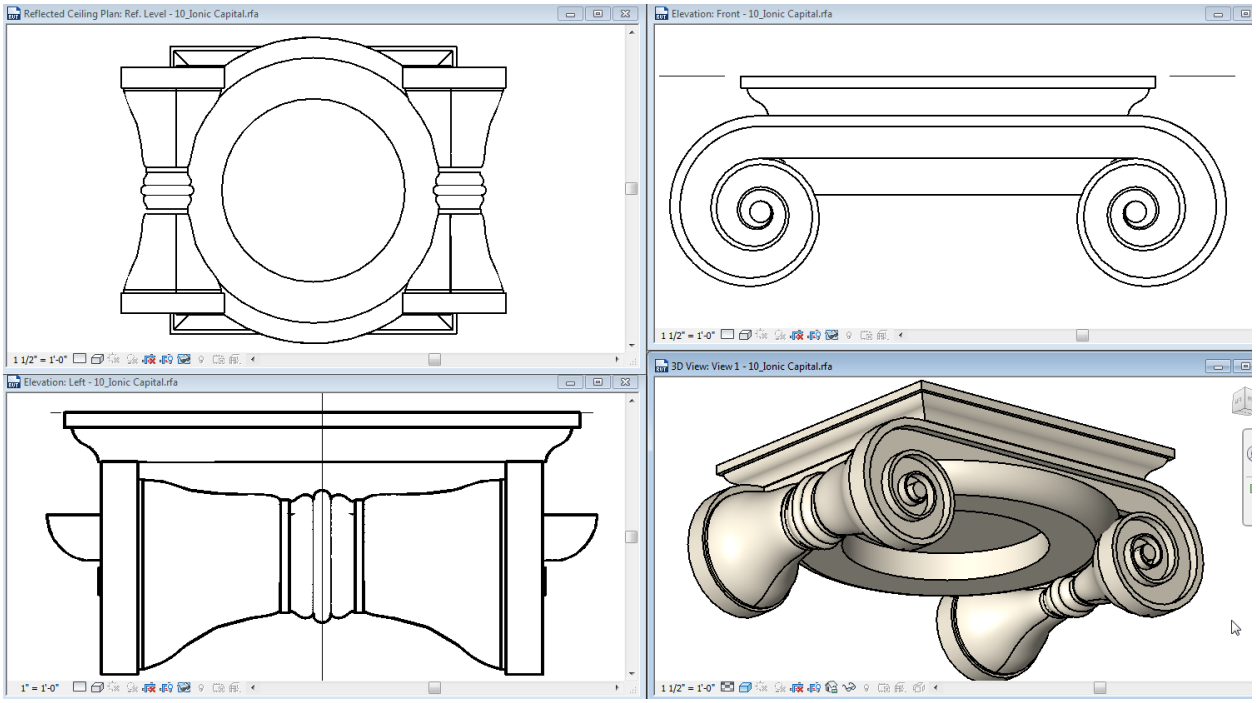


Figure 46 Ionic Capital in the family editor

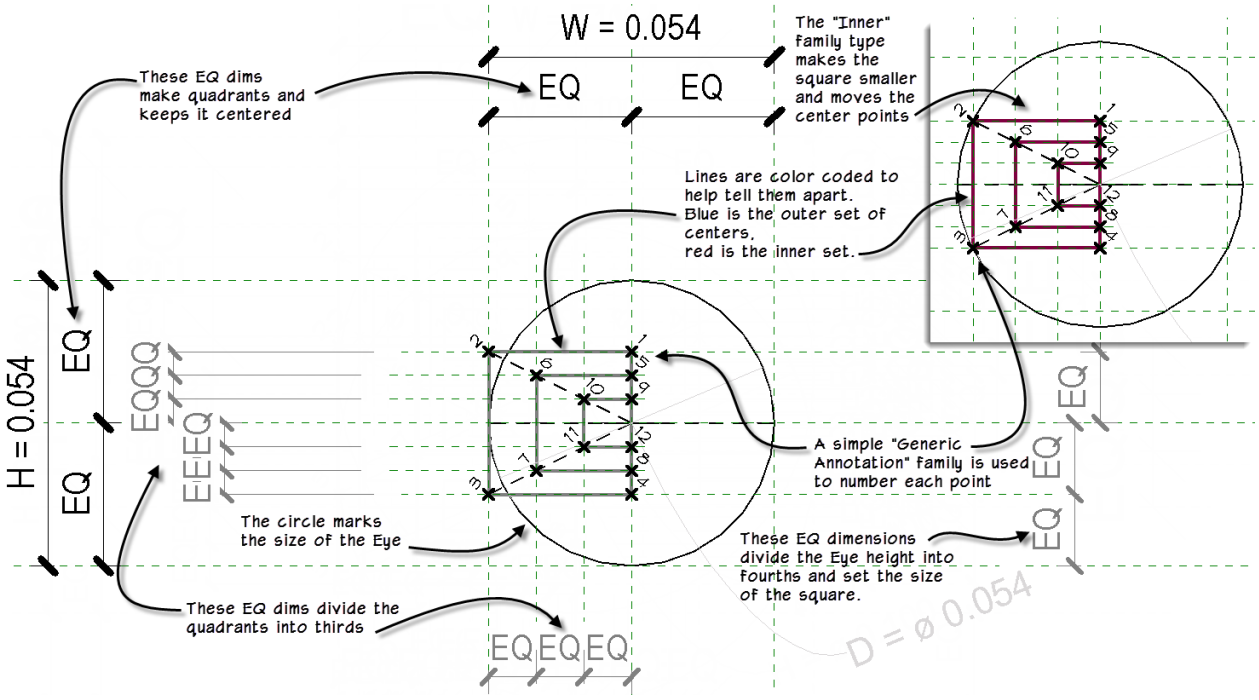


Figure 47 Developing the volute scrolls of the Ionic capital. These rely on nested detail item rigs

However, this does not mean it is not complex. If you look closely at the spiral on the volute, you see that the inner and out spirals taper in towards the eye. So you cannot do a sweep along the path in the traditional family editor. It might be possible to use a swept blend but the path would have to be a spline. But getting a spline path to follow the spiral exactly is tricky. The volute is a series of diminishing arcs; each one a quarter of a circle. The spline can come close to approximating this, but not exactly.

I opted instead to sweep it the other way. Instead of sweeping along the spiral, I swept instead along a short line perpendicular to the spiral plane. This allows for the two spirals to become an enclosed shape that form the profile of the sweep. Constructing the spirals uses a fairly complex rig (see Figure 47) and several arcs (see Figure 48). Each one has to be constrained carefully. Note the detail item rig from Figure 47 at the center of the arc spiral.

