ES122256

Conceptual Structural Design using Revit Adaptive Components and Dynamo

Desirée Mackey, PE, SE
Martin/Martin

Brian Mackey
BD Mackey Consulting

Learning Objectives

- Learn how to utilize repeaters in adaptive components
- Learn how to use divided surfaces for structural applications
- Learn how to use Dynamo to extract structural information that can be input directly into structural analysis software
- Discuss how a conceptual design model was used as a tool for design and analysis

Description

This session will share the design process of conceptual design of a signature pedestrian bridge and an arena roof. Both structures are unique, and somewhat complicated. The iterative nature of the designs prompted the use of both Revit and Dynamo for a coordinated, efficient conceptual phase. To facilitate the concept development, the geometry of the structures was created using Revit conceptual massing and adaptive components. Then, to help minimize the rework associated with verifying the feasibility of each design iteration, the Structural Engineer utilized Dynamo to help automate some of the structural analysis – pulling information from Revit, into a spreadsheet, and ultimately into structural analysis software. The use of several tools, utilizing applicable technology, allowed these projects to efficiently move through the conceptual phase, and into design. This session will share, in detail, how all the models and scripts were created.

Speakers

Desirée Mackey, PE, SE
Desirée has been in the AEC industry since the1990s. After obtaining her bachelor's and master's degrees from University of California, Davis and Massachusetts Institute of Technology, she perpetuated her nerdy tendencies with Revit. She started her career in California with a construction company, then with a structural engineering firm, and now is a practicing structural engineer and BIM Manager at Martin/Martin in Denver, Colorado. Desirée is a regular speaker at many conferences, she co-founded the Rocky Mountain Building Information Society, is the Chair of the Structural Engineers Association of Colorado's BIM Committee, served as an AUGI board member, Treasurer, and Vice President, and is a member of the BILT North America Committee. Finally, as if that is not enough Revit in her life, she is married to “The Revit Geek” and acts as a partner in his BIM consulting firm, BD Mackey Consulting.

Brian Mackey
Known as “The Revit Geek”, Brian has spent more than 25 years in the industry, more than 10 of which have been focused on Revit. Over a decade of working with Architects and Engineers to advance BIM in their companies, Brian started his BIM consulting company in 2011 to focus on custom high-level training/mentoring. Brian has clients all over the US/Canada that generally tolerate his sarcastic nature in exchange for his wide breadth of BIM knowledge. Brian showcases his love of talking about Revit, or maybe just his love talking, in a monthly light-hearted, occasionally irreverent, free Q&A webcast, Revit Radio.

Introduction

While structural engineers don't always participate in early conceptual phases, it does happen when the structural elements themselves are a primary part of the overall aesthetic. Collaborating as a team early on has its obvious benefits, but working on these early phases together also provides an avenue to have more detailed, functional models earlier (without having the "throw away" conceptual model that is rarely realistic). However, the challenge to advancing to this level of structural detail so early is that the major design decisions are still undecided, which leads to an iterative process that can become complicated and time-consuming. This leads to the generation of detailed, yet flexible, models that can be flexed and changed until a final design is decided upon.

The two examples shared here are projects where the structural engineer participated in producing these conceptual models.

Pedestrian Bridge – Geometry, Constraints, Expectations, Goals

The geometry of the bridge was generally set by the owner – they had a very specific look they were going for. The idea was that the bridge would have an inner and an outer spiral, which the owner referred to as "helixes". The outer helix would "wrap" around the bridge the opposite direction as the inner helix. There were other basic geometric constraints such as general size, slope, etc., but the primary geometrical constraint was the look of the helixes.

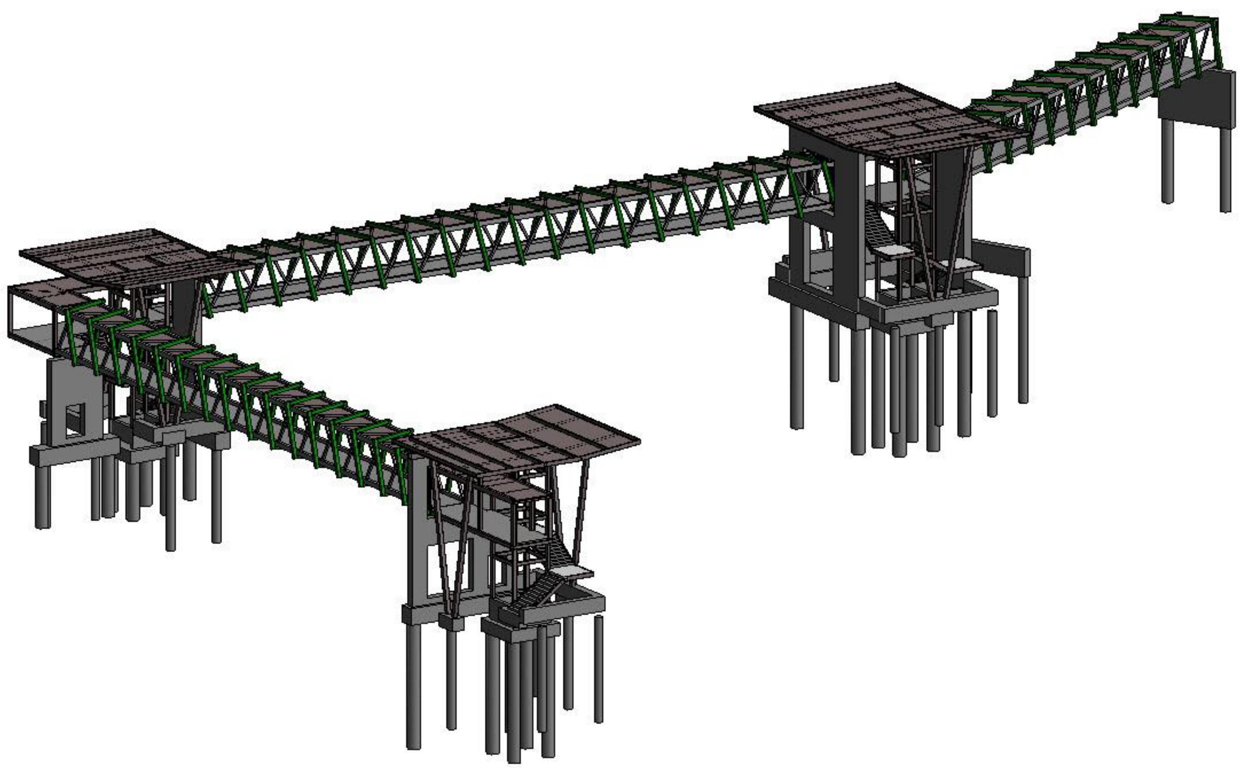

Figure 1: Pedestrian Bridge

The architect wanted the model to be flexible to help the owner understand how the helixes would look. They wanted to optimize how many times the helix "wrapped" around, the angle of the various portions of the helix, and the orientation of the bottom members (perpendicular to the bridge deck, or angled like the rest of the helix members.

Internally, we hoped that the model would not be a "throw away". While we value the creation of models and other visual aids for design purposes, it would certainly be more efficient for the family and model to be used beyond the conceptual phase and into the design phases.

Arena Roof – Geometry, Constraints, Expectations, Goals

This example was a result of our client asking, "hey, could something like this work?" We asked some follow up questions about design intent/desired aesthetic and decided to produce a design aid that our client could use to manipulate the aesthetic of the roof, then could send back to us for feasibility studies.

The geometric constraints were few, and preliminary. There was a general overall shape that looked something like a rhombus or a trapezoid, and the desire to adjust both the overall curves, as well as the vertical offset over the length of the shape to create a bit of a dome effect. Finally, there was a desire for a "honeycomb", for lack of a better description, look to the framing. The image below is a preliminary sketch explaining the general aesthetic goals.

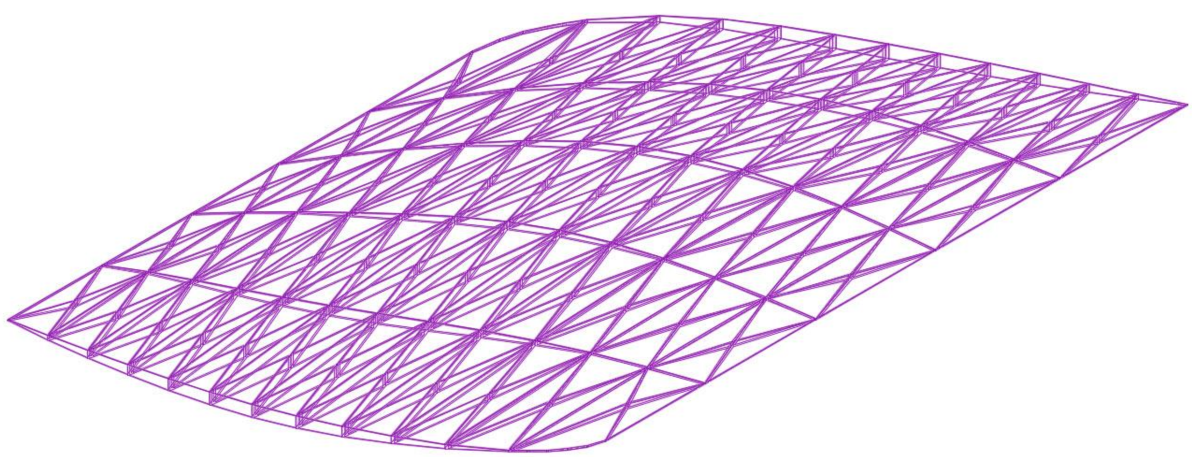

Figure 2: Arena Roof

In addition to providing our client with a design aid, we wanted to be able to extract the coordinates of the structural members to import them into an analytical program to quickly run these feasibility studies.

Pedestrian Bridge

General Approach

The bridge is composed of a few layers of nested families, all of which will be described in further detail below. A profile is nested into an adaptive component. That adaptive component is then nested into the host adaptive component family. The host family utilizes divided paths and repeaters to place the nested component. Once completed, three versions of the family were created for the three segments, and these were used in the project environment.

Each of the locations where elements came together will be castings, so an additional adaptive component family was created and loaded into the project for the castings.

Finally, to enable the Dynamo workflow, a simple “analytical node” family was created and placed into the bridge family. Dynamo then pulled the coordinates of the “analytical nodes” out of the family and put them into a spreadsheet. The spreadsheet could then be loaded into the analytical software for quick modeling and analysis.

Nested Family

The nested family is a basic two-point adaptive component, with a rectangular profile. However, the two-point family has several enhancements to make it more user friendly.

First, while there are the two basic insertion points, additional points are hosted on the reference line that connects the two placement points. Because the profile is hosted on these additional hosted points, properties in the family can be created to allow the profile to be placed along, or even beyond the line. Thus, the shape can have either a cut back, or an extension.

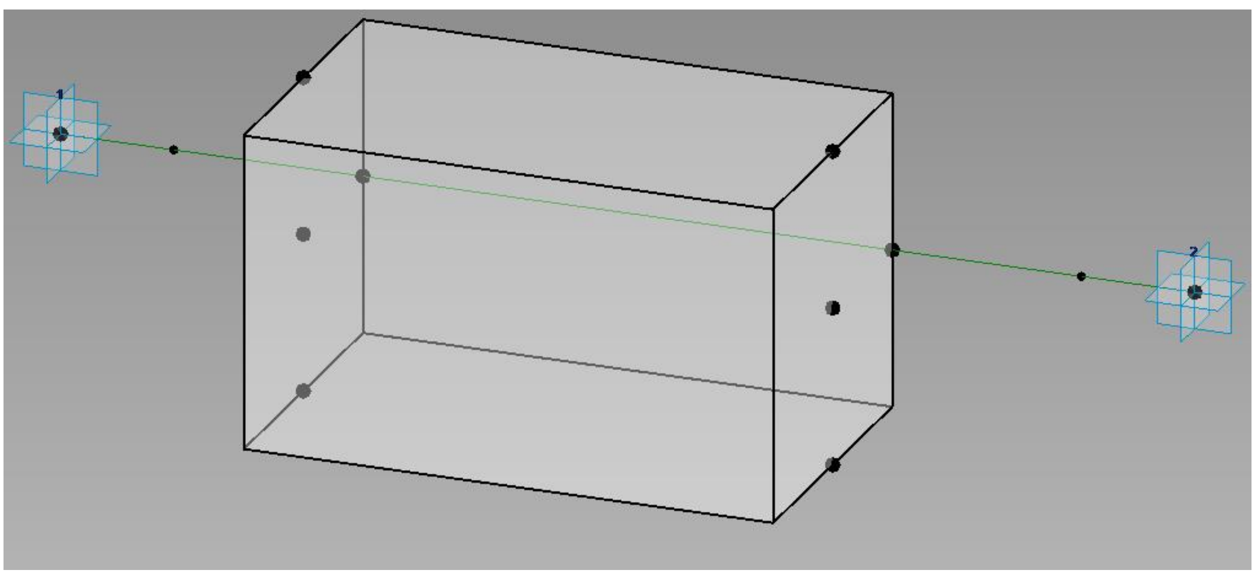

Figure 3: Two-point adaptive component

The second modification is that parameters were added to the family to adjust where the profile is hosted on the line – this gives a similar functionality as the y and z justifications and offsets of structural framing families.

Host Family

The host family is also an adaptive component. Originally there were two placement points, with a reference line between them. Additional points were then hosted on these points, and then lines between those points were added as well. Eventually the second placement point was changed to a reference point so that the overall length could be set within the family.

Once the points and reference lines were modeled, there were a total of 8 lines, 9 if the original is included. These 8 lines were then divided using the Divided Path tool, with the quantity of divisions being parametrized. Like the two-point adaptive family, the ability to offset the first and last points was also added.

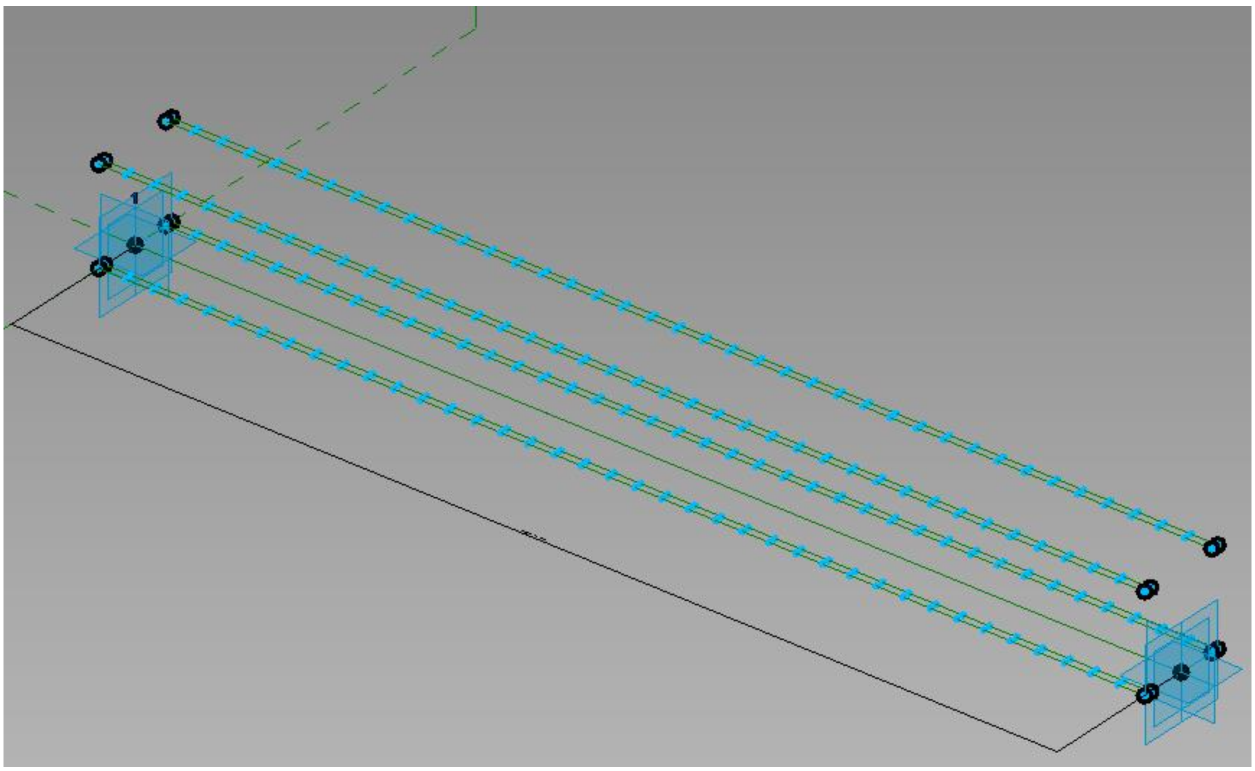

Figure 4: Divided Paths

Building the Family – Repeater

Next, the two-point adaptive component was placed, hosting on various points to create the desired look. The justification points and offsets were flexed to create the overall geometry. The interesting item to note here is that the justification points were not always intuitive, so it was a bit of guess and check to get the family in the desired location. Once two "revolutions" of the "helix" were in place, so eight instances, the repeater command was used to repeat the pattern along the path. (If only one instance was placed the spacing would not be correct.)

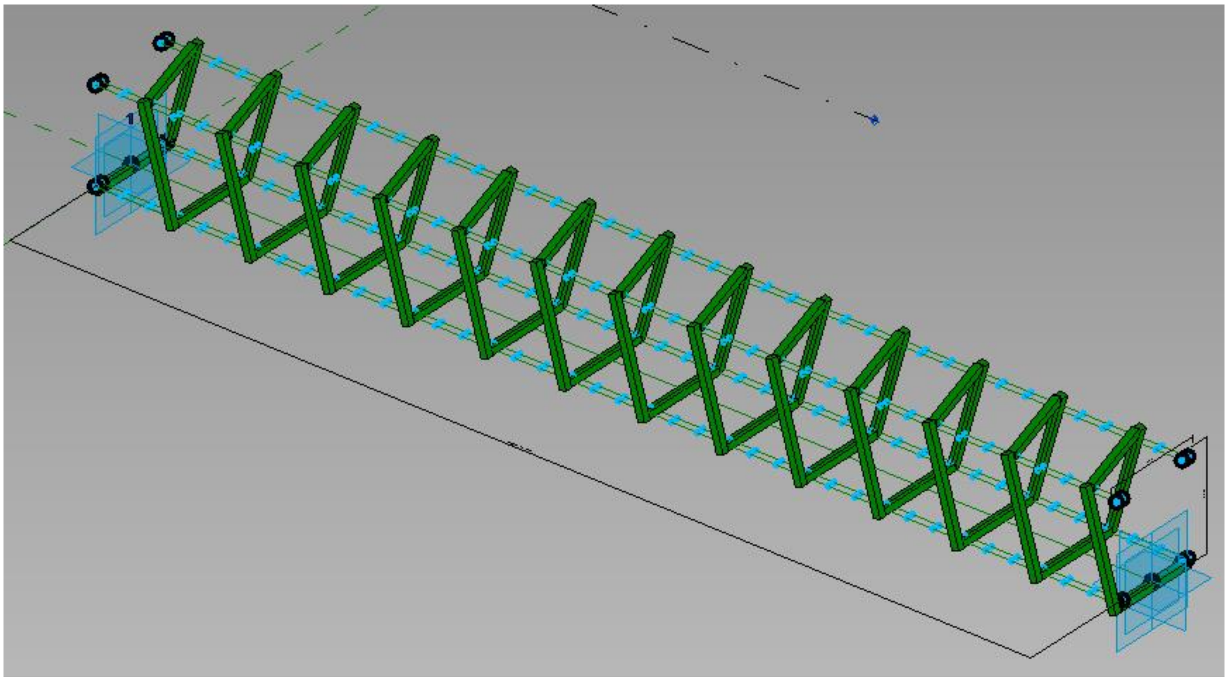

Figure 5: Repeater

Eventually all instances were placed, manipulated and repeated, as shown in the image below.

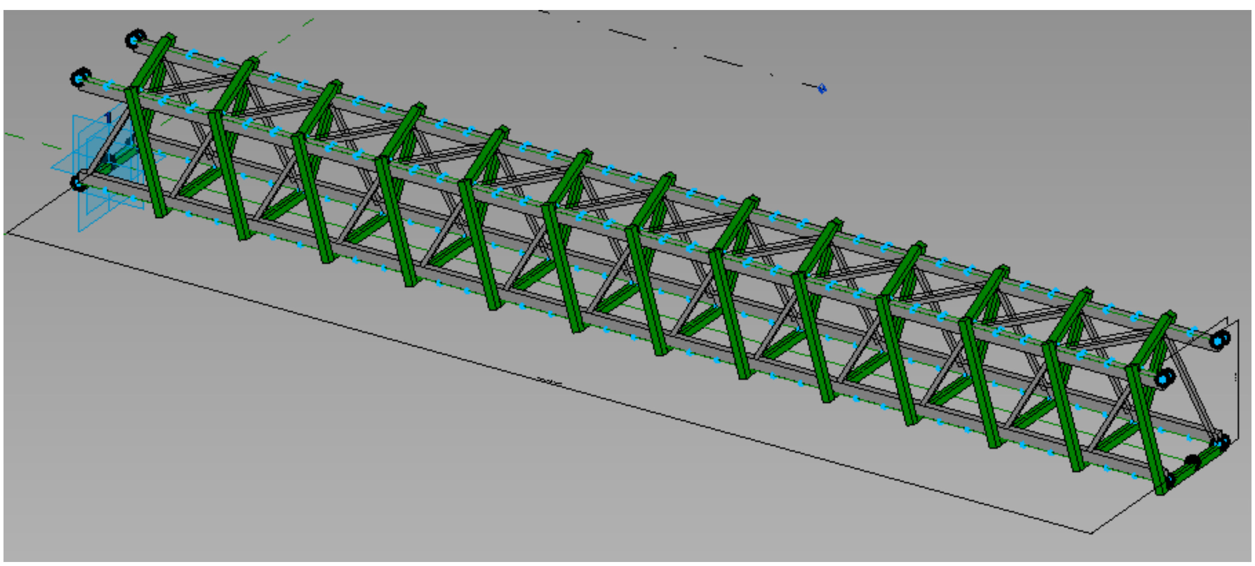

Figure 6: All Repeaters

Iterations of the Family

Early in the design process, there were several different aesthetics the client wanted to test, so a few different versions of the family were created: One where the bottom member of the helix was perpendicular to the axis of the bridge; one where the bottom member was at an angle; one where all the members were tight against each other; and one where there were gaps between the inner and outer helixes.

Placing and Manipulation

Originally the goal was for one adaptive component family to be placed in the project three times, one for each span. In addition, the early iterations of the family were meant to be manipulated to understand how many "revolutions" of the helix, the angles between members, and other design decisions.

Most design decisions were made early, so some of the parametric behavior of the family became less important. In addition, manipulating the parameters within the family turned out to be a little less user friendly than we had hoped. For example, we found that the repeater grew from the center, so changing the number of "revolutions" had to be done in increments of two. End-users were also struggling to place the family in exactly the length that worked for the desired revolutions, and the length of each bridge segment was known early on. For these reasons, additional iterations and modifications of the family were created.

A separate family was created for each bridge segment, and the second placement point was changed to a normal reference point, thus "baking in" the length. Also, while the parametric divisions were no longer needed within the project, they were useful when making the additional iterations of the family.

Once in the project, additional elements, such as floor and roof decks were added to the model.

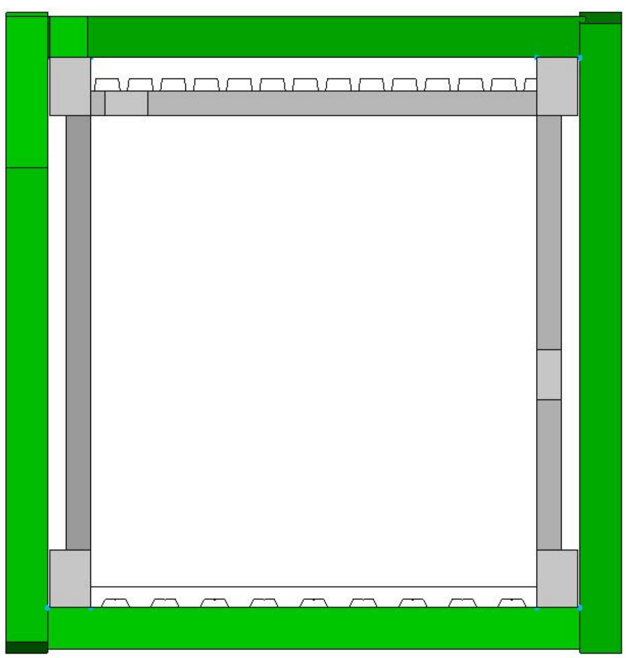

Figure 7: Bridge Cross-Section

The "Castings"

The places where the members came together, the joints, were intended to be castings. One goal was to minimize the number of castings created, so symmetry was a big factor. Once bridge geometry was set, we went about modeling the castings. The client wanted a smooth overall look, with no visible change in diameter of the members. Unfortunately, given the geometry and angles of the members and they turned the corners, this was hard to achieve, and difficult to explain to the client. The crux of the challenge was getting the end of a square-cut member to align to an angled, biased-cut member.

We used various massing tools to come up with iterations showing different ways to blend the members around the corners. At one point, we considered 3D printing the various options for a visual aid, so we even took the time to fillet the edges and hollow out the members. To do this we found it easiest to use AutoCAD…we tried repeatedly in Revit but failed.

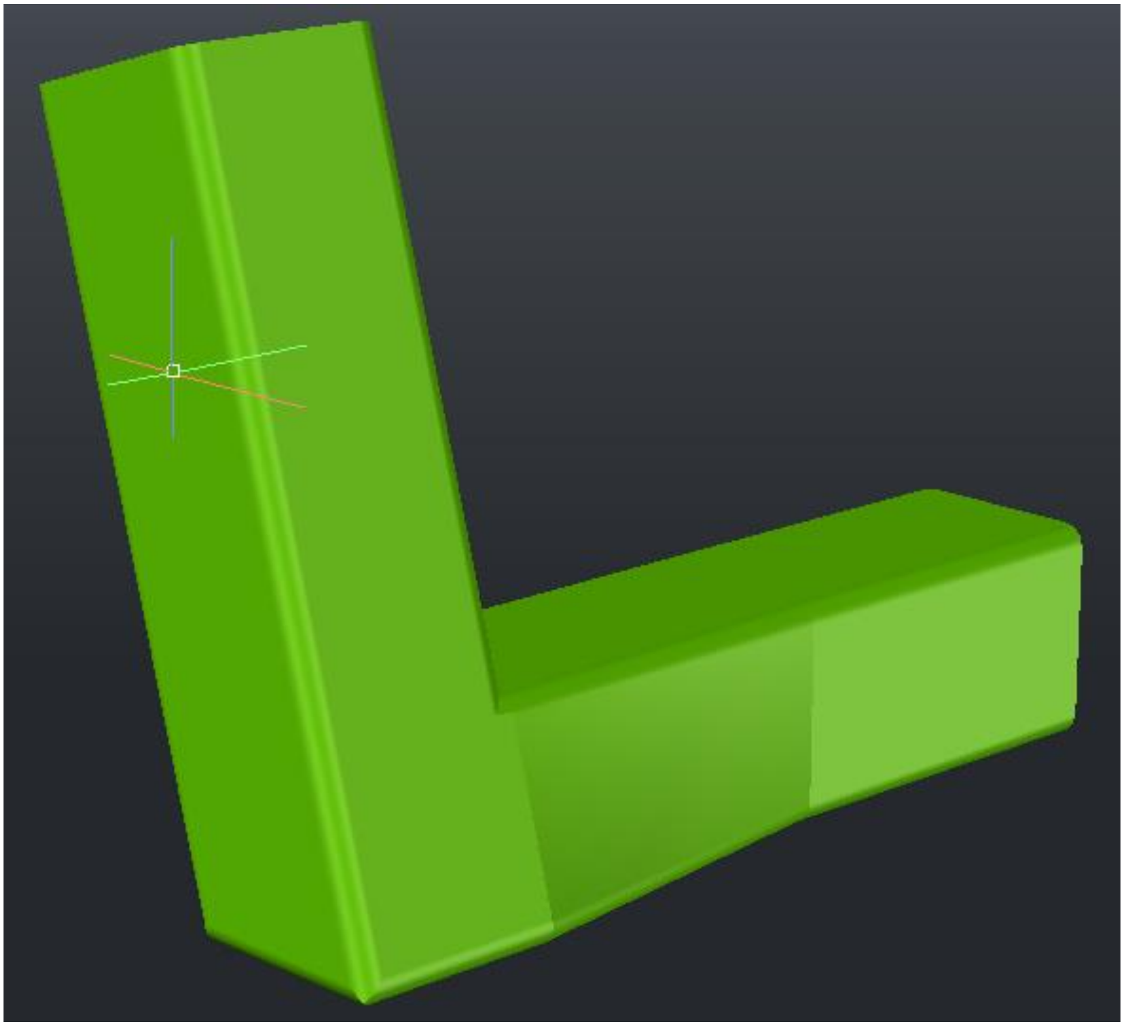

Figure 8: "Castings"

A fabricator was brought on-board relatively quickly, and they took over some of these finer details.

Dynamo

Along with the obvious geometric decisions made during the conceptual phase, there was also a certain amount of structural engineering during this phase. Since we participated in this phase and generated the model, the expectation is that each iteration is generally structurally feasible. Given the somewhat complicated design – the structural members were located based on a desired aesthetic rather than structural efficiency and load path – most of the preliminary feasibility studies were done using a quick run through in an analytical software. Given the iterative nature of the process, and the amount of time it would take to build a separate analytical model for each iteration, we came up with a different approach. In addition, the family was built as a generic model, so the analytical properties were non-existent, so a simple export to analytical was not really an option.

The general approach was to write a Dynamo script that extracted the coordinates of the "analytical nodes" and/or members, dumps those coordinates into a spreadsheet, and then the spreadsheet could be loaded into the analytical software to recreate the analytical node and member locations. The first attempt was to achieve this by extracting the coordinates of the endpoints of each member. Essentially, the script was: Get all the members>>turn members into lines>>get endpoints of lines>>send coordinates to excel. However, given the type of members used – generic geometric shapes – the Dynamo functions that dissolve members into lines gave lines defining the profile instead of the centerline.

The workaround for this issue was that an "additional node" family was created and hosted on each member intersection. Then, Dynamo extracted the coordinates of the "analytical node" family only. A screen shot of the script is below.

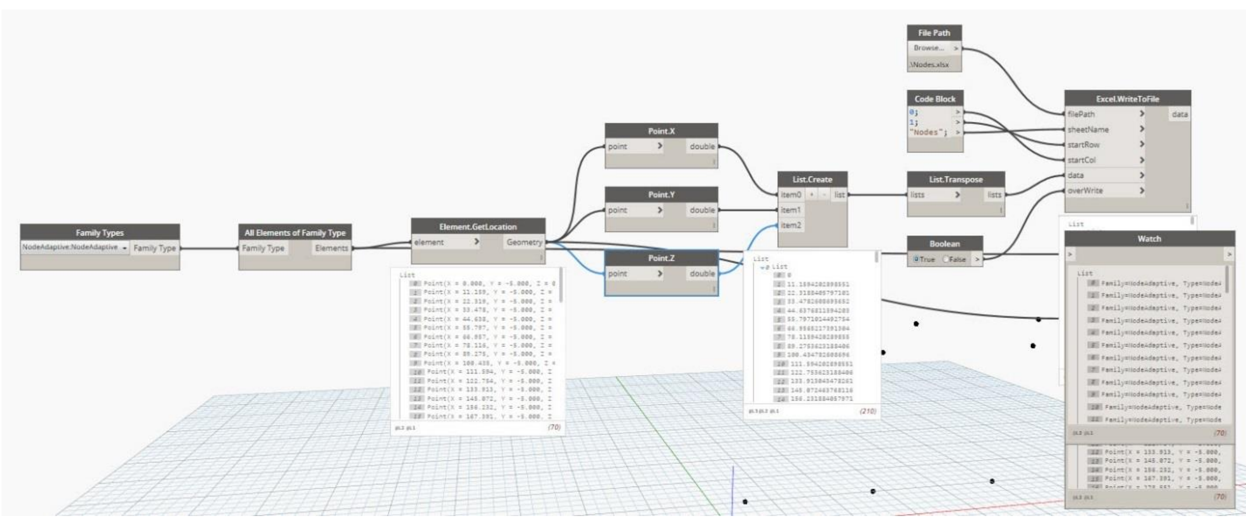

Figure 9: Dynamo to extract coordinates of analytical nodes

Since creating these variations of this script for this bridge, we have created several additional iterations and have improved on this process.

Visualization

For visualization and communication, and let's be honest, for fun, we took a couple things out to Enscape, a common visualization program. Below are two QR codes that you can scan to take a look.

Figure 10: Bridge QR Codes

Arena Roof

General Approach

Our client's main goal for the model of this roof was to be able to quickly visualize various framing schemes for the aesthetic. Our goal was to just as quickly be able to tell the client if the framing schemes were structurally viable. There was not an initial goal for the model to be useable past this crude conceptual phase.

Given the basic visualization and analytical requirements, and no documentation requirement, we decided to use a curtain panel to build a roof family. We created a basic rectangular 2-pt adaptive component to represent a framing member, and nested it into a curtain panel by pattern family, hosting the adaptive component on the nodes of the chosen pattern. This was then nested into a mass family.

The roof itself was created within the mass family. We had a rough "outline" of the roof shape from the client, so we created three curves that followed the desired shape. We lofted the curves to create the surface. We then divided the surface, and applied the nested pattern. Manipulating the chosen pattern and division rules allowed the client to create a plethora of iterations.

We wrote a dynamo script to extract the coordinates of the end points of each of the members. Once the client selected various "favorites", we ran the script and produced excel output of member coordinates. This output was then manipulated and loaded into analytical software for quick feasibility studies.

Nested Families

The 2-pt adaptive component started out as the same one as was used in the bridge, however, due to an issue with the Dynamo script, and to increase regeneration speed while manipulating, we reduced the family to just a 2-pt adaptive line.

The 2-pt adaptive line was then nested into a curtain panel by pattern family and then hosted on a selected pattern. A few different combinations of patterns, with and without the geometry of the 2-pt family, were created and tested.

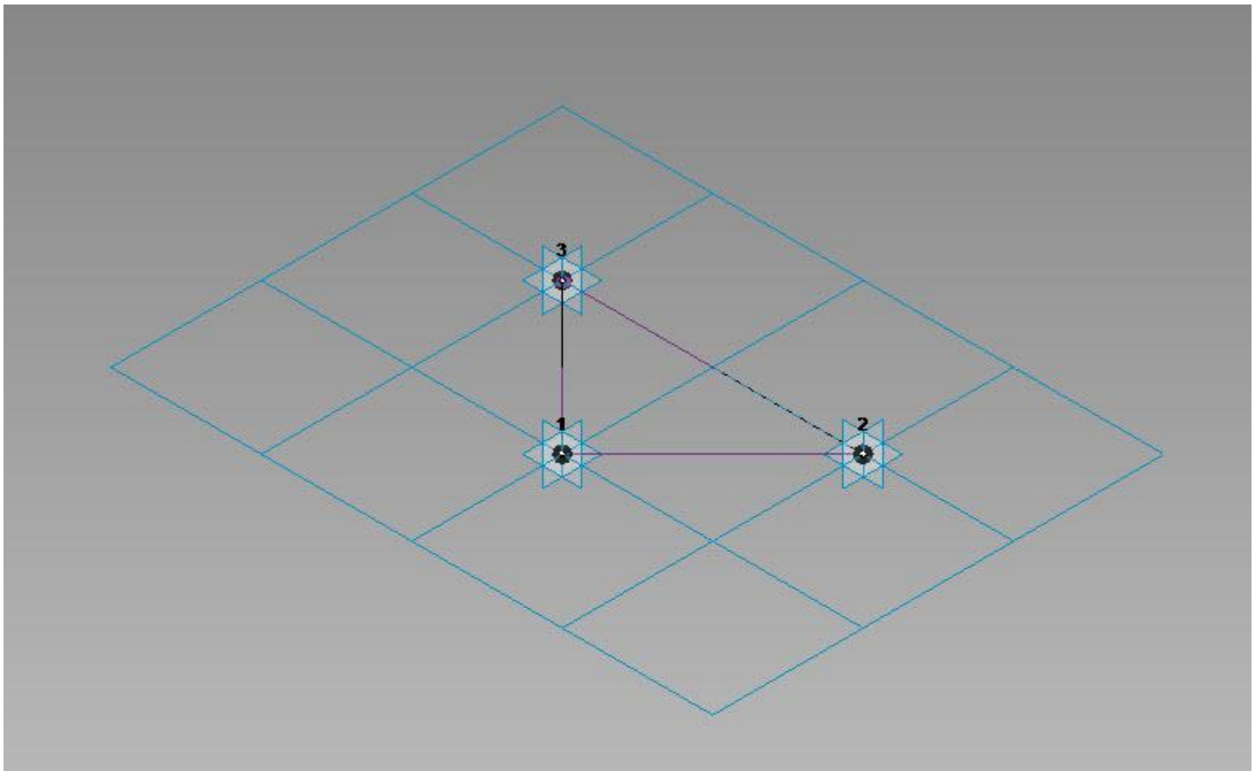

Figure 11: Curtain panel by pattern

Host Family

The host family is a mass family. To create the mass, three curves were created to roughly match the desired roof shape.

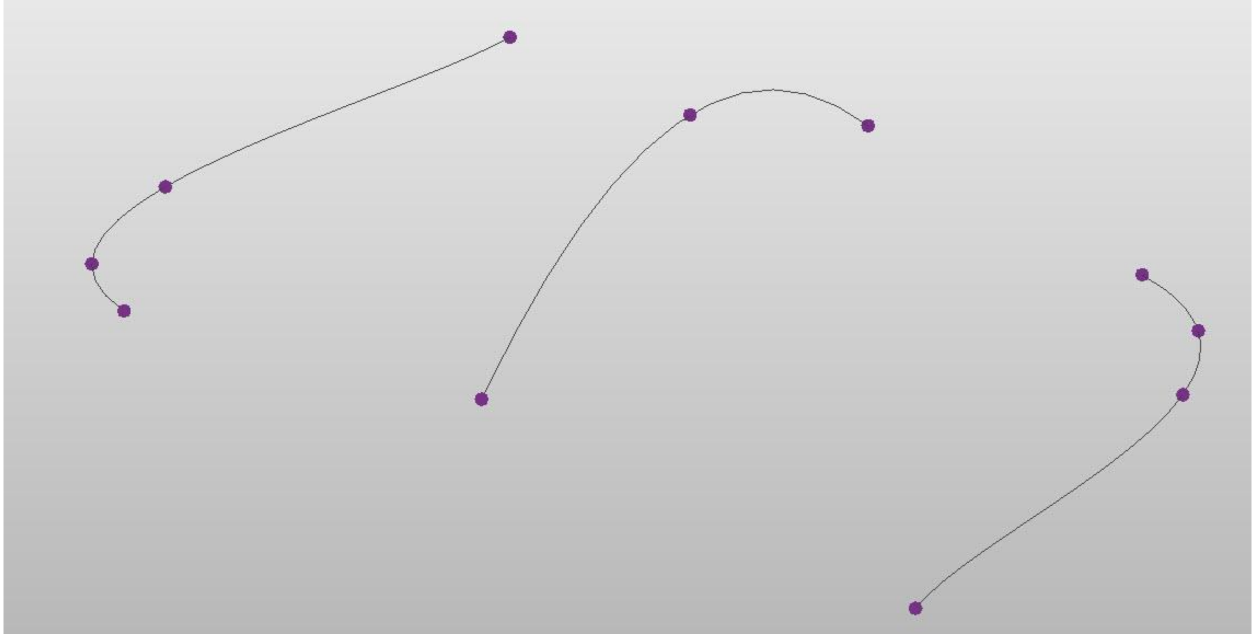

Figure 12: Three defining curves

The three curves were then lofted to create a surface. These curves could be adjusted later to adjust the shape of the roof.

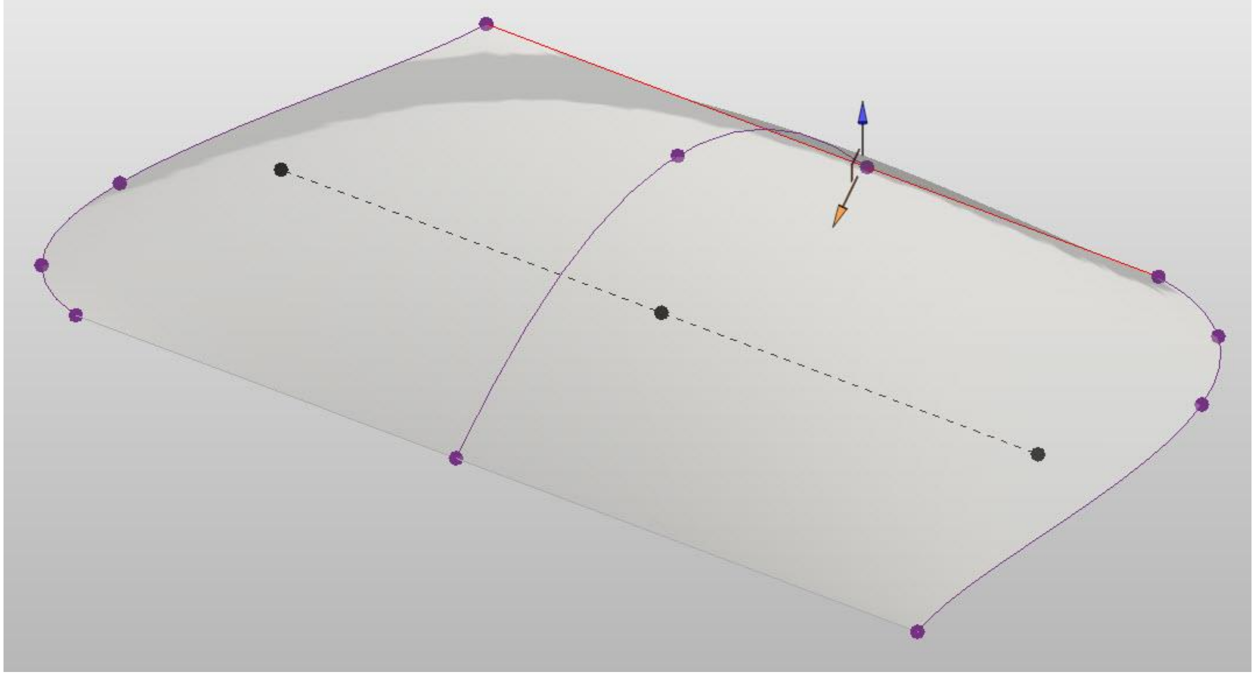

Figure 13: Lofted Surface

Building the Family

Once the surface was complete, that surface was then divided.

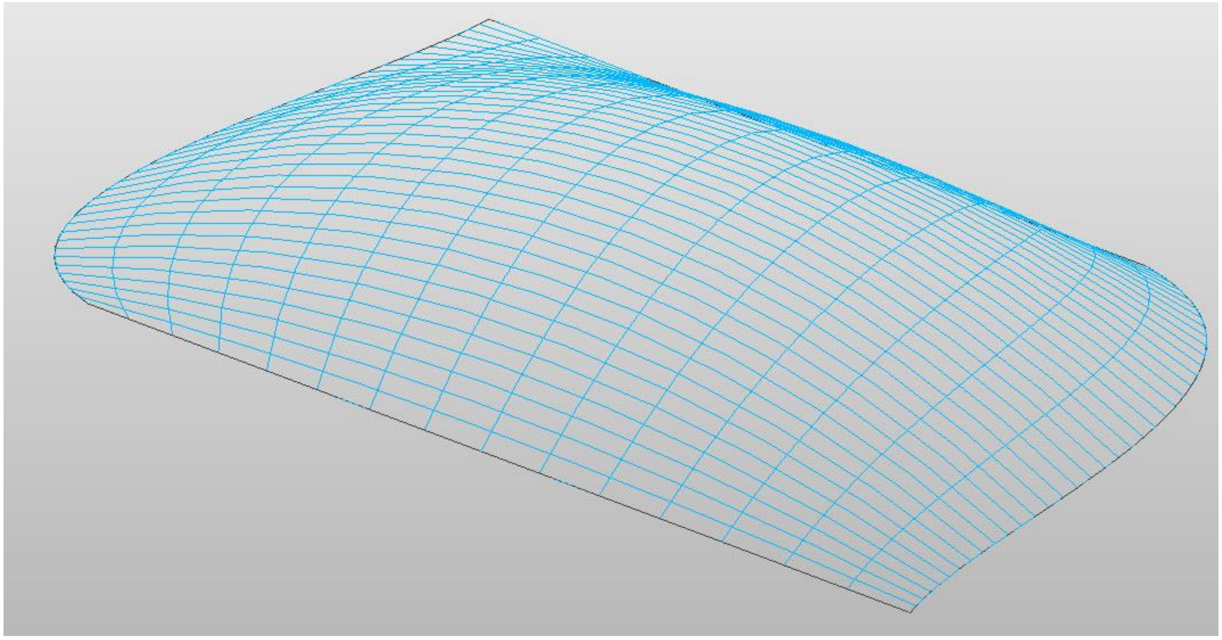

Figure 14: Divided Surface

The curtain panel was applied to the divided surface. The aesthetic of the roof could be manipulated by changing the pattern, or by adjusting the number of divisions.

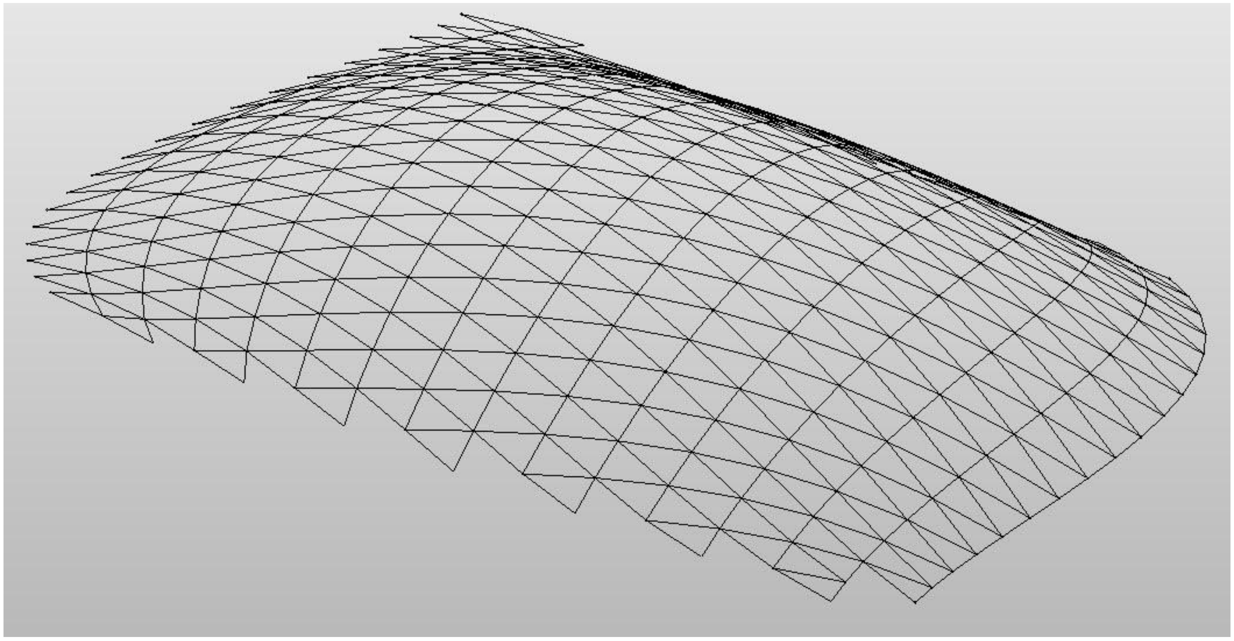

Figure 15: Applied curtain panel

AUTODESK UNIVERSITY

Manipulation

Manipulating the roof, within the mass family, is relatively simple. There are three options to change the aesthetic of the roof. The first is to adjust the curves to change the overall shape, which is accomplished by selecting the surface and adjusting the curves or points making up the curves. The x-ray tool is helpful for this process. The second option is to change the number of divisions of the surface, which is simply done by selecting the curtain panel and adjusting the division numbers in the properties, or on the options bar. Last, the pattern applied to the divided surface, making up the curtain panel, could also be swapped out for a different pattern. This is done by selecting the curtain panel and choosing a new pattern from the type selector.

We found that the curtain panel patterns that had an element nested into it, especially if that element had geometry more than a line, took an extremely long time to regenerate after changes. The single line component, or even no nested geometry at all helped speed up the iterations.

Dynamo

The goal structurally was to quickly extract the coordinates of the end points of the framing elements into a spreadsheet that could be manipulated and quickly put into an analytical software.

As was discovered with the bridge family, limitations with Dynamo and generic model families complicated the goal to extract the coordinates of the framing elements – extracting a line/curve from a generic model yields curves for the profile, not just the location line. There were two solutions for this. First, we used the same solution as with the bridge – nested in an “analytical node” to the curtain panel by pattern family and then extracted the coordinates of the nodes. Second, once we decided to use only a single line framing element to aid with manipulation speed, extracting the end points of that line became possible. The script for the member coordinates is below.

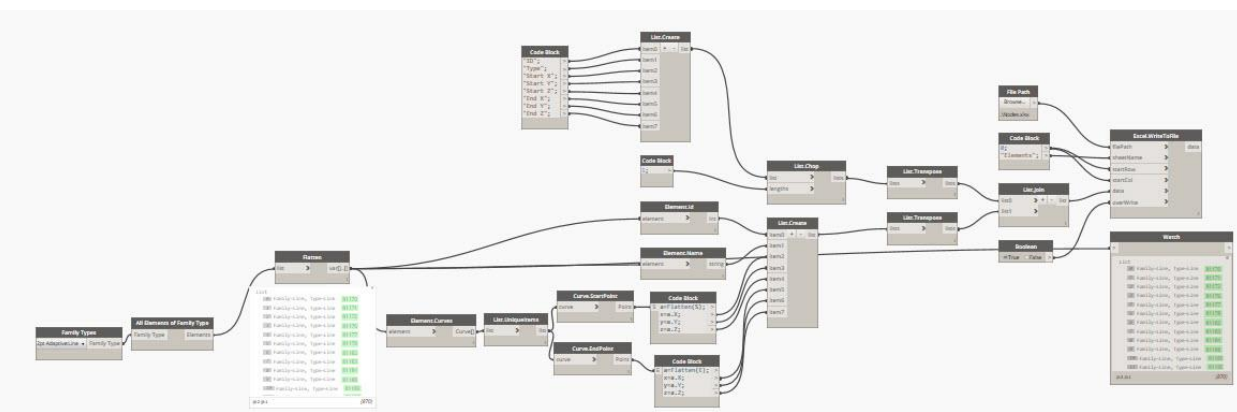

Figure 16: Dynamo to extract member end-point coordinates

Conclusions

Successes

Generally, both projects were successful. For the number of iterations, no other approach would have been easier with the bridge, due mostly to the complicated geometry. The arena roof was quick and easy, especially after learning from the bridge dynamo. Both were very good visual aids for the client.

Challenges

Bridge – There were too many parameters that the client wanted to flex, and very few known values. Building in that much flexibility made it hard to manipulate the family for anyone who didn't create it. We struggled in Dynamo with the curves, but through various iterations created a menu of solutions that could be applied to other projects. Finally, client understanding of the geometry of the castings was a challenge, so it was hard to explain why certain configurations weren't possible given geometric constraints.

Arena Roof – Partly due to the lessons learned on the bridge project, the roof was less challenging. The primary difficulty was the time it took to manipulate the geometry of the roof. Simplifying the geometry to be more representative, instead of actual, was helpful for processing speed.

Other Projects

We have used the Revit-Dynamo-Excel-Analytical process a few times now, on various structures, for various reasons, and with a few different analytical software. What we have found if we are using actual structural framing members, we can use the analytical lines to extract the coordinates of the end points. Here is a screenshot of one other of these projects that is interesting; this project used Rhino as well.