Computational Fabrication

CS 491 and 591
Professor: Leah Buechley
https://handandmachine.cs.unm.edu/classes/Computational_Fabrication_Spring2021/

Weekly Artist: Piotr Waśniowski

https://www.instagram.com/piotr__wasniowski/

What we did last class

1. Write code to generate these 2D lattices, illuminating some fundamental tiling geometry
2. Use our lattice generating code to generate 2D tiles and tilings

What we did last class

2. Use our lattice generating code to generate 2D tiles and tilings.

   Adding some Escher-like tile manipulation

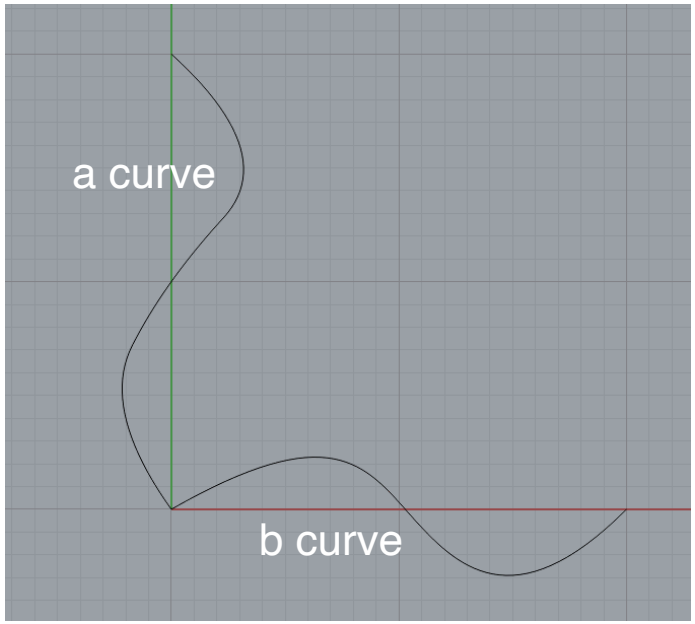
Approach

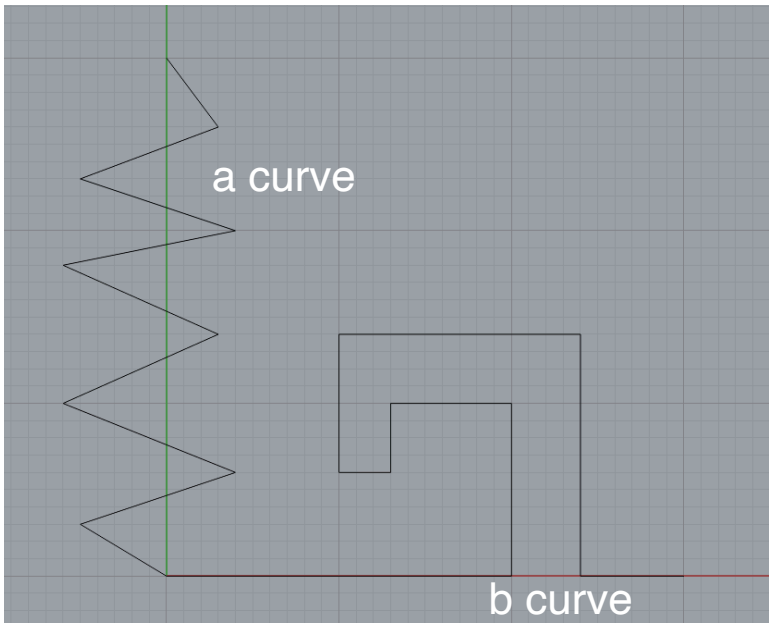
1. Allow Escher input curves as **a** and **b** curves of lattice.
2. Input curve requirements:
   - **a** curve: begins at origin and ends at point on y axis
   - **b** curve: begins at origin and ends at point on x axis
3. Edit first Python block
   - Accept Escher curves as input
   - Output appropriately scaled and rotated Escher curves.

questions?

Draw Curves in Rhino

- **a** curve: begins at origin and ends at point on y axis
- **b** curve: begins at origin and ends at point on x axis



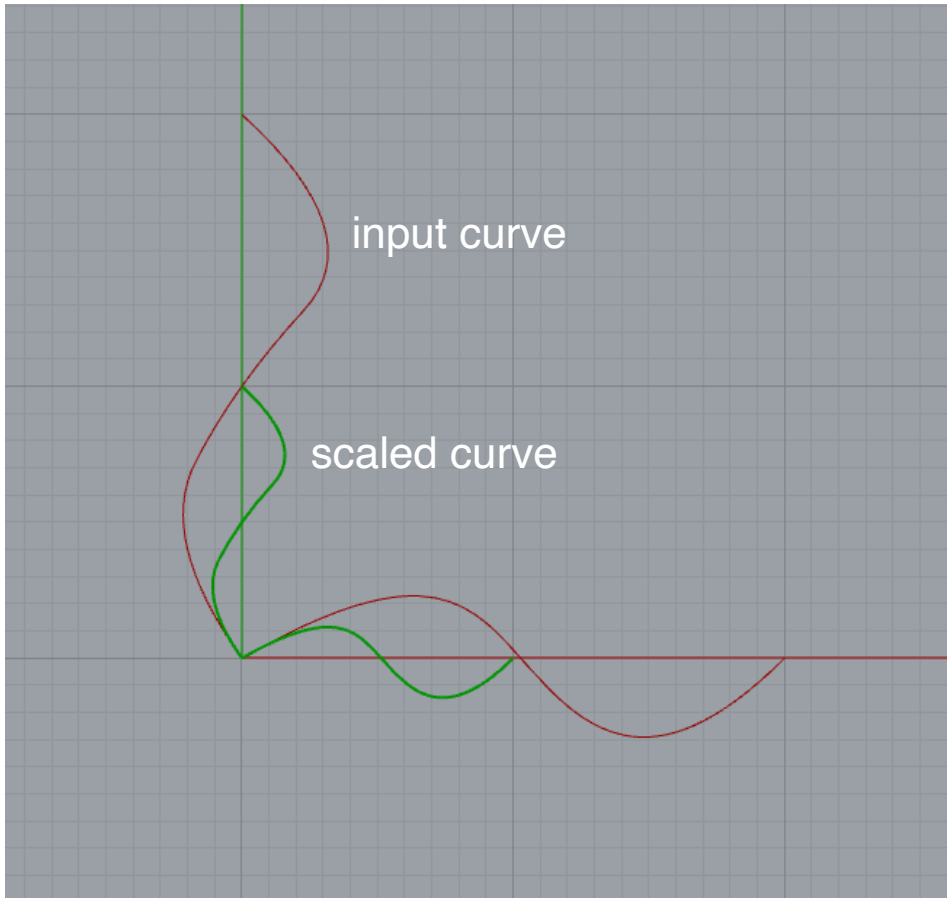


Scale Curves to fit Lattice

1. Use **rs.CurveEndPoint()** to find end points of curves.
2. What does the end point tell us about the length of curve **a**?
3. Use **rs.ScaleObject()** to scale each curve
4. What is the scale factor for curve **a**?

```
17 #scale curves to match magnitude inputs
18 curve_a_length=rs.CurveEndPoint(curve_a).Y
19 a_scale = a_length/curve_a_length
20 rs.ScaleObject(curve_a, point, rs.CreatePoint(a_scale,a_scale,1))
```