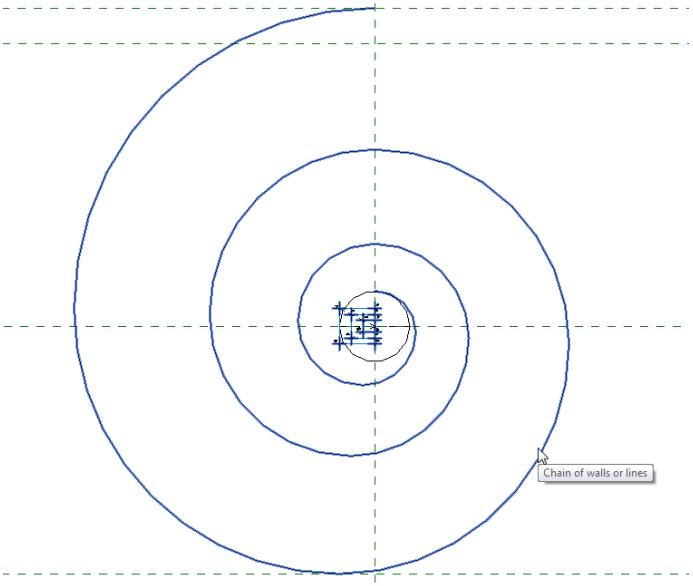


Figure 48 You end up with a chain of lines for the inside (not shown) and another for the outside (shown)

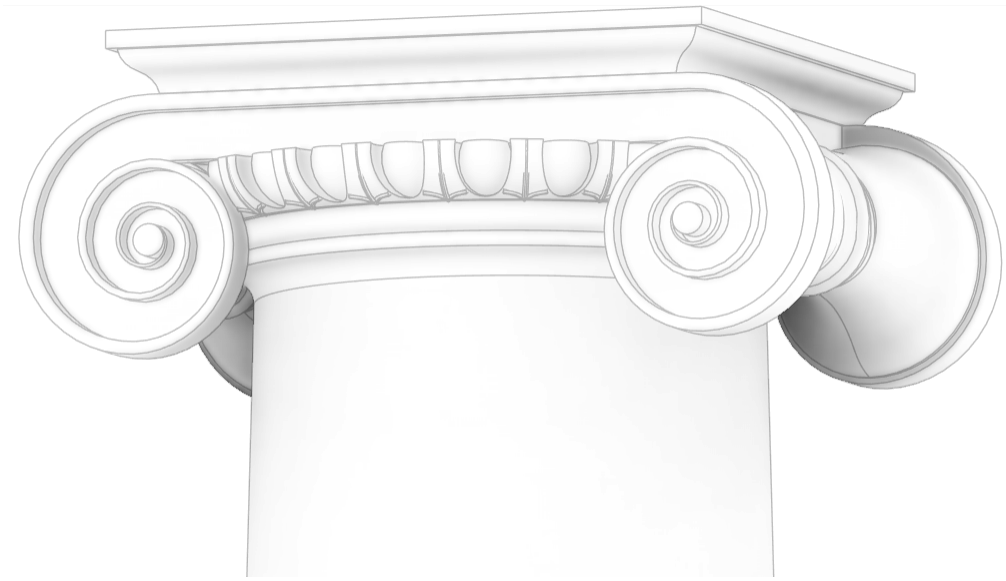


Figure 49 The final Ionic Capital

This image shows the final capital (see Figure 49). It also shows the egg and dart, which I did not include in the book. I was having a difficult time getting the egg and dart to flex. So the book version is left smooth. Maybe I'll revisit in a future edition. Also, this image was created by a happy accident. I love the look in Revit with ambient shadows turned on. But the actual shadows were displaying a little ragged on the edges. It turns out that if you turn on Reveal Hidden Elements (the small light bulb on the View Control Bar) it softens everything, including the linework at the edges. Very nice! I just used SnagIt to take a high res screen capture and the result is what you see here.

The horns (behind the volute scrolls) are created using swept blends. To get the smooth curve, the profile families use splines. This give a nice organic form and keeps you in the traditional family editor.

When you combine the capital with the other components, you end up with the completed Ionic order shown here in Figure 50.

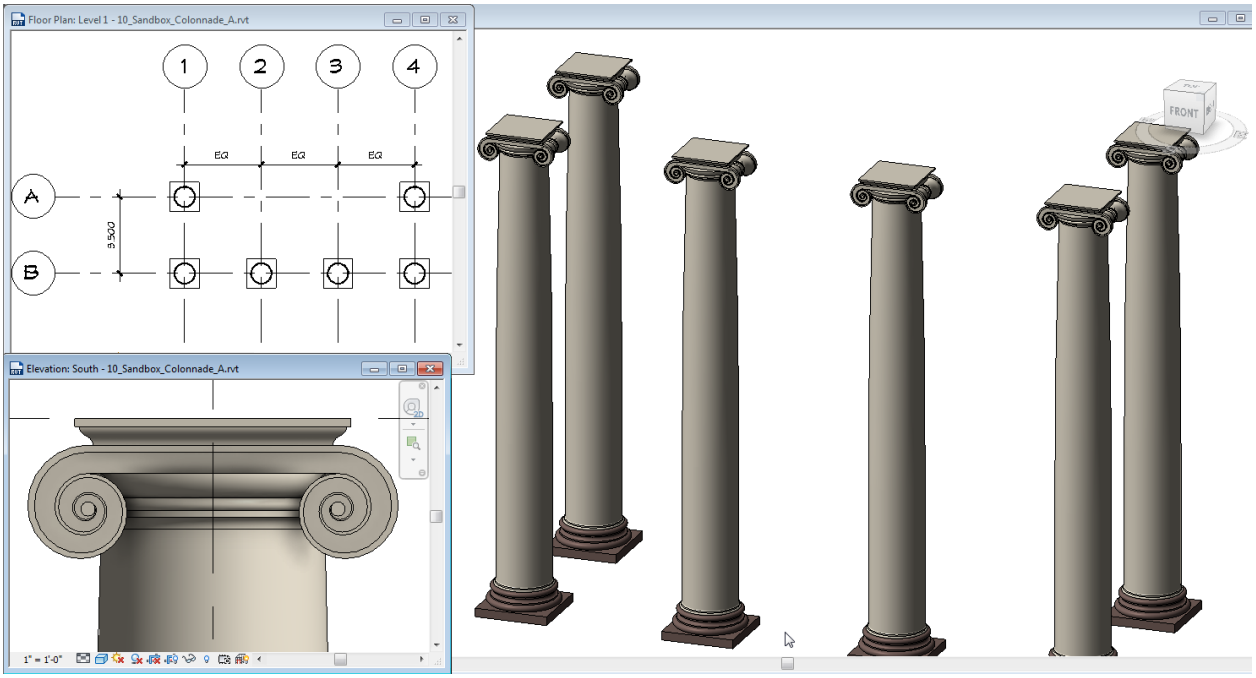


Figure 50 Completed Ionic with a smooth shaft

Ionic Order Summary

- ⇨ Laying out the Volute in a Profile Family
- ⇨ Using Sweeps for volutes
- ⇨ Using Swept Blends for the horns
- ⇨ Using a Spline to help shape the collars

The Corinthian Order

The first three orders were built completely in the traditional environment. Only the fluted shaft and the Corinthian capital is built in the massing environment as an adaptive component.

