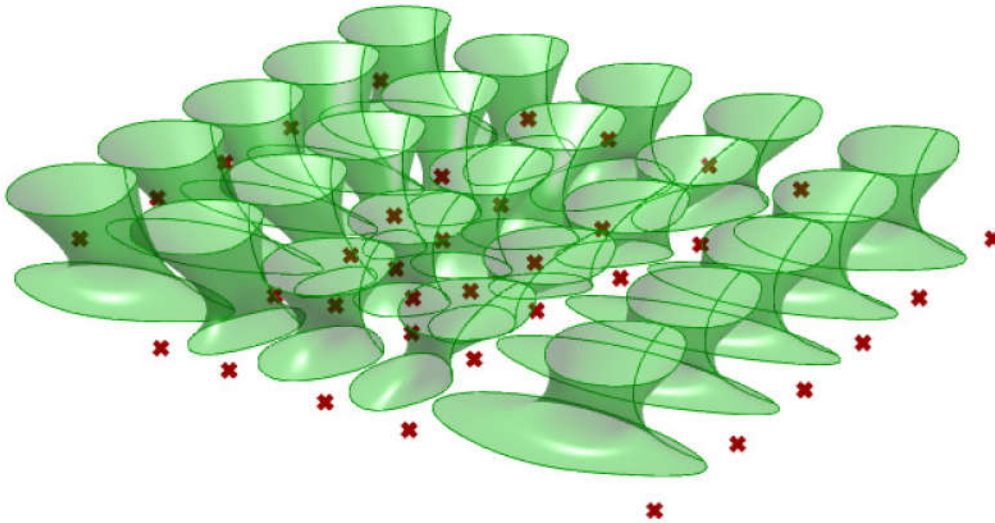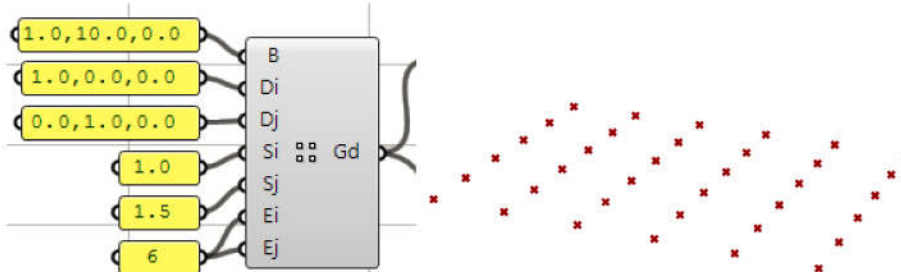## Loft Morphed Curves

This tutorial shows the following:

- How to create attracted grid.

- How to morph module curves in 3D space.

- How to create 3D modules from morphed curves.

Start the GH definition with creating a rectangular grid using ptPanar component.

B *(base point) = (1.0,10.0,0)*, **Di** *(row dir) = (1.0,0.0,0.0)*, **Dj** *(col dir) = (0.0,1.0,0.0)*, **Si** *(row spacing) = 1.0*, **Sj** *(col spacing) = 1.5*, **Ei** *(row number) = 6*, **Ej** *(col number) = 6.*

Next add an attractor point and change grid points to attract towards the attractor point.
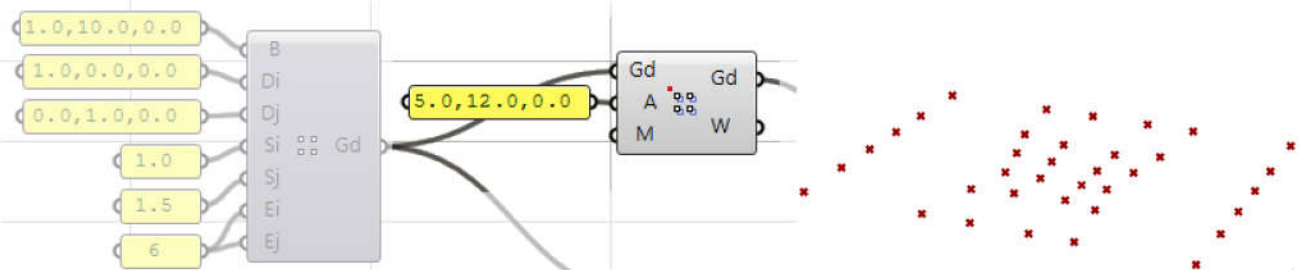
Gd *= grid to be attracted*, **A** *(attractor point) = (5.0,12.0,0.0)*, **M** *(Magnitude of attraction) = 1 (default)*, **Gd** *(output) = attracted grid*, **W** *= weights grid (attraction degree for each grid point 0-1).*