Scale Curves to fit Lattice

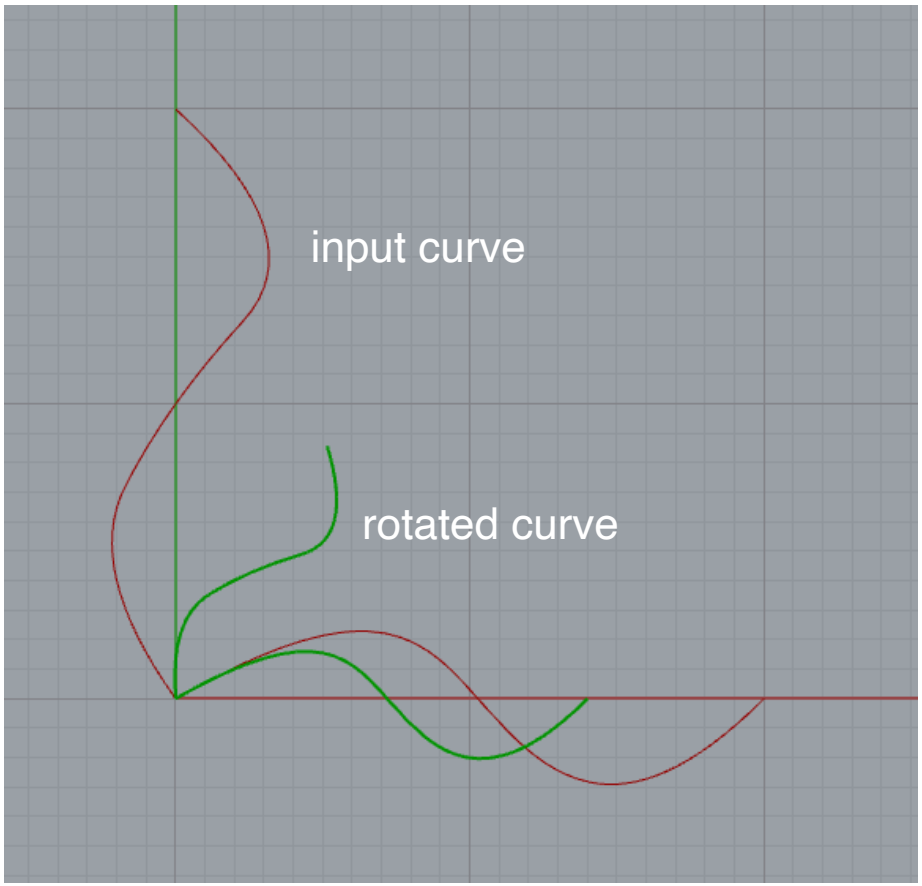


Rotate Curves to fit Lattice

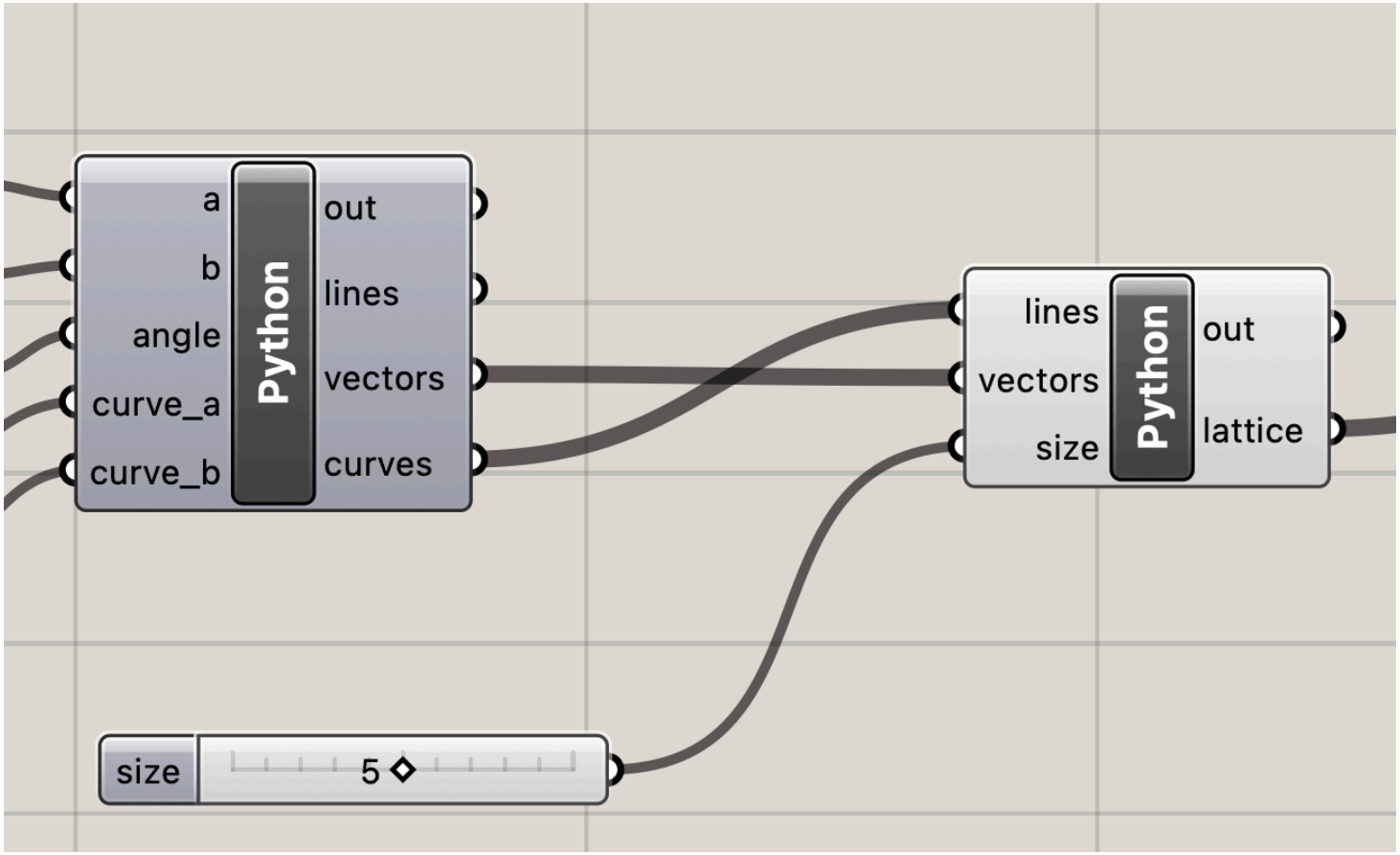
1. Which curves do we have to rotate?
2. What is the rotation angle in terms of the input angle?

```
22 rs.RotateObject(curve_a,point,angle-90)
23
```

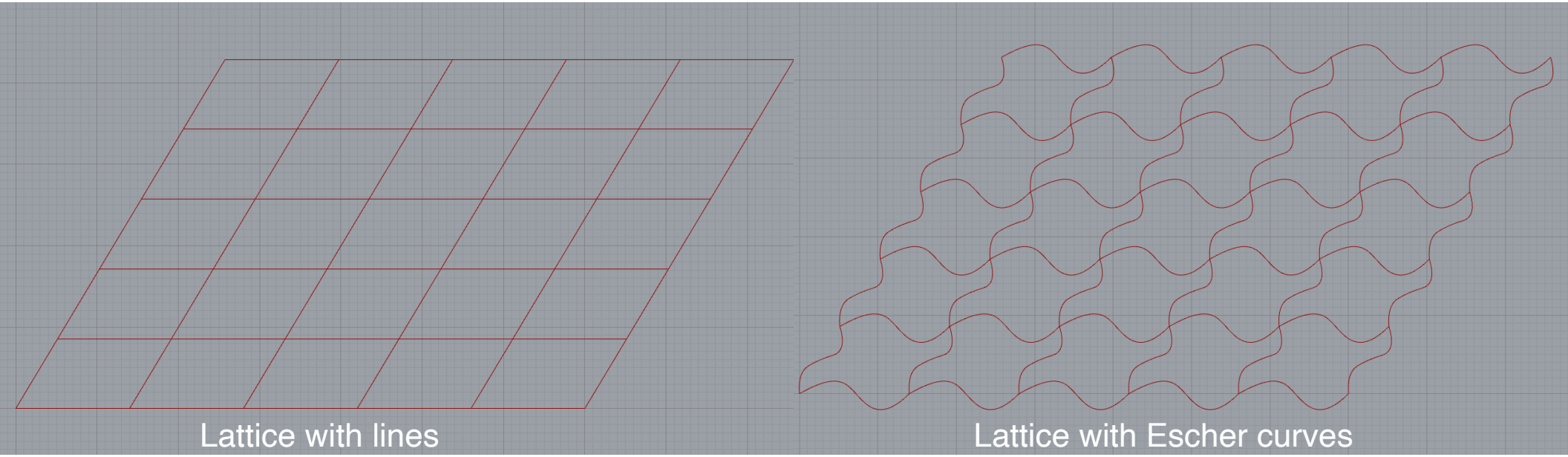
Rotate Curves to fit Lattice



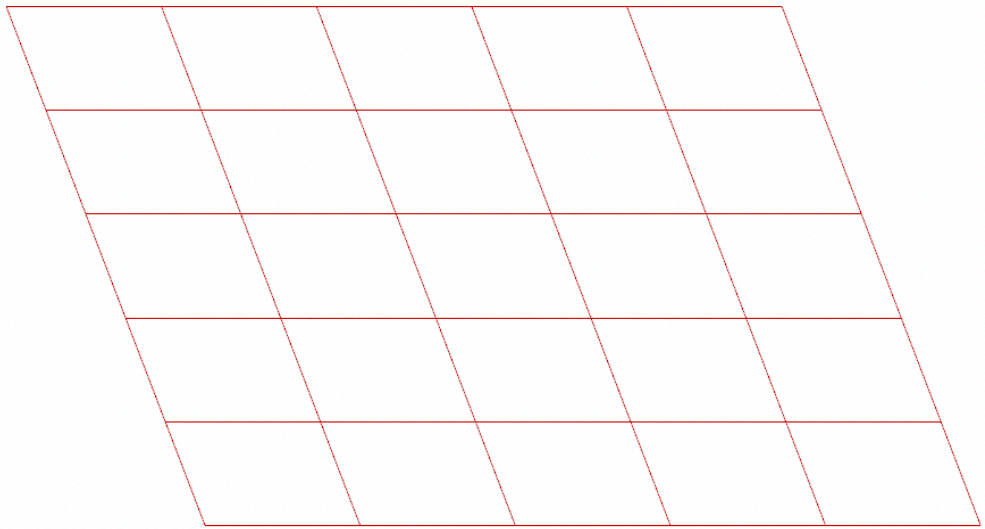
Connect Curves to Lattice Code

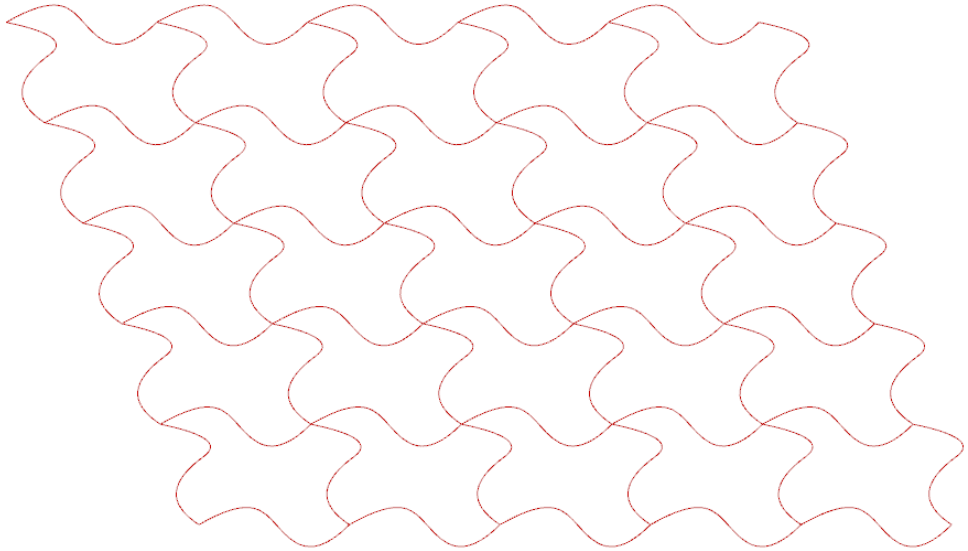


Connect Curves to Lattice Code



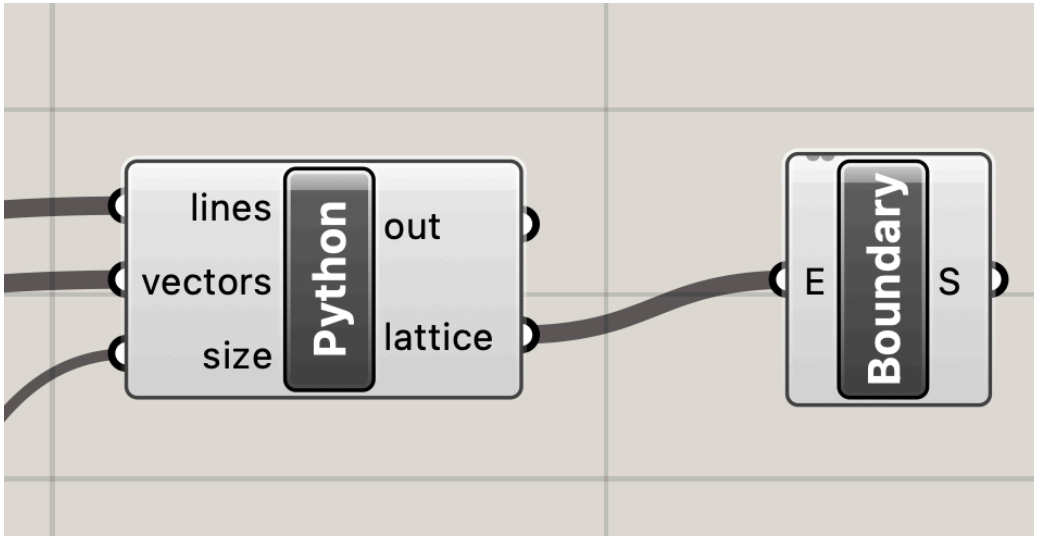
Connect Curves to Lattice Code

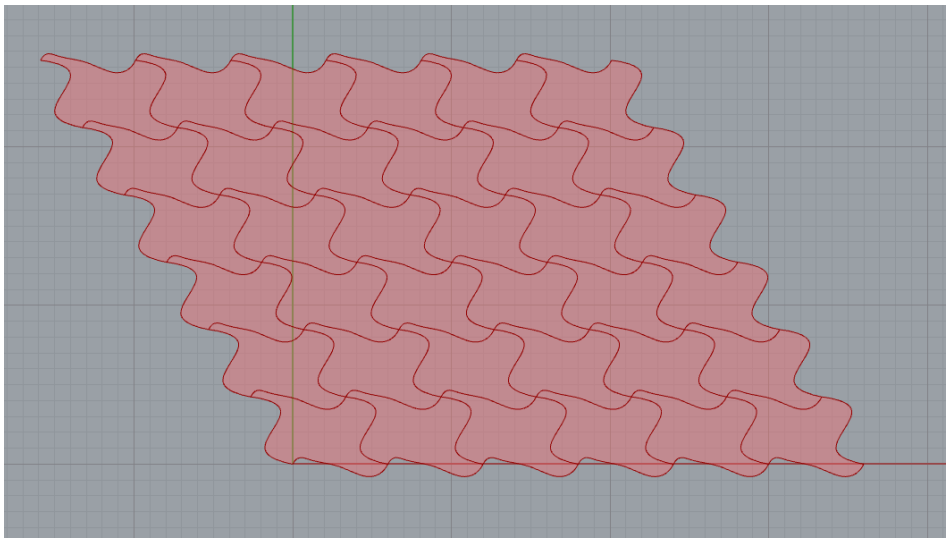




Rendered view in Rhino

Create surfaces from tile Outlines



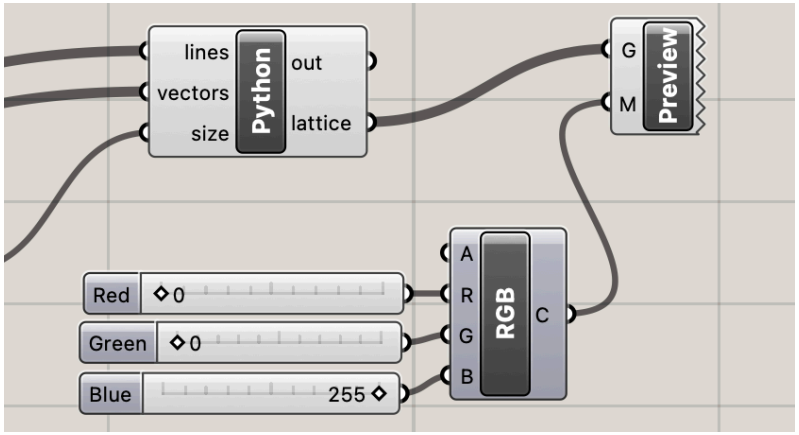


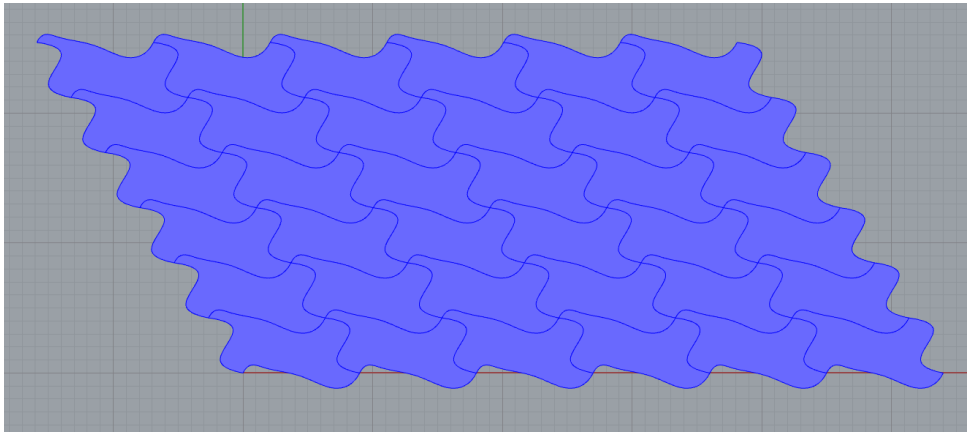
questions?

Add Some Color

Color in Grasshopper

- Color:
  - standard RGB: (red, green, blue)
  - each number ranges from 0-255
- To render color, use a “Custom Preview” block

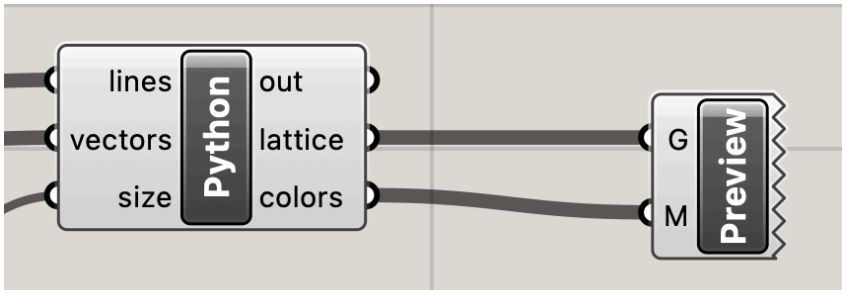




Color in Python + Grasshopper

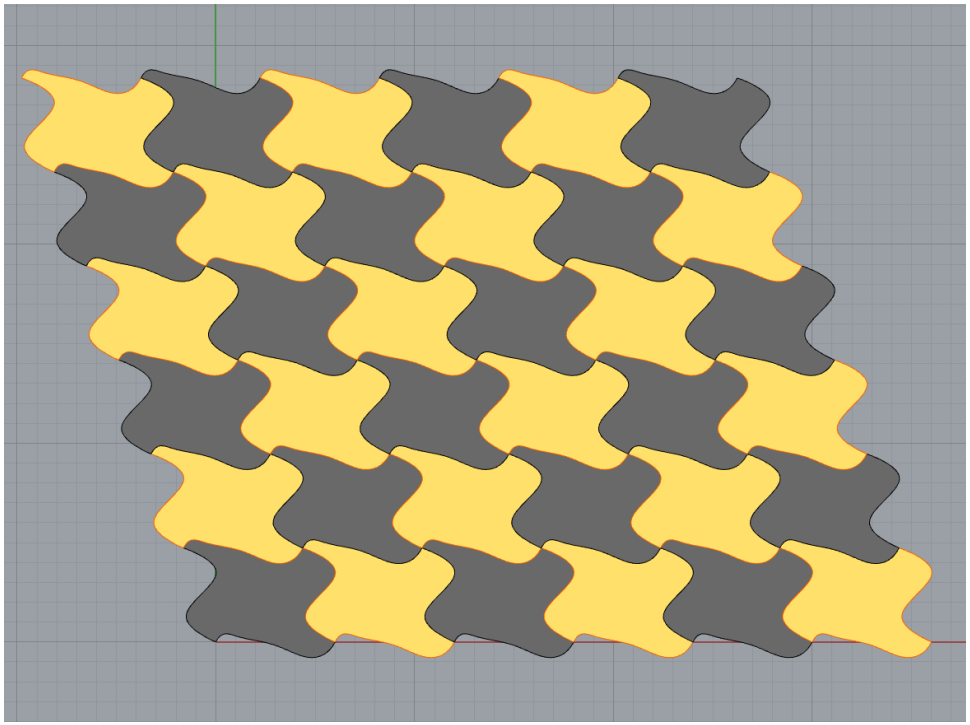
- Create a color array, where each color is added as a string
- Connect array to M input on Preview block

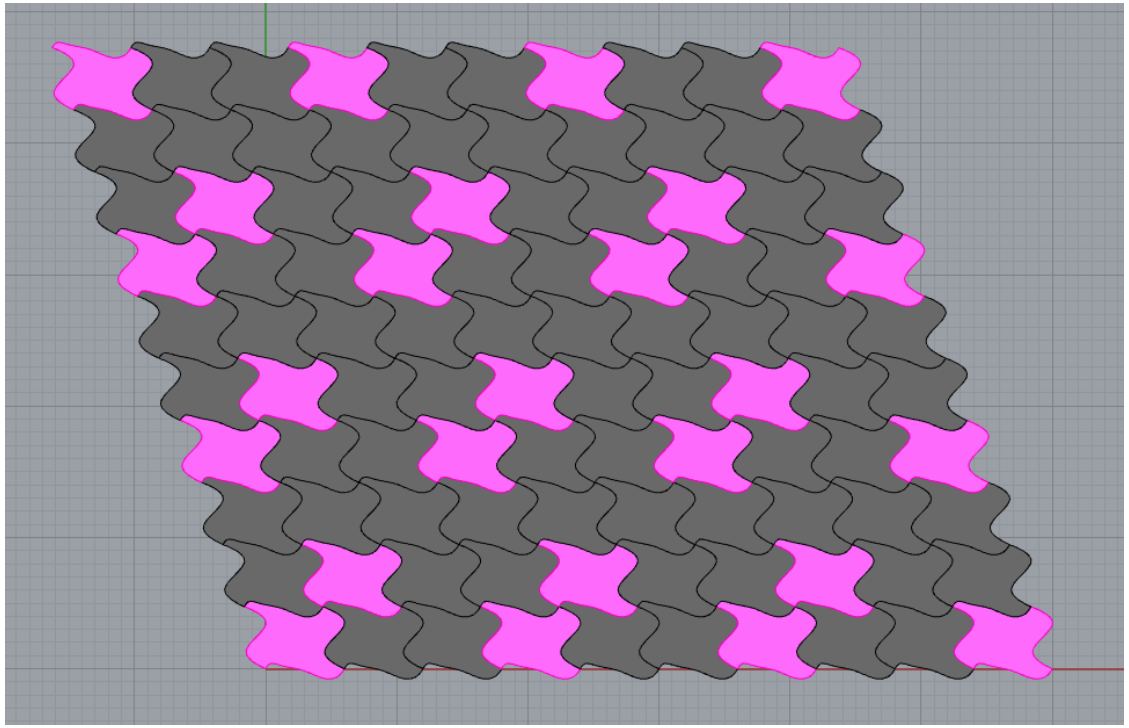
```
tiles = []
colors = []
for i in range(size):
    for j in range(size):
        edge0 = lattice[i][j][1] # bottom
        edge1 = lattice[i][j][0] # left side
        edge2 = lattice[i+1][j][1] # bottom
        edge3 = lattice[i][j+1][0] # bottom
        tile = rs.JoinCurves([edge0,edge1,edge2,edge3])
        tiles = tiles + tile
        colors.append("0,0,0")
```

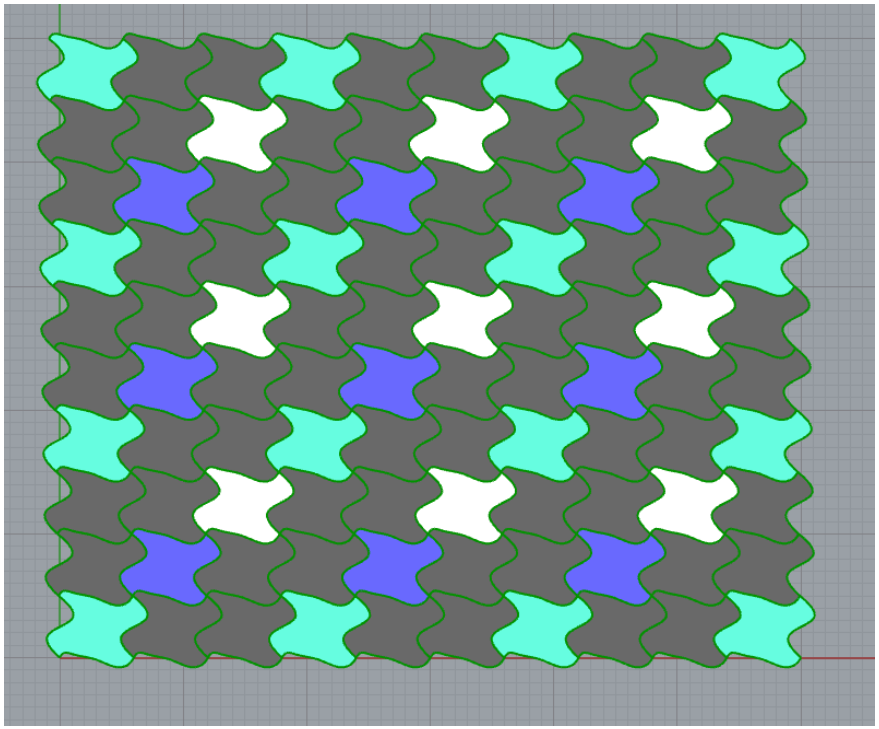


Color Tiles in Patterns

```
17 # generate tiles
18 # flatten data structure to list
19 tiles = []
20 colors = []
21 for i in range(size):
22     for j in range(size):
23         edge0 = lattice[i][j][1] # bottom
24         edge1 = lattice[i][j][0] # left side
25         edge2 = lattice[i+1][j][1] # bottom
26         edge3 = lattice[i][j+1][0] # bottom
27         tile = rs.JoinCurves([edge0,edge1,edg
28         tiles = tiles + tile
29         if (i%2==0 and j%2==0):
30             colors.append("0,0,0")
31         elif(i%2==1 and j%2==1):
32             colors.append("0,0,0")
33         else:
34             colors.append("250,100,0")
```





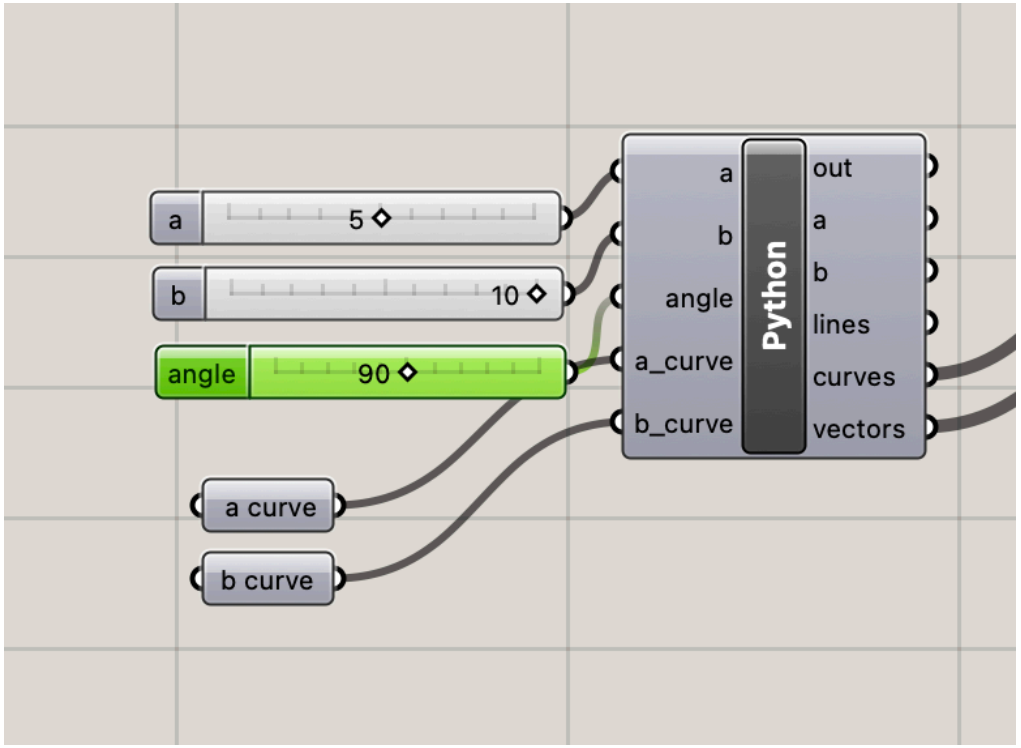


questions?

Tiling Complex Surfaces

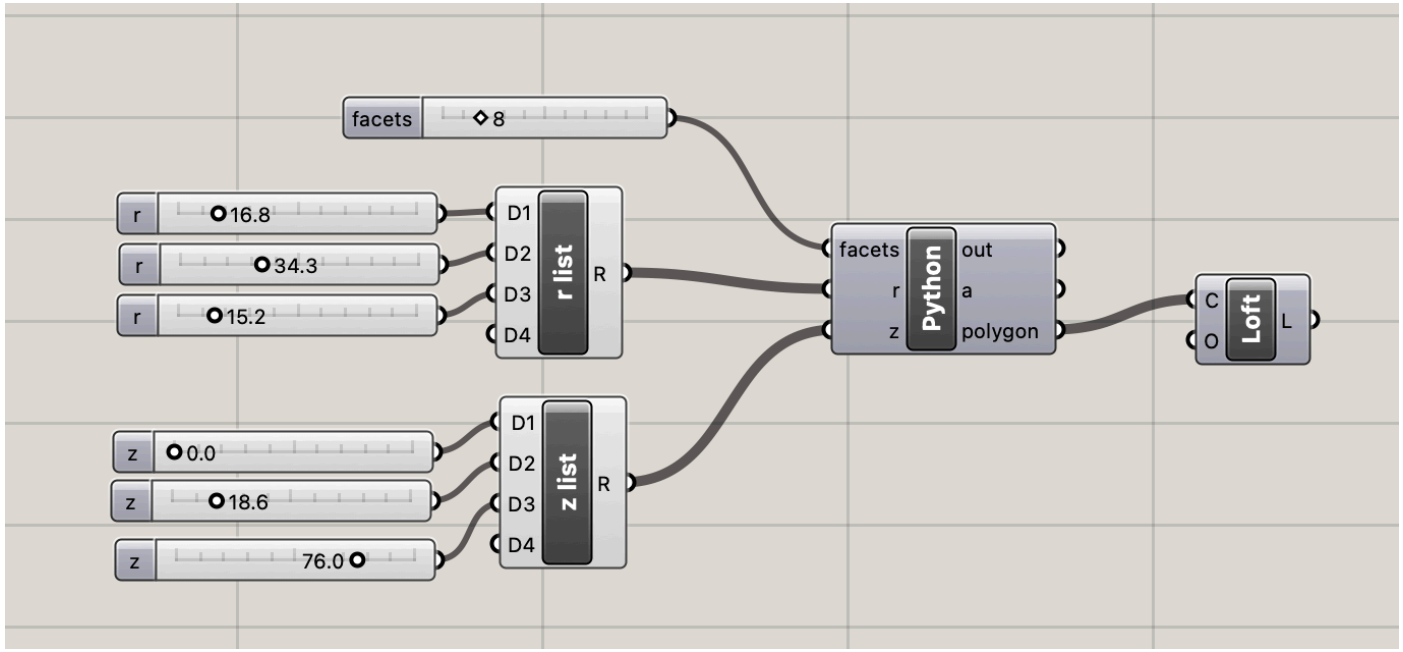
We will morph our tiling across a surface.
Note: can morph any 3D geometry across a surface

First: set your angle to 90°

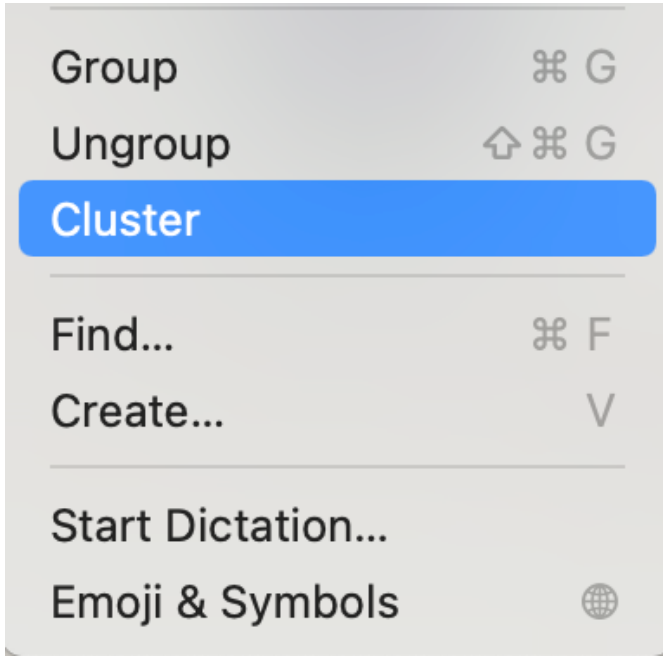


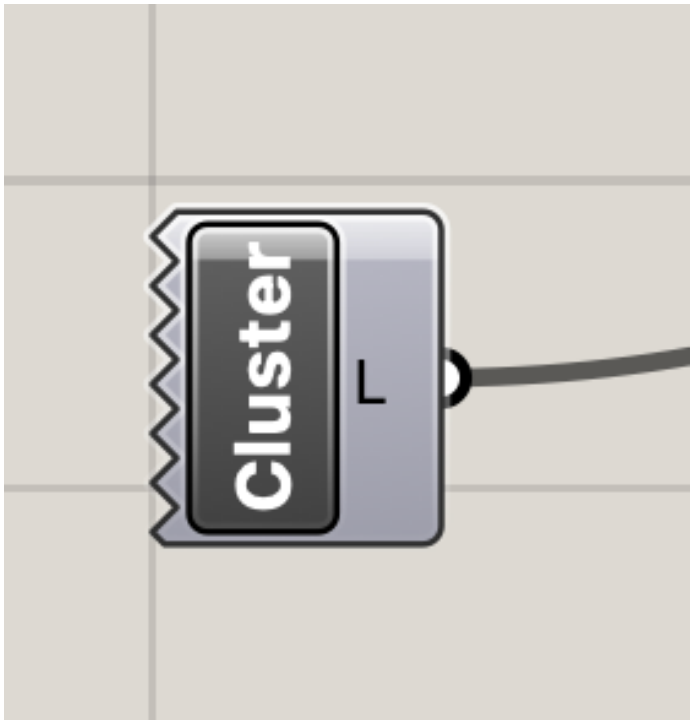
Create A Surface.
Open the Code from the Vessel Assignment

Copy and Paste into Tile Program

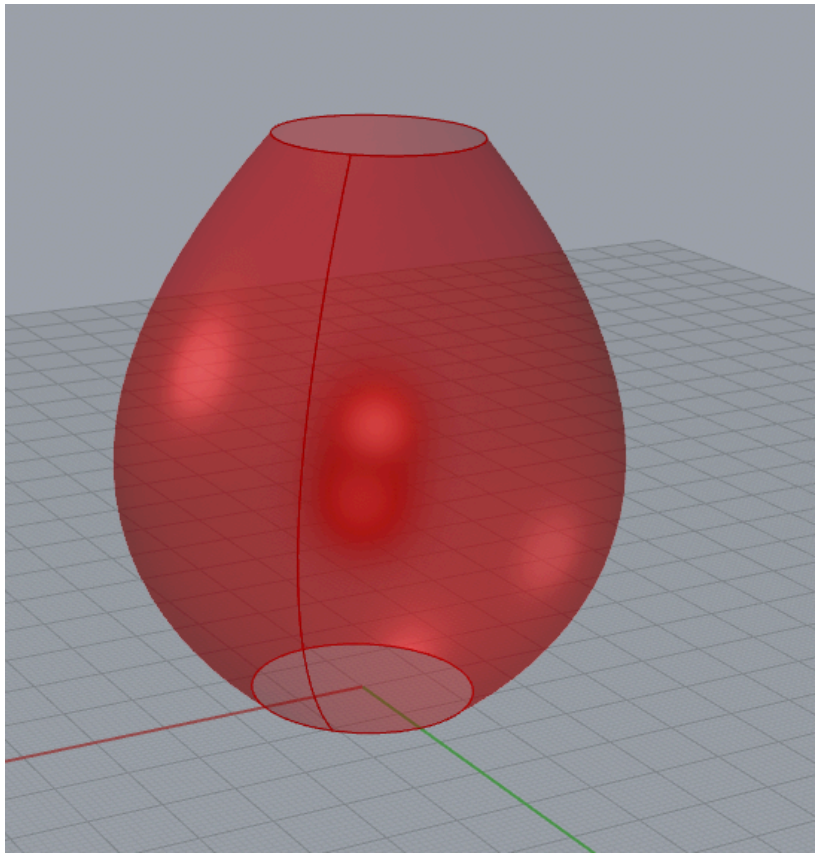


Optional: Save As Cluster



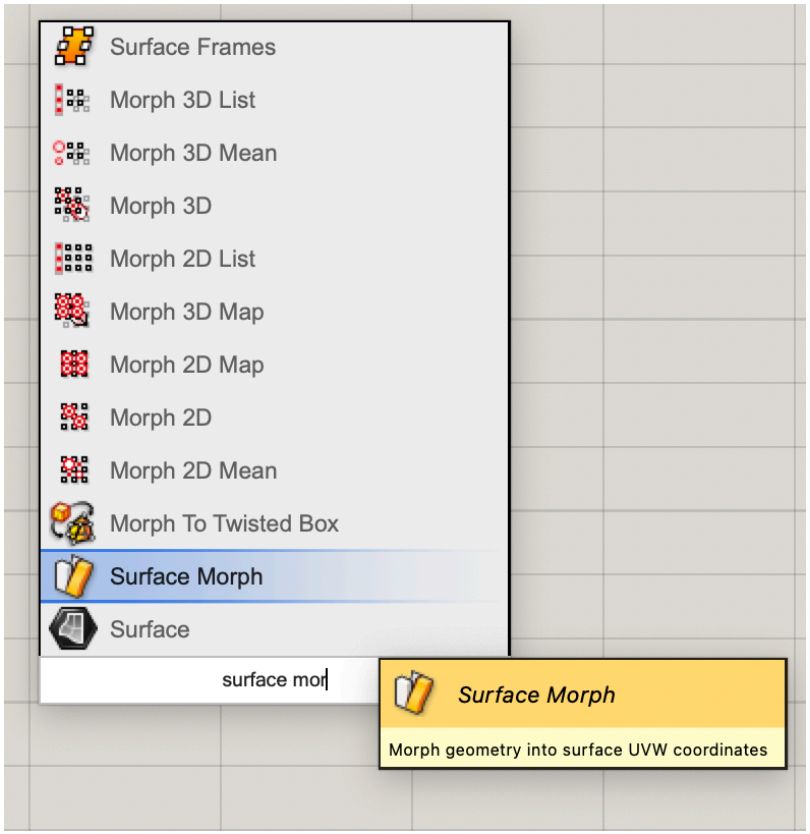


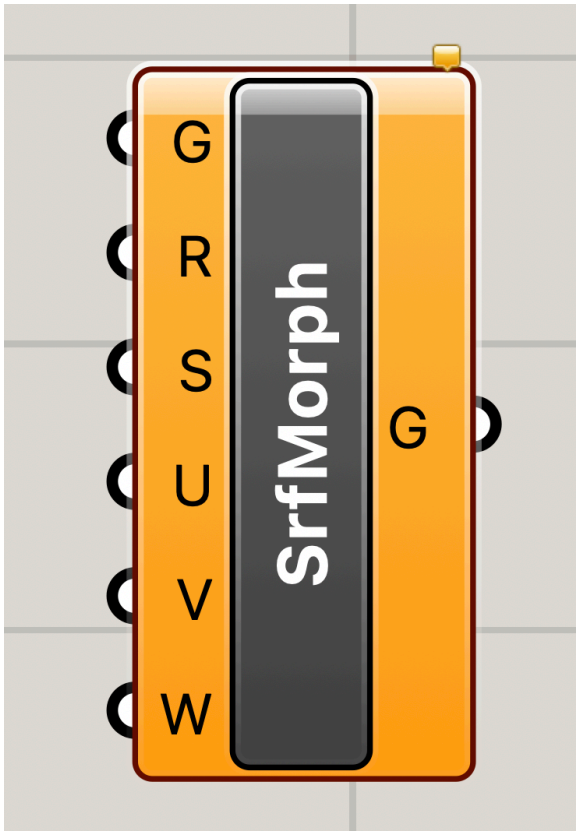
Surface



questions?

Drag out a Surface Morph block

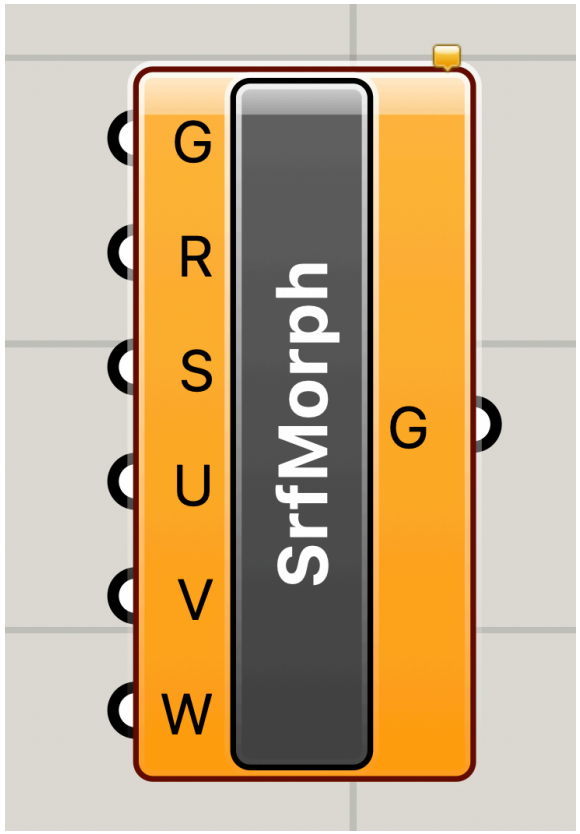




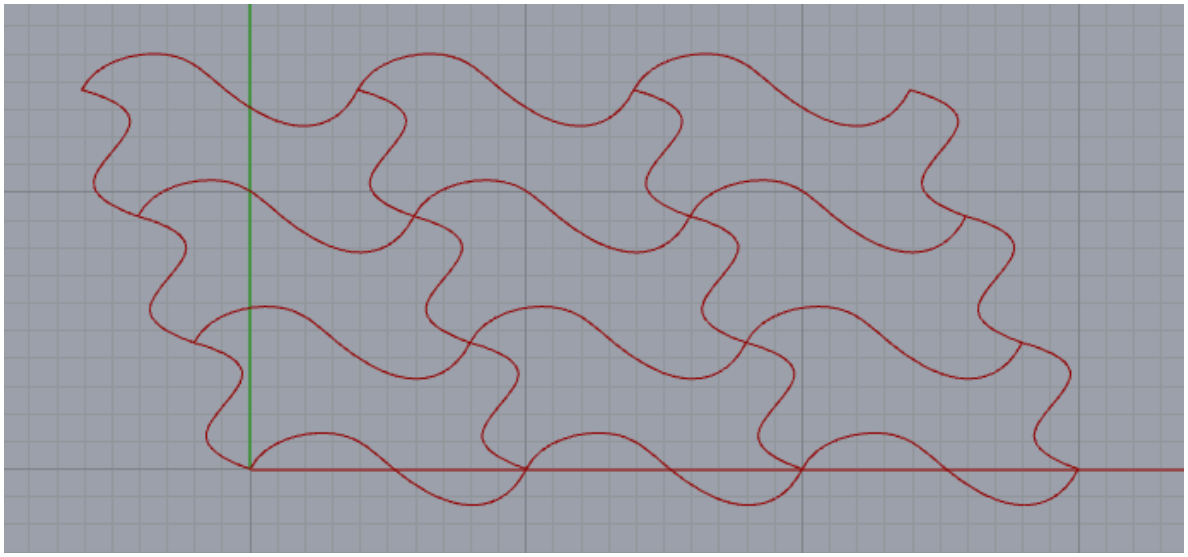
Surface Morph

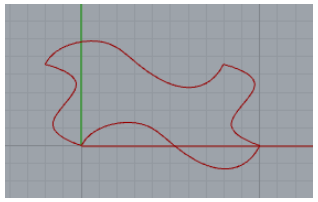
The surface morph block takes as input:

- A geometry (G), we'll use our complete tiling
- A size reference for the input (R), we will use one basic lattice cell
- A surface (S)
- U,V,W = a size reference that determines how the geometry is stretched across the surface in the x(u), y(v), and z(w) dimensions

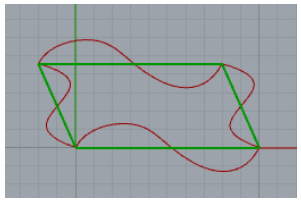


Some Observations About Sizes

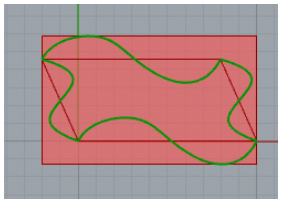




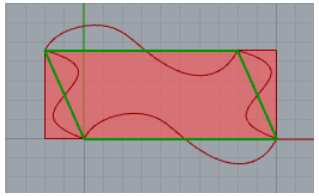
tile



+ lattice cell



+ tile
bounding box



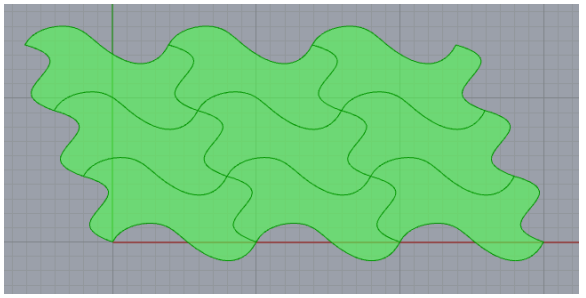
+ lattice cell
bounding box

Tiling is generated across the **lattice**.
Lattice cell determines tile translations.
Lattice cell edges: **a** and **b**

questions?

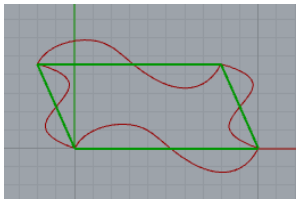
Surface Morph

- A geometry (G), we'll use our complete tiling



G

- A size reference for the input (R), we will use one basic lattice cell



R = lattice cell

- U,V,W = a size reference that determines how the geometry is stretched across the surface in the x(u), y(v), and z(w) dimensions

U,V = 1/N N = size of array
one tile takes up 1/N
of the surface area

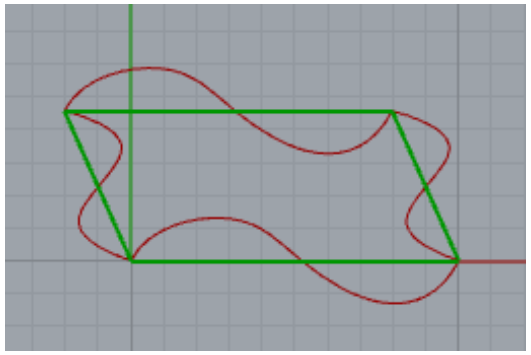
W = thickness of surface

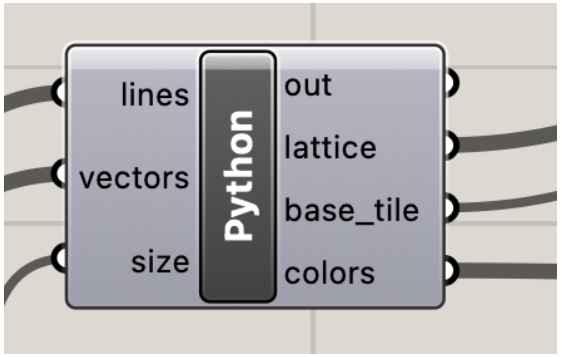
questions?

Create a Base Tile

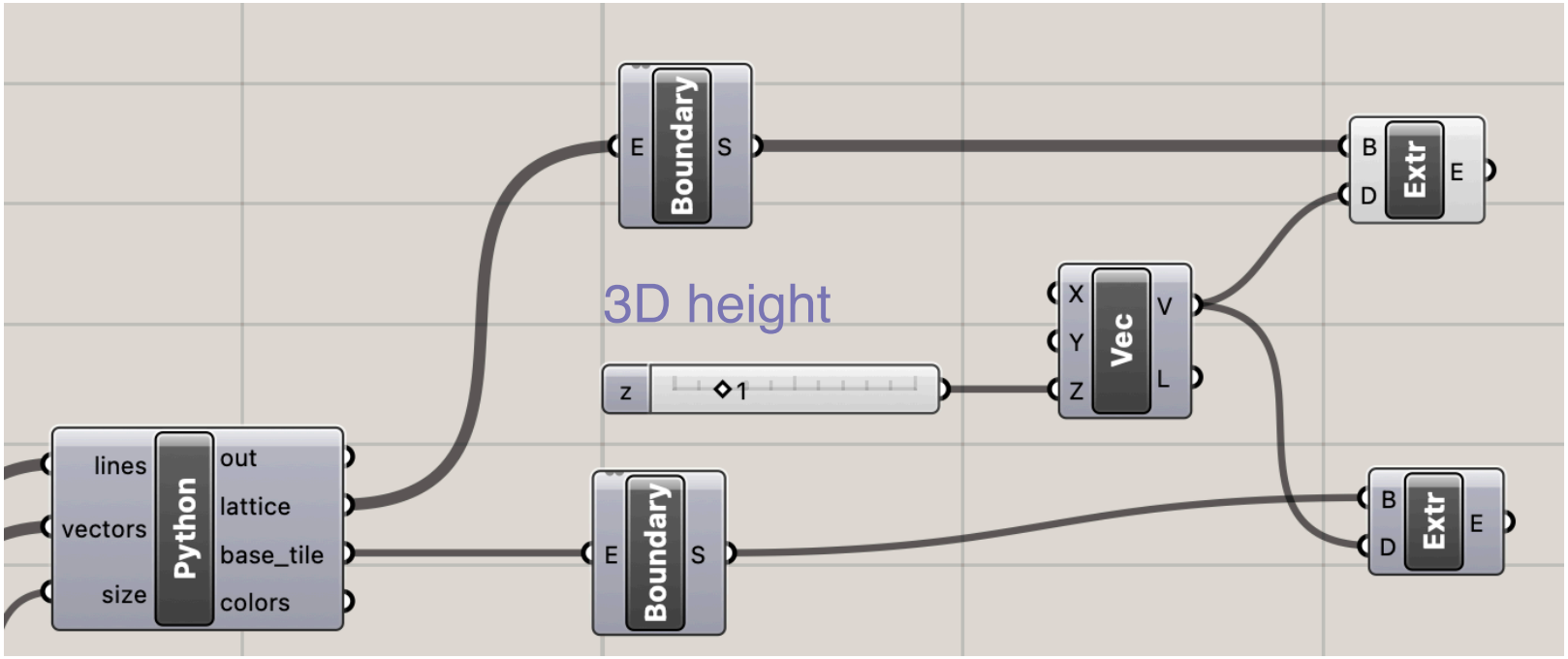
```
base_tile = []
edge0 = lattice[0][0][1] # bottom
edge1 = lattice[0][0][0] # left side
edge2 = lattice[0+1][0][1] # bottom
edge3 = lattice[0][0+1][0] # bottom

base_tile.append(rs.AddLine(rs.CurveStartPoint(edge0),rs.CurveEndPoint(edge0)))
base_tile.append(rs.AddLine(rs.CurveStartPoint(edge1),rs.CurveEndPoint(edge1)))
base_tile.append(rs.AddLine(rs.CurveStartPoint(edge2),rs.CurveEndPoint(edge2)))
base_tile.append(rs.AddLine(rs.CurveStartPoint(edge3),rs.CurveEndPoint(edge3)))
base_tile = rs.JoinCurves(base_tile)
```