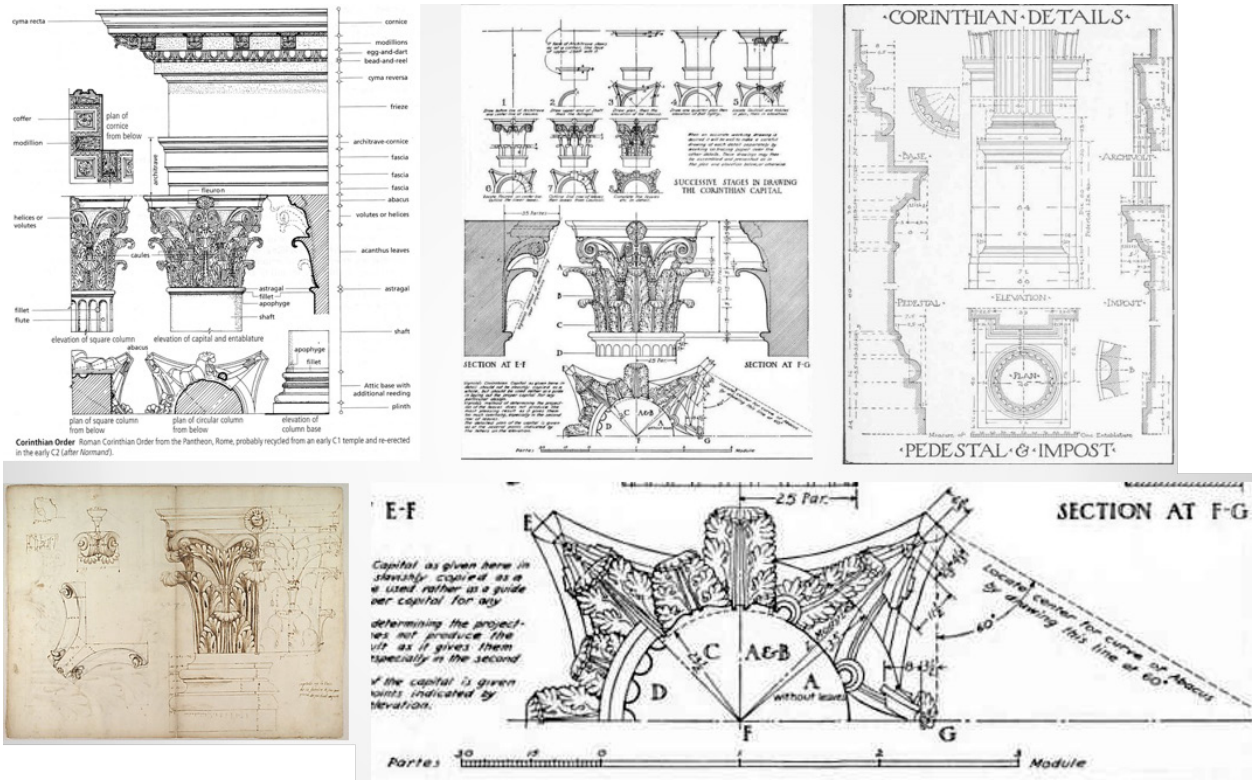


Figure 51 Some of the Corinthian resources I used

When using the Massing Environment, there are a couple key trade-offs. First, you can only change categories and build "components" if they are created as Adaptive Components. File size is the next issue. I have not addressed file size at all so far, but believe me, these files are breaking ALL the file size rules! So at some point I will have to try to optimize them. As much as I would like the Corinthian to be a standard family, I just don't feel that the forms would be successful using the traditional family editor.

My first Corinthian version was a whopping 150MB!

My current version is about 30MB. Still *very* large, but much better.

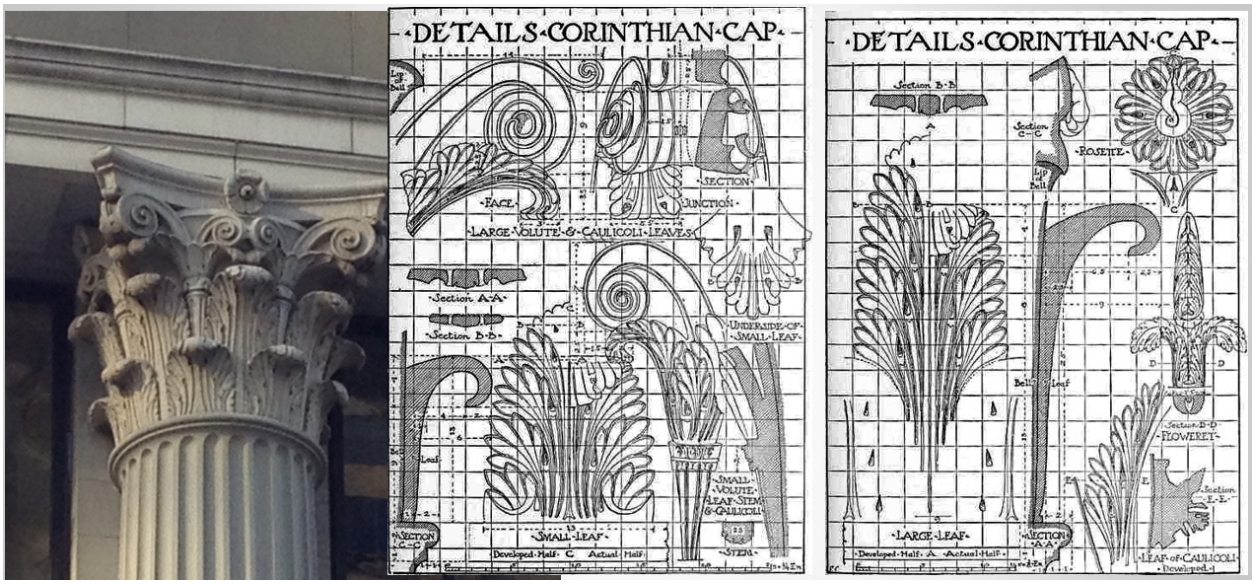


Figure 52 Alternative sources to help with the details

Most sources I have seen do a good job of giving the overall proportions (see Figure 51), but few tackle the proportions and sizes of the smaller details like the leaves, buds, coliculi, etc. So, in addition to Chitham and my other sources, I scoured the web for resources and took photographs of local buildings here in Chicago (see Figure 52).

Early Attempts

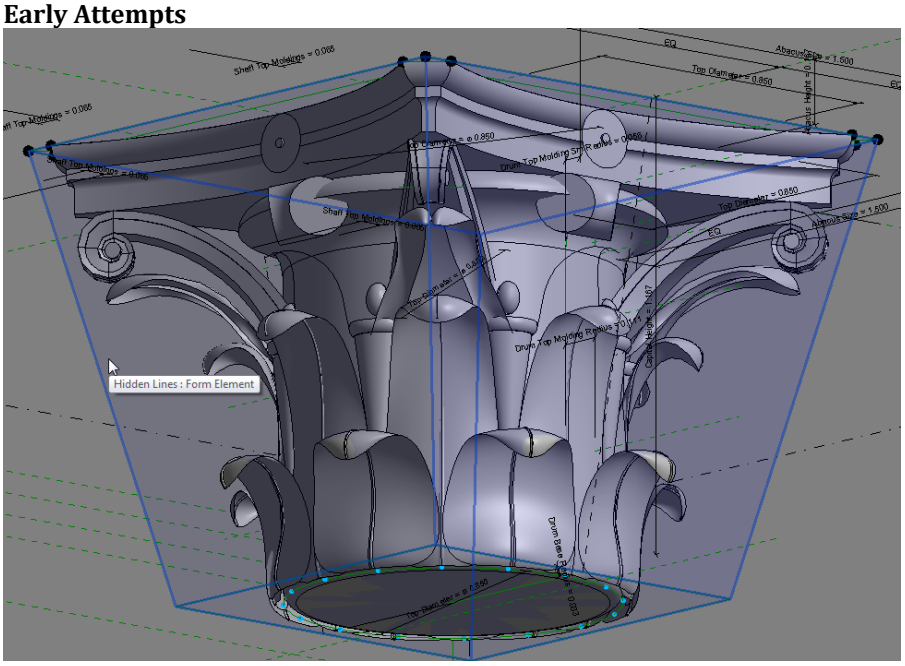


Figure 53 This is the first version I built

The first version (see Figure 53) that I built used a variety of techniques. These included parameters driven by a Base Diameter, rigs and various nested components. The rigs are really the key to making complex forms flex proportionally. If you do a little searching on adaptive components on the Internet,

