

Grasshopper ARCHICAD: Freeform Surface Grid

Description

A freeform surface grid is a standard task in parametric modeling. And a good reason to use ARCHICAD together with Grasshopper. See here some of the basics.

The forms I use are only meant as examples – please feel free to build alternate structures.

The workflow I show you centers around Grasshopper. But we use also Rhino3D and ARCHICAD. Rhino3D supplies us with editable *NURBS* curves on which we build our surface grid. (And of course it serves as viewport for our Grasshopper geometry.) ARCHICAD takes all this parametric stuff and turns it into BIM-manageable building geometry.

To achieve a similar result in Grasshopper you can also use Map to Surface, a technique <u>I extensivey</u> describe in another article of mine.

Contents

- Rhino3D: New Document, Meter Scale
- Standard Freeform Surfaces
- Surface A (Loft): Draw Curves in Rhino3D
- Reference Curves in Grasshopper
- Turn Curves into Loft Surface
- Divide Loft Surface
- Alternative Surface: Curve Network (Surface B and C)
- Loft Surface: Create Diagonal Lines
- Loft Surface: Create Panels
- Start ARCHICAD and connnect to Grasshopper
- Grasshopper Diagonals to ARCHICAD Beams
- Edit ARCHICAD Beams
- Grasshopper Panels to ARCHICAD Shells
- Edit ARCHICAD Shells, Part I
- Correct Panel Geometry in Grasshopper

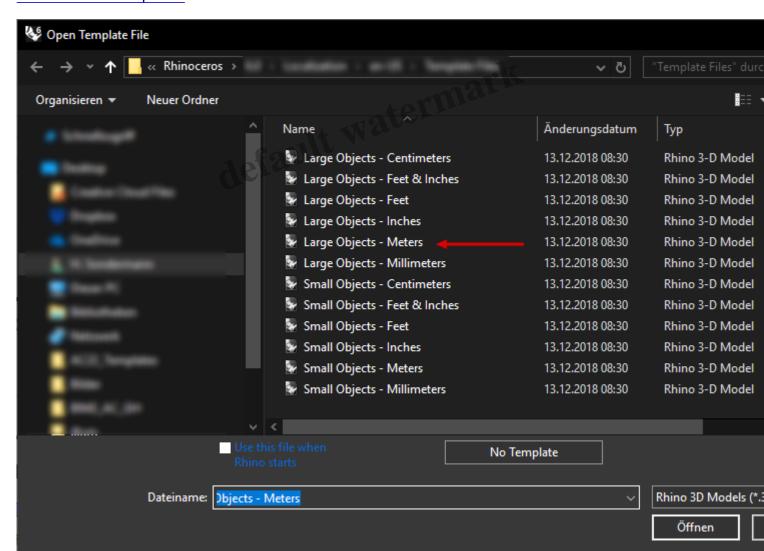
- Edit ARCHICAD Shells, Part II
- Add Border Frame Beams
- Make ARCHICAD Geometry Standalone
- Parametric Playground
- Roundup and Links

Now let's start.

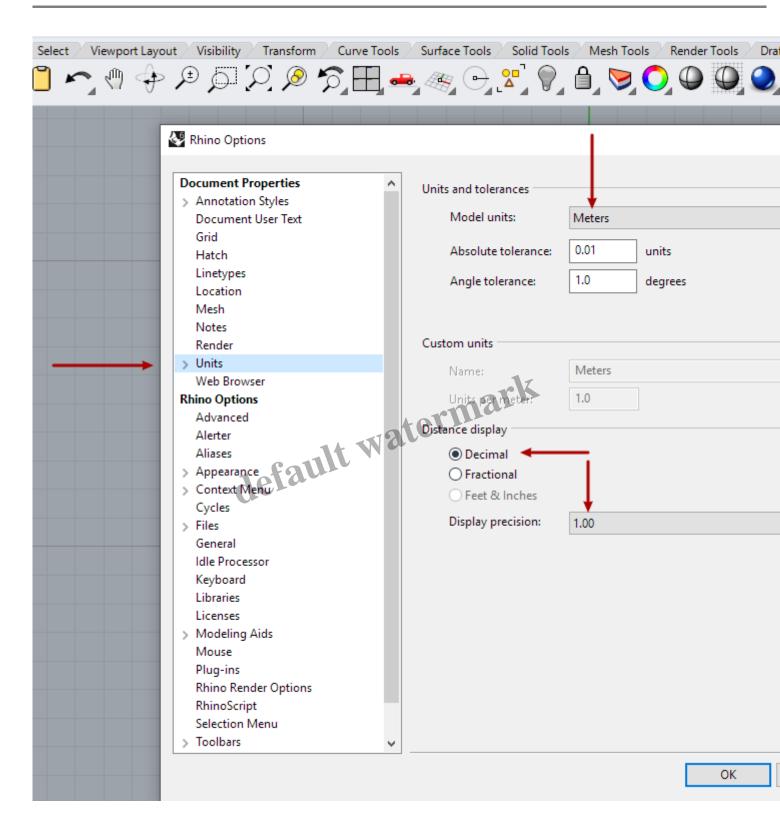
Rhino3D: New Document, Meter Scale <

Open Rhino3D and form there, open a new document.

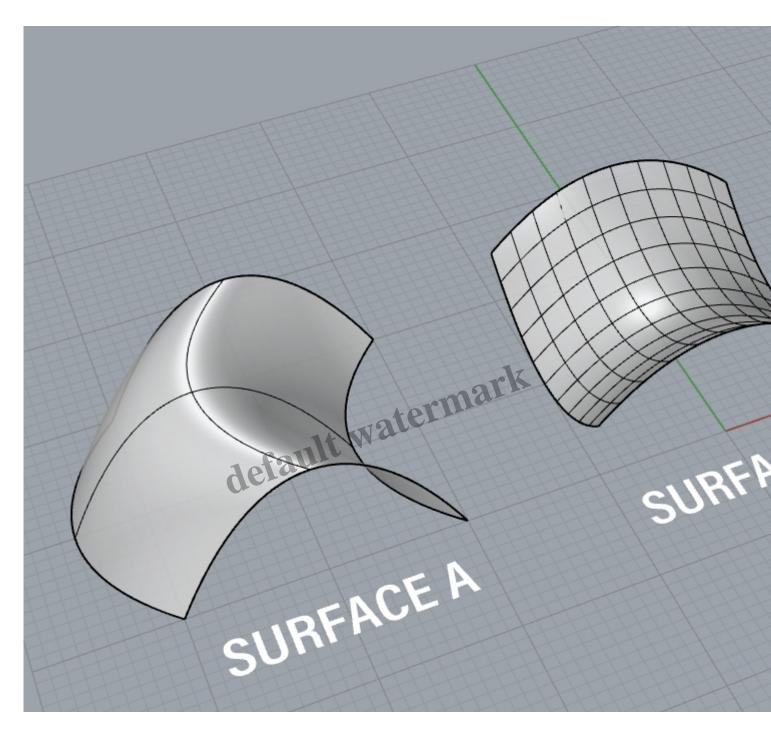
Beware of the *units!* Definitely check out you are working in *meter* units and use the right *grid.* The best way to secure this is to choose a *template* named *Large Objects – Meters*. See also my article on Rhino default templates on this matter.



In Rhino3D options you can check if units and grid fit your needs:



Standard Freeform Surfaces <

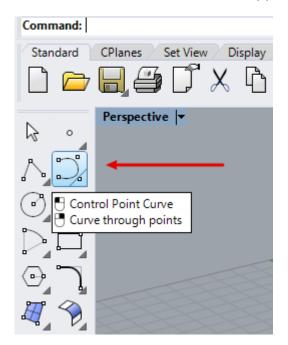


Above you see 3 common freeform surfaces. Surface *A* is constructed via *Loft* on 3 curves. Surfaces *B* and *C* are made with *Curve Networks* based on 4 border curves.

In the following I will explain how to construct surface A. I am also giving hints on how to construct surfaces B and C. Anyway, feel free to make your own choices while following along.

Surface A (Loft): Draw Curves in Rhino3D <

To construct surface A in Grasshopper we first need to draw some NURBS curves in Rhino3D:



Just to keep things a bit more complex, I use 3 curves that are not parallel:

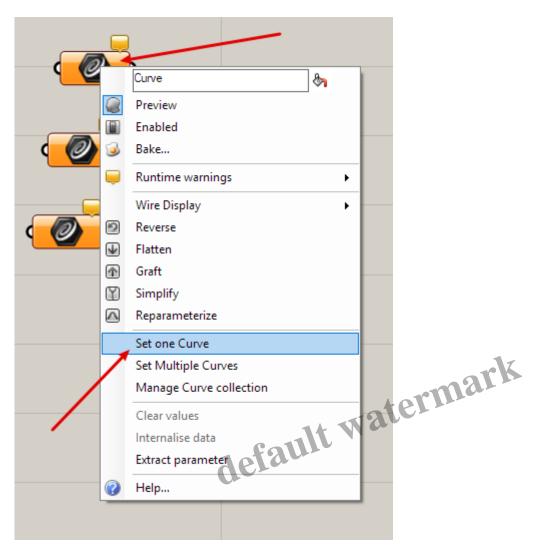
default watermark

Reference Curves in Grasshopper <

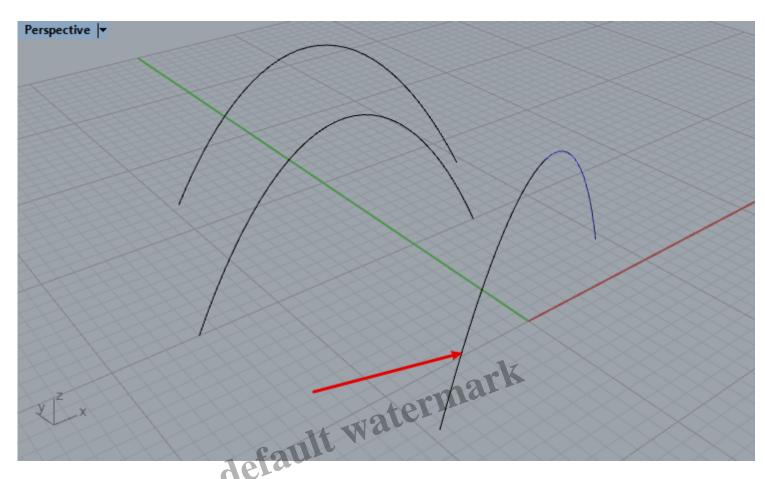
To start in Grasshopper we need 3 *Curve Input* components. (We could also use just 1, but this way things are easier to handle.)

Grasshopper - ARCHICAD_Grasshopper_LoftSrf_DiagonalBeams_1218*





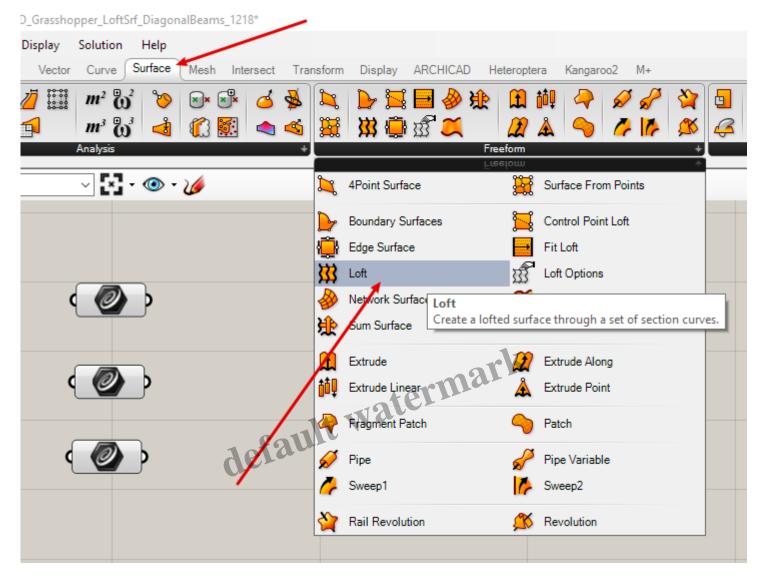
Now right-click every component, choose *Set one Curve* and click on one of the Rhino3D curves. (If you chose to use only 1 curve input, you have to *Set Multiple Curves*.)



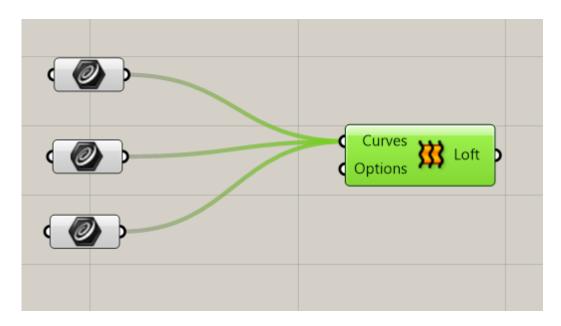
Now Grasshopper has your 3 curves referenced. Which means that you can change these curves howand whenever you like and Grasshopper will process the respective changes.

Turn Curves into Loft Surface <

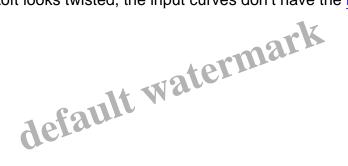
Now it's time to build the surface. For surface A (which we deal with) a Loft is the right component:

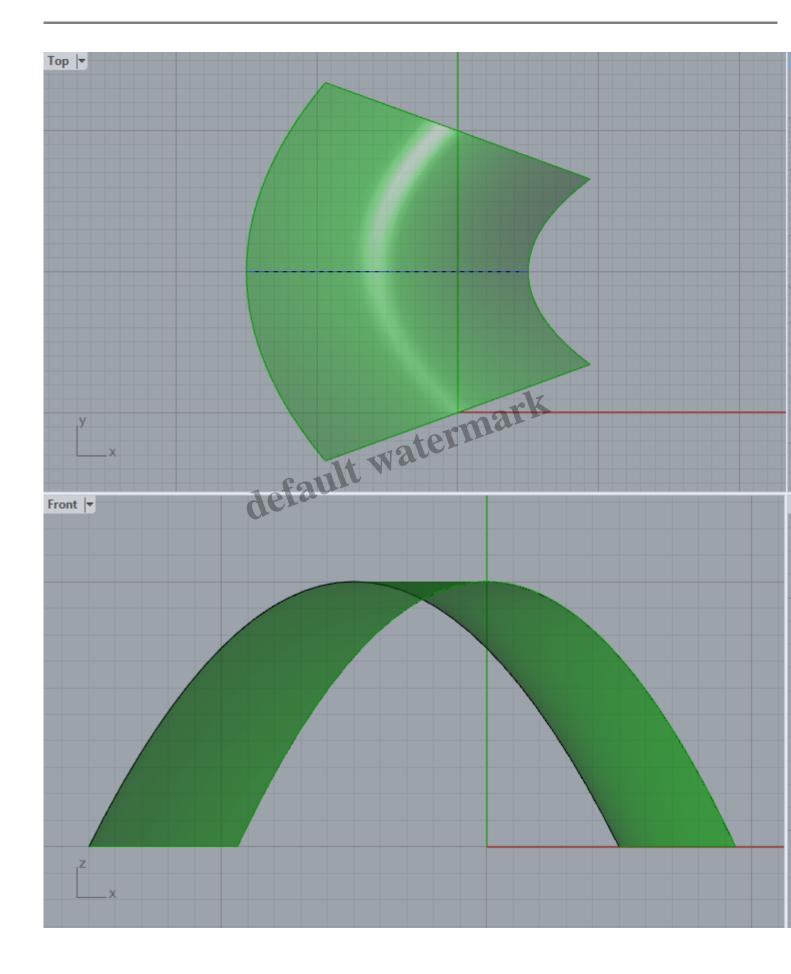


Connect the 3 *Curve* inputs to Loft's *Curves* input. Press *Shift* so you can tune *several* streams into 1 input.

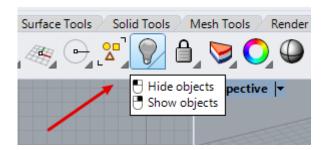


This is the result. If your Loft looks twisted, the input curves don't have the <u>right order</u>.

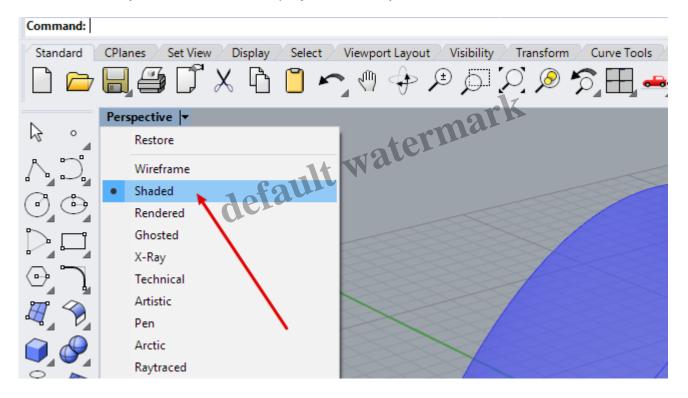




By the way – the Rhino3D viewport shows both your *original Rhino curves* and the *new Loft* produced by Grasshopper. You can hide the original curves via *Hide Objects* to keep things tidy:

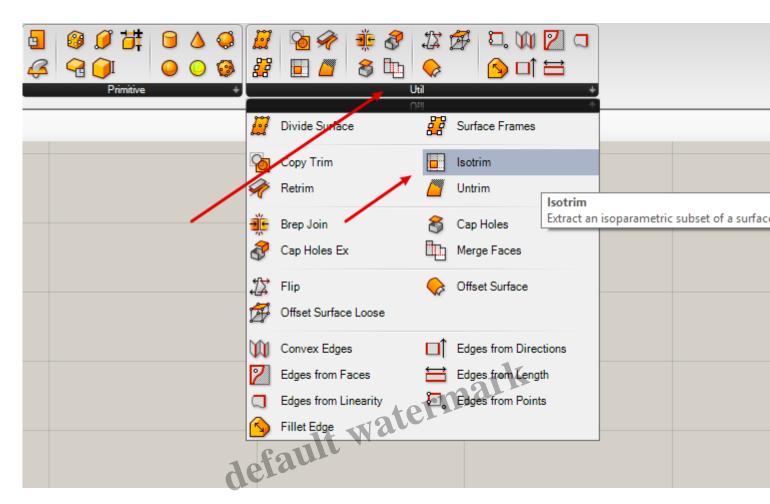


Also, be aware you can choose a display mode that you like:

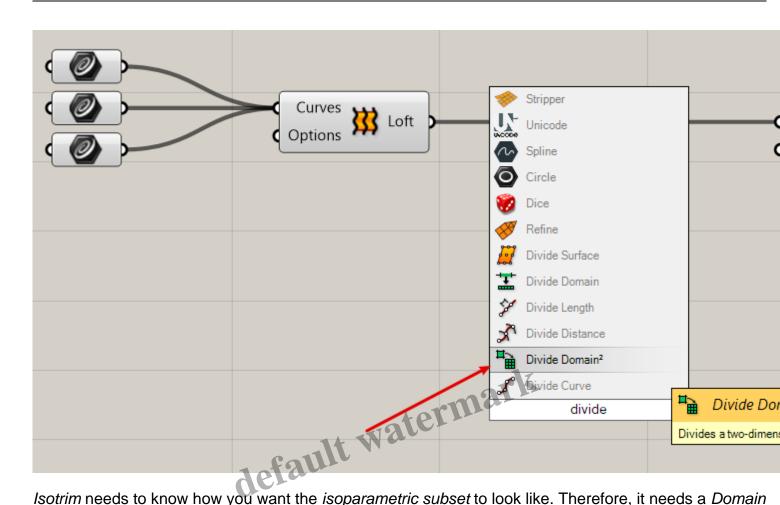


Divide Loft Surface <

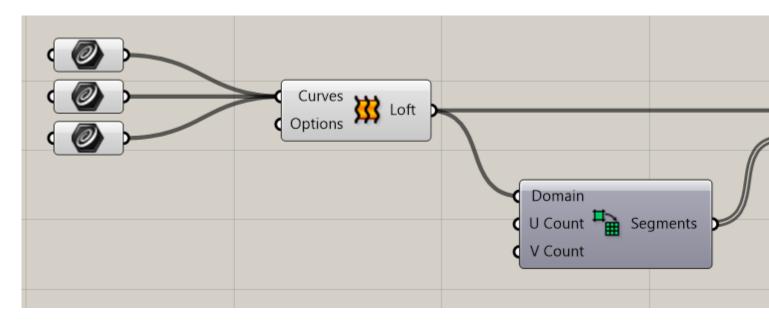
Up to now, your Loft is only an abstract piece of geometry. To add a structural grid you need to subdivide it. Follow along, this is standard procedure. First of all you will need an *Isotrim* component:



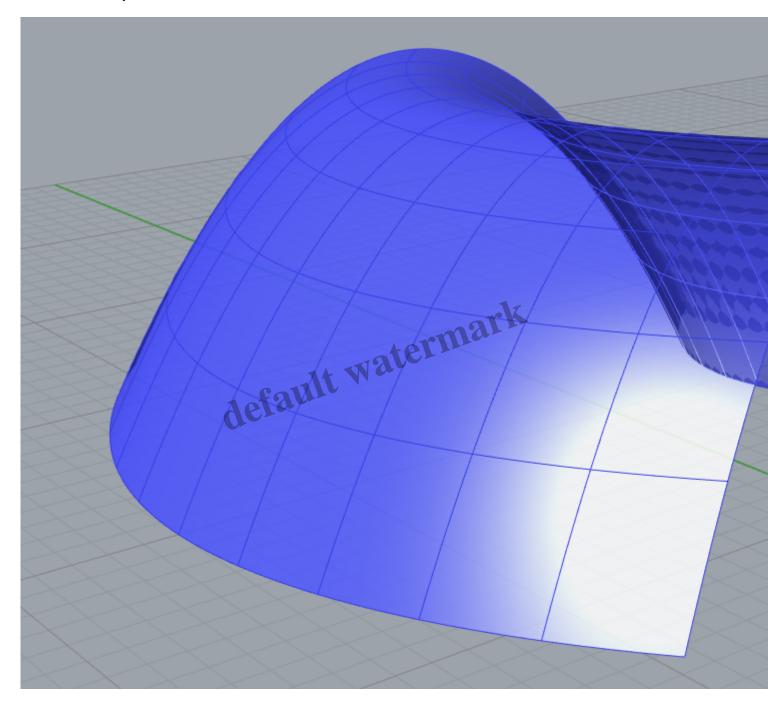
Connect your Loft output to the new component's Surface input:



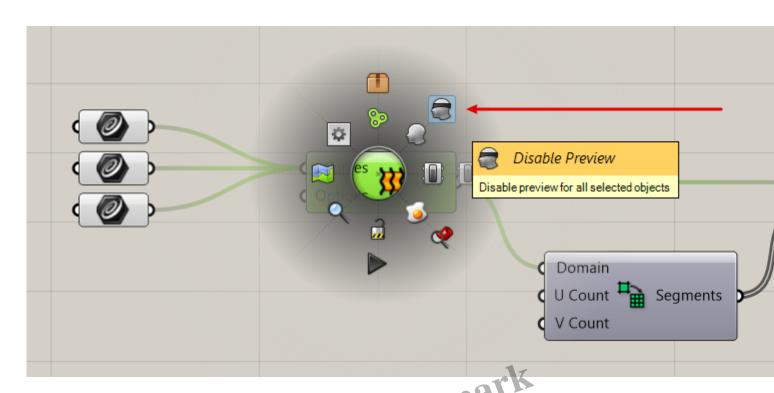
Isotrim needs to know how you want the isoparametric subset to look like. Therefore, it needs a Domain input – basically, information about the range and scale of the desired subdivision. So pick a Divide Domain2 component. Connect its Domain input to your Loft output. Connect its Segments output to the Isotrim's Domain input:



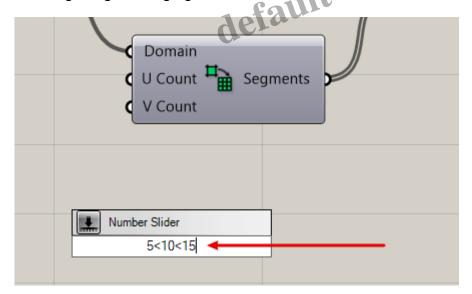
By default, the subdivision in U and V direction is set to 10. So this is what you see, a 10 \times 10 subdivision of your Loft:



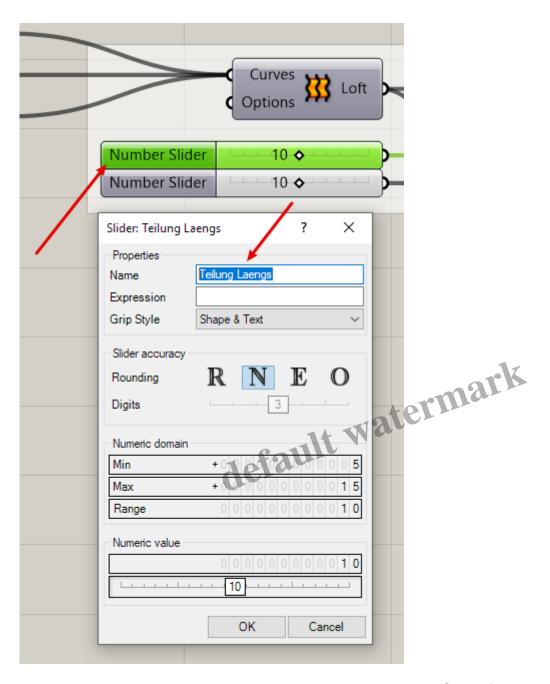
And again, Rhino's viewport shows double geometry. Select the Loft component, press the middle mouse button and choose *Disable Preview:*



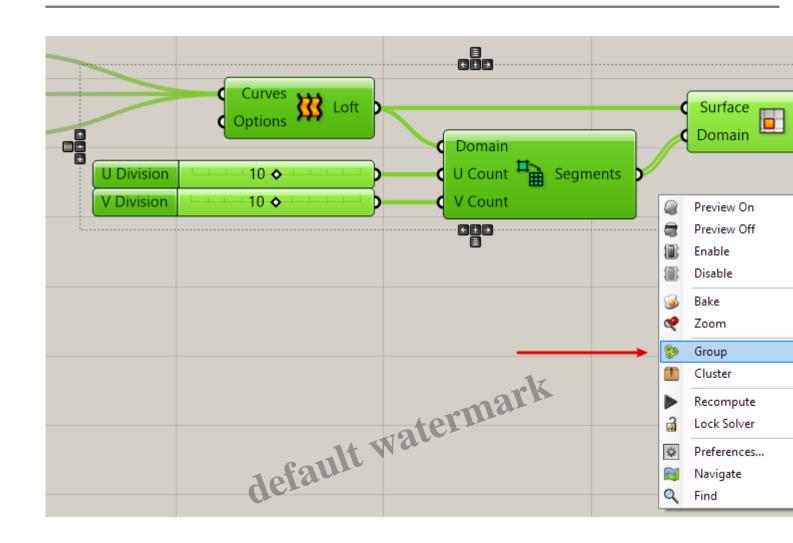
Now all you see is the *Isotrim* instance of your Loft. To gain more choices concerning the U and V division, double-click on the Grasshopper canvas and write *5*<*10*<*15*. This will produce a *Number Slider* showing *integers* ranging from *5* to *15* with *10* as default value.

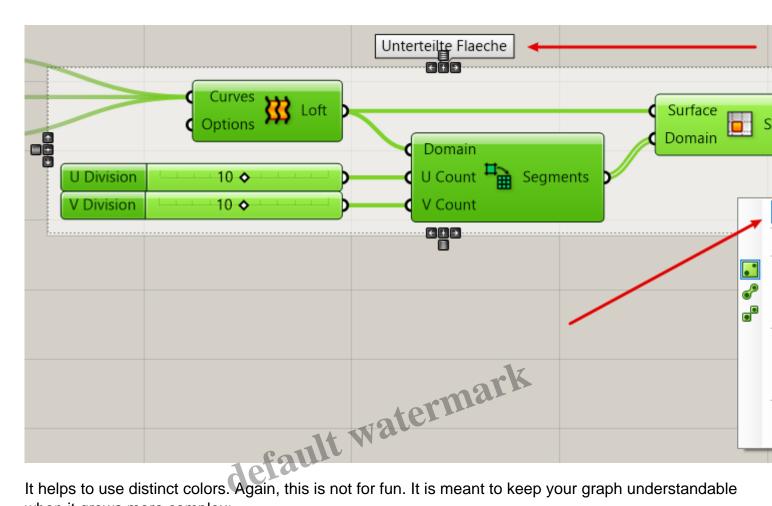


Make a copy of the *Number Slider*. Connect both sliders to the *U* and *V* inputs of the *Divide Domain* component. Rename the sliders by double-clicking their names. Choose names that help you understand your Grasshopper definition later on.

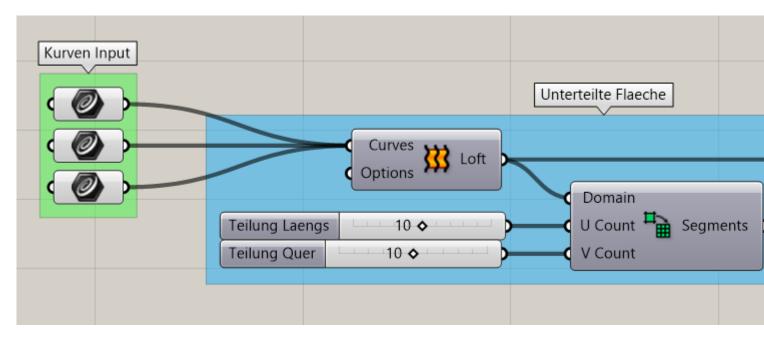


To add even more readability you should always utilize the *Group* feature:





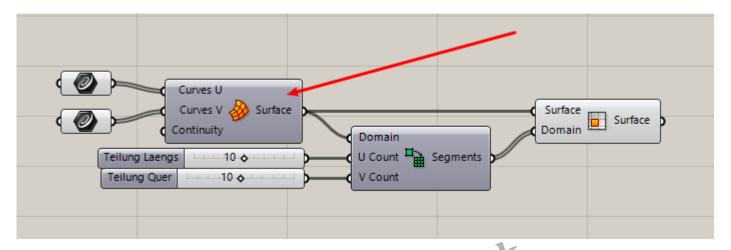
when it grows more complex:



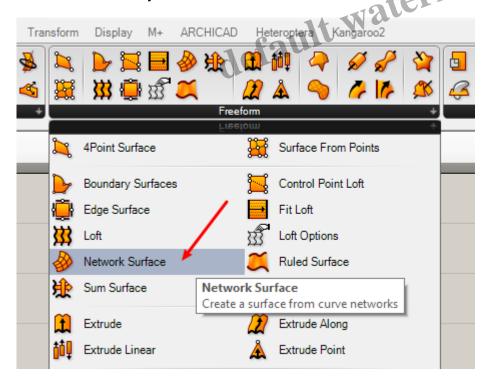
Alternative Surface: Curve Network (Surface B and C) <

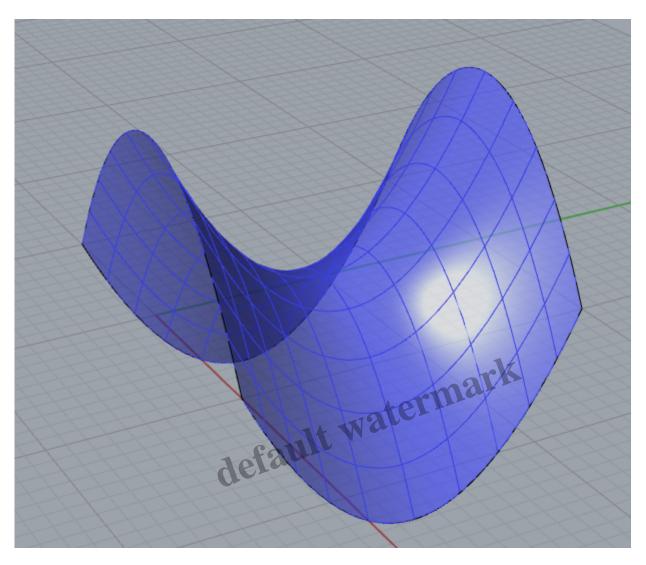
As I said above, a Loft surface is only one way to produce a freeform surface. My examples B and C base on 4 curves which define the surface's border contour in a different way.

The according Grasshopper component to use here is *Network Surface*. The 2 *Curve Input* components each contain 2 curves:



As you see, only the first part is different – the surface generation. Subdividing the resulting surface is done the same way as for the Loft above.

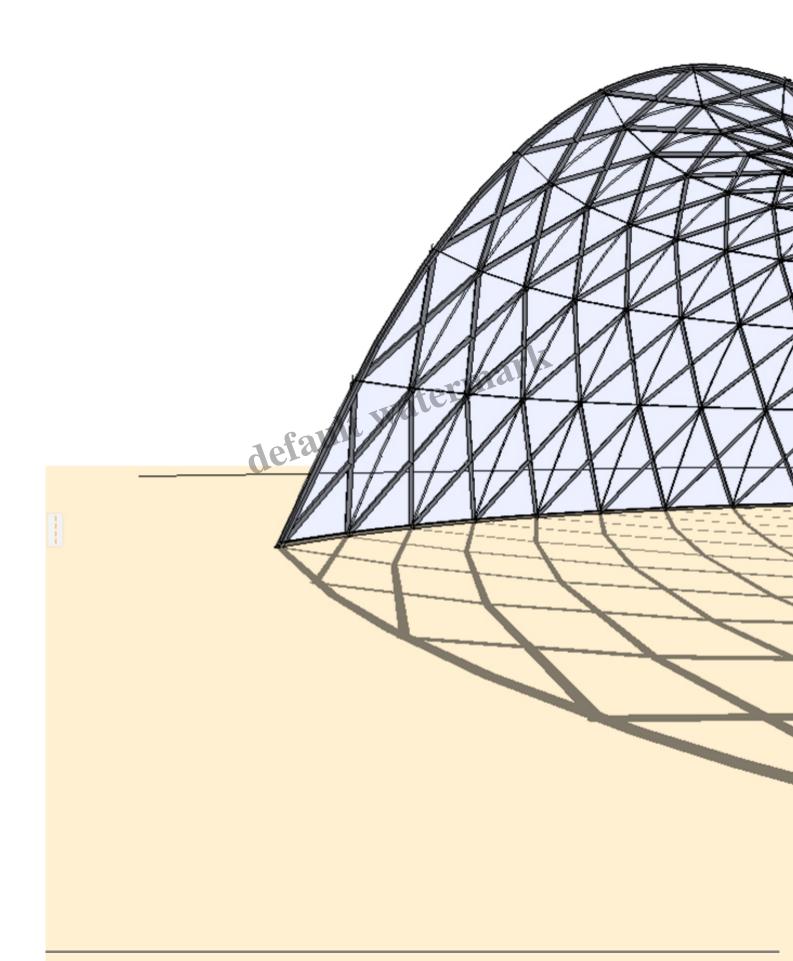




So if you want to follow this path go along. For the rest of us, let's turn back to our Loft.

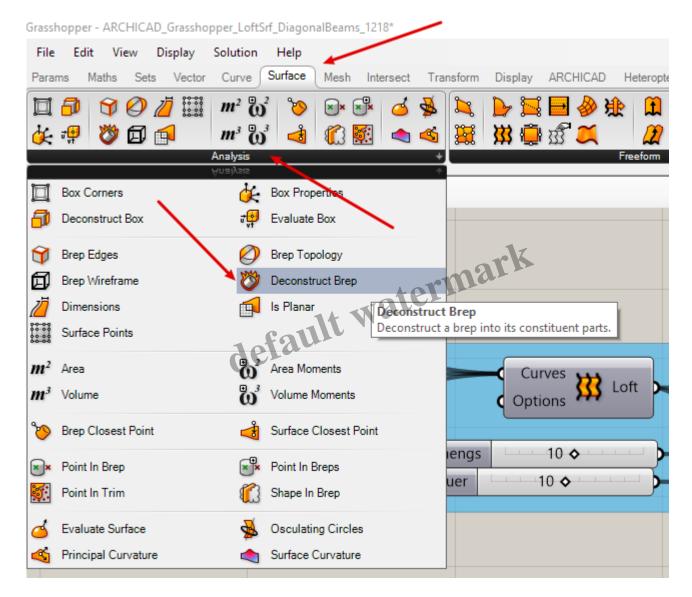
Loft Surface: Create Diagonal Lines ≤

default watermark

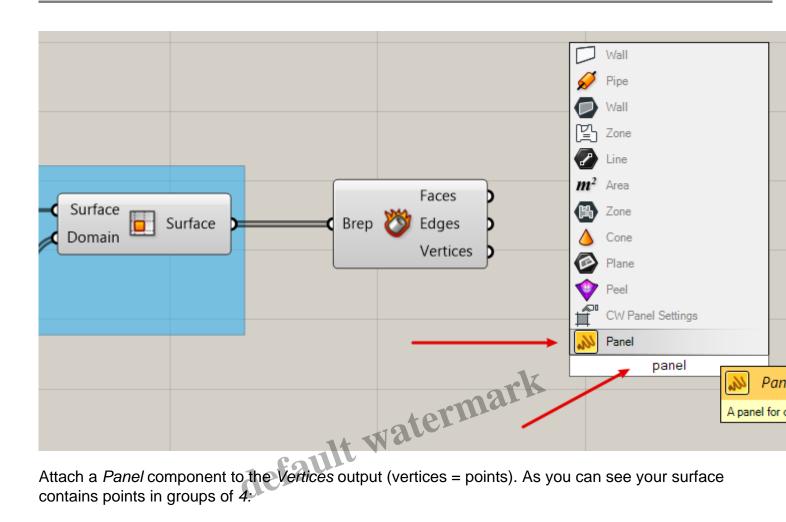


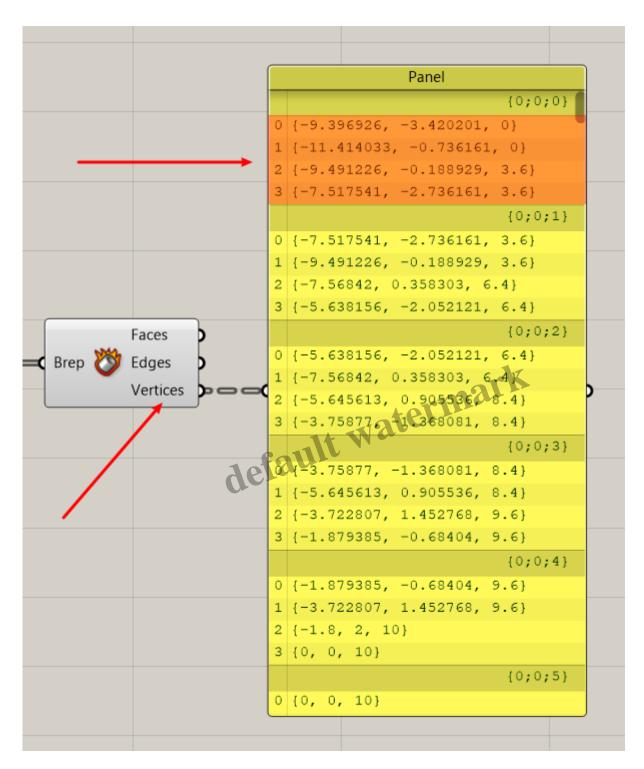
As can be seen in this image, one of the structure's basic features is it's diagonal grid. For Grasshopper, this is only a set of lines.

To produce these lines we need points on our surface. These points have to be retrieved in the first place. One common way to do this is to use the *Deconstruct Brep* component:

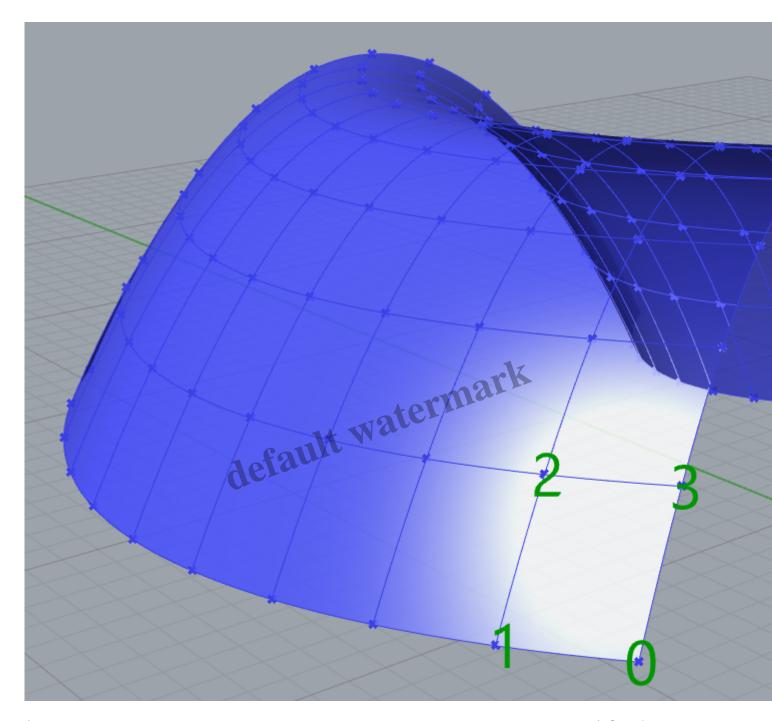


Connect it to the *Isotrim* component like this:

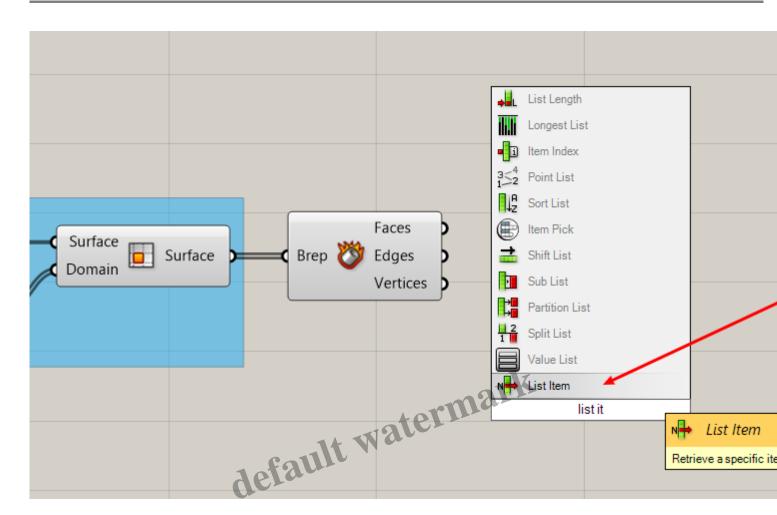




Obviously, those groups of 4 points build the subdivison rectangles on your surface. Each group has points named 0, 1, 2 and 3:



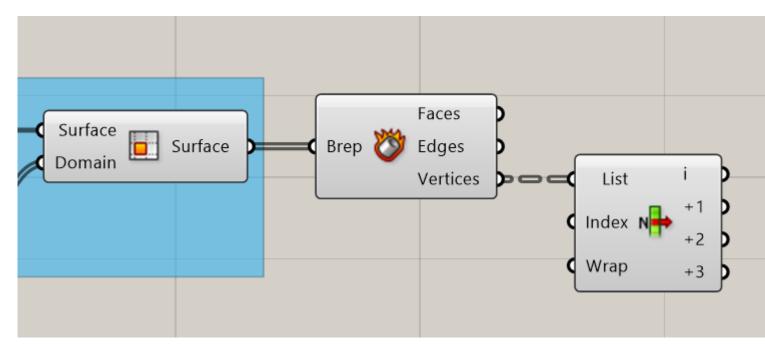
(I used a special component to display those point numbers, in case you wonder.) So, if you need diagonal lines, you want to connect points 0 and 2, and 1 to 3 respectively.



To retrieve points from a list you use the *List Item* component. Produce one, zoom into it and press the *Plus* sign 3 times.

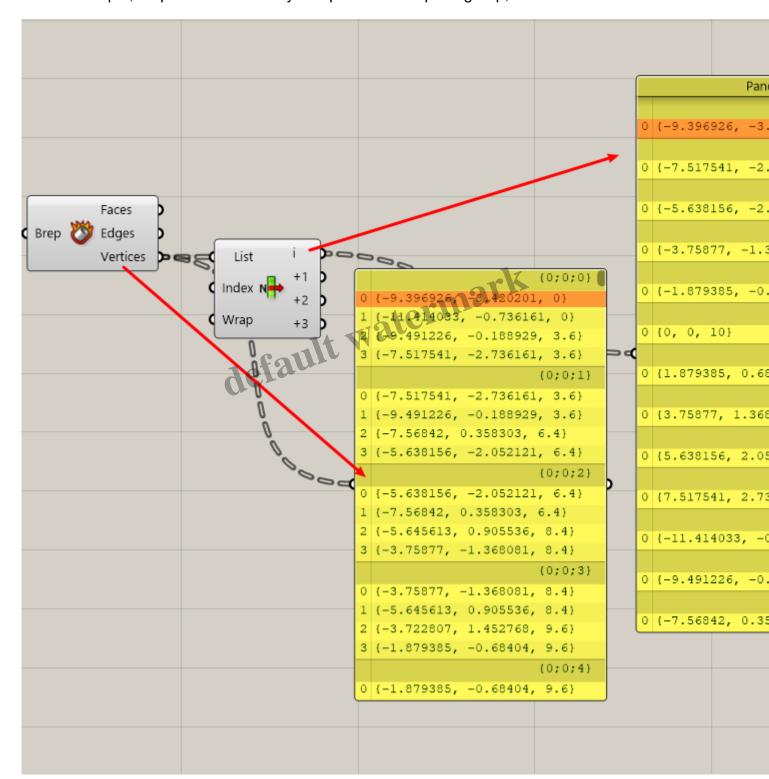


Now connect your List Item component to the Deconstruct Brep Vertices output as shown:

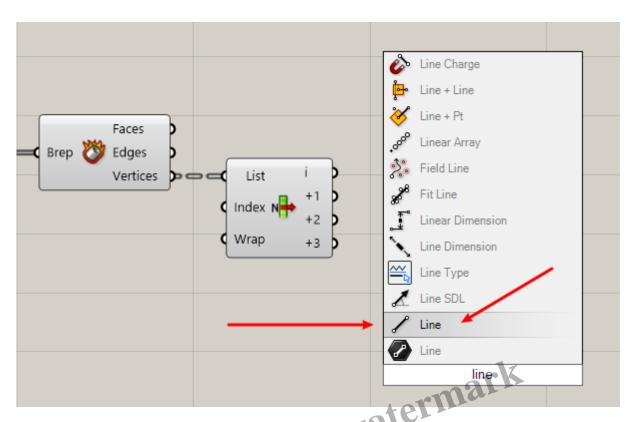


As you see, your *List Item* component has 4 outputs: i = 1st list item, +1 = 2nd list item, +2 = 3rd list item, +3 = 4th list item.