We need a second bounding grid to populate our module in between. Copy the original planar grid in the Z direction.
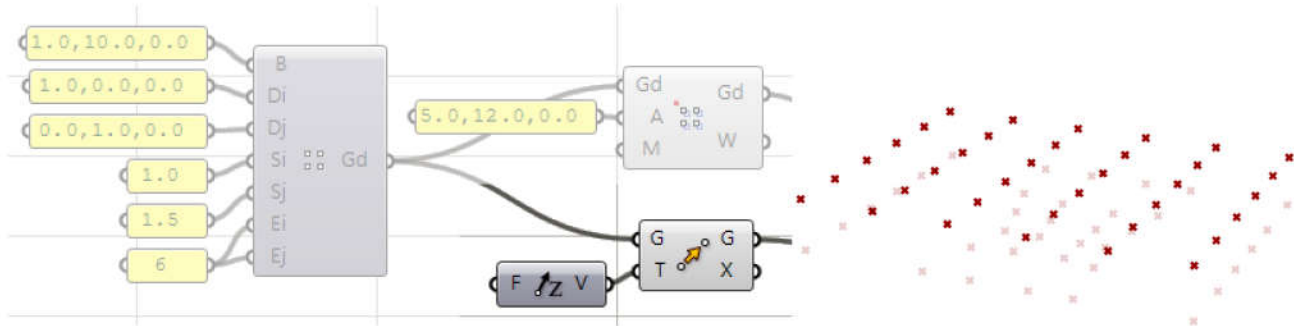

G = input geometry, T = input vector.

Next create the module curves in Rhino. In this case, we use three curves to define a loft surface. We will morph the curves rather than the lofted surface because it is faster and more efficient.
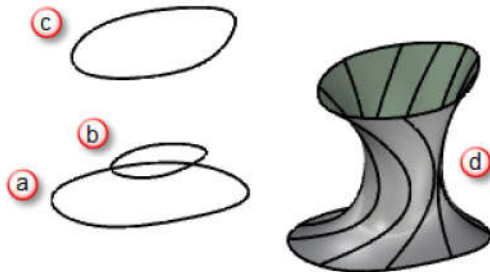
First module curve (a), second module curve (b), third module curve (c), lofted surface (d).

Reference the module curves in the next step to morph between our two bounding grids using the 3D morphing component.
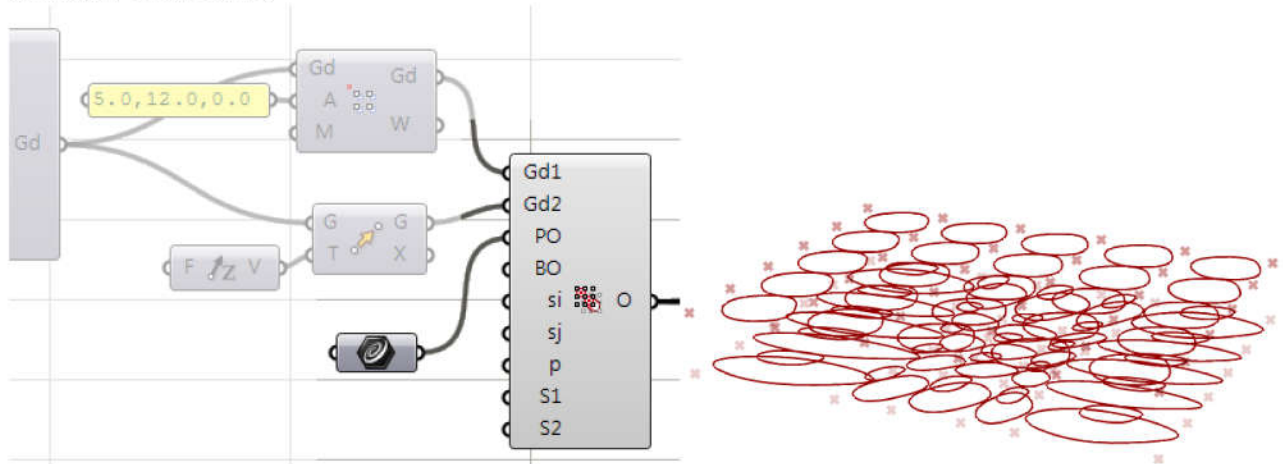