you are likely to find many sources that discuss the use of rigs. These can be line rigs, rectangle rigs or even box rigs. The idea is simple, you create one of these basic forms and constrain its proportions using just a few dimensions. For example, with a rectangle rig, you only need width and length. Then you use hosted points placed on the edges of your rectangle. Since the Normalized Curve Parameter maintains its position as a percentage of the length along the host line, it will scale proportionally when the rig changes shape. Check out Andy Milburn's blog for some really good tutorials on this technique.

In the first attempt, I succeeded in making a believable Corinthian capital, but there were many issues that I needed to solve to improve it in general, and more importantly to be able to convey *how* I did it in a tutorial. In other words, make it repeatable. File size was also a big concern as I already noted.

I noted that rigs can be 1, 2 or 3-dimensional. In the first attempt, I used box rigs (3-dimensional). The leaves were built within these boxes. The form was acceptable, but the leaves were actually created from 3 splines that attached to these boxes. There were two problems. First, 3D splines are very heavy and make big files. Second, when you create form from a few splines, you get a surface form. Yes it was organic looking and believable, but it was impossibly thin which was noticeable at certain angles and certainly would pose a problem later if we try to 3D print the object.

In the next version I began experimenting with making solid leaves (see Figure 54). If you look at the left side of the figure, these leaves are still quite thin, but they do have some thickness. They are also quite a bit smaller in file size. So progress was definitely being made. To make these leaves, a 2D spline forms the center path of the leaf. Then several "ribs" are created along this path. These are used to create 3D form. The result is a nice smooth, and more importantly, solid leaf form. (More details below).

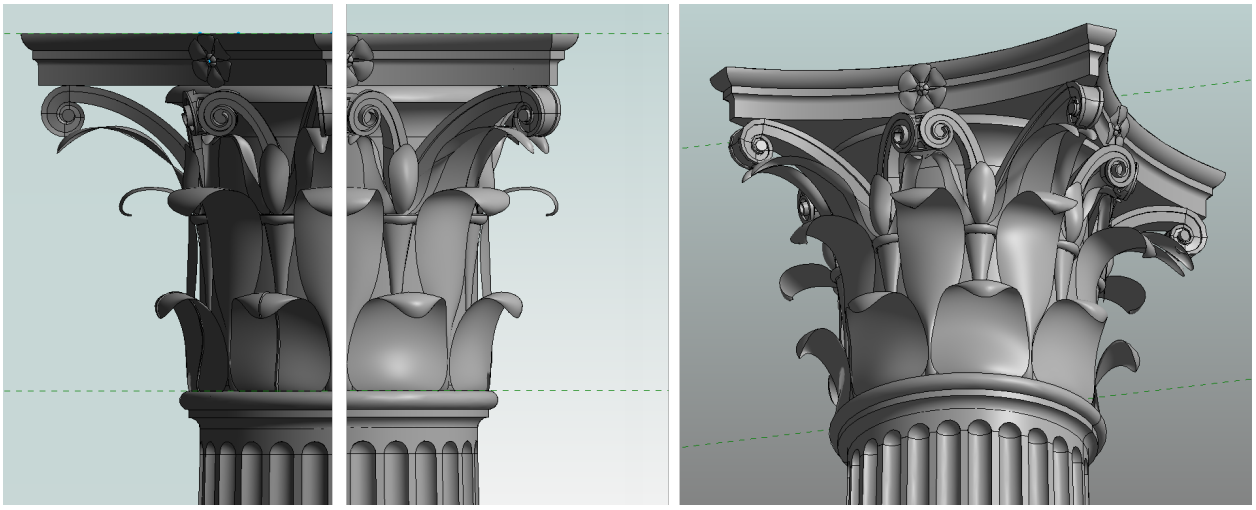


Figure 54 An intermediate version with better lower leaves

Current Version

I figured out much of what I wanted to do in the early version, but completely rebuilt it from scratch for the book. The full column is an adaptive component. The schematic version is the simple block family like the other orders. The base is shared by both medium and fine levels of detail and it is a traditional family. The shaft for medium detail is smooth (also a traditional family) with lines on the surface to represent the flutes. The fluted version is displayed in fine and is an adaptive component (see Figure 55).