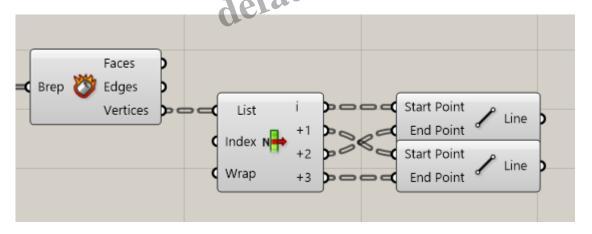
So for example, output *i* extracts every first point in each point group, like so:



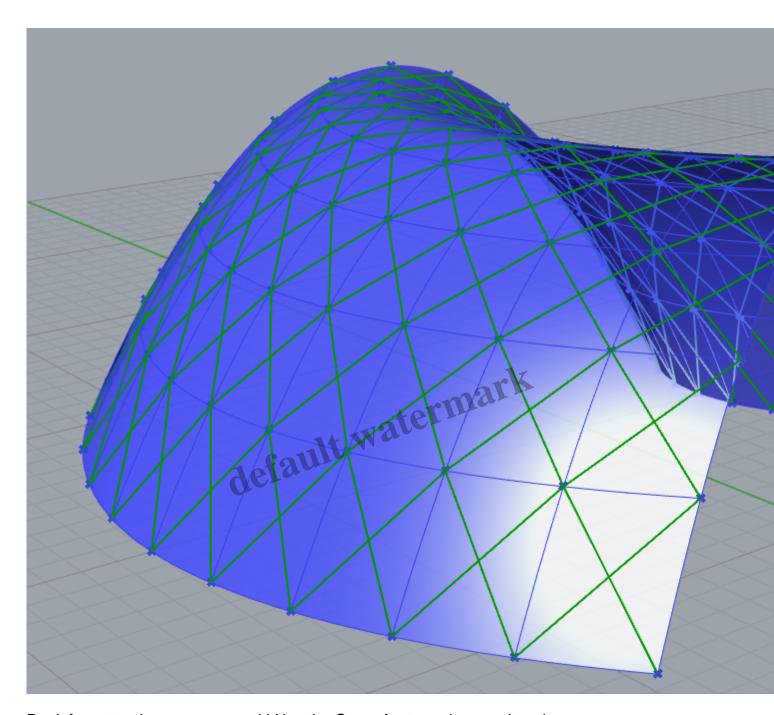
Produce two *Line* components:



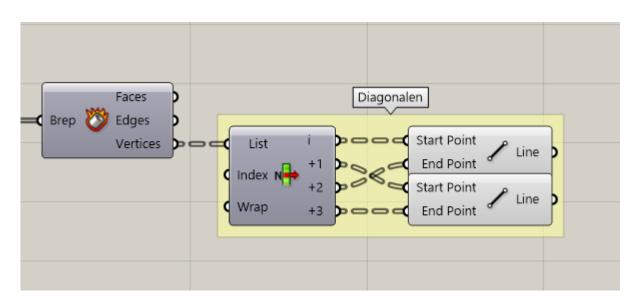
Connect the first Line component to List Item outputs i and +2. Connect the second Line component to List Item outputs +1 and +3:



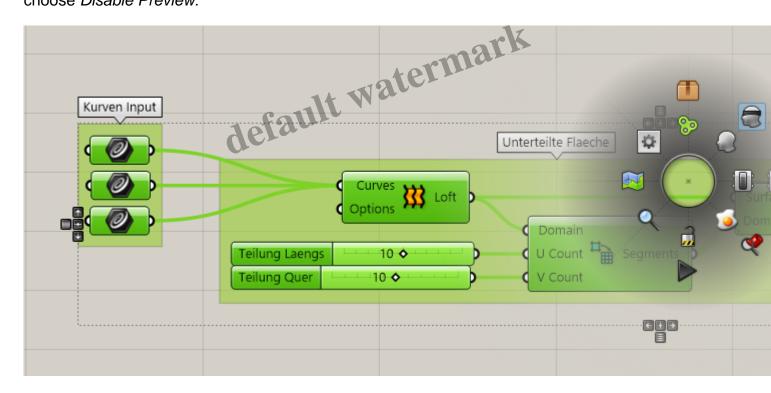
Here you are - the diagonals show up:



Don't forget to clean up your work! Use the Group feature wherever it makes sense:



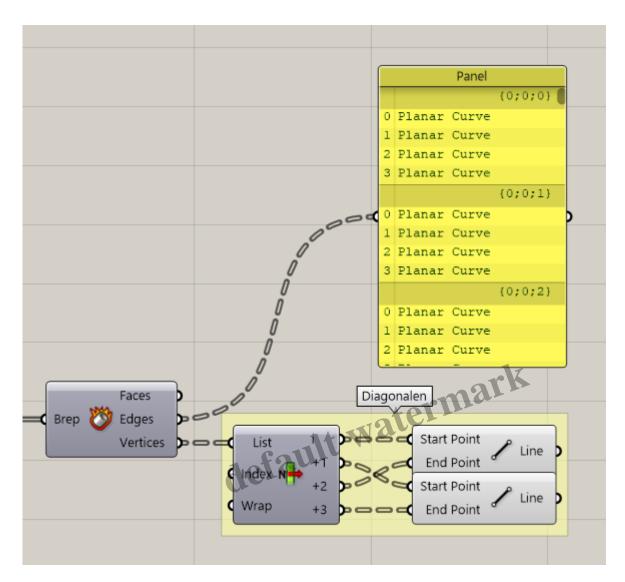
On your canvas, select everything but the last 2 Line components. Press the middle mouse button and choose *Disable Preview:*



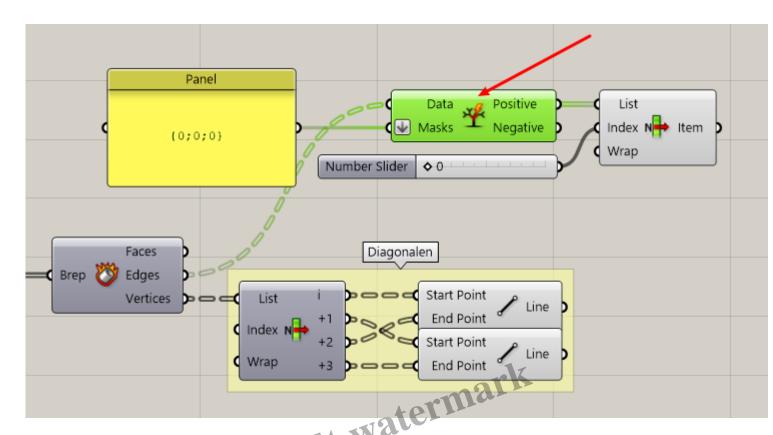


Loft Surface: Create Panels <

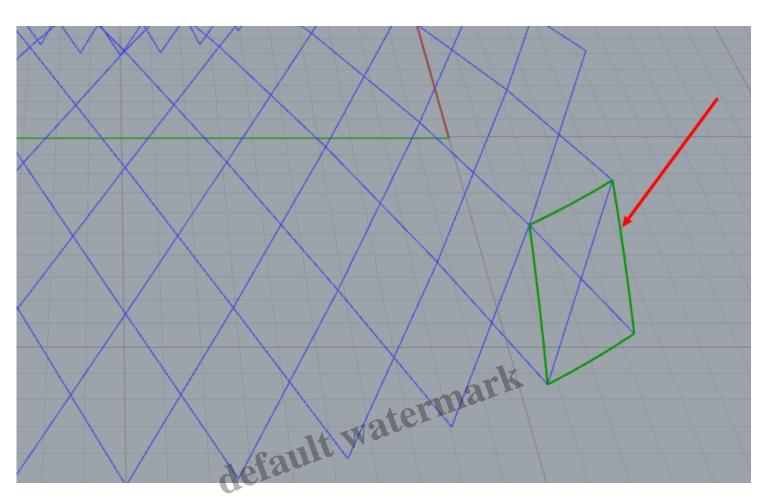
Now for the panels. They won't follow the diagonals, but will fill the U-V-pattern you created with *Isotrim*.



For the roof panels we will utilize the *edges* that were produced by the *U-V* subdivison. When you connect a *Panel* component to the *Deconstruct Brep Edges* output you see again groups of *4.* Only this time it's edges, not points, that generate the U-V rectangles.



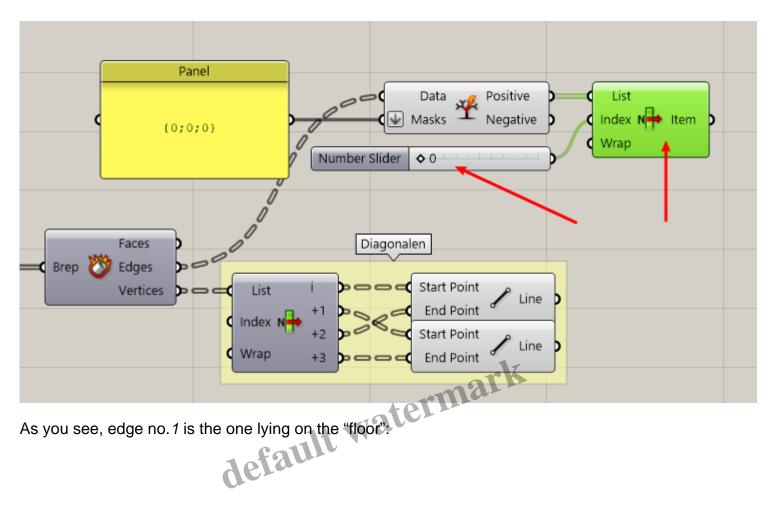
By using a small sub-definition in Grasshopper I can show you exactly what this looks like. (Don't do this yourself – this is only for demonstration.)



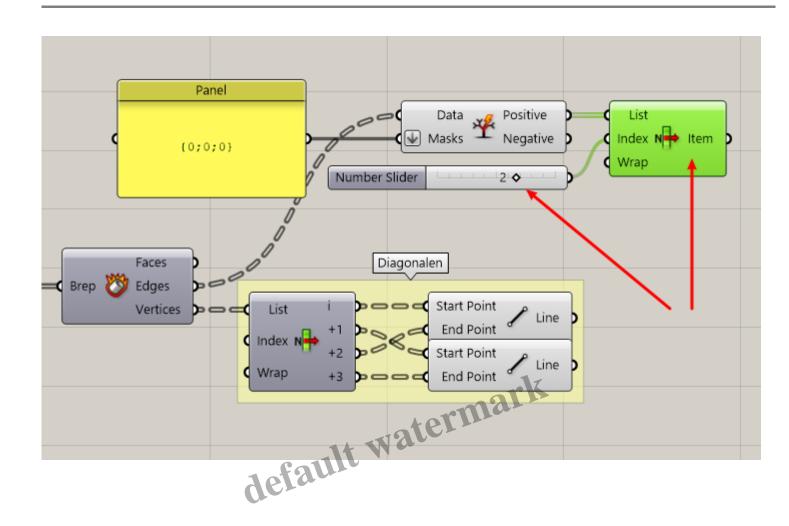
Above you see one of the edge groups. Below you see my small sub-definition I built for demonstration purposes.

Basically, it extracts one edge group and then allows me to pick one of the 4 edges inside this group – in this example edge no.0.

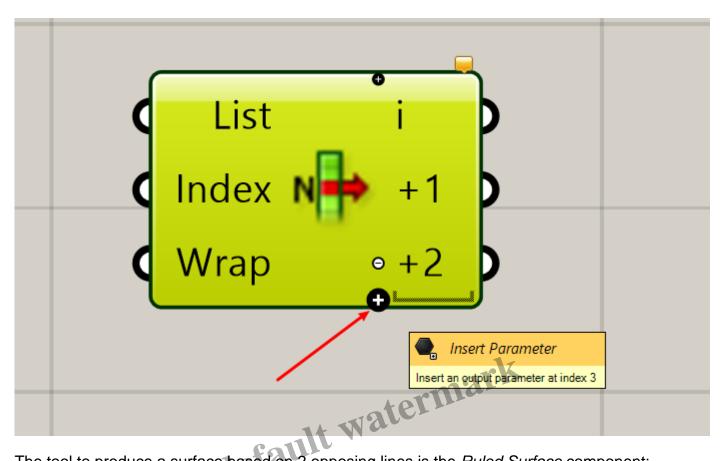
Again: You don't need to do this yourself:



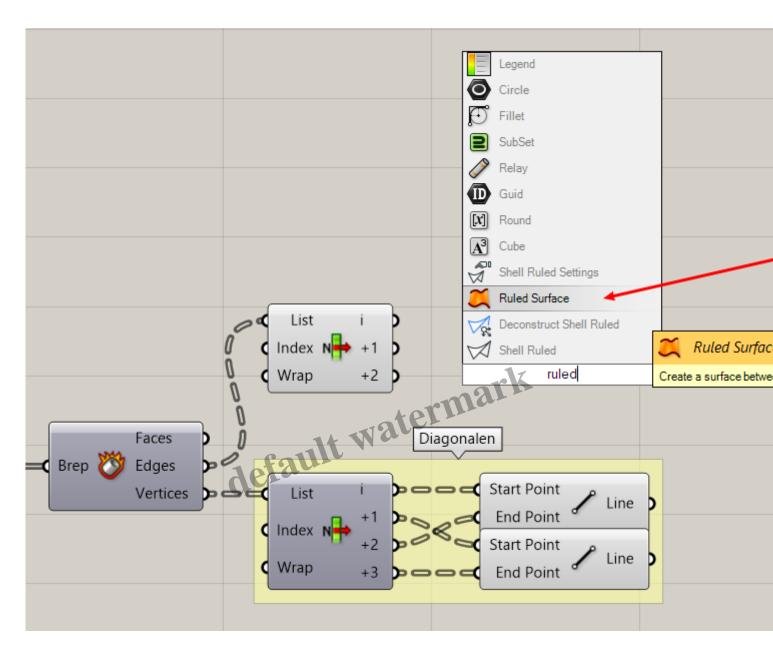
And edge no.2 is the one on the opposite side:



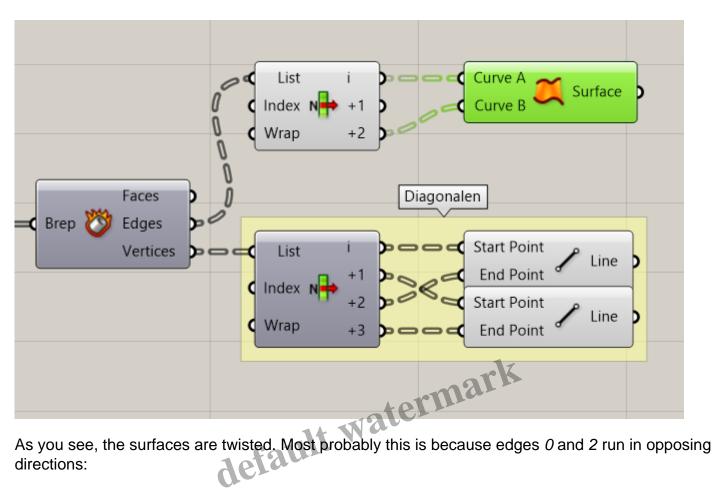
So again, for generating surfaces you will want to connect only some members of the edge groups. Let's try to connect edge 0 and 2. It makes perfect sense to use a *List Item* component again – only this time 3 outputs will be enough:

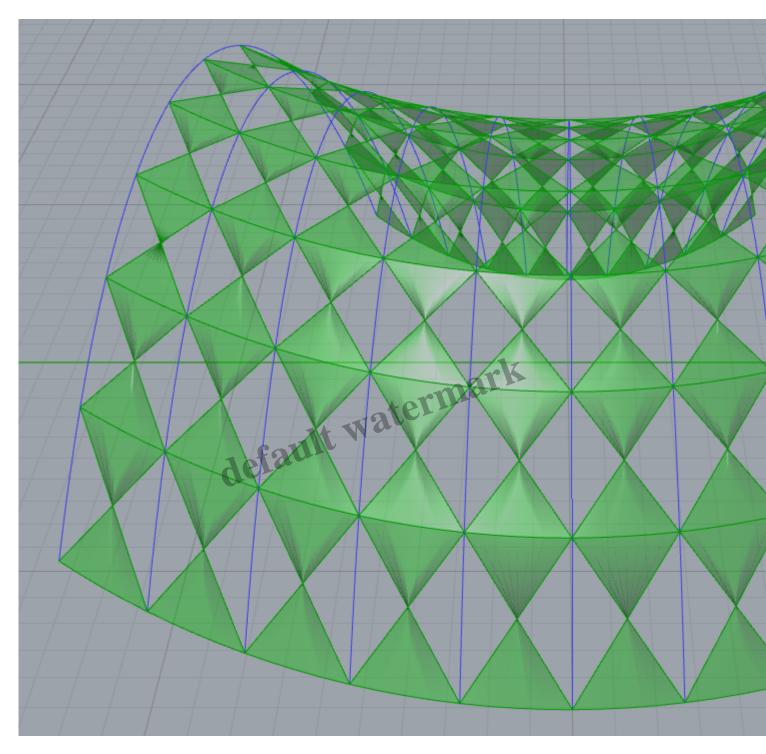


The tool to produce a surface based on 2 opposing lines is the *Ruled Surface* component:

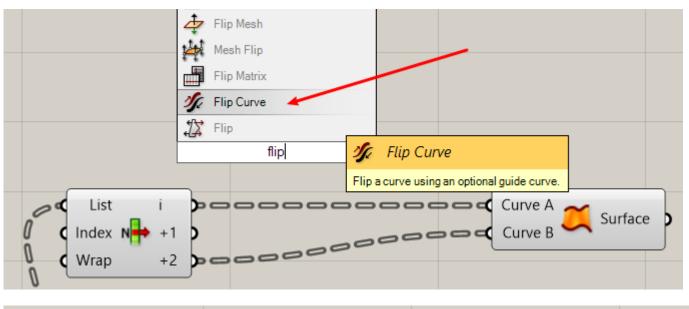


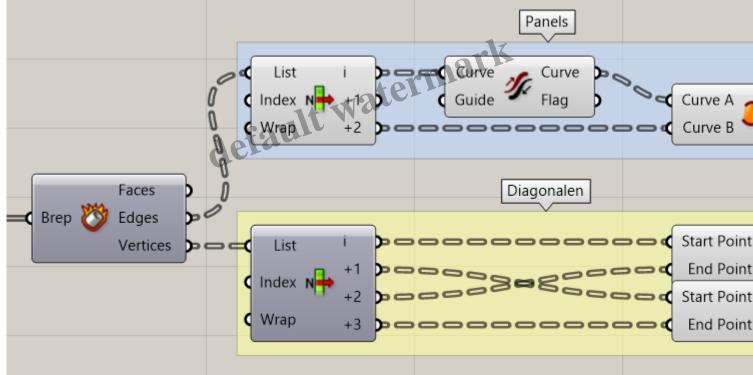
Connect it to the List Item outputs for edge no.0 and 2 of each edge group like this:



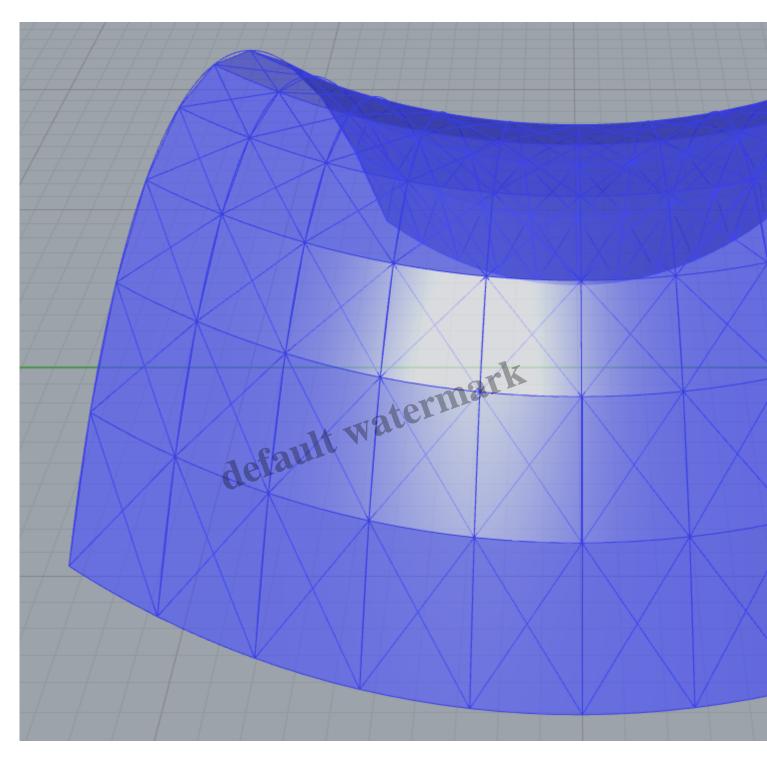


In this case, the Flip Curve component helps turning one of the two edges around:





Now things looks alright:

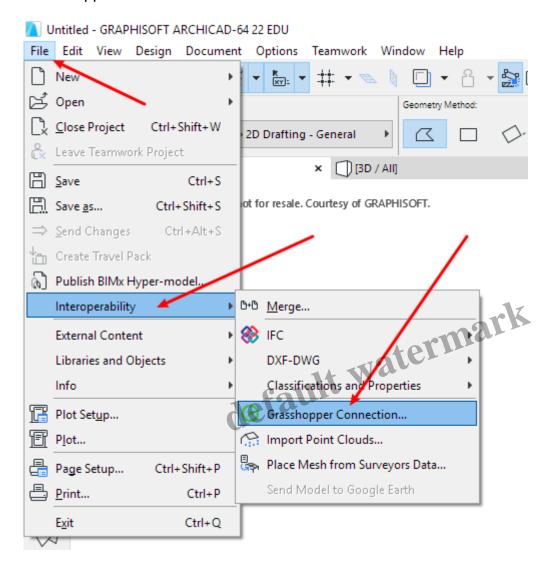


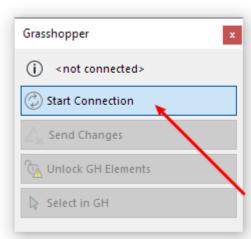
As said in the beginning, there is an alternative to achieve a similar result – <u>read this article to learn</u> more about it.

Start ARCHICAD and connnect to Grasshopper ≤

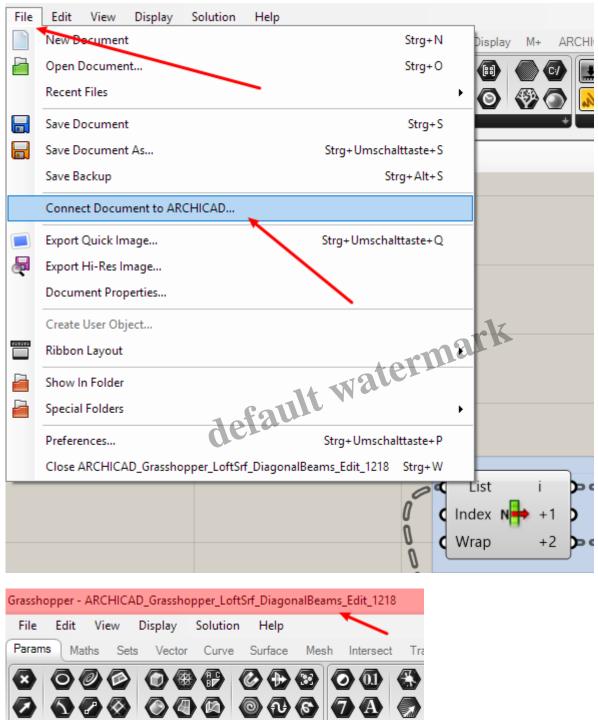
What we have achieved so far can be considered a geometric skeleton. To add some real building substance we will now turn to ARCHICAD.

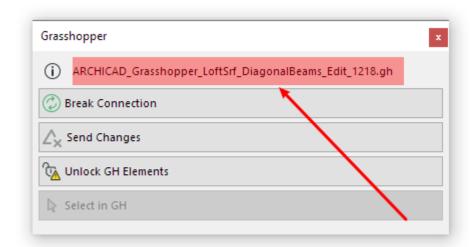
First of all, start ARCHICAD, connect it to Grasshopper and vice versa. Be sure the correct Grasshopper file is referenced in ARCHICAD:





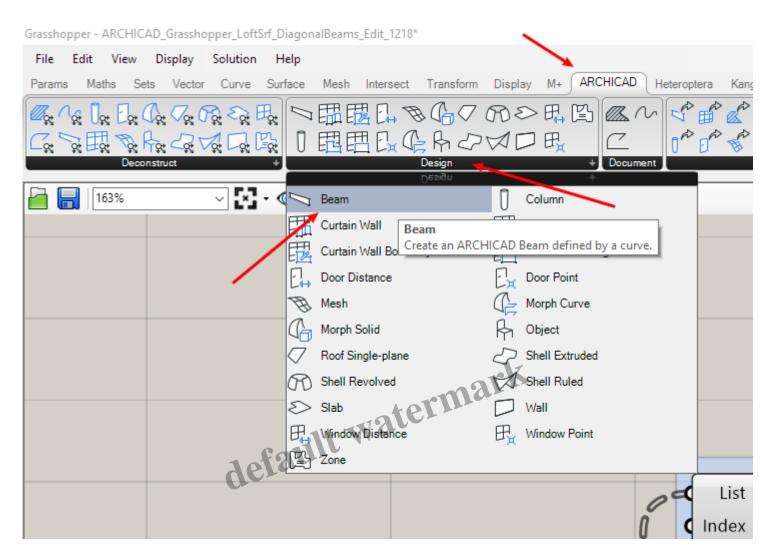
Grasshopper - ARCHICAD_Grasshopper_LoftSrf_DiagonalBeams_Edit_1218



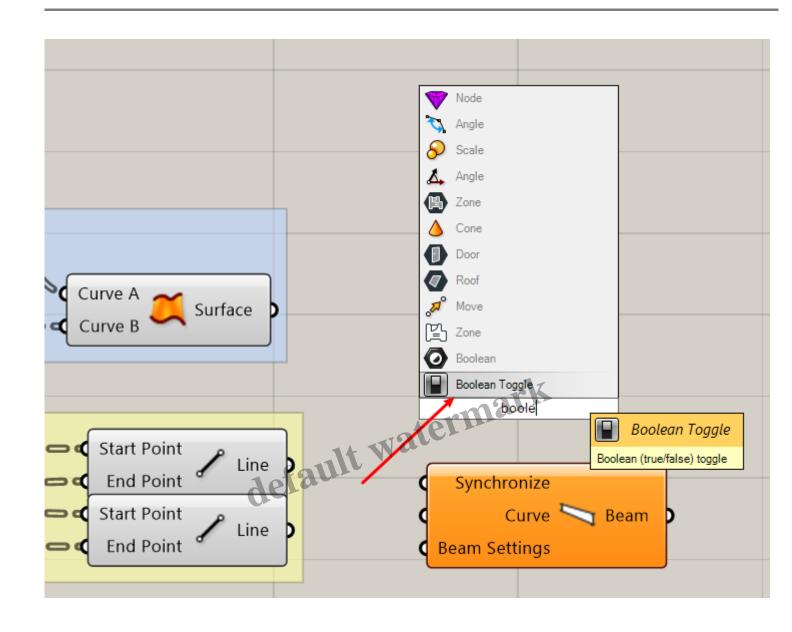


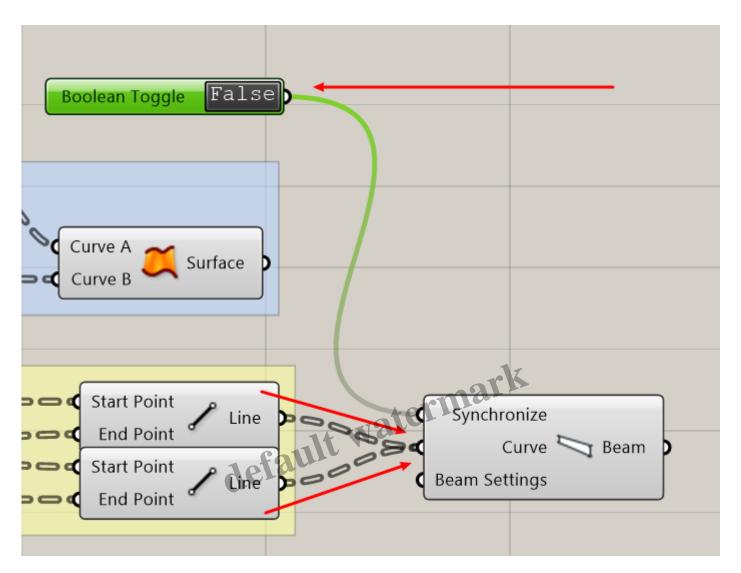
Grasshopper Diagonals to ARCHICAD Beams <

To fill the Loft's diagonal grid with 3D substance you may want to use ARCHICAD *beams*. In Grasshopper, choose the *Beam* component from the *Design* parameters:

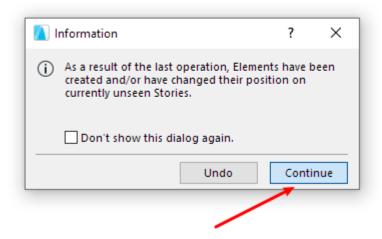


Then, before connecting anything, produce a *Boolean Toggle*, connect it to the *Beam* component *Synchronize* input and make sure it is set to *False*:





This way, Grasshopper will only generate ARCHICAD geometry when you say so. Connect the two *Line* components to the *Beam's Curve* input. Press *SHIFT* to be able to connect more than one input.



Set the Boolean Toggle to True (double-click). ARCHICAD starts working the geometry. Ignore

messages concerning any *story* issues. The result depends on your ARCHICAD *beam* tool default settings. In my example, it looks like this:



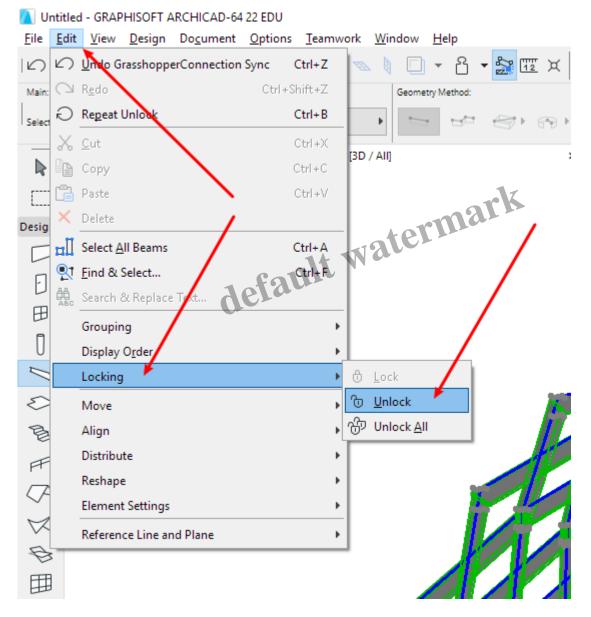
Edit ARCHICAD Beams <

Now these beams are a bit clunky. However, it is possible to edit geometry in ARCHICAD even when it is generated by Grasshopper.

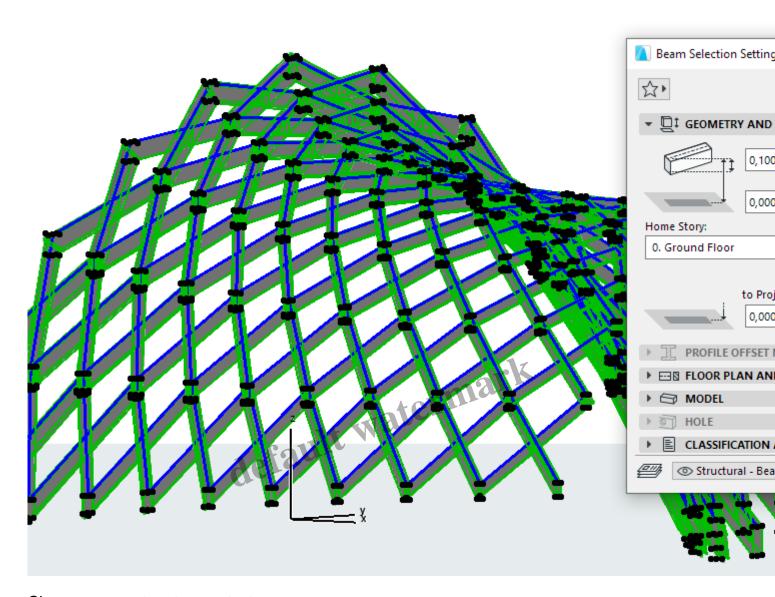
Just pick the Beam tool ...



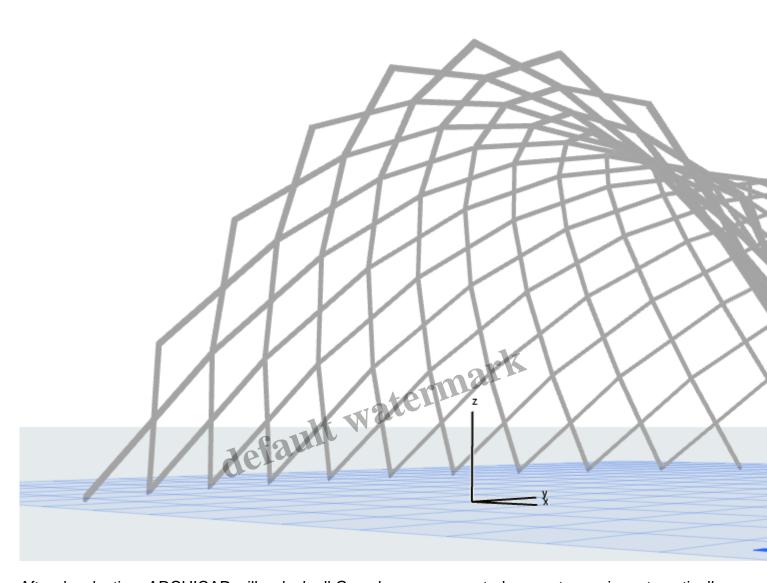
... and press *CTRL-A* to select all beams. By default, Grasshopper-generated geometry will be locked in ARCHICAD. So choose *Edit – Locking – Unlock:*



You may have to press *CTRL-A* again because ARCHICAD tends to deselect things on the way. Hit *CTRL-T* when all beams are selected:



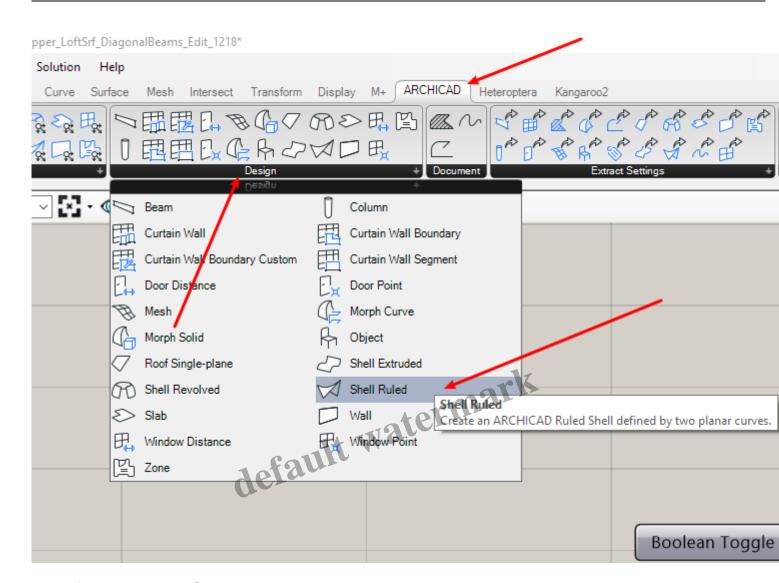
Change parameters to your taste:



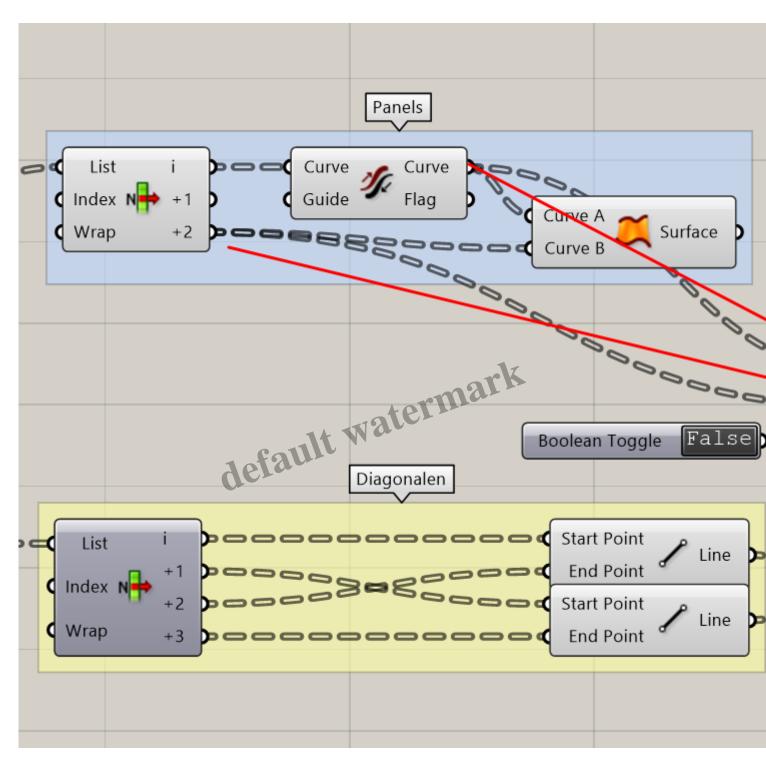
After deselecting, ARCHICAD will re-lock all Grasshopper-generated geometry again, automatically.