Gd1 = *first bounding grid,* **Gd2** = *second bounding grid,* **PO** = *pattern objects,* **BO** *(optional) = bounding objects for the pattern objects,* **si** = *shift in the i direction = 1 (default),* **sj** = *shift in the j direction = 1 (default),* **p** = *pull for smooth morphing = false (default),* **S1***(optional) = grid1 surface,* **S2** *(optional) = grid2 surface.*

Note that the output curves from the 3D morphing component are organized into 3 branches.
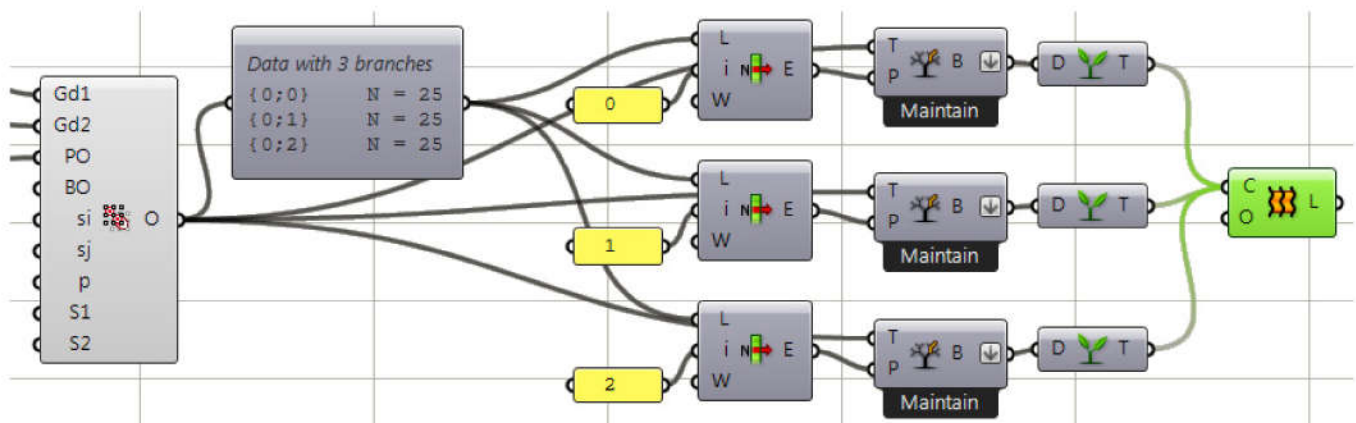
{0;0} holds morphed "a" curves.

{0;1} holds morphed "b" curves.

{0;2} holds morphed "c" curves.
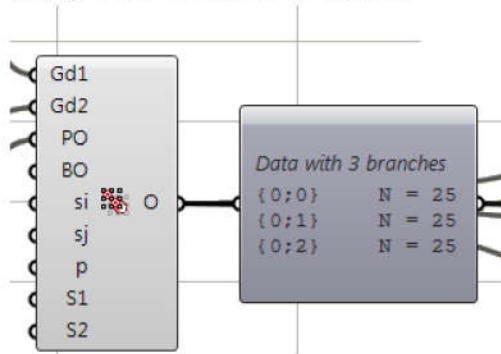
In order to loft morphed curves, we need to separate the three branches before feeding them into the GH "Loft" component. You can do that by separating the branches of the tree, then graft each branch before feeding into the GH "Loft" component as in the following:

This how the lofted modules look: