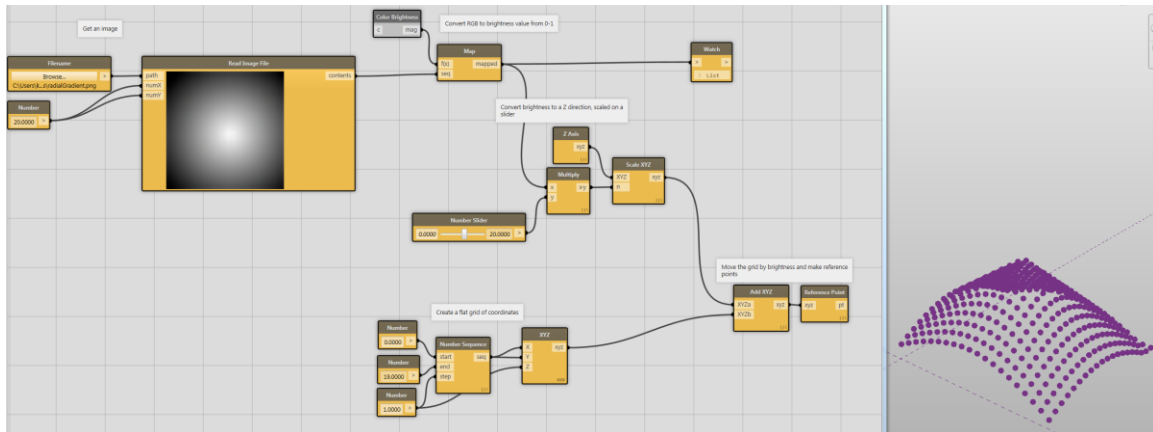# Programming for Non-Programmers – Dynamo Visual Programming

Class Description

This lab will demonstrate how to use Dynamo Visual Programming Tool for Autodesk® Revit® software and Autodesk Vasari. The lab will provide users with resources and step-by-step examples for automating geometry creation, adjusting family parameters using external data, and sharing information with different design platforms.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

Dynamo Visual Programming

Computational Design refers to the ability to link creative problem solving with powerful and novel computational algorithms to automate, simulate, script, parameterize, and generate design solutions. Computational Design has had a profound impact on Architectural practice in recent years. Design practices, large and small, have begun to invest in new computational capabilities that allow them to customize their process and pursue new, innovative design agendas. Computation might be leveraged for a variety of tasks such as automating a redundant production process or to construct an expressive form-generator. Regardless of the end-use, what is clear is that designers need frameworks that let them construct their own tools.

"Visual Programming Language" is a concept that provides designers with the means for constructing programmatic relationships using a graphical user interfaces. Rather than writing 'code' from scratch, the user is able to assemble custom relationships by connecting pre-packaged nodes together to make a custom algorithm. This means that a designer can leverage computational concepts, without the need to write code.

Dynamo is an open source Add-in for Autodesk Vasari and Revit. Dynamo allows designers to design custom computational design and automation processes through a node-based Visual Programming interface. Users are given capabilities for sophisticated data manipulation, relational structures, and geometric control that is not possible using a conventional modelling interface. In addition, Dynamo gives the designer the added advantage of being able to leverage computational design workflows within the context of a BIM environment. The designer is able to construct custom systems to control Vasari Families and Parameters

Dynamo exposes a fundamentally new way of working with geometric information within Autodesk Vasari and Revit. Users can create control frameworks for creating, positioning, and visualizing geometry. The Visual Programming framework lets the user create unique systems and relationships and expand how BIM can be used to drive design ideation.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

Getting Around in Dynamo

The Basics

Dynamo is a plug-in for Revit and Vasari. It works in Revit 2013, 2014 and Vasari Beta 2 and 3 and the installation of one of these applications is a pre-requisite.

Dynamo is a free, open source software that can be downloaded from http://dynamoBIM.org, or from http://autodeskvasari.com

The Dynamo Interface
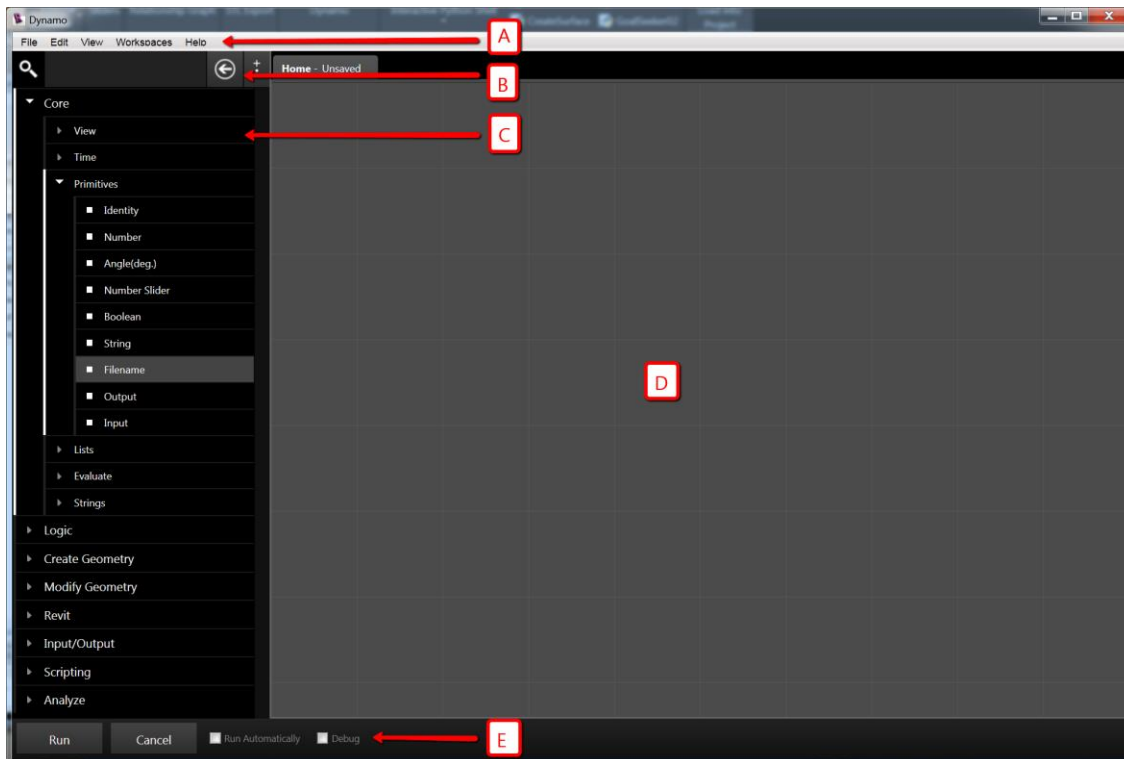
A. Pulldown Menus

Use the File pulldown to Open dynamo files, make new ones, Save-As a new file name, and export an image of your current workspace. Use edit to do copy/paste operations, create new custom nodes, and add

comments. Use the View pulldown to activate background previews, view the console (log), and change wire appearance.

B. Search Bar

Use the Search Bar to find loaded Nodes

C. Node Library

Browser for picking nodes. Click on a node title to add to the Dynamo workspace

D. Workspace

The Workspace is the main environment for creation of Dynamo visual programs. The Home Workspace is the default workspace. When a user creates or edits a custom node, it will appear as a new tab.

E. Execution Bar

The Execution Bar allows the user to run or execute the current workspace The "Run Automatically" checkbox will cause the workspace to run if the user changes the workspace or any of the watched Revit Elements in Revit or Vasari The "Debug" checkbox will invoke a more detailed method of execution

Concepts, Definitions, Terminology

Workspace

The active **Workspace** is the area where you interact with the elements of your visual program. When you start Dynamo you are in a blank **Home Workspace**. You start creating your visual program here by placing **Nodes** and connecting with **Wires**.

You can save the Workspace as a **.dyn**, or Dynamo file for later reuse. Dynamo files can be opened from the File menu or the **Samples** menu.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

Opening a Dynamo file clears the previous workspace and opens the file as the new active Workspace.

A Workspace is executed when you press the Run button.

Nodes

Nodes are the objects you place and connect together with **Wires** to form a visual program.

Nodes can represent Revit Elements like Model Lines or Reference Points.

Nodes can also represent operations like Math Functions.

Nodes have inputs and outputs.

The colors of Nodes change to indicate state.

a. Orange Nodes are well-connected and have all of their inputs successfully connected. They are part of the active Program Flow.

b. Grey Nodes are inactive and need to be connected with Wires to be part of the Program Flow in the active Workspace.

c. Red Nodes are in an Error state. You can see what the error is by hovering your mouse over the node and reading the tooltip.

d. Currently selected Nodes have an aqua highlight. You can drag selected nodes around the workspace or right-click to see their properties.

e. Nodes have input **Ports** on the left side and output Ports on the right side. Directionality of execution and program flow usually goes left to right.

Wires

Wires connect between Nodes to create relationships and establish a program flow. You can think of them literally as electrical wires that carry pulses of information from one object to the next.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

Wires connect the output Port from one Node to the input Port of another node.

You create a Wire using the mouse left-clicking on an output Port and dragging with the mouse button held down, then connect to the input port of another node. Wires appear as dashed while being dragged and solid lines when successfully connected. To disconnect a Wire, left-click on the output Node and pull the Wire away.

Ports

Ports are the light rectangular areas on Nodes, they are the receptors for Wires. Information flows through the Ports from left to right. a. Inputs Ports are on the left side of the Node. b. Outputs Ports are on the right side of the Node.

Ports are expecting to receive certain types of data, for example a Node might work on Point objects or Line objects. Try to connect like outputs to like inputs (XYZ->XYZ for example). Passing a Line object into the input Port of a Node that is expecting a Point will result in an error.

You can tell what a specific Node is expecting to receive or to return from the text label. If the name is cut off, simply hover over the port and a tooltip will appear.

Program Flow

Workspaces have a Program Flow that represents where to start program execution, what to do in the middle and how to know when program execution is complete. When you press "Run" Dynamo executes the visual program according to the established Program Flow.

Directionality of Execution

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

Typically Dynamo visual programs are executed from left to right and you can read the program that way to understand the Program Flow.

There are some exceptions as you get into advanced topics like recursion, but for the purposes of this class simply read from left to right.

Custom Nodes

You can create your own reusable Nodes in Dynamo without programming. These Nodes can be used in the current Workspace or in other Workspaces on your machine. You can also share these Nodes with others. Create new nodes from File>New Custom Node or by selecting existing nodes in the workspace and in Edit>Create Node From Selection (or right click in the canvas). Give the node a name and a category (selecting from the pulldown or by entering a new name. Notice the background color changes when you are editing a user-created node. Custom Nodes appear in the **Node List** like the other nodes. You can double click to edit these nodes at any time or by selecting their name from the View menu pulldown.

Custom Nodes are graphically distinguished from other Nodes by looking like a stack of nodes. When you see a node with a dotted shadow underneath, it is an indication you can double-click on it to edit the contents.

You can share nodes you create with other colleagues by using the Save-as command under Menu. They can open the .dyn file you give them and Dynamo will import the node into the local definitions directory.

Custom Nodes can be nested inside of themselves to create recursive functionality, such as Fibonacci sequences or fractals.

Workflow The active environment that will be executed. The aggregation of the active workspace, python scripts and all dependent user-created nodes.

Examples

Example 01: Create a Point or "Hello World!" In Dynamo

Creating a reference point is the most basic operation you could want to do in Dynamo. It's the venerable "Hello World!" for the application. To get there, you will need to learn some high level concepts and understand some of the basic principles of working in the Revit and Vasari environment

- Learn how to launch Dynamo in the right environment for your needs
- Tour the User Interface, Understand search/browse, and navigate the Dynamo Workspace
- Place nodes and wire them together
- Learn about the difference between geometry and Revit Elements
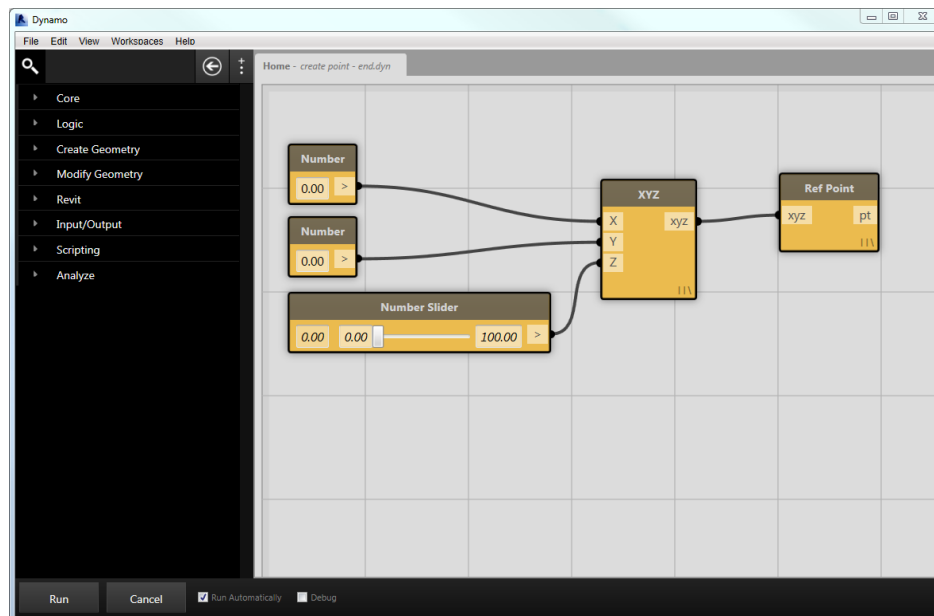
A workspace for creating a dynamically controlled Reference point