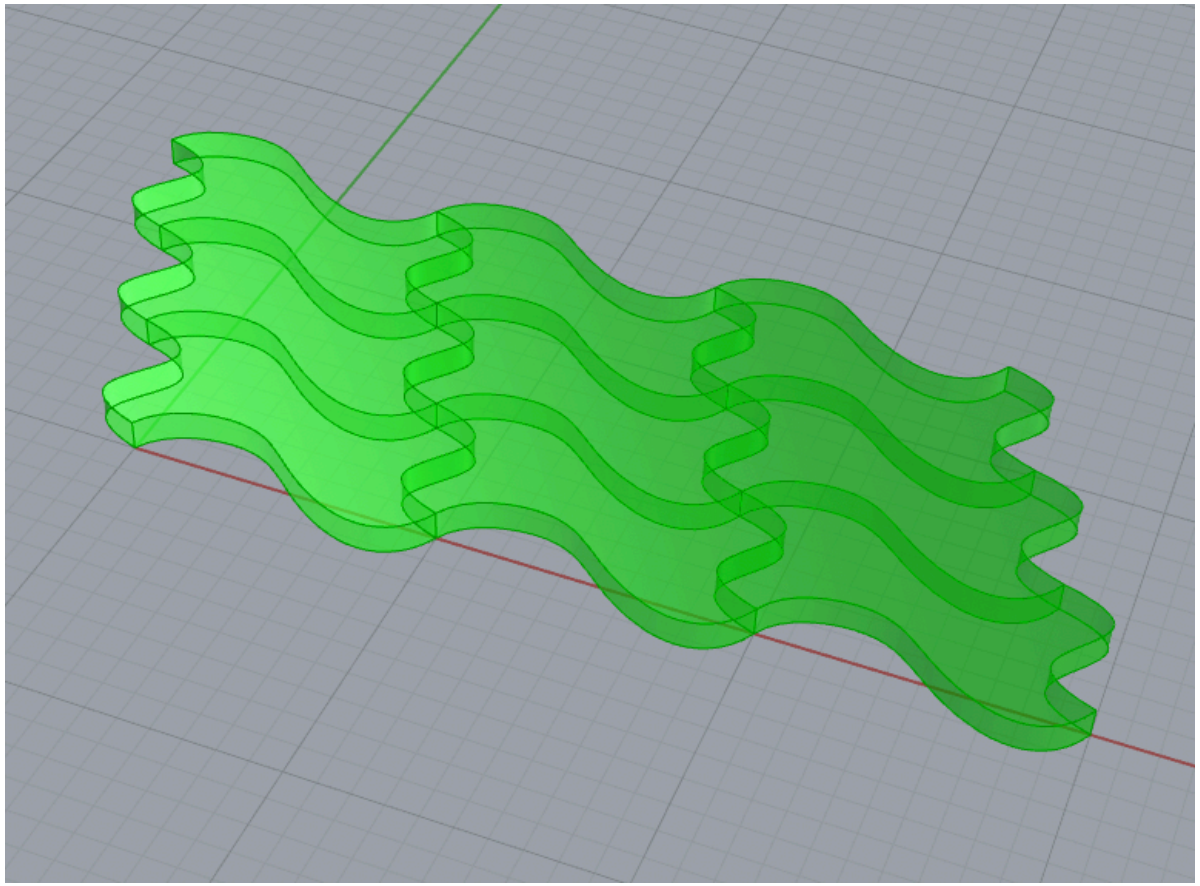




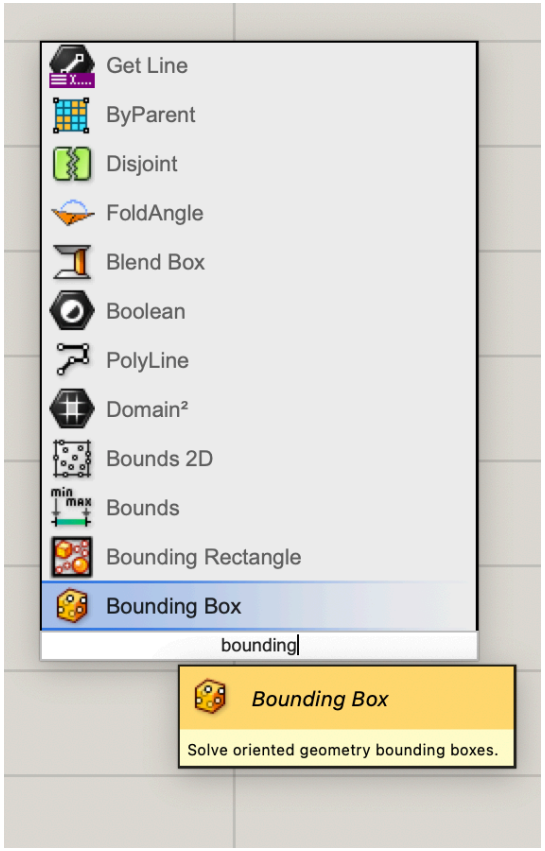
Make Tiling and Base Tile 3D

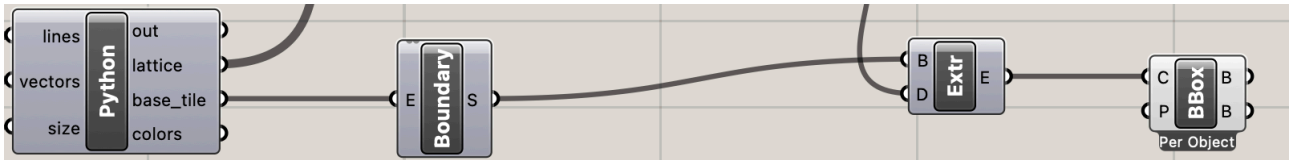


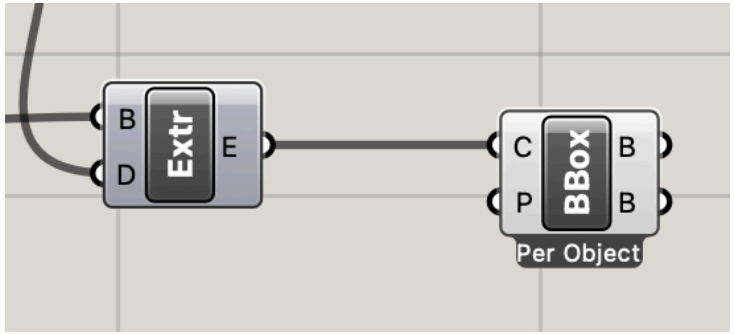
Make Tiling and Base Tile 3D

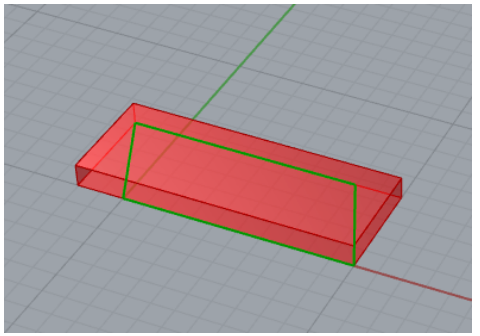


Get Size of 3D Base Tile



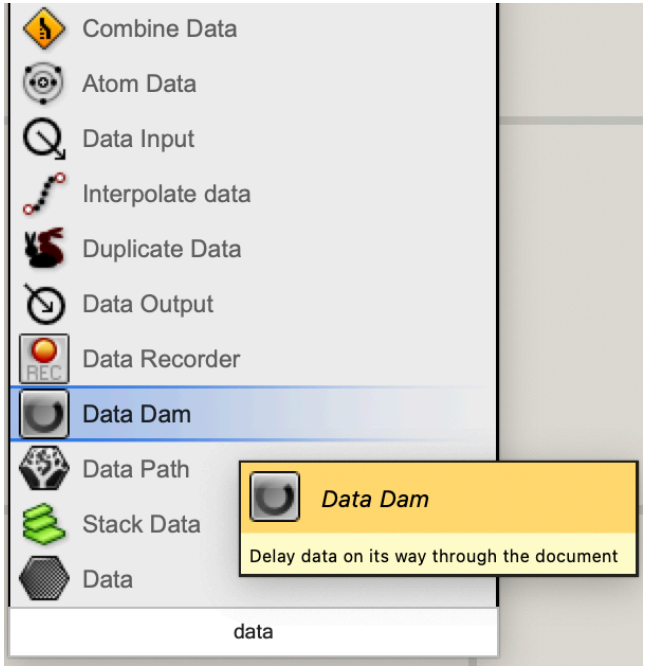




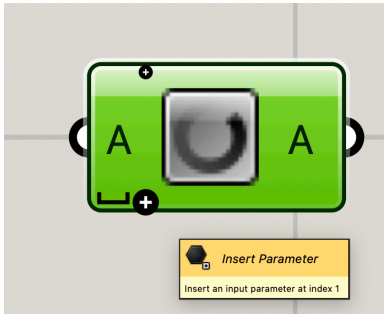


Connect Bounding Box to 3D Base Tile

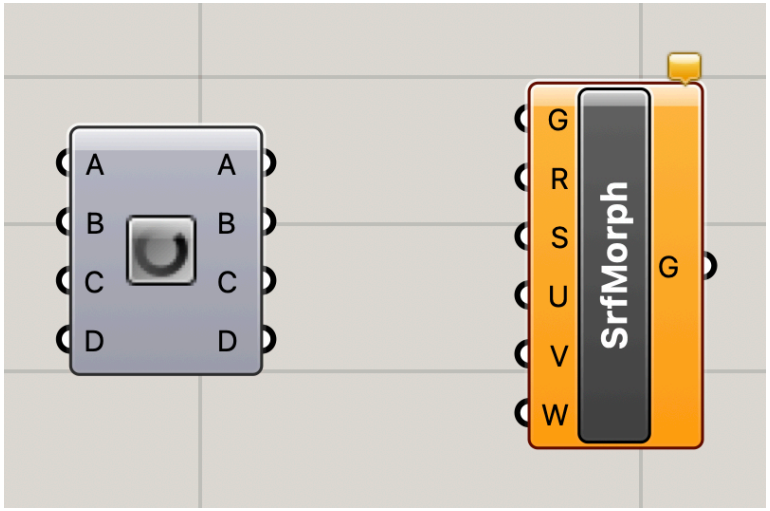
Create a data dam to avoid triggering the computationally expensive Surface Morph