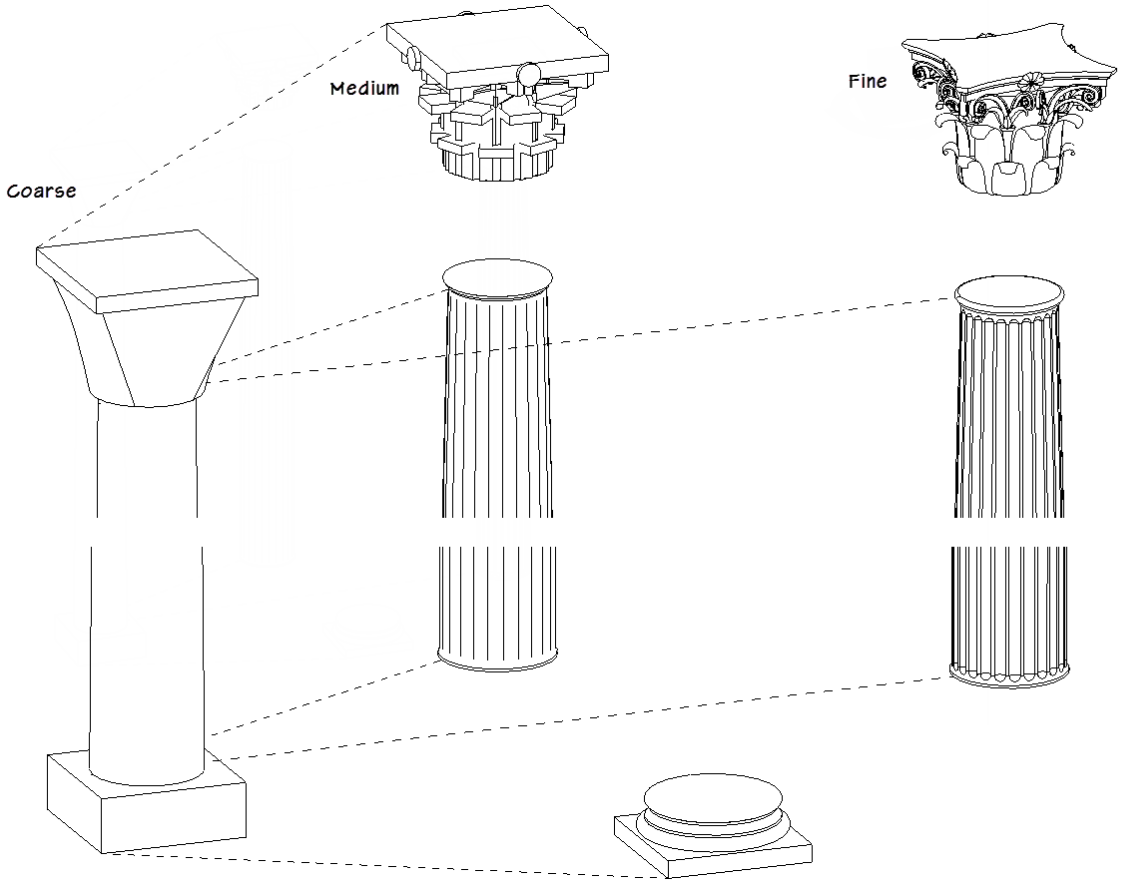


Figure 55 The current version uses three levels of detail

There is a medium detail capital and a fine detail one. Both are adaptive components. The medium could have been a traditional family since none of its forms are complex, but the approach I took was to build the medium version first as simple block forms and then do a save as and modify each simple form directly to create the complex form. In this way, I had already figured out the orientations, proportions, scaling and repetition. All I had to do for the fine detail was create the complex form. The rest would take care of itself when reloading the nested families.

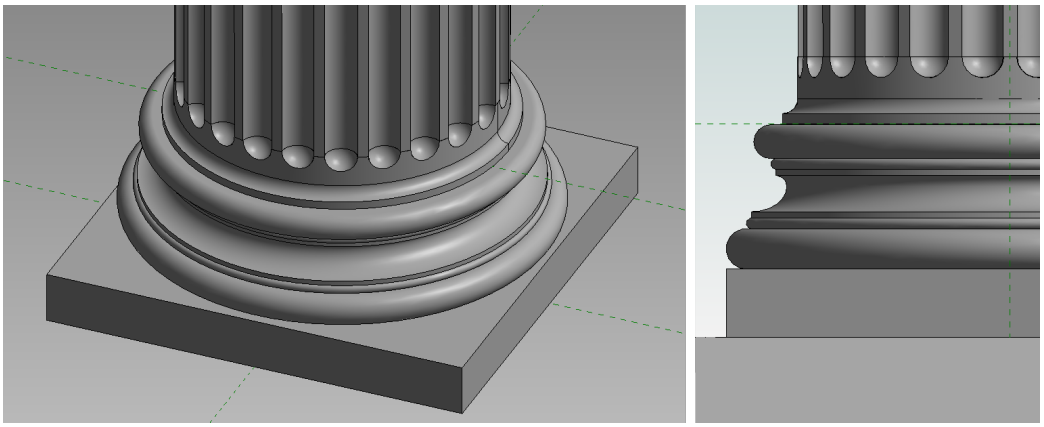


Figure 56 The base is very similar to the bases of the other orders

The base was the simplest part (see Figure 56). It remains a traditional family and I simply did a save as from the Ionic version and modified the moldings as required. This is the real advantage of the approach. Once you build the framework, much of it can be easily resued to create the other families.

This brings us to the capital. I began with the abacus and bell. These provide the backup for all of the other elements like the leaves, scrolls and flowers (see Figure 57).

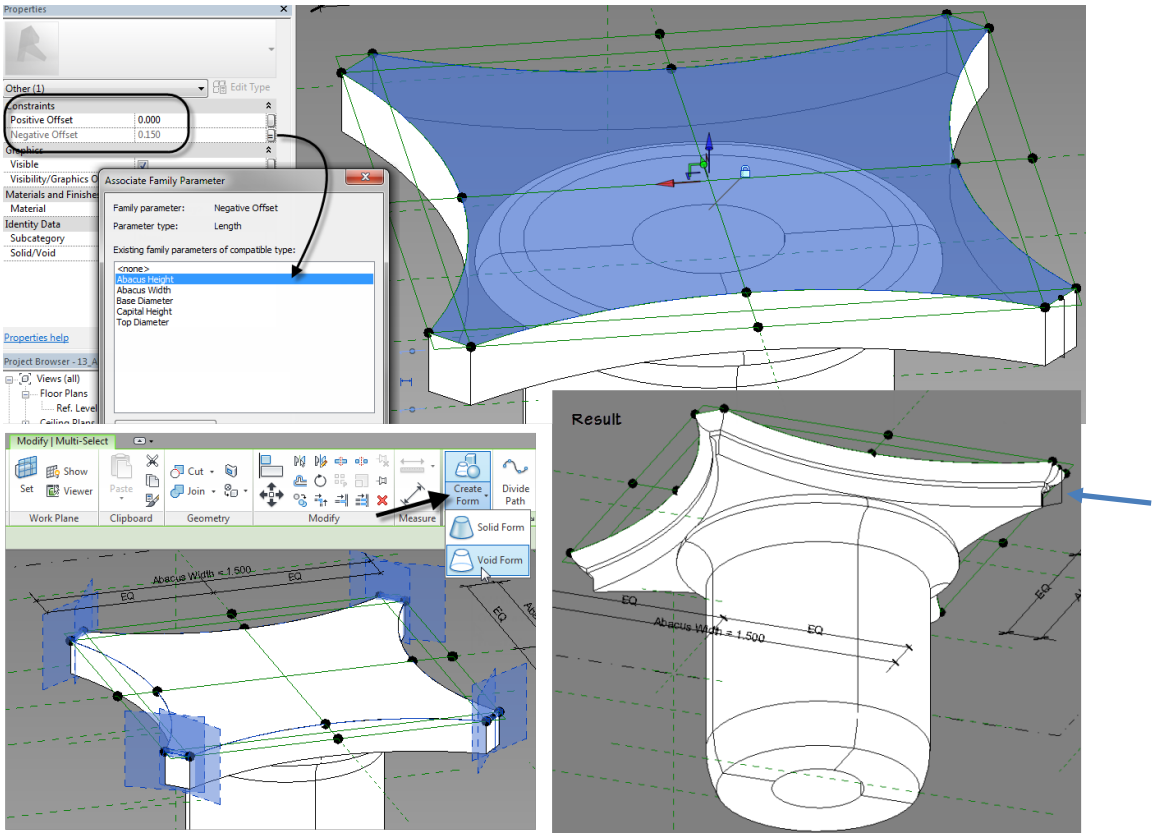


Figure 57 The bell and abacus provide the backup for the other elements

The abacus is an extruded form with a swept void to carve the molding. To do a sweep with a profile in the conceptual massing family editor, you have to use a nested **generic model** family instead of a profile family. Profile families cannot be used in the CME. The generic model family contains a closed model line shape. It has to be model lines. Detail lines and symbolic lines are not recognized in CME. If you look carefully at the “Result” panel in the figure, you can see the nested generic model family near the top right side of the abacus.

As noted above, the leaves use a spline for a path and then have several closed rib shapes placed along this path. This allows the ribs to be lofted to form an organically shaped 3D form (see Figure 58).

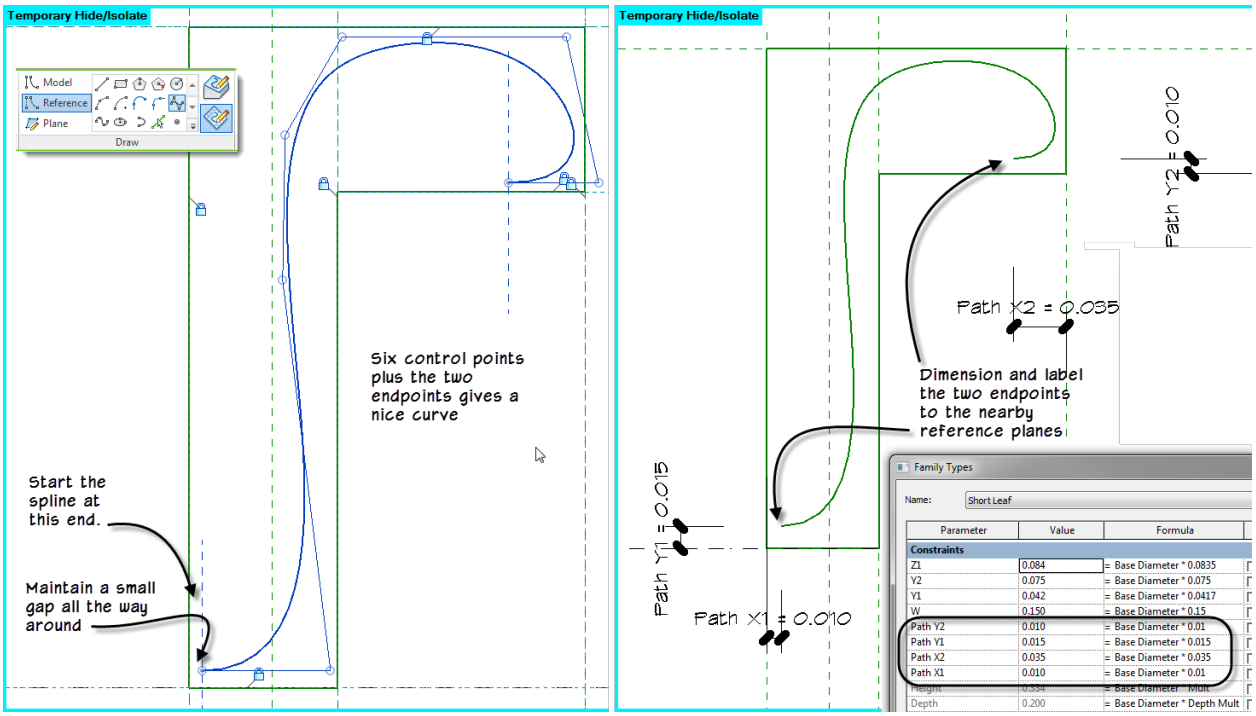


Figure 58 The Leaves are lofted forms in an adaptive family. There is a spline path

Splines have a unique super power. You only need to lock down the two endpoints. When these endpoints flex, the shape of the spline is maintained as it resizes!

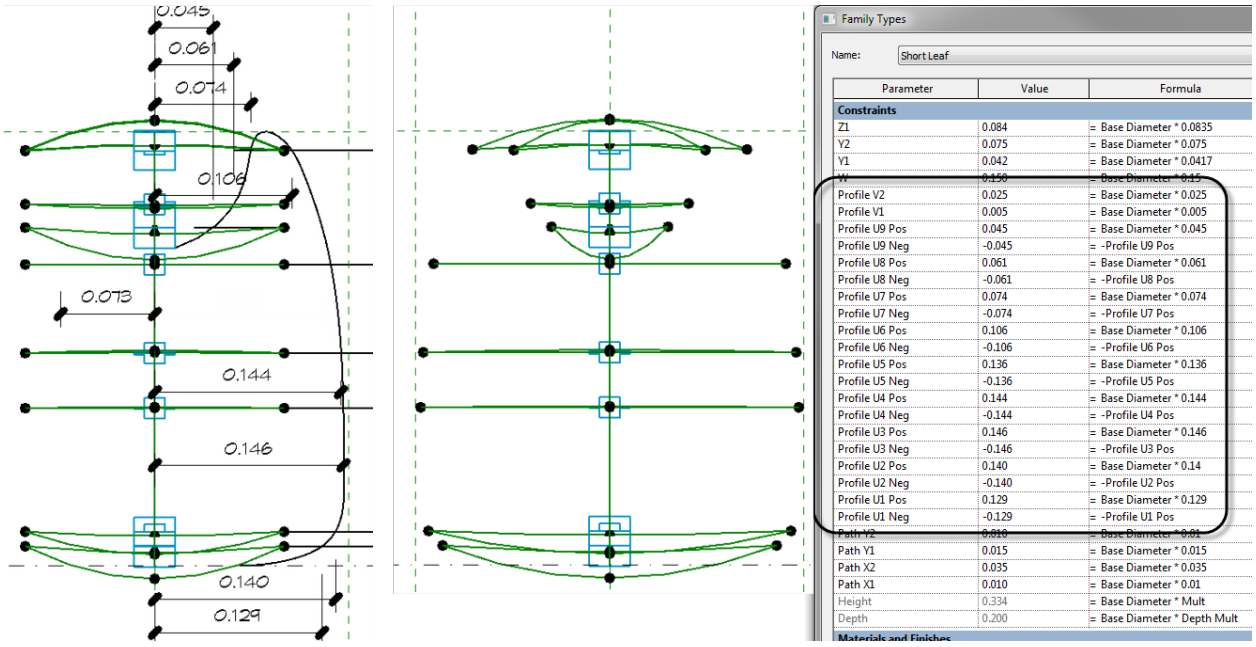


Figure 59 Using hosted points with parameters to ensure that it properly flexes

The next feature to leverage is hosted points. When you host a point on an edge or line or spline, Revit maintains the position of that point with the properties on the Properties palette. One such property is the "Normalized Curve Parameter." This is measures the location of the point along its host as a ratio from 0 to 1. The points thus hosted, can then in turn host other points! This second tier of hosted points have positive and negative offsets relative to host. They can be placed in the X, Y or Z planes. So, we place a point hosted on the spline, then add other points hosted to the first one. Finally we build splines from these. It's like "Dem bones": "the hip bone connected to the thigh bone, the thigh bone connected to the knee bone…"

Use parameters to control the offsets of the points with respect to their hosts. This in turn makes the whole rig parametric and controllable from the single Base Diameter parameter (see Figure 59).

With the leaves, I built everything in the same family. With the scrolls, I used nested generic model profiles instead. But the concept is exactly the same. You are basically creating a parametric loft. The path of the loft is a spline, so you only need to constrain its endpoints (see Figure 60). The shape is maintained automatically. The points hosted on this path can carry other hosted rigs, or nested components. When you link up all the parameters, the entire thing flexes from a single parameter!