1. Launch Vasari.
2. We are going to be placing Reference Points, which can only be done in the Mass, Curtain Panel by Pattern, or Adaptive Component family environments (not the "project" or .rvt

environment). Click [icon] > New > Family> Mass.rft. Or, in the recent documents screen, under Families, Click on "New Conceptual Mass".
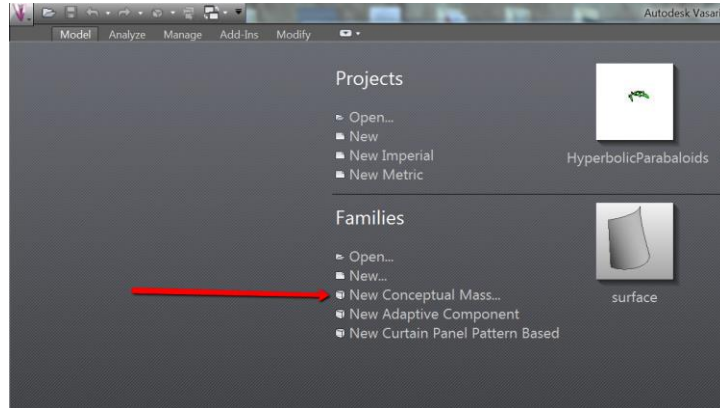
3. Dynamo will operate on the .rfa or .rvt file that is active at the time Dynamo is launched. Now that we have a Mass file open, go to Add-ins tab and launch Dynamo

4. From the Dynamo File Menu, go to **File/Samples/ 1. Create Point / create point.dyn**
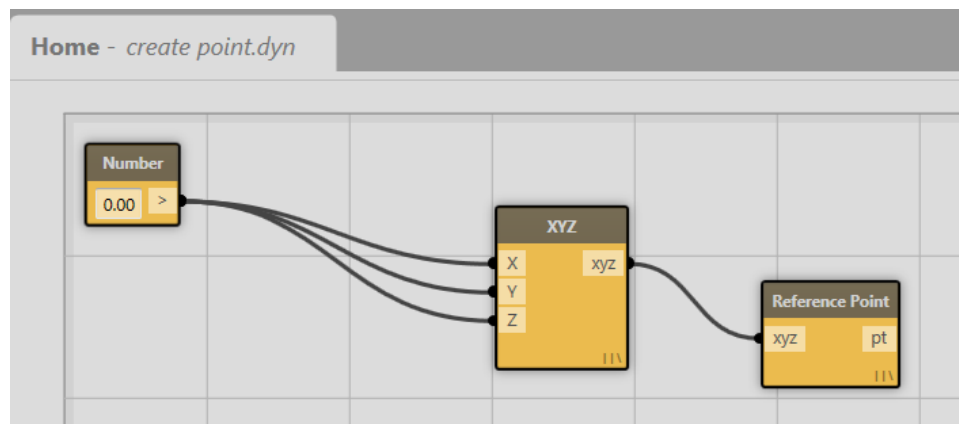
5. Notice a few nodes (**Number**, **XYZ** and **Ref Point**) in the workspace. **Run** to create a single Reference Point at 0,0,0.

   a. Note: there is a difference between an XYZ and a Reference Point. An XYZ is a coordinate point in space, while a Reference Point is a full-fledged Revit Element with many aspects and associated meta-data.

6. Select and move nodes by using the left mouse button.

   a. Type Delete in order to delete a node or right click and click Delete.

   b. The right click menu will also show a number of other functionalities. Click on the help button to see more information on a selected node

   c. Select all the nodes and right click to set their alignment.

   d. Zoom in and out using the mouse wheel and pan using middle click and hold

7. Note that currently Dynamo doesn't support Undo/Redo, so make sure to save your work often.

8. Now we will learn how to re-wire the workspace to add more inputs:

   a. Type "Number" into the search bar to find the **Number** node to add it into the Workspace. This can be done by either typing enter with the number node selected in Search or clicking the node in the node lib

   b. Find the **Number Slider** node and add that as well

   c. On the **XYZ** node, select the end of the wire connecting to the **Y port**. Drag it off into space to disconnect. Do the same for the **Z port**.

   d. Connect the new **Number** node to the **Y port** and the **Number Slider** node to the **Z port**.
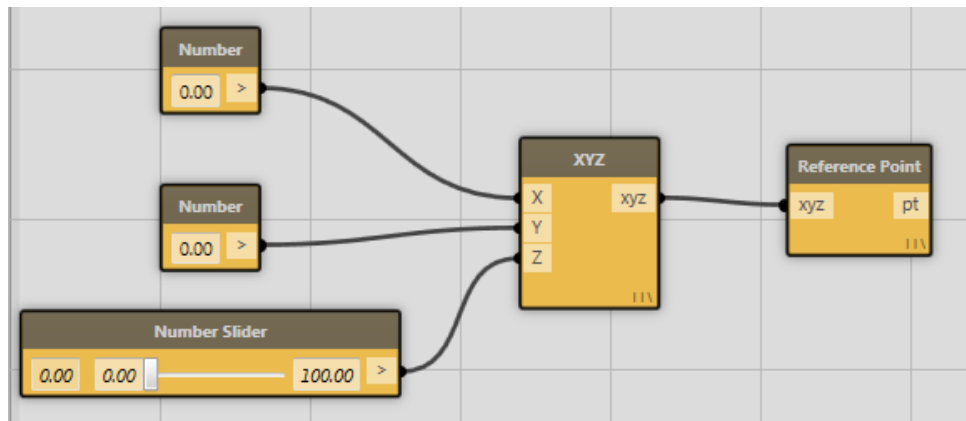
e. At the bottom of the canvas, check "Run Automatically".
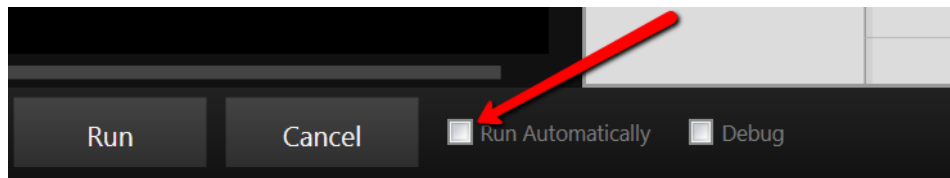
f. Move the slider to see the point move around

g. In the Vasari toolbar, pick Model> Draw>Spline Through Points, and draw a spline with one of the points using the Dynamo created Reference Point. Move the Number Slider and see both Dynamo created stuff and manually created stuff update.

h. Select the Dynamo created Reference Point and, in Dynamo, right click in the canvas and pick "Find Nodes from selected elements"

9.   Create a Custom Node by Selecting the XYZ and Reference Point nodes, then right click out in the canvas, and pick New Node from Selection. Name your custom node something meaningful.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc
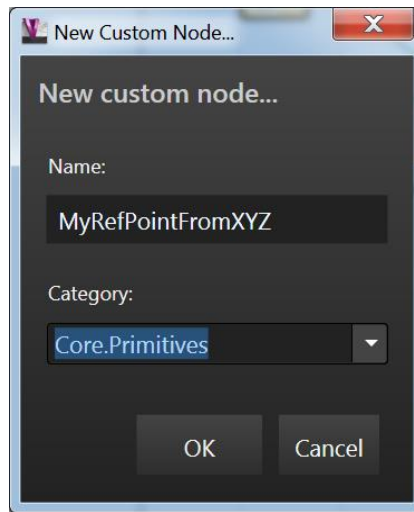
10. In the Edit Menu, pick Create Note to annotate your workflow (or type Cntrl-W). Double click on the Note to edit it (or edit from the right click menu)
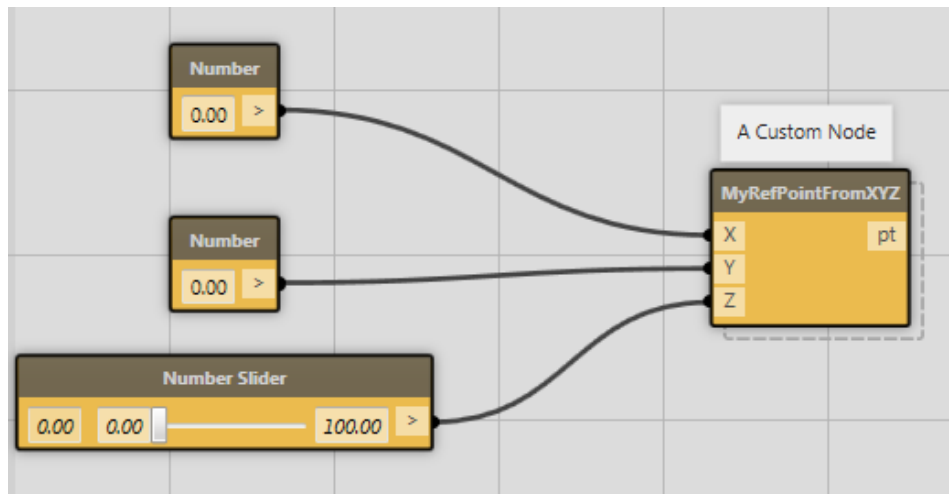
Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

Example 02: Creating and Laying Out Geometry on Lines, Grids, and Lattices

This tutorial aims to introduce the following:

- How to use node **Lacing** to evaluate the members of a list in different ways
- Generate **lines, grids, and lattices** of Reference Points
- **Place family instances** with Dynamo
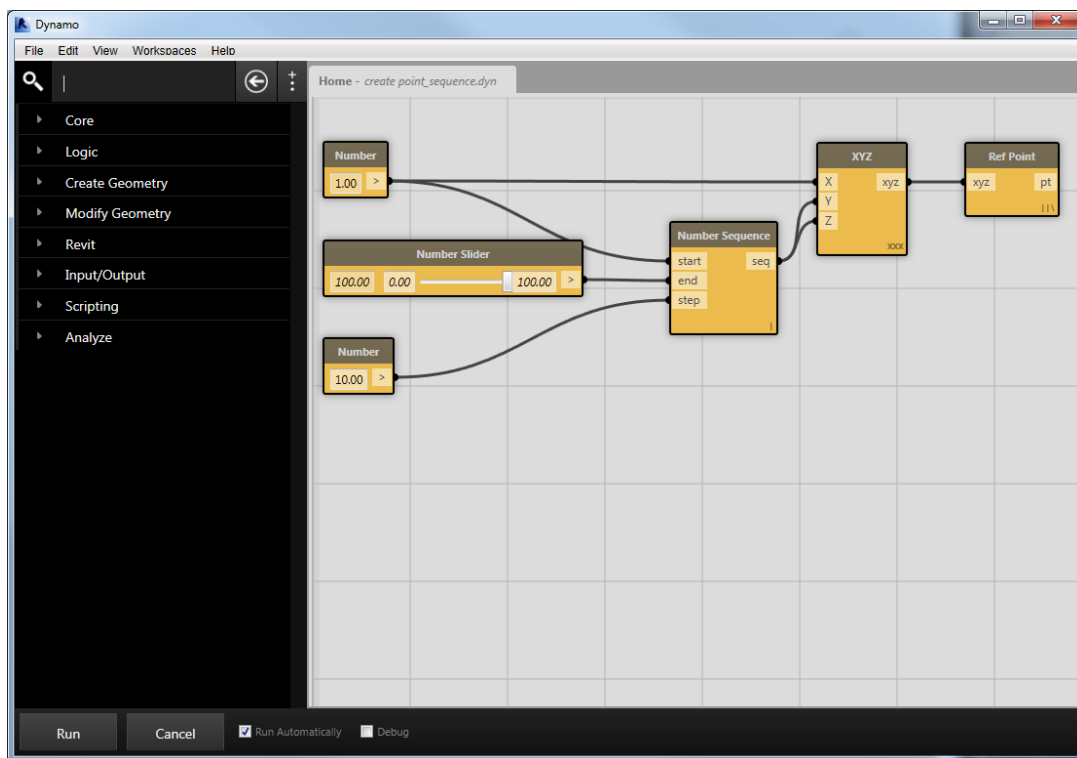
Dynamo after opening the create point_sequence.dyn file

1. From Vasari, Click > (Open).
2. Navigate to **C:\Autodesk\Dynamo\Core\samples**
3. Open **Mass with Loaded Families.rfa** from the **Samples** directory
4. Go to Add-ins tab and launch Dynamo. If you already had it open, close and re-open it to re-associate Dynamo with the newly opened .rfa file.

5. From the Dynamo Help menu, go to **Samples/ 1. Create Point /
   create point_sequence.dyn**
6. Hit the "Run" button to see the following:
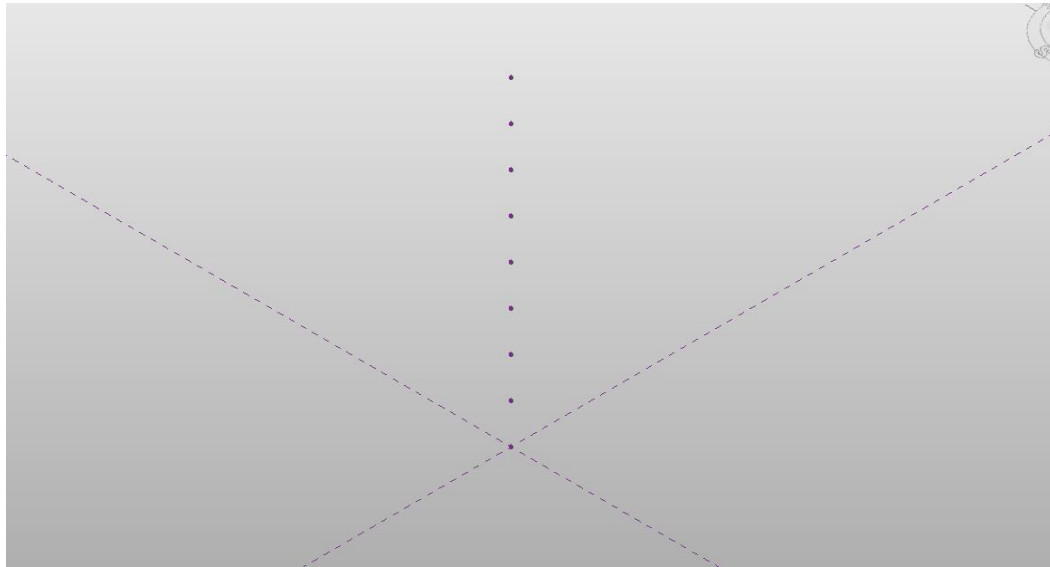
A sequence of points created by running this example

7. Right click on the **Number Sequence** and select **Help...** to see what
   kinds of inputs and outputs the number sequence expects.
8. Notice the little icon in the bottom right corner of the XYZ node. This
   indicates the *Lacing* for this particular node. This concept,
   borrowed from McNeel's Grasshopper, allows you to automatically
   apply the node to the sequence created by the Number Sequence
   node.
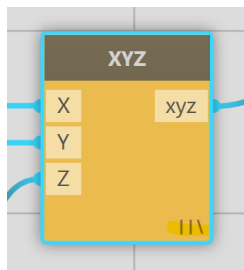
The XYZ node with lacing set to "Longest"

9. Try changing the Lacing strategy to **First** by right clicking on the XYZ node and selecting First. You should see the icon in the bottom right corner of the node change.