Create a Data Dam

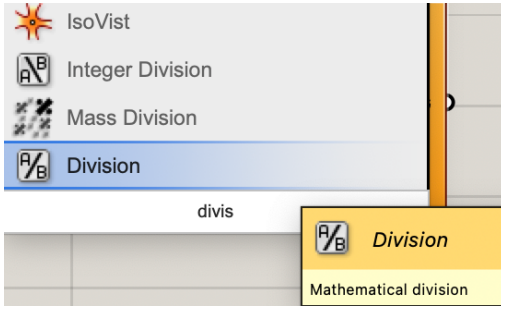


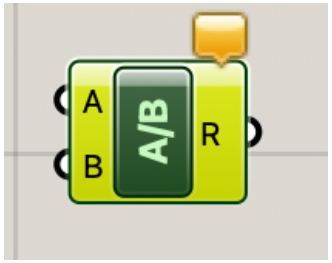
Zoom into the block to add parameters, just like we do with Python blocks



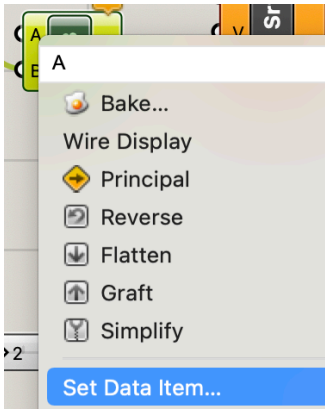
Add 5 parameters to the Data Dam block

Generate U,V information for Surface Morph



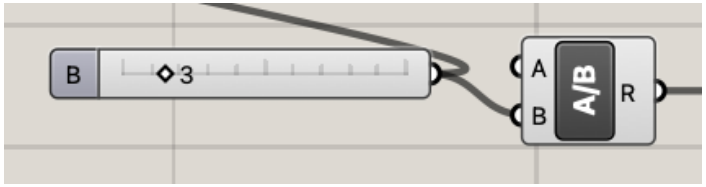


Create a Division block



Right click on the A parameter on the Division block

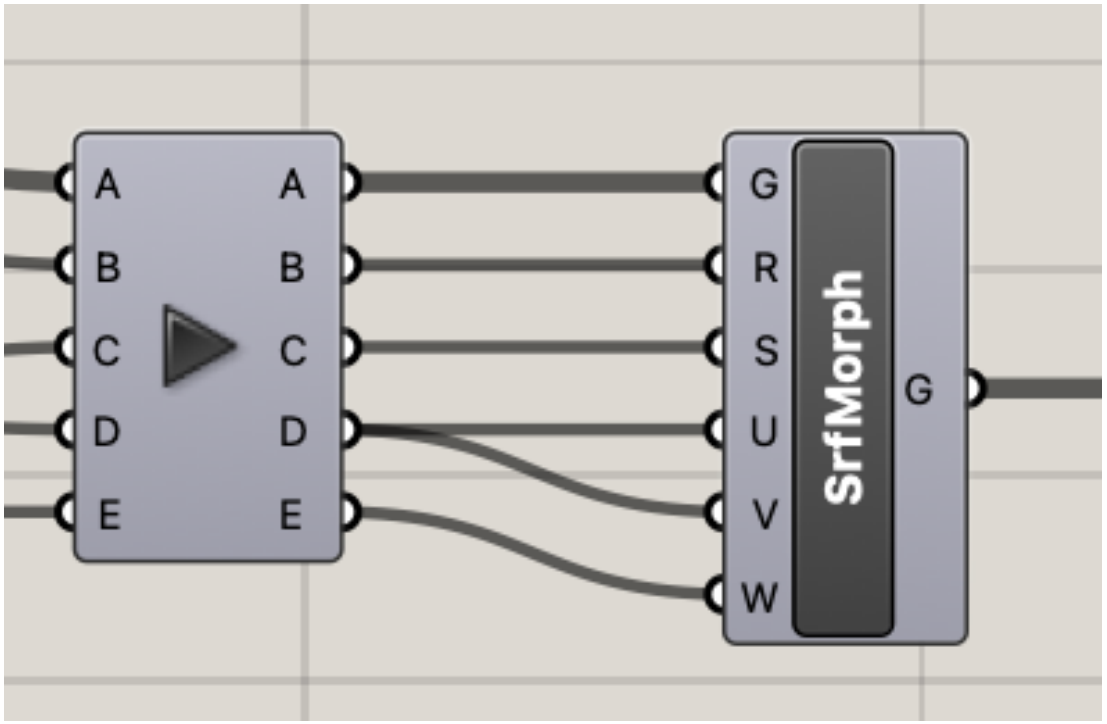
Enter 1 as a Data Item



Connect the size slider to B on the Division Block

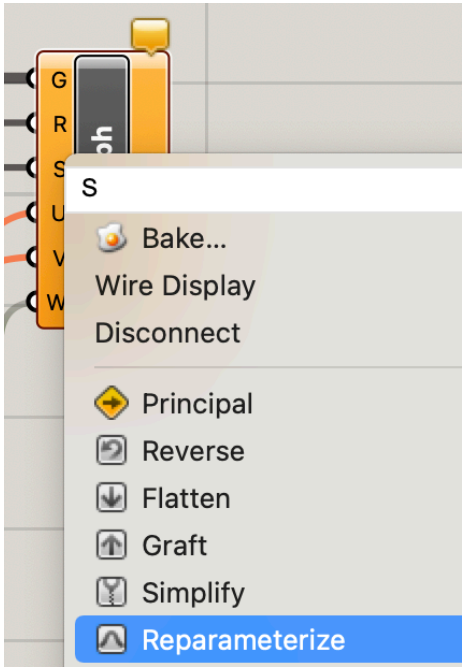
R = 1/number of tiles

Connect inputs through the Data Dam



- tiling goes to G
- Bounding Box goes to R
- Surface goes to S
- 1/size goes to U and V
- number slider goes to W

Reparameterize Surface Morph

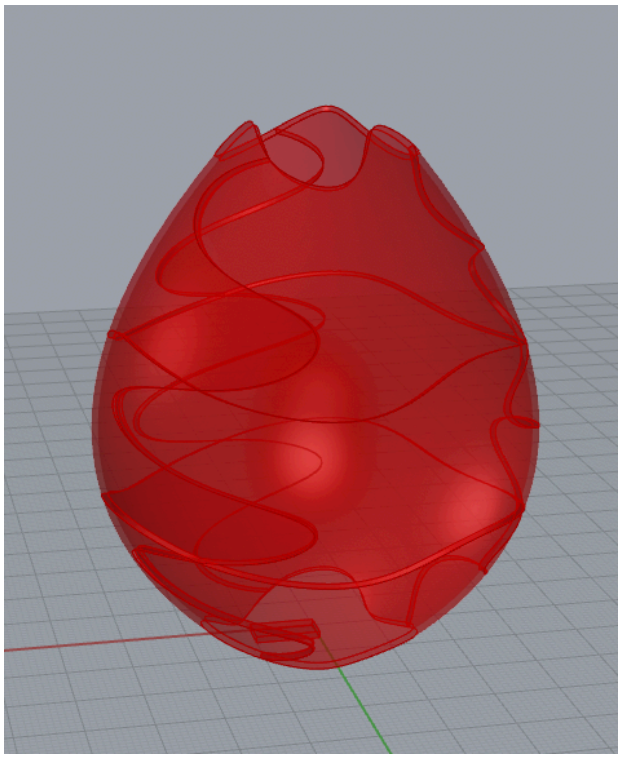


Right click on the S input to the Surface Morph block and click Reparameterize.

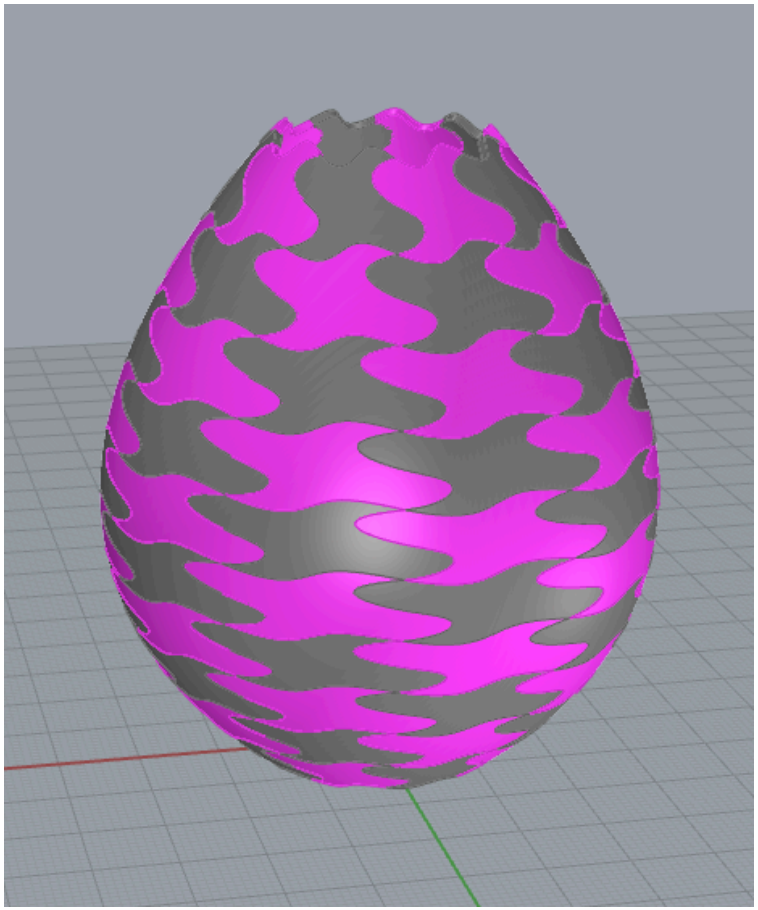
This will tell the block that U, V, and Z are percentages instead of absolute values.

Save your Grasshopper file
in case the next step crashes Rhino.

Click the Play button on your Data Dam to trigger Surface Morph



Add Some Color



questions?

Thank you!

CS 491 and 591
Professor: Leah Buechley
https://handandmachine.cs.unm.edu/classes/Computational_Fabrication_Spring2021/