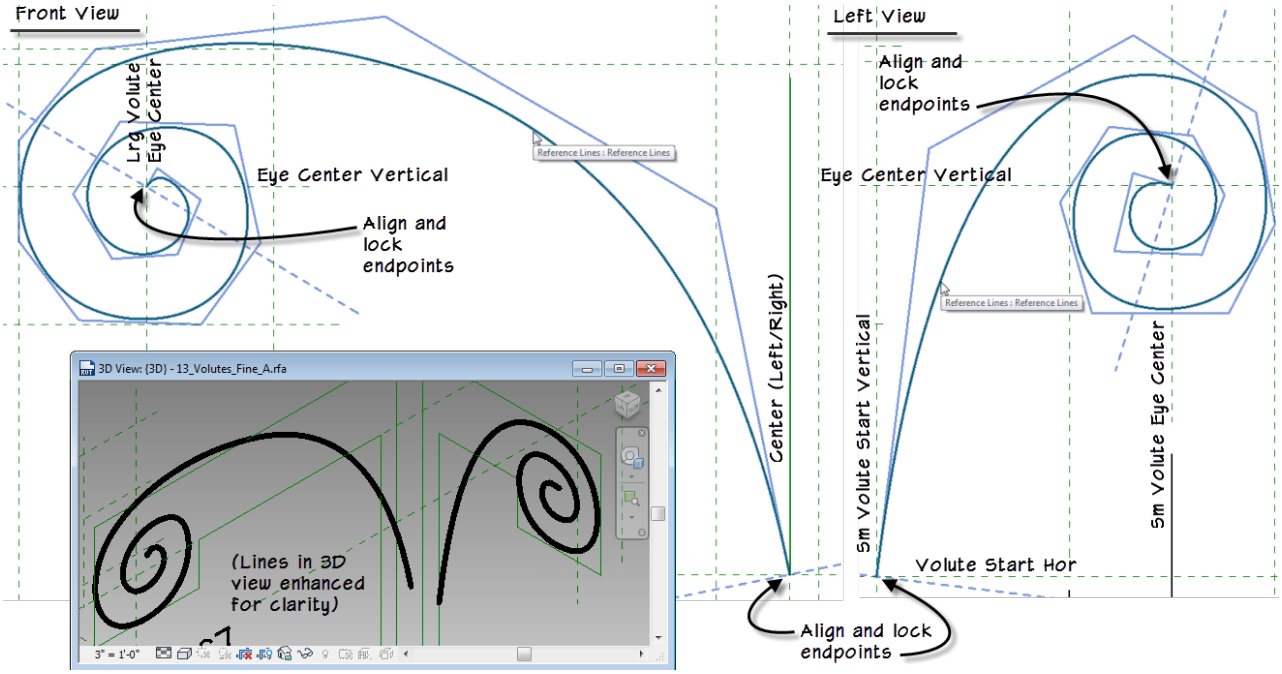


Figure 60 Here I used a single spline to make the path of the volutes. These are constrained and locked

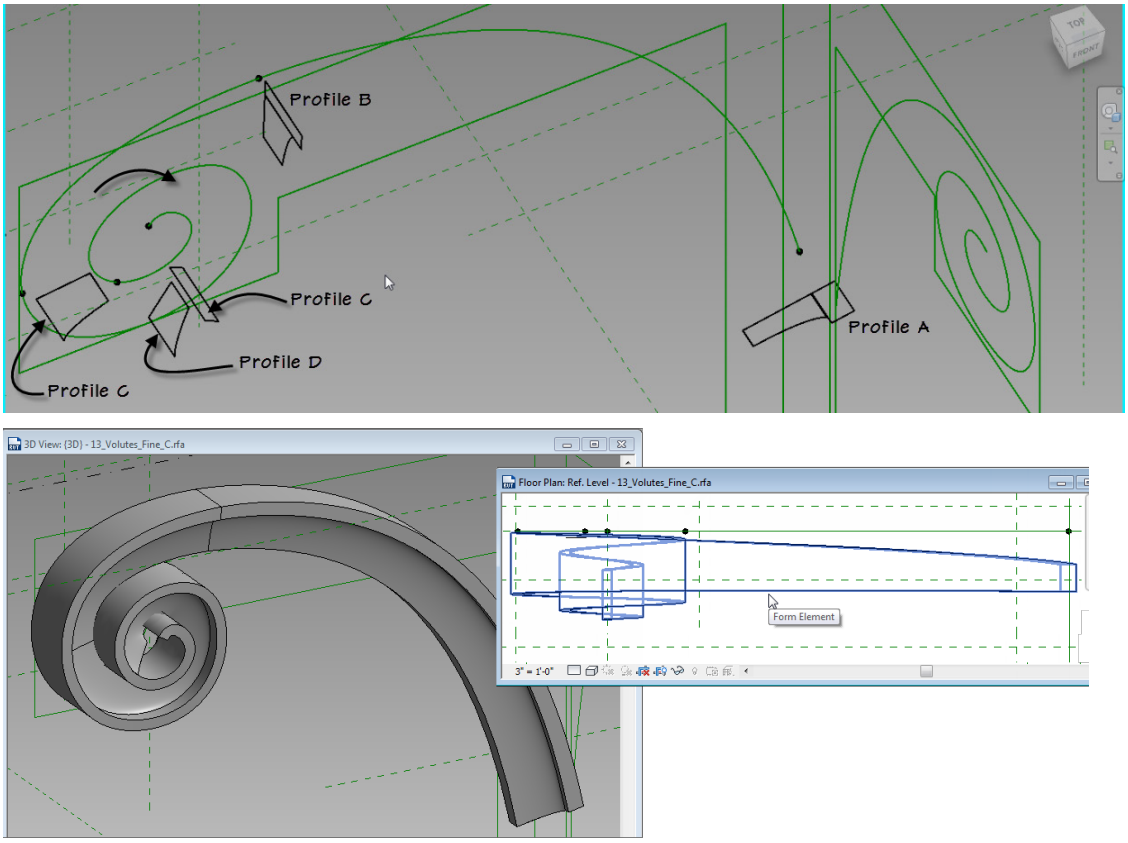


Figure 61 Again hosted points along the spline. But this time, generic model profiles are hosted on these points

You can loft a form between the hosted profiles (see Figure 61).

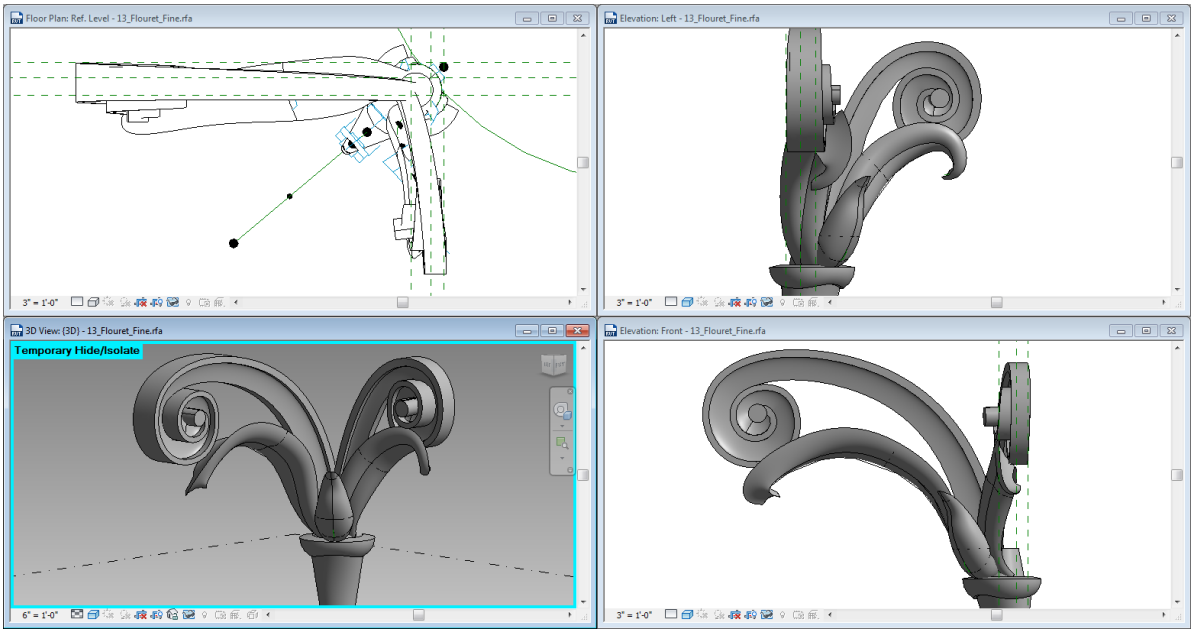


Figure 62 Combine several of the previous techniques to create the remaining pieces

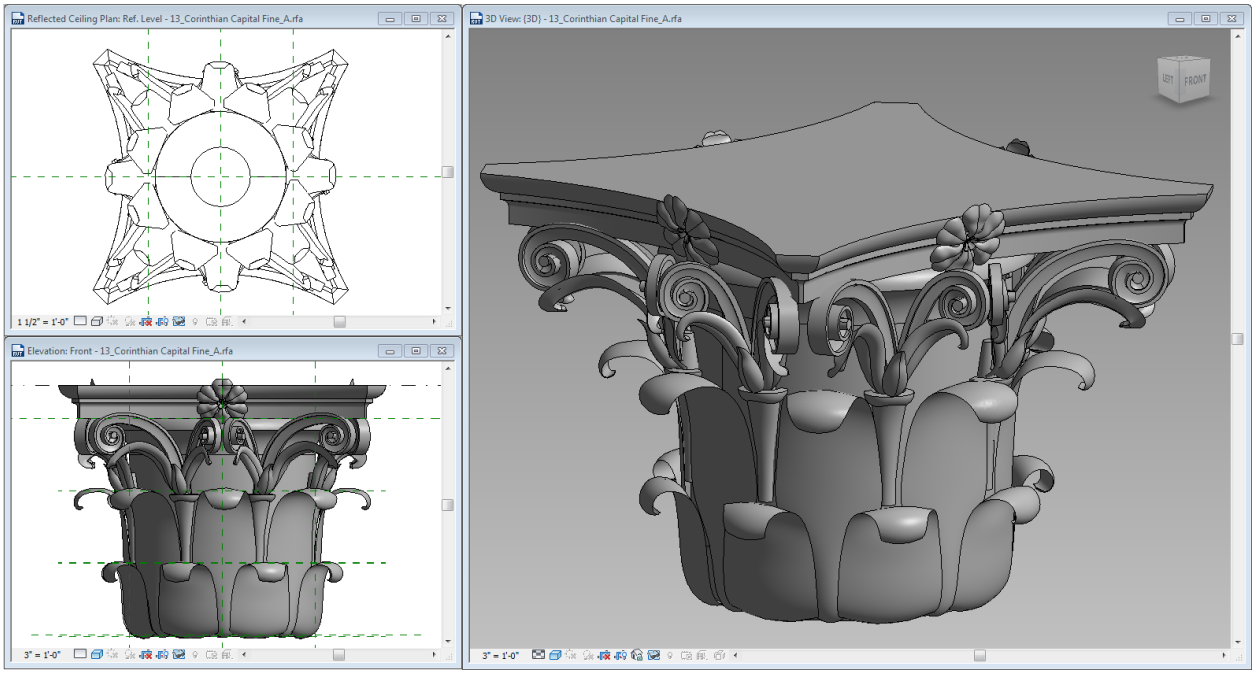


Figure 63 Pull in all of the nested pieces and copy them around the bell

I decided against divide and repeat in this version. Repeaters can be a little difficult to control after they are placed. So I liked the flexibility of simply having free-standing copies.

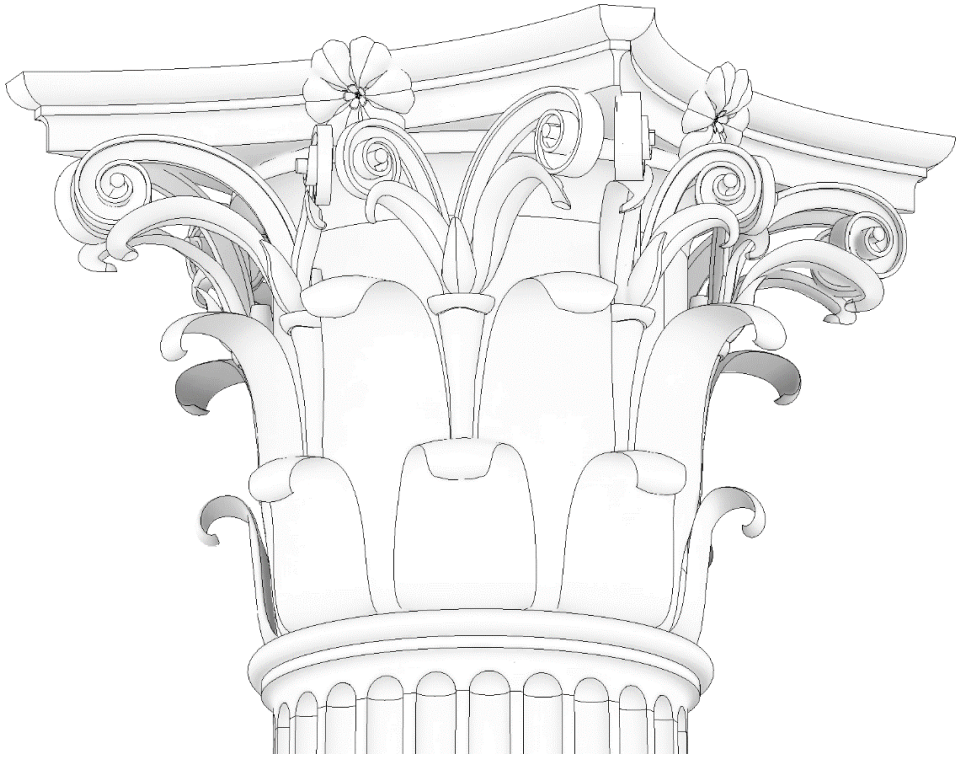


Figure 64 The final version with ambient shadows

Corinthian Summary

- ⇨ Using the Massing Environment
- ⇨ Adaptive Component
- ⇨ Divide and Repeat (Opted not to use)
- ⇨ Using Generic Models as Profiles
- ⇨ Building a Reference Line Rig
    - ⇨ Awesome way to scale…
- ⇨ Creating Solid Leaves

Wrap up

Figure 65 A few more of my sketches from my Italian semester. (Regrettably I have not had much time to sketch since then)

Thank you very much for attending! I sincerely hope you enjoyed it. Please feel free to contact me at:

- **www.paulaubin.com**
- **paubin@paulaubin.com**
- **@paulfaubin**

Learn more about the book:

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

Thank you!