10. Hit Run again. You should see a single point located at the origin. This is because the XYZ node is only evaluating the first element in the list created by the Number Sequence node.

11. Change the lacing on the XYZ node to **Cross Product**. If you hit Run again, you should again see a vertical line of regularly spaced points.

12. Click the output port button on the Number Sequence node and connect it to the XYZ node by clicking on the Y input. Your workspace should look like this:
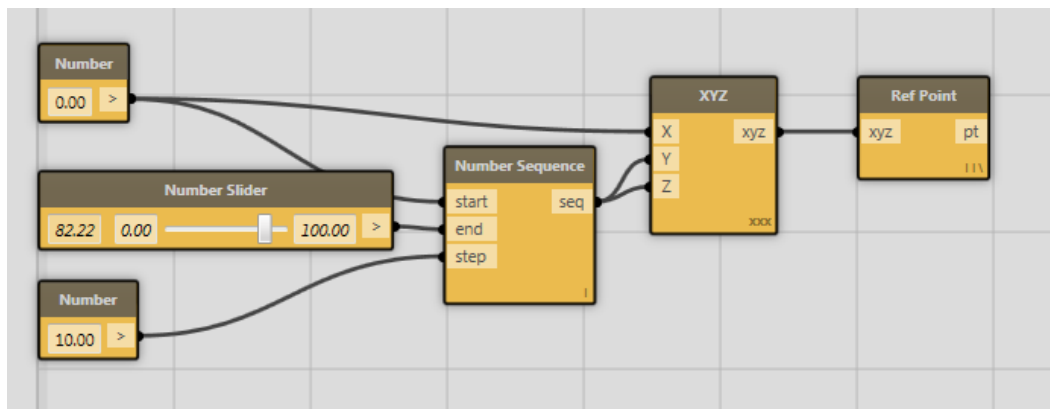
A workspace for creating a grid of XYZ points in the YZ plane

13. By running again, you should see an orthogonal grid in the YZ plane, like this:
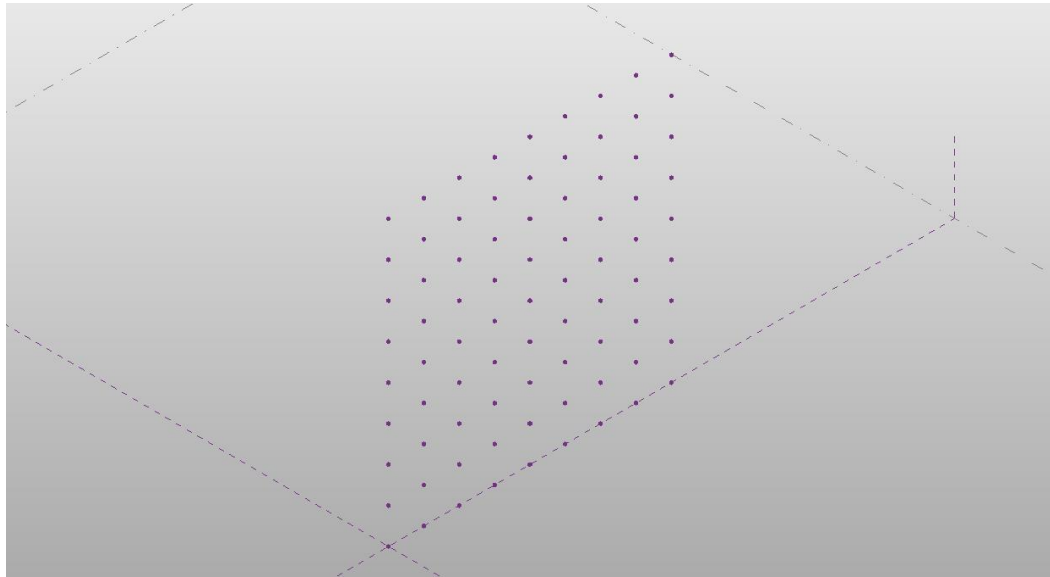
A square grid of XYZ points.

You can experiment by connecting the sequence to the X and Y ports to get a plane in XY plane.

14. Connect all 3 input ports of the XYZ node to the output of the Number Sequence Node and Hit Run. You should get a cubic 3d lattice:
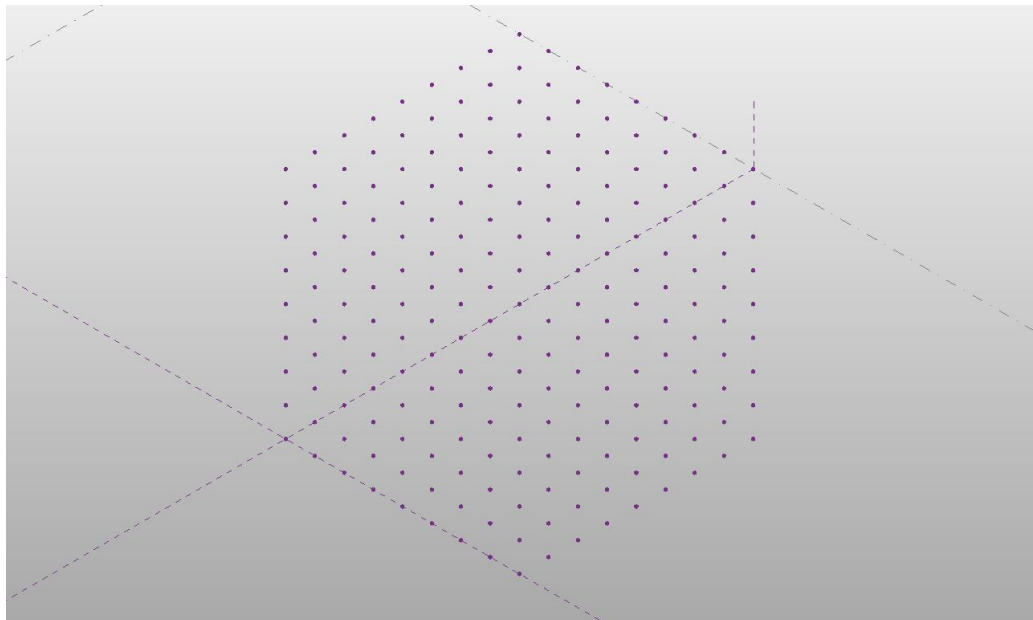
A cubic lattice of XYZ points.

15. Let's scale back a bit and go back by connecting the Number node, set to 0, to the Z port of the XYZ node. We'll **have just a grid**

in the XY plane.

Now we will extend this workspace to do something more useful than just creating points. We'll place Family Instances!

16. Go to the **Search Bar** and type in **"Family",** this filters the available nodes down and allows easier access from the **Node List.** You should now only see nodes related to "Family".

17. Add one **Create Family Instance** node and one **Select Family Type** node into the workspace. Look at the Help menu for both of these nodes by right clicking.

18. Connect the **Select Family Type** output to the **Create Family Instance** "typ" input port.

19. Now select the **Cone** Family Type from the pulldown on Select Family Type and connect the XYZ output from the **XYZ Node** to the XYZ input of the **Create Family Instance** node. Your workspace should look like this:
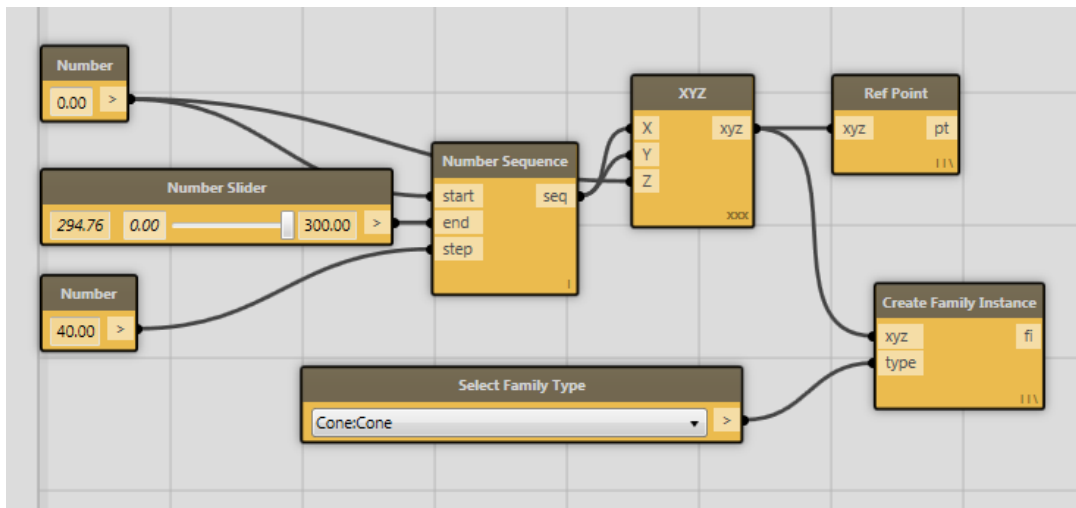
The workspace after adding the Select Family Type and Create Family Instance nodes

20. Hit **Run** and you should see something like this:
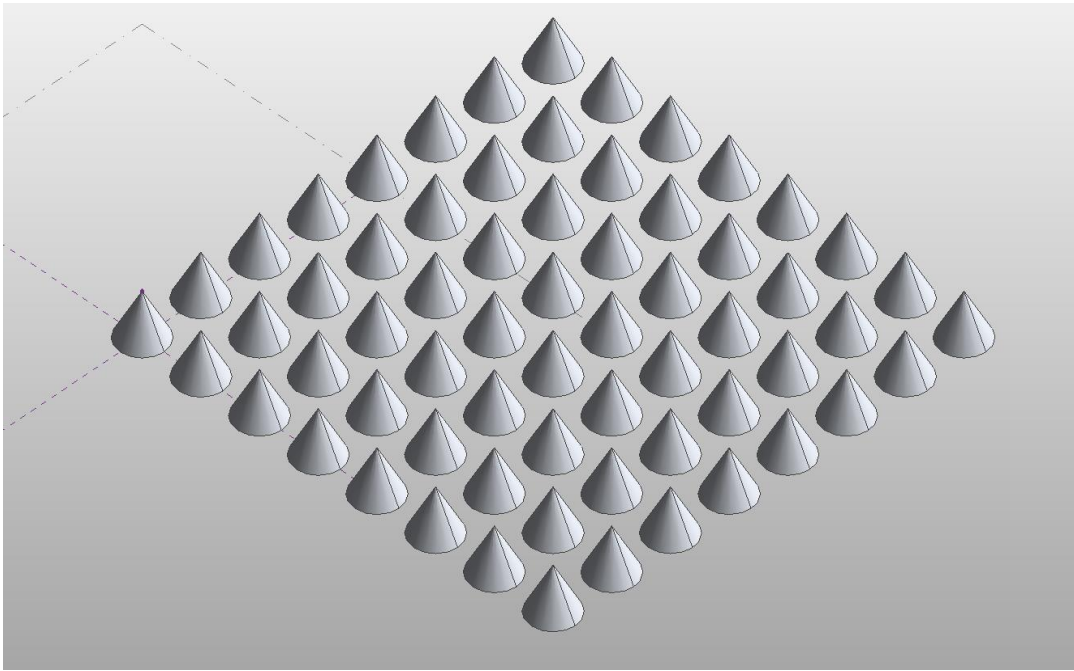
A grid of cone families laid out on the XY plane.

21. Experiment with different values for the number sliders or different family types. By turning on **Run Automatically** you can do this interactively.

Example 03: Synchronizing Family Instance Parameters

This tutorial aims to introduce how the user can:

- Use Dynamo to select **Family Instances**
- Synchronize **Family Instance** parameters across multiple instances using **Run Automatically**

1. From Vasari, Click ![icon] > ![folder] (Open).
2. Navigate to:
    C:\Autodesk\Dynamo\Core\samples\08 Get Set Family Params
3. Open:
    inst param mass families.rvt

You should see the following:

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc
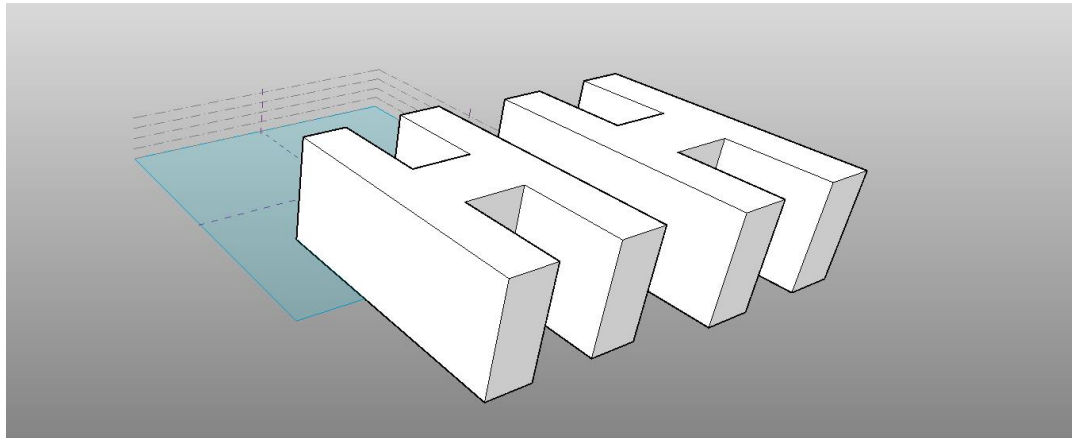
The contents of inst param mass families.rvt

4. Go to Add-ins tab and launch Dynamo. If you already had it open, close and re-open it to re-associate Dynamo with the current document.
11. From the Dynamo Help menu, go to:

   Samples/ 08 Get Set Family Params / inst param 2 masses driving each other

5. You should see the following in the workspace:
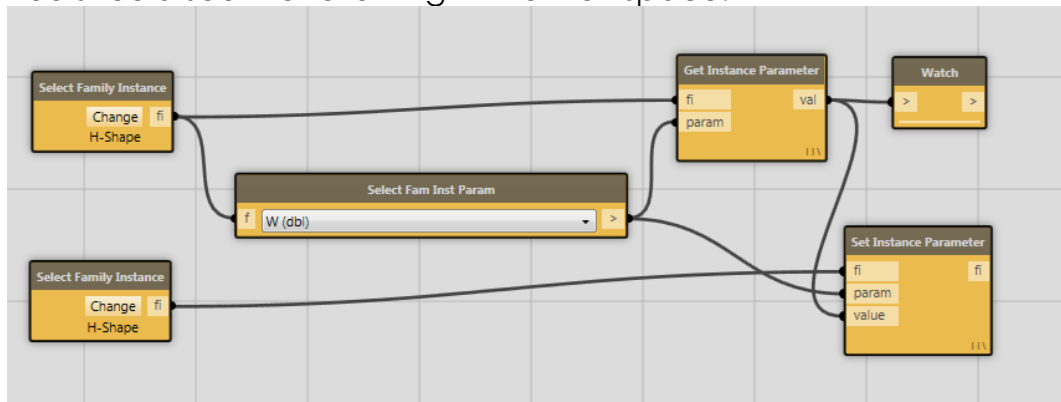
Dynamo after opening the create point_sequence.dyn file

6. Note that the two **Select Family Instance** nodes have automatically associated themselves with the two family instances (a pair of H-shaped building masses)
7. Set the **Select Family Instance Parameter** node to H (dbl).
8. Click **Run**. You should see the two instances match in height.
9. In Vasari, edit the height of the H instance closer to the origin.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc
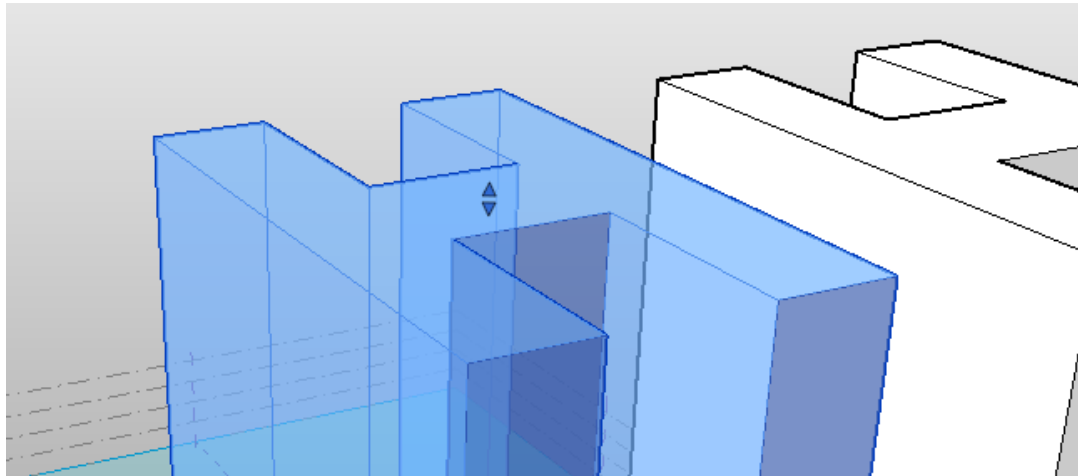
The manipulator to change the height of the family instance

10. Click **Run,** again. You should see the family instances update.
11. Set Dynamo to **Run Automatically.**
12. Edit the same Family Instance's height again. You should see the other instance update.

Now, you know how to synchronize Family Instance parameters!

Example 04: Doing Basic Math with the Formula Node

Dynamo has many useful tools for doing math. This tutorial aims to demonstrate teach you how to:

- Use the Formula node to simplify writing mathematical expressions
- Generate points that follow circular or elliptical paths from a mathematical formula.

1. From Vasari, Click > (New) > Family.
2. Use Mass.rft in the Conceptual Mass folder.
3. Go to Add-ins tab and launch Dynamo. If you already had it open, close and re-open it to re-associate Dynamo with the newly opened conceptual mass
4. From the Dynamo File Menu, go to **Samples/ 19 Formulas / Scalable Circle.** You should see the following in your workspace:

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc
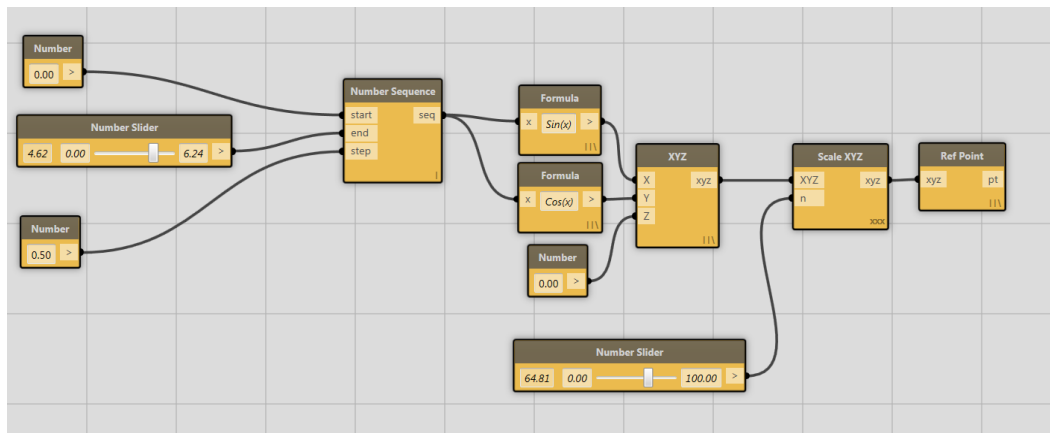
The nodes in Scalable Circle.dyn

5. To get an idea of what this workspace does, hit **Run.** You should see an arc of XYZ points centered at the origin:
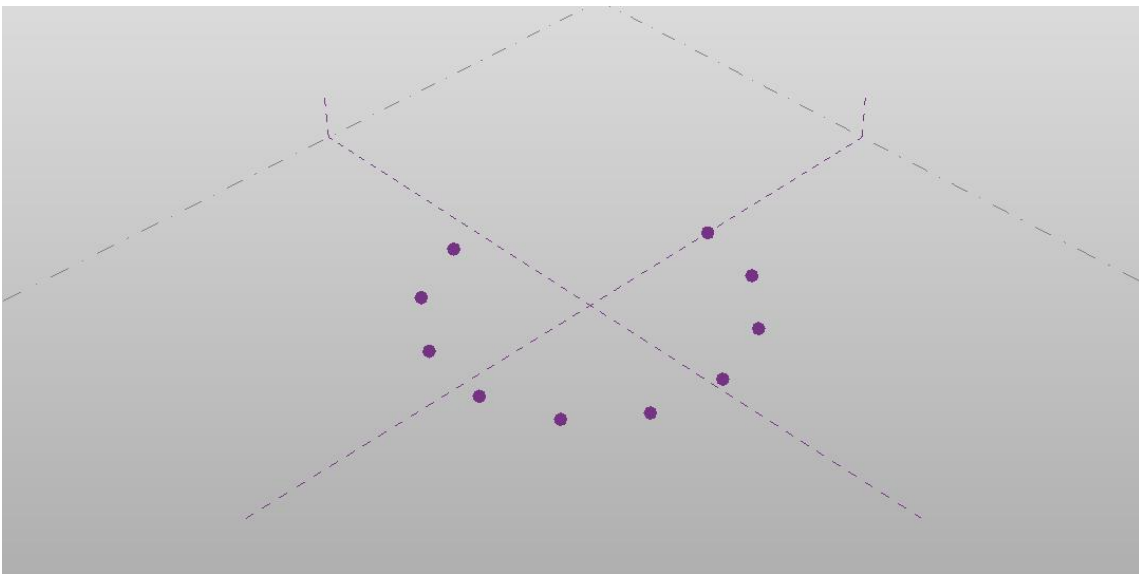

The result of running Scalable Circle.dyn

We're going to edit this workspace to demonstrate a few concepts of the **Formula node**.

6. First, select and delete the Scale XYZ node. You should see the XYZ and Ref Point node turn grey.
7. Let's edit the formula in the first Formula node, which currently contains "Sin(x)". Change the formula so it says "a * Sin(x)" and hit

enter. You should see a new port on the Formula node called (appropriately) "a". You could've called this variable anything. It should look like this:
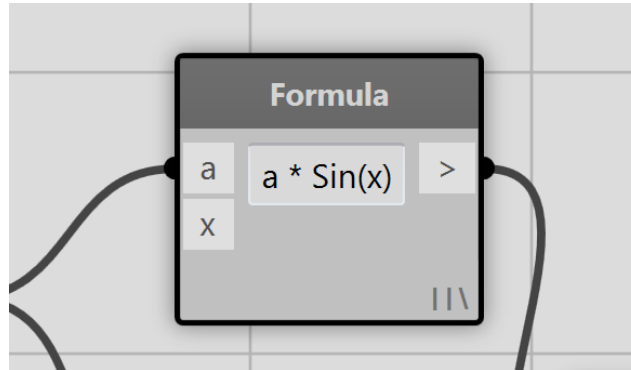
The Formula node after editing the expression

8.   Connect the output of the Number Slider node to the "a" input of your freshly edited Formula node and reconnect the output of the Number Sequence node to the x input of the same node. Connect the XYZ output again to the Reference Point node. Your workspace should look like this:
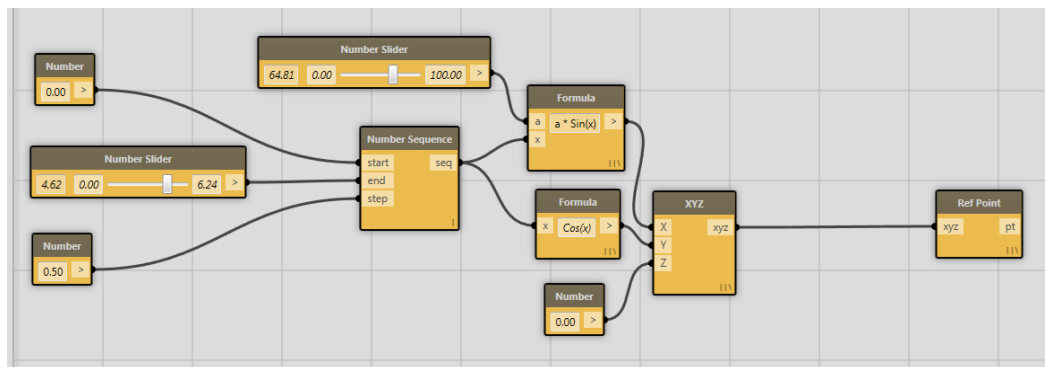
The new configuration of the workspace after editing the first Formula node.

9.   Hit **Run**. You should see your circle of points disappear and a very narrow ellipse will replace it. Play with the sliders to change it.

10. Let's do the same thing to the other formula node which currently has "Cos(x)" in it. That is, let's add a multiplier to the second

Formula node and connect a number slider to that. Now, your
workspace should look like this:
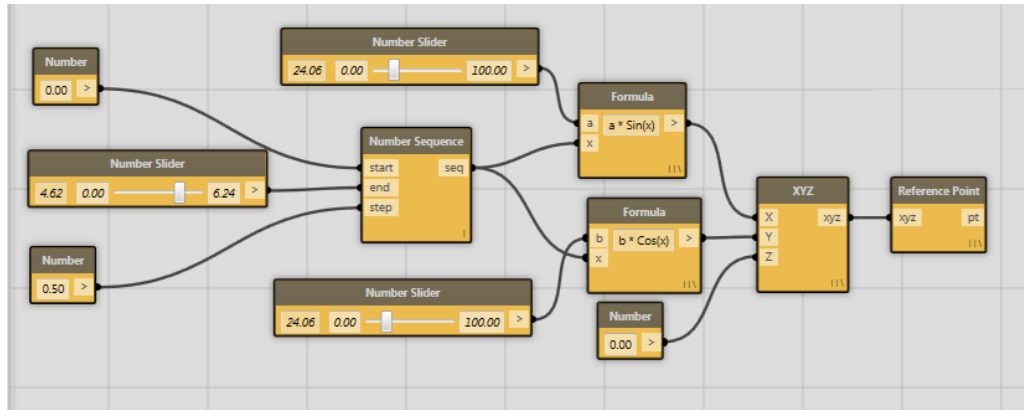
The new configuration of the workspace after editing both Formula nodes.

11. Now, turn on Run Automatically and experiment!
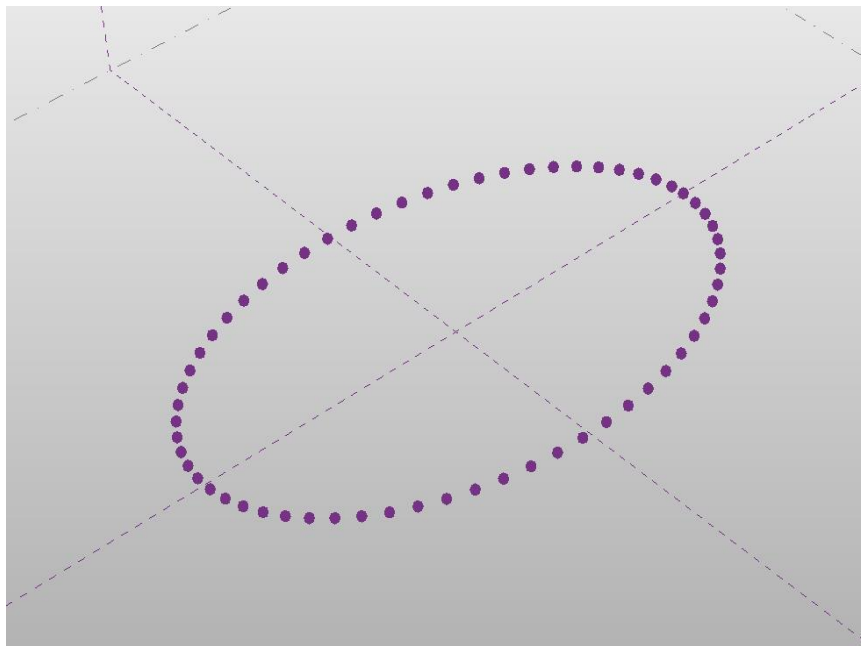

An ellipse created using two Formula nodes.

Note: The formula node is based on the open source **NCalc** library. It has
an amazing amount of features – it includes many mathematical
operators, functions, and you can even pass your own functions into the

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

node! For a full description of operators, functions, and more check out**:**
http://ncalc.codeplex.com/

Example 05: Attractors
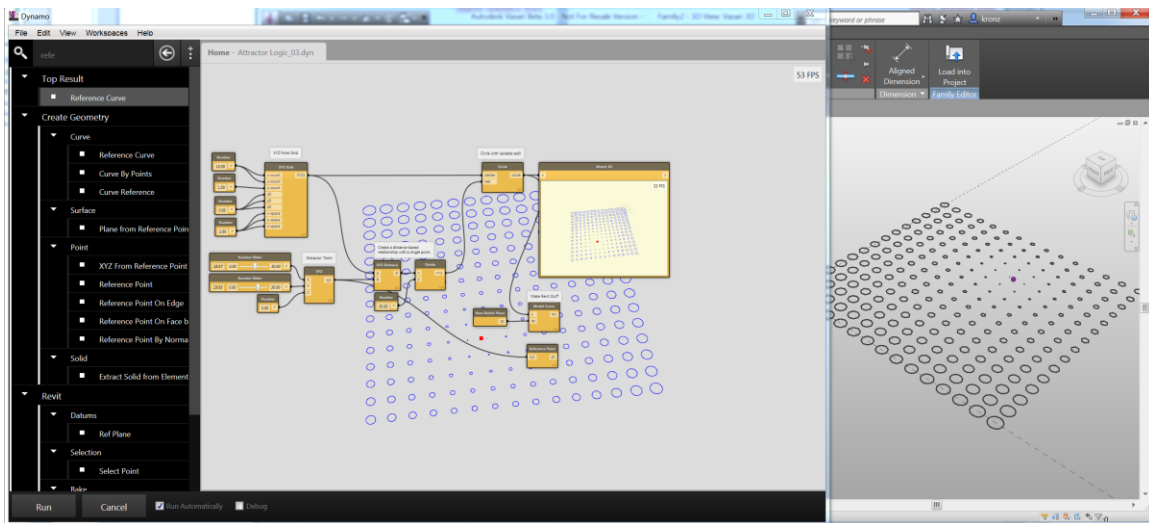
Dynamo after opening and running the CSNM_SimpleAttractor.dyn file

This exercise will guide the user through a simple computational design problem using Dynamo. The user will construct an "attractor" logic wherein the distance between points will be used to drive a geometric variation.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

- Learn how to compute relationships between XYZ geometry.
- Learn how to construct a basic mathematical relationship with Dynamo Nodes.
- Learn how to create variations in geometry with distance-driven values.
- Convert abstract Geometry to Model Lines
- Visualize geometry in a number of ways

This example creates Revit elements that can be made in any environment, so you can work in any Revit or Vasari context.

1. Navigate to
C:\Autodesk\Dynamo\Core\samples\10 Attractor
2. Set to Run Automatically, and hit Run
3. Zoom to Fit Watch 3d
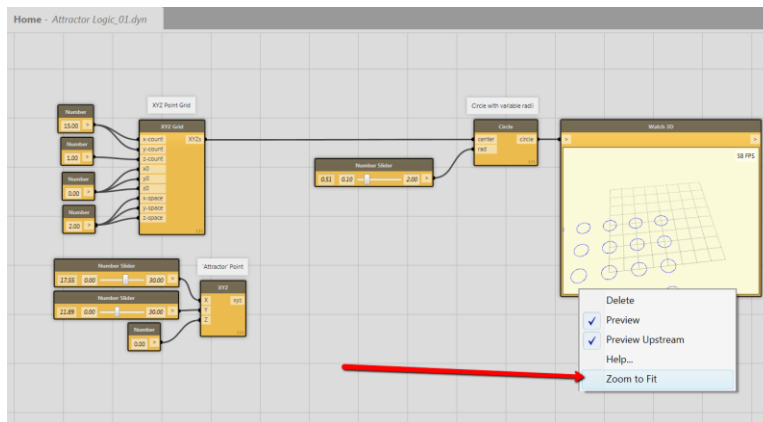
4. Adjust the slider wired into the Circle generating component. Notice that although you are getting different sized circles, you are not creating Revit geometry.
5. Adjust the Sliders for the 'Attractor" Point. Notice that there is no geometric representation although the Watch node registers a change
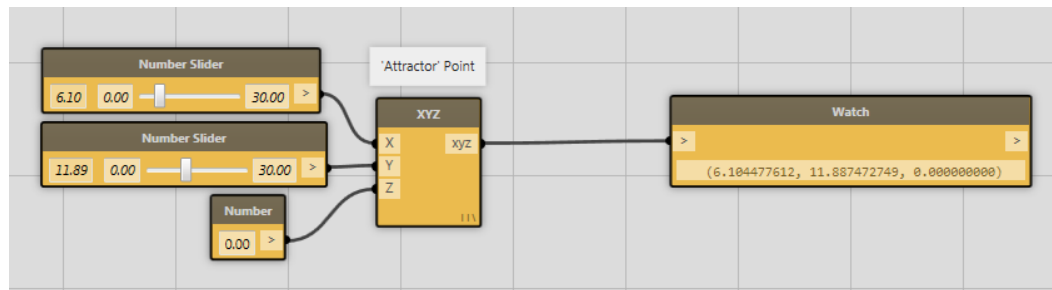
6. In the View menu, click on Preview Background. All geometric entities are rendered in the canvas.
    a. Toggle Navigation by clicking cntrl-G. Zoom out with the scroll wheel, orbit with right click, and pan with middle mouse
    b. Hit cntrl-G to exit navigation
    c. Change the Attractor Point sliders, see XYZ and Circles in the same view.
7. Now we need to connect the proximity of the Attractor Point to the radius of the circles. In the lower portion of the screen, find a cluster of nodes: XYZ Distance, Divide, and a Number.
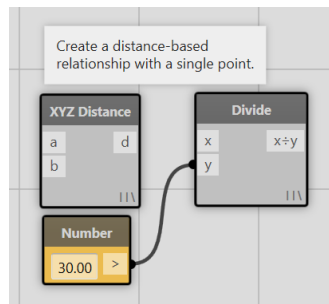
8. Measure the distance between each grid point and the attractor point. Pass the position of the Attractor point and the position of each XYZ Point grid to the XYZ Distance node and then connect the resulting distance to the circle's radius.

Programming for Non-Programmers
Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc
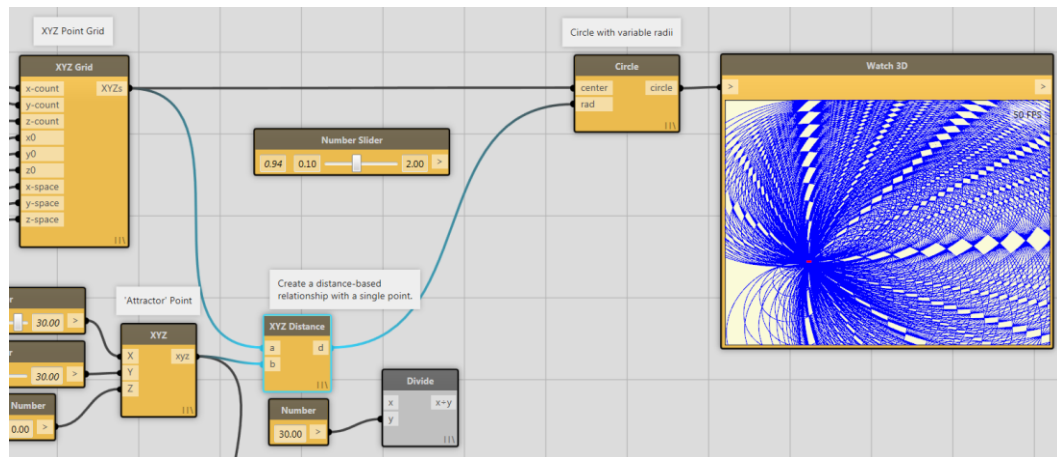
9. The resulting Geometry is a bit of a mess, as each circle has the same radius as the distance of the attractor point. Let's moderate this by passing the XYZ Distance Node through a Divivision operator first
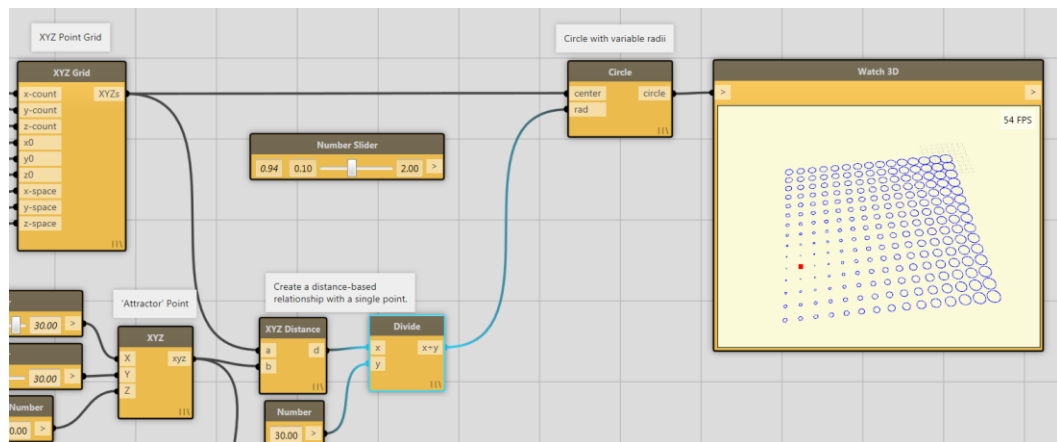

10. So far we have just made abstract geometry. We can dump out this data into Revit Model Lines by attaching the Circle or Watch 3d outputs to Reference Curve and New Sketch Plane nodes
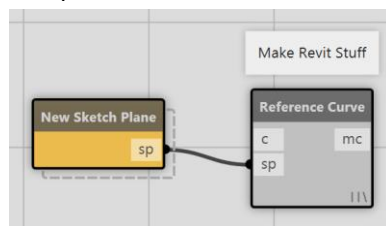
11. The resulting workflow results in this Revit Geometry which is still associated with the graph.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

a. Experiment with the sliders to get a feel for changes in model update with and without Revit geometry.

b. If you save the Dynamo file and the Vasari file, the Dynamo file will "remember" the geometry it has previously created and manipulate its parameters later, not create new stuff.
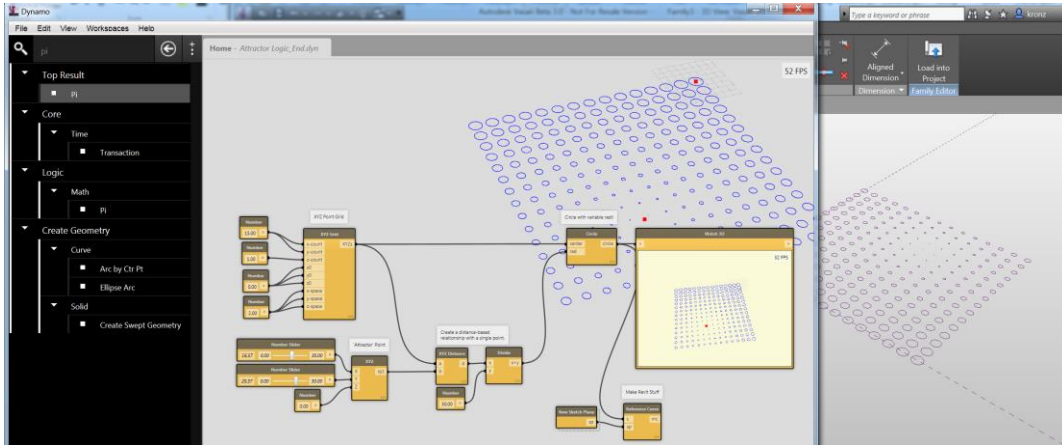
Example 06: Using Revit Geometry in Dynamo and Adaptive Components

Dynamo can use geometry extracted from the Revit environment and perform lots of different operations on that geometry. That geometry can be manipulated in various ways using list operations. Finally, you can take that information to place adaptive components. In this example, you'll learn how to:

- Select and use Revit geometry in Dynamo
- Place Adaptive Components.

1. From Vasari, Click [icon] > [icon] (Open).
2. Navigate to:
    C:\Autodesk\Dynamo\Core\samples\18 Adaptive Components
3. Open **Adaptive Component Placement.rfa**
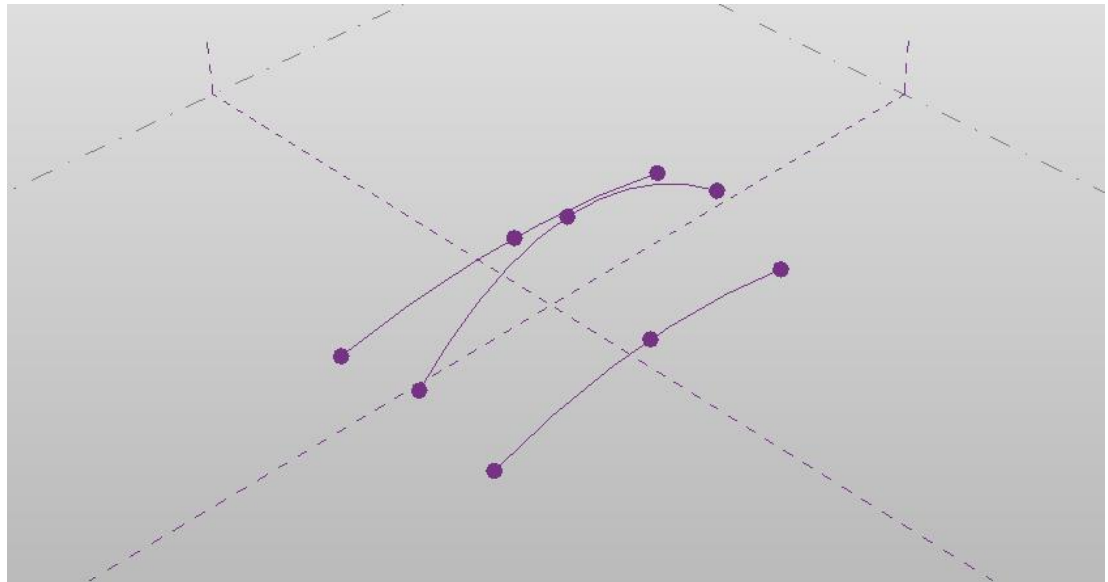
The contents of Adaptive Component Placement.rfa.

4. Go to Add-ins tab and launch Dynamo. If you already had it open, close and re-open it to re-associate Dynamo with the newly opened .rfa file.

5. From the Dynamo **Help menu**, go to:

Samples/ 18 Adaptive Components   / Adaptive Component Placement
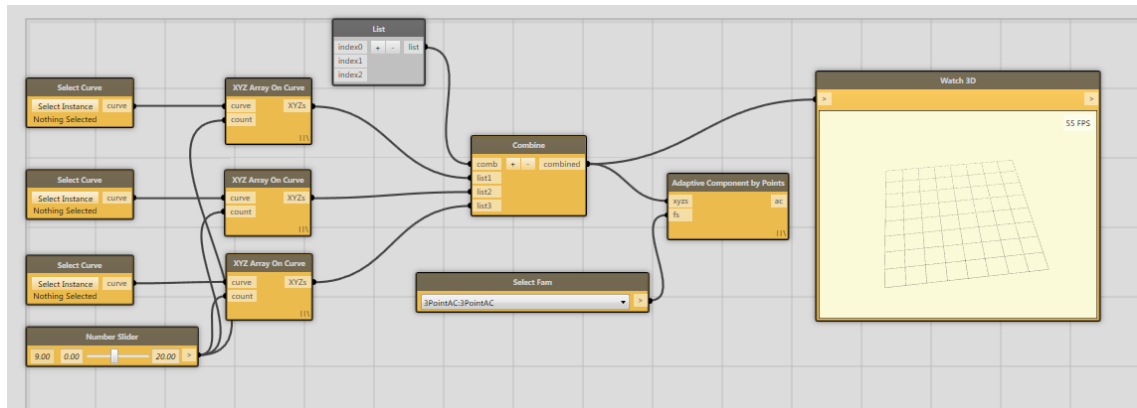
The nodes in Adaptive Component Placement.dyn

6. You'll notice that each of the Select Curve nodes has a button associated with it. That button can be used to select a curve inside of Revit. Click the first Select Curve nodes "Select Instance" button and select the first curve (Leftmost on the preceding image). You should see the node change to this:
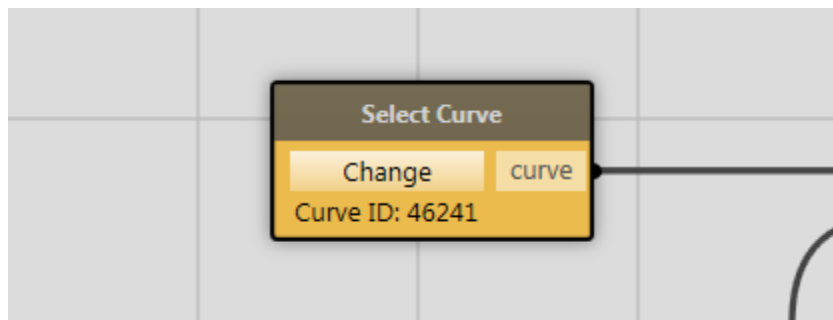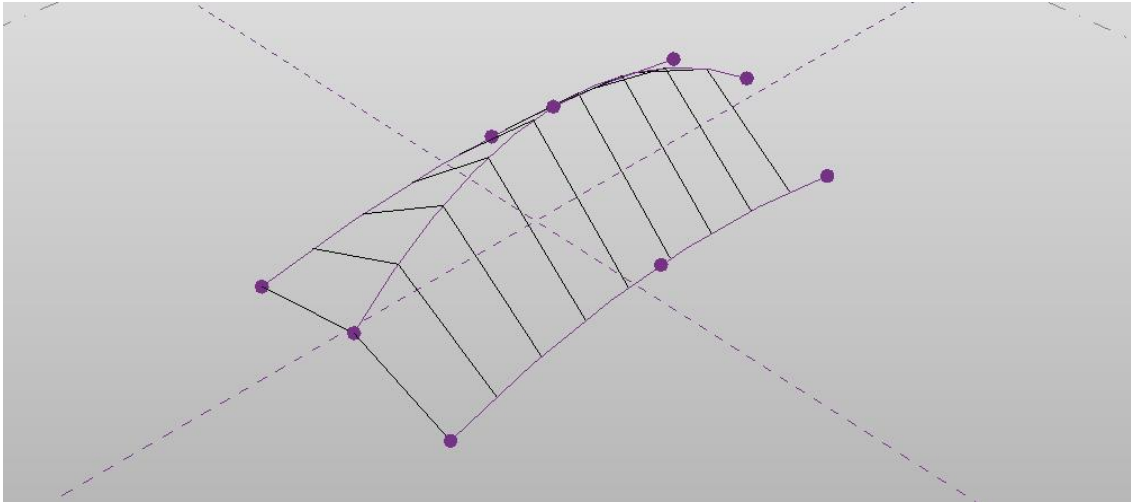

A curve has been selected

7. Do the same on the other two nodes, continuing with first with the middle curve. Now, we have selected three curves for use in Dynamo.

8. Now, hit Run. You should see the following:

A series of three point adaptive components placed along the three curves.

9. Change the family type on the Select Family Type node to "3PointAC_wireTruss" to place a series of trusses along these three curves.

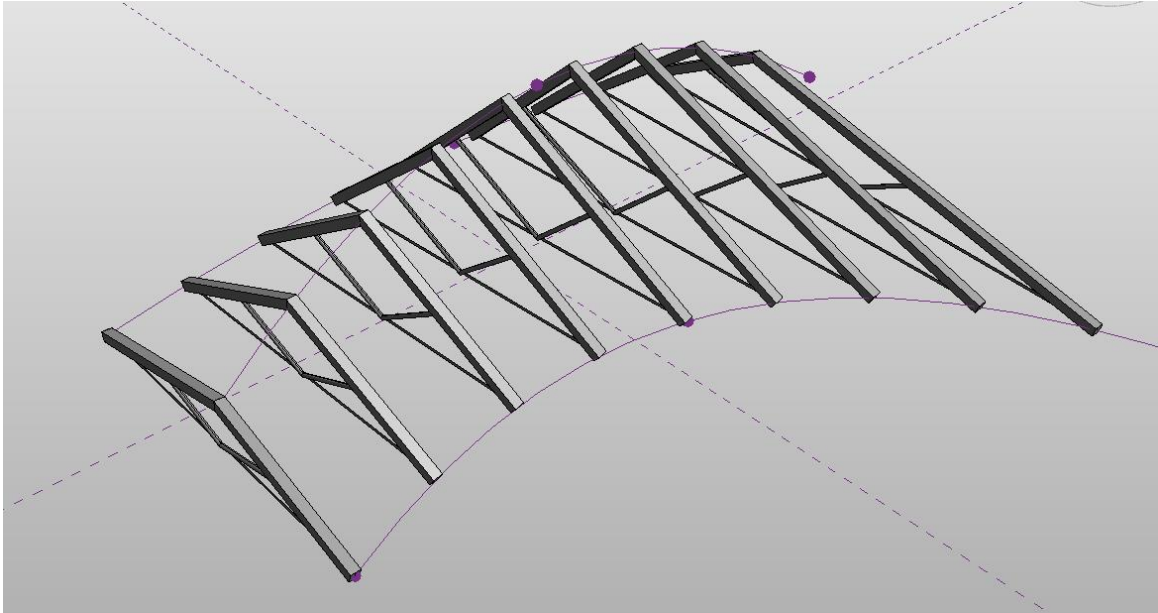10. Select one of the reference point controlling the three curves and edit it.
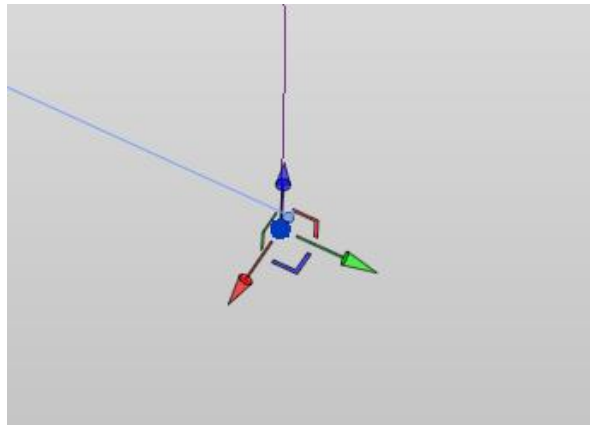
Edit the reference point

11. After editing the curves, re-run the calculation in Dynamo. Note that the adaptive components have changed shape to match new location of the curve:

The adaptive components arrayed onto the three curves

We might need to pause a moment to understand what's been done here. This workspace begins by taking three curves from the **Select Instance** and the **XYZ Array On Curve** node to get a regular list of XYZ's along the curve. Then, we use the **List** node to combine the XYZ's into a nested list. This list contains three lists of XYZ's, a 2D list, each one sampled from the curves we selected at the beginning. Then, we use the **Transpose Lists** node. This node returns a list of length-three lists of XYZ's. This is the result of replacing the rows with the columns in our original 2D list.

This workspace gives us the three XYZ's that are necessary to place the Adaptive Component's.

Example 07: Using Python to Build a Sine Wave in Dynamo

The aim of this tutorial is to show you how to:
- **Edit and run a python script** in Dynamo
- Show how to **use changes in the document to update your Python script.**

1. From Vasari, Click ![V] > 📂 (New) > Family.
2. Use Mass.rft in the Conceptual Mass folder.
3. Go to Add-ins tab and launch Dynamo. If you already had it open, close and re-open it to re-associate Dynamo with the newly conceptual mass
4. From the Dynamo File Menu, go to **Samples/ 06 Python Node / create sine wave from selected points.**   You should see the following in your workspace:
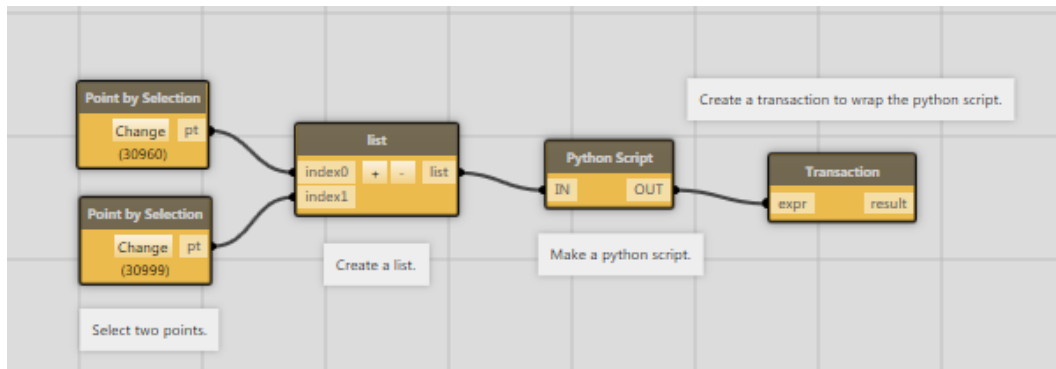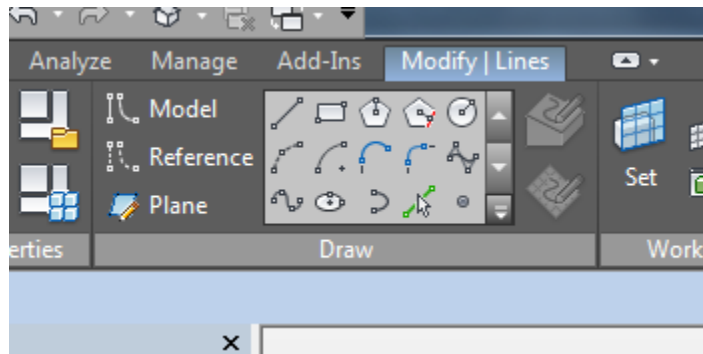

A python node in a workspace.

5. Now place two points using the Vasari Sketching Gallery The reference point button is in the bottom right corner.
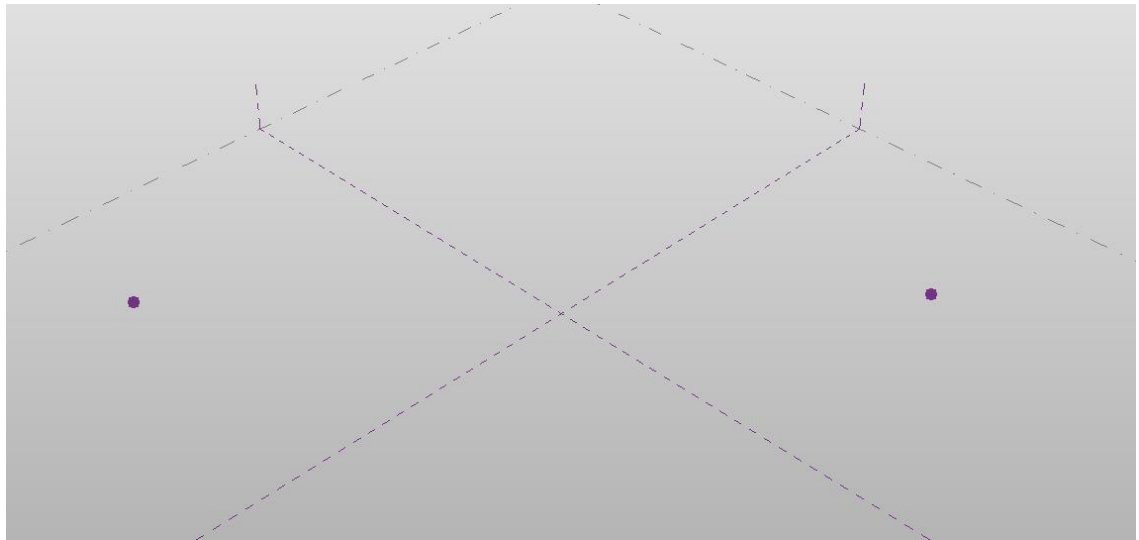
Programming for Non-Programmers
Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

The Vasari Draw menu.

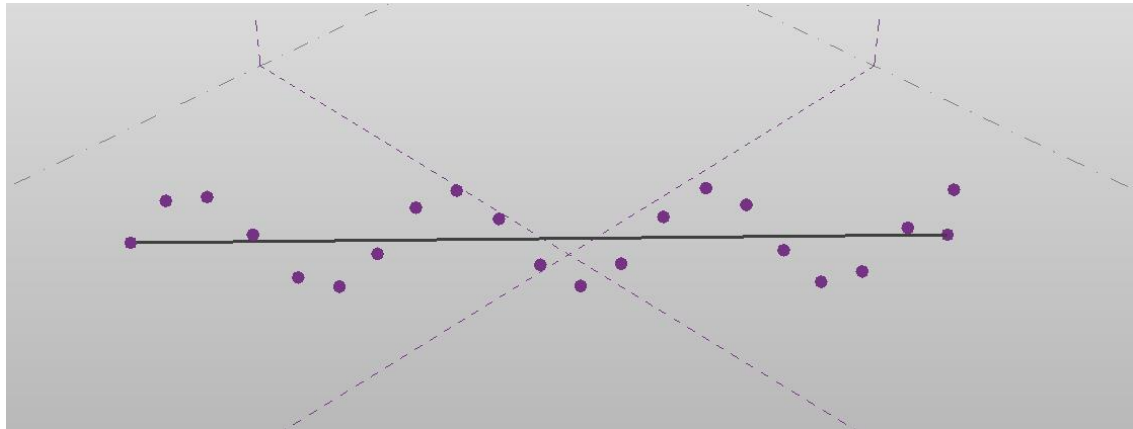6. You should have two Reference points in the Vasari view:

Two reference points

7. Select each point using the **Select Point** nodes in Dynamo.
8. Press **Run**. You should see a line and a sine wave plotted between the two entered points. These elements were created by a Python script!

Programming for Non-Programmers

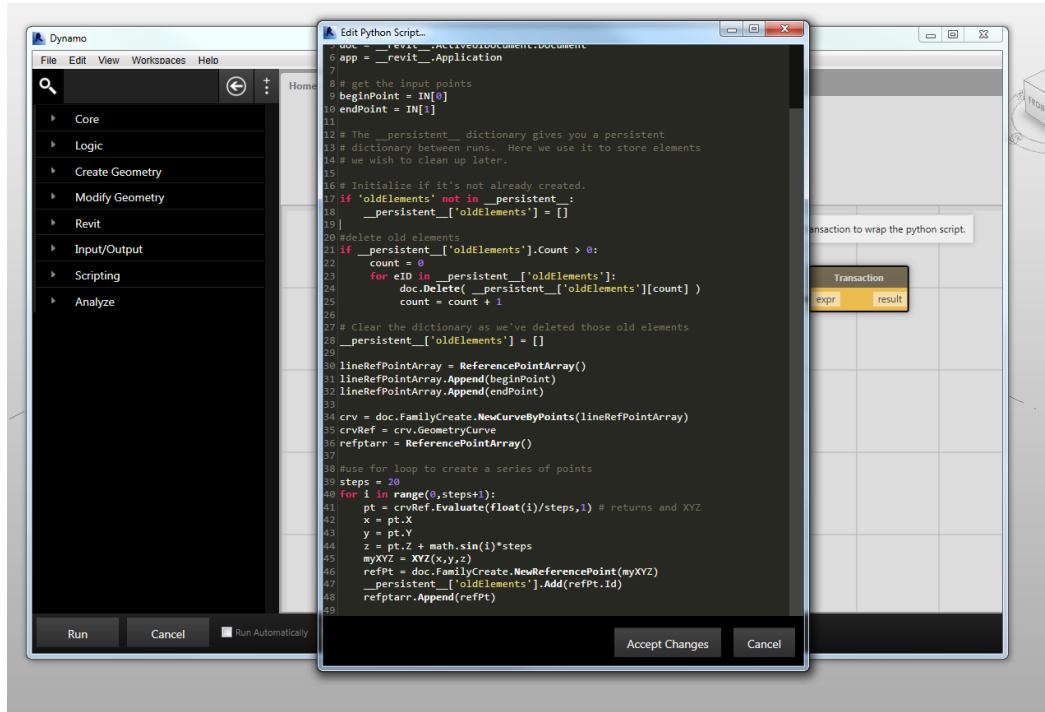Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

The output of the Python script

9. Set Dynamo to **Run Automatically**
10. Experiment with moving the points or the straight line, notice how the sine wave is updated to follow the new position? This is because **Dynamo and the Python node can watch specific elements and then re-run the workspace to keep things in sync based on your changes.**
11. Right click on the Python node and click Edit… to show the script editor:

The python script editor in Dynamo

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

12. We will not dissect it in detail now, but take a look at how the **beginPoint** and **endPoint** variables, get assigned objects from **IN**. The **IN** and **OUT** ports map to variables in the script.
13. In the script editor, go to line 39 and edit the value of steps.
14. Click **Run** again. You'll notice that the sine wave has changed. Feel free to experiment with this script.

You don't have to use Dynamo to edit your python scripts! If you feel more comfortable in a different text editor, you can use the Python Script From String node. Combined with the File Path node and Read file, you can read your files from a text file and Dynamo will automatically update. Check out the DynamicMobius example under Help > Samples > Dynamic Python Editing for an example of this.
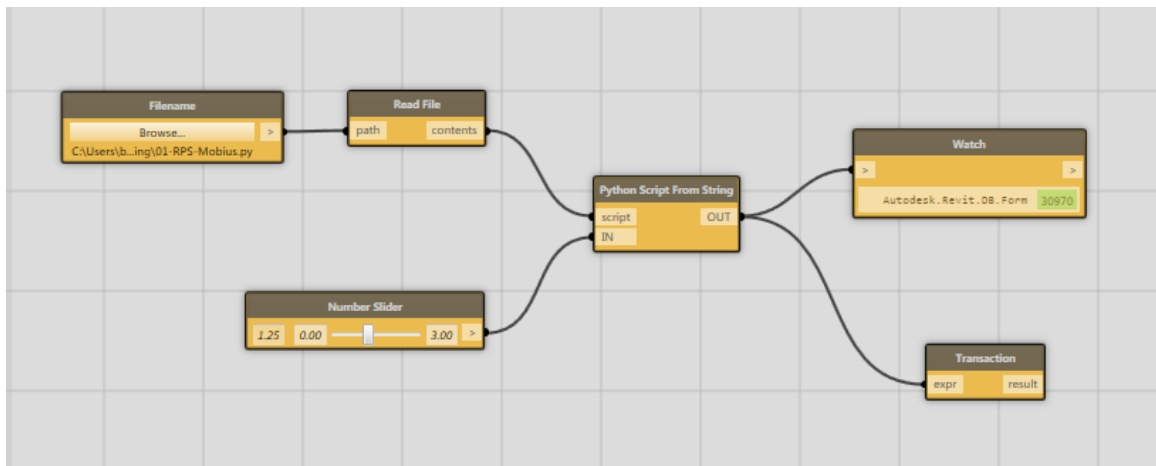
An example of using Python Script From String to edit a python script with a different text editor

Example 08: Using Custom Nodes and Recursion to Create a Fibonacci Sequence
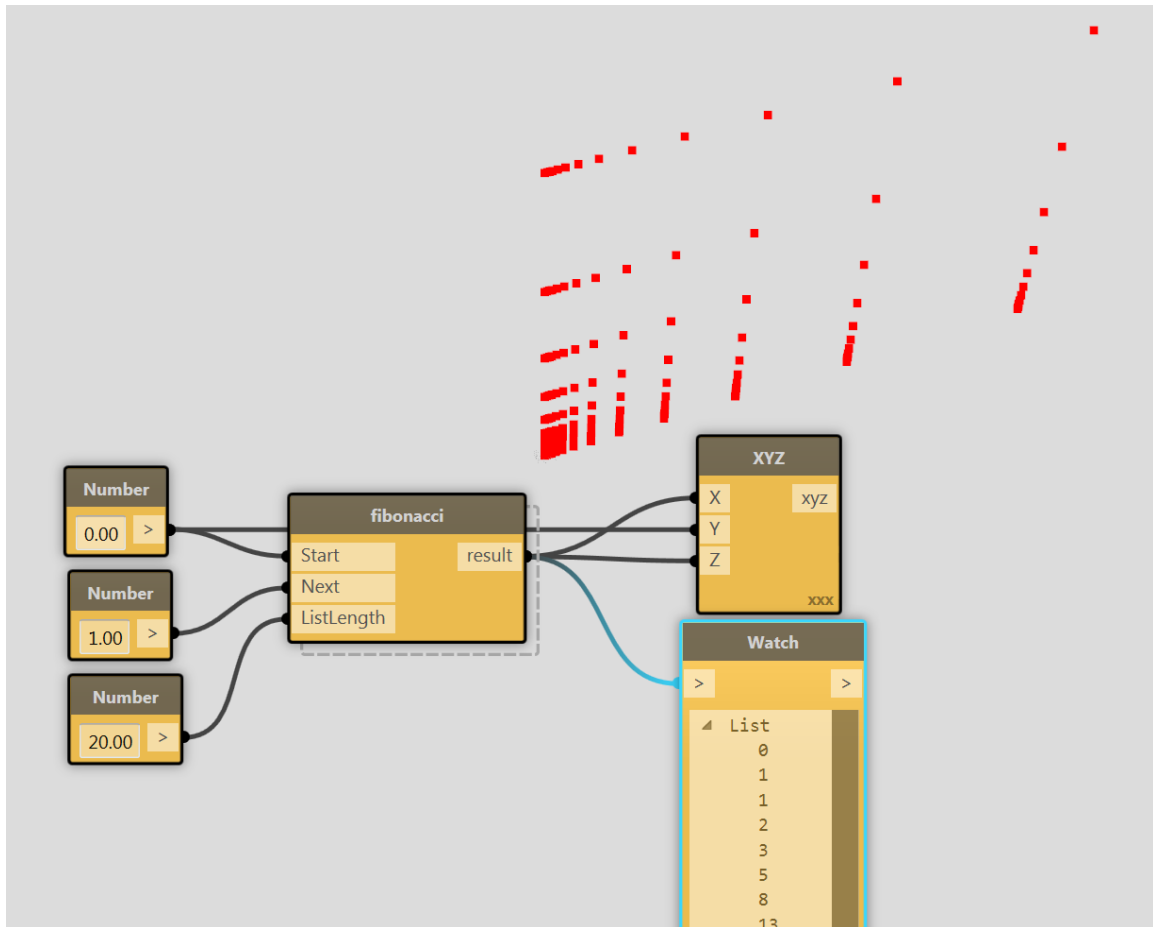
As shown in Example 01, Custom node creation allows users to make compact representations of groups of functionality. Users can identify modular sections of their workflows for reuse either in the same project, or in other projects. But Custom nodes also allow for more advance functionality in Dynamo by accessing recursive functionality.

- Learn how to do iterative processes by nesting Custom nodes inside of themselves
- Create a Fibonacci sequence

1. If Dynamo is open, close down the session.

Programming for Non-Programmers

Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

2. In the class datasets, find the file called "fibonacci.dyf" and copy it to your Dynamo definitions folder:
   C:\Autodesk\Dynamo\Core\definitions

3. Start Dynamo. In the Workspaces menu you should now see "fibonacci". Click on it.
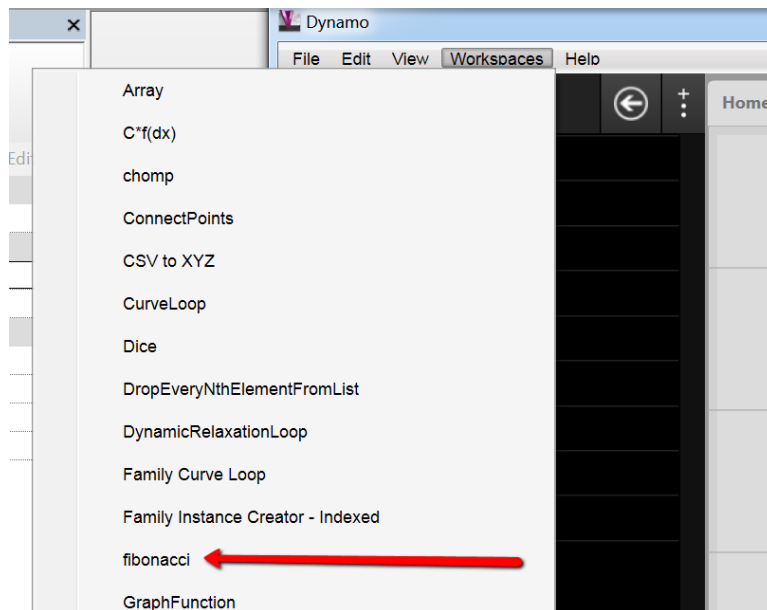

4. You will now have a second tab open which shows the contents of the custom node
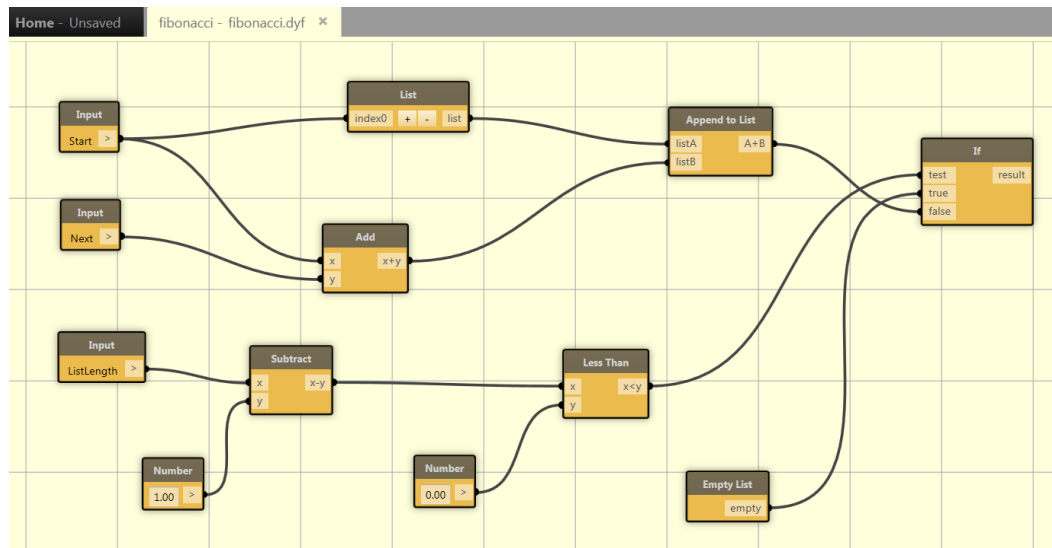
5. The logic of this node is incomplete. To create a fibonnaci sequence, the computation needs a starting sequence (usually 0,1) which is then recursively added. The second number is added to the first, then the resulting third number is added to the second, and so on. In the search bar, enter "fibonnaci" and you will now see your loaded custom node appear. Place it INSIDE of the active fibonacci.dyf workspace.
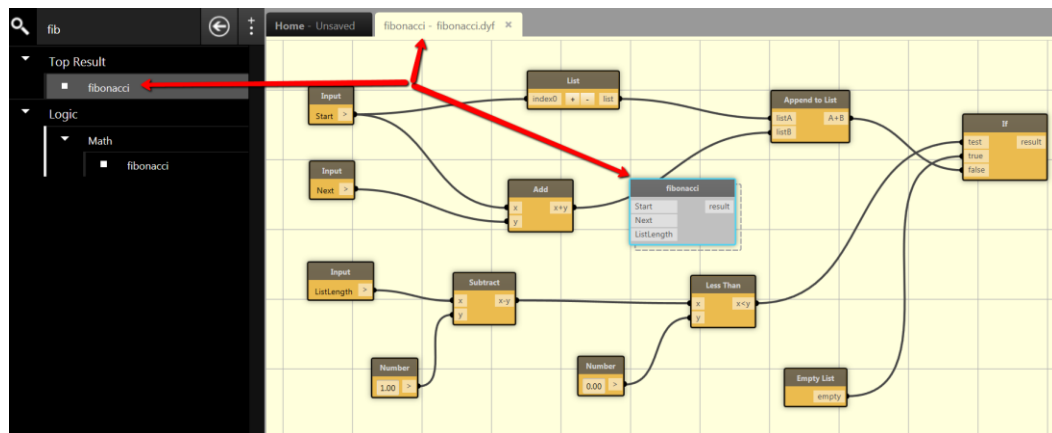
6. The second number in the Fibonacci sequence becomes the starting number for the next calculation. We can now pass the "next" input to the "start" input of the recursive node. The additive result of Start and Next becomes the subsequent calculation's "next" input, and the "Result" of this calculation gets appended to a growing list of numbers.
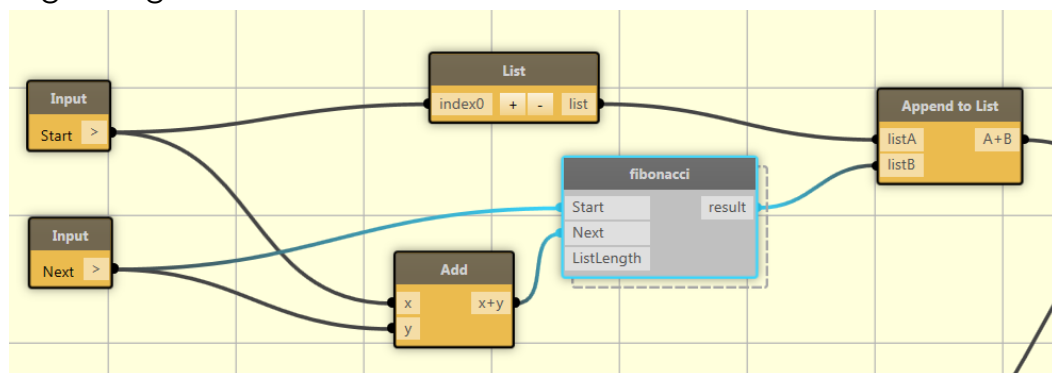

7. This calculation would continue infinitely if there was not condition specified to stop it. The ListLength creates a countdown,

subtracting one from every loop of the recursive function, until it becomes zero, at which point a conditional statement tells the loop to cease. Wire the countdown ListLength into the recursive node. See how the "IF" node creates a condition wherein the appended list is passed on until the LengthList passes a value less than zero.
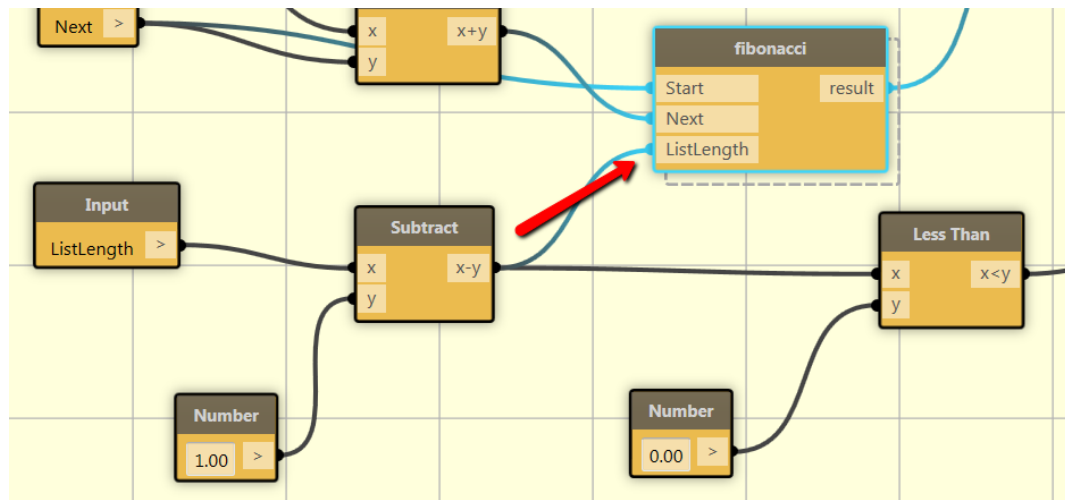
8. Save the file, click on the Home workspace tab, and place your fibonnaci node in the workspace
9. Wire it up with Start, Next, and ListLength nodes, and monitor the output with a watch.
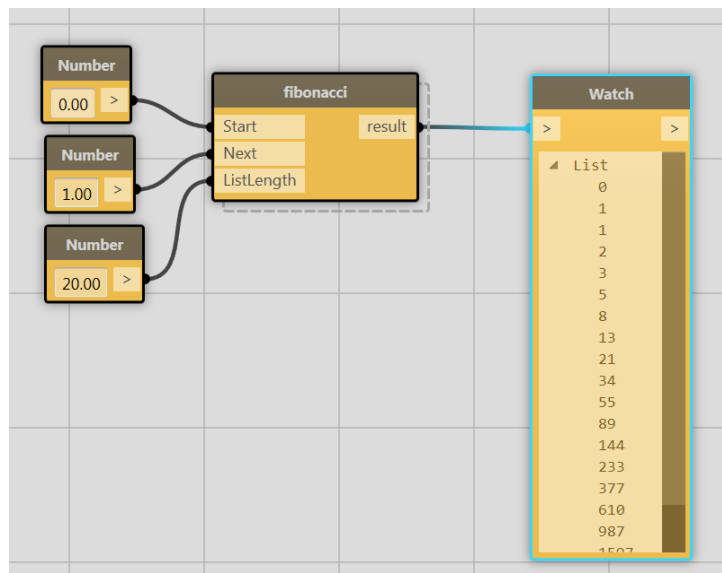
What Else Can Dynamo Do?

The above examples all demonstrate many of the individual capabilities of Dynamo working on top of either Vasari or Revit. By combining these and other techniques, users can build up complex and robust parametric building systems. The following images demonstrate one such project, a Stadium enclosure and seating bowl. The seating bowl is formed by a recursive algorithm that optimizes the view for each seat based on the seat below it. The enclosure creates a self-adjusting shading system based upon the orientation of the panels relative to the sun. The structural system is a combination of manually defined and Dynamo placed truss systems, leveraging both the rule based and hand modelled strengths of Dynamo and Vasari.
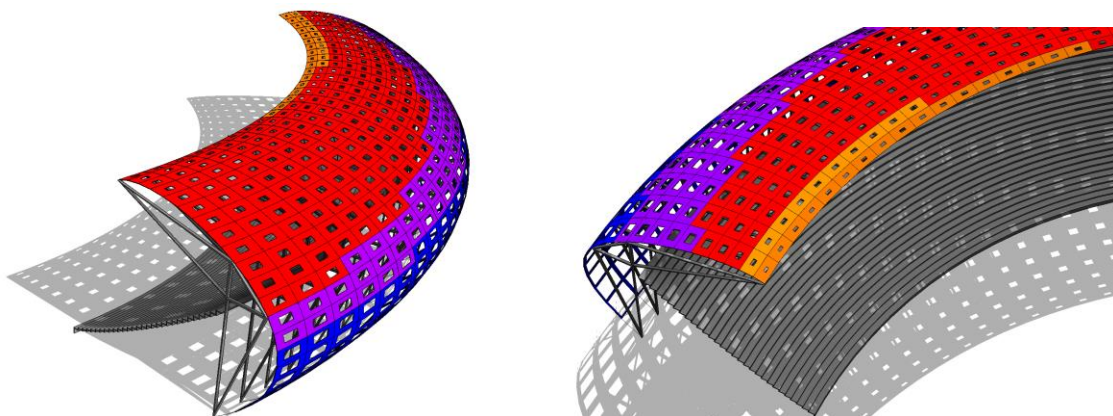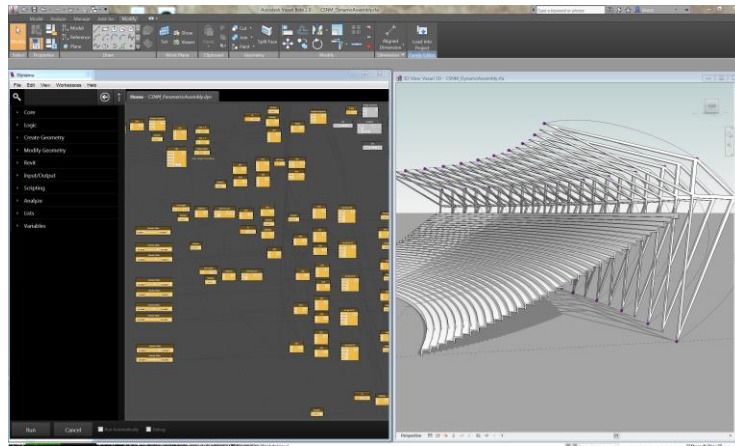
Matt Jezyk, Peter Boyer, and Zach Kron, Autodesk, Inc

Where to Learn More about Dynamo

Dynamobim.org
Get an official build.

github.com/ikeough/Dynamo
Watch development as it takes place

github.com/ikeough/Dynamo/issues
Submit bugs, comments, and improvements for Dynamo