Grasshopper Panels to ARCHICAD Shells <

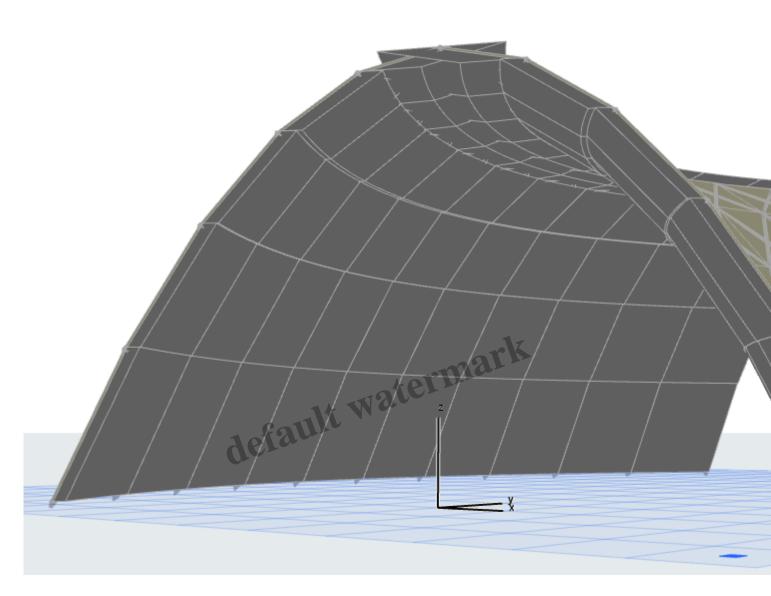
For the roof panels you produced in Grasshopper you will want to use ARCHICAD *Shells*. In Grasshopper, pick a *Shell Ruled* component:



First of all connect it's *Synchronize* input with your *Boolean Toggle* which should be set to *False* again. Then, connect the *flipped edges* (*Curve* output) to *1st Profile Curve* input as shown. Then connect the List Item's +2 output to *2nd Profile Curve* input:



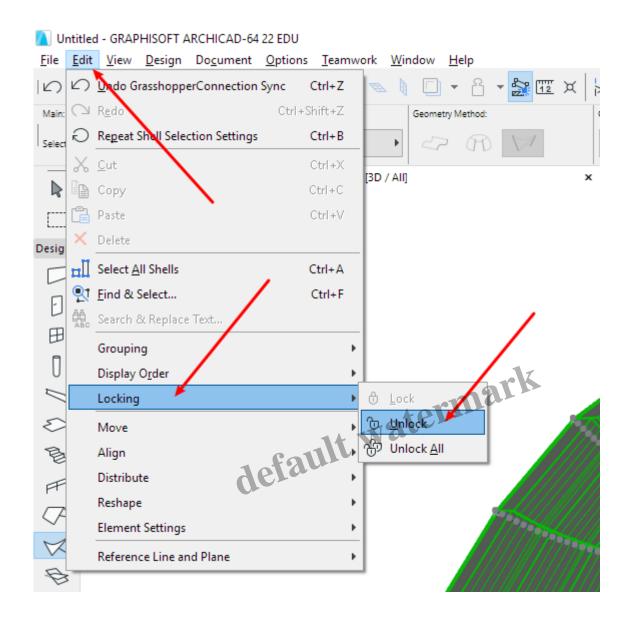
When you switch Boolean Toggle on again you may see something like this:

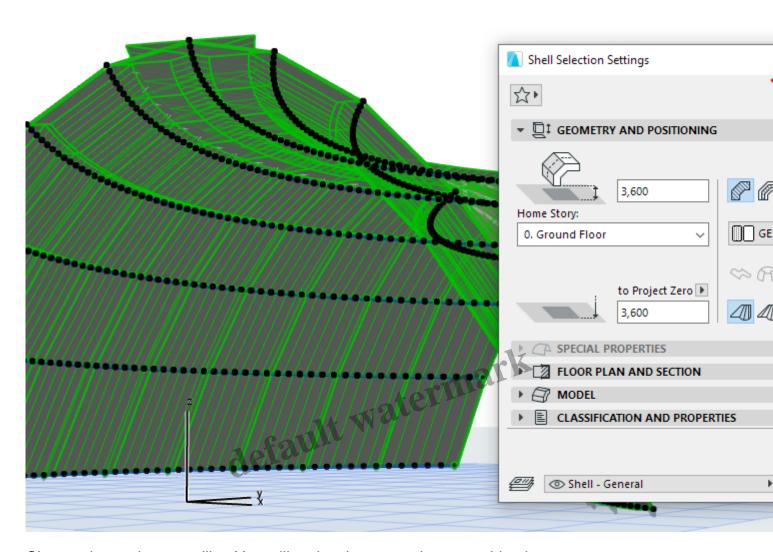


Edit ARCHICAD Shells, Part I <

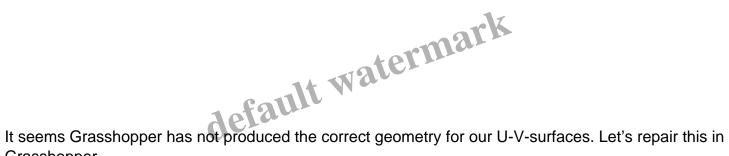
Like with the beams, you may change the shell's looks in ARCHICAD. Use the *Shell* tool, press *CTRL-A* and *unlock* all elements:





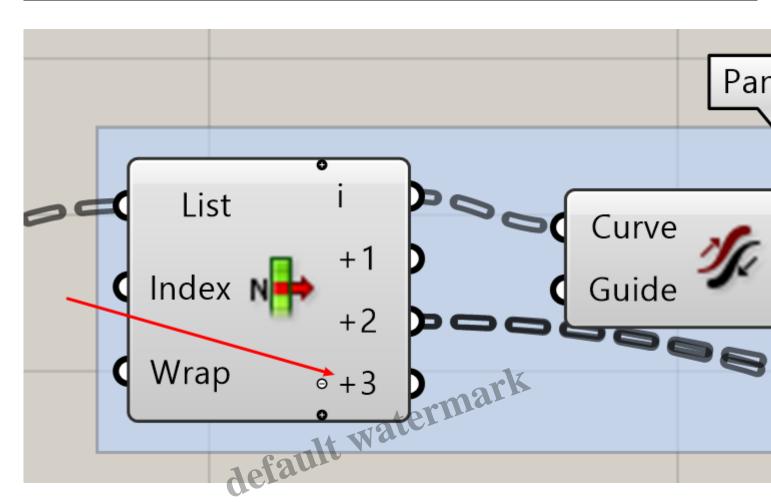


Choose the settings you like. You will notice, however, that something is wrong:



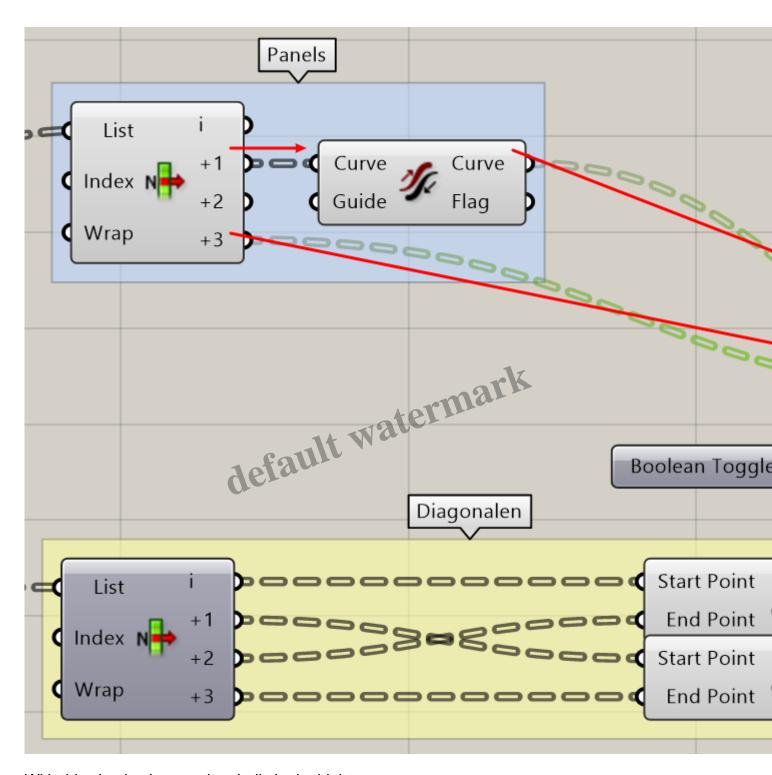
Grasshopper.

Correct Panel Geometry in Grasshopper <

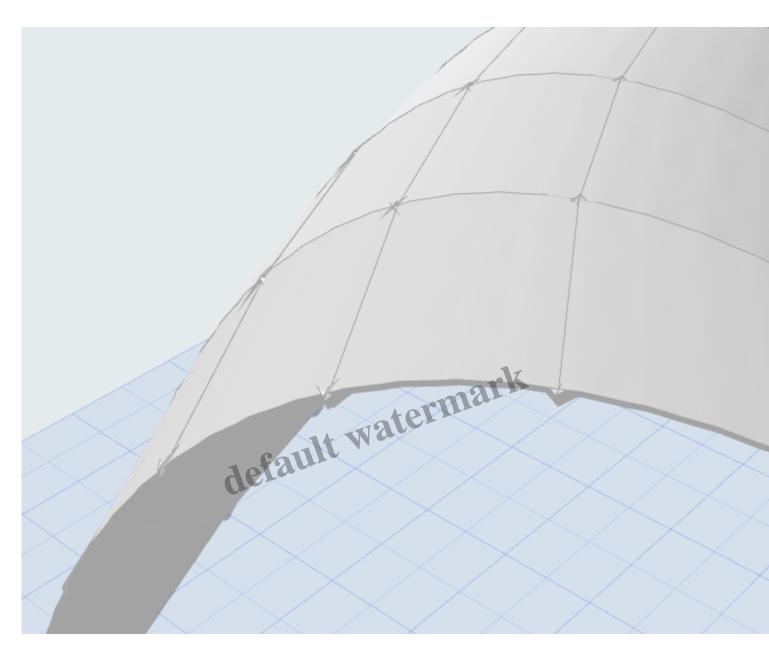


To produce the roof panels, I told you to extract every 1st and 3rd edge in each edge group and connect them via the Ruled Surface component. To do this we used the List Item component extracting item 0 (i) and 2 (+2).

Now that the result in ARCHICAD looks messy, let's try to connect edge 1 and 3 instead. Zoom into the *List Item* component and add a 4th output. Then reconnect everything, this time using List Item outputs +1 and +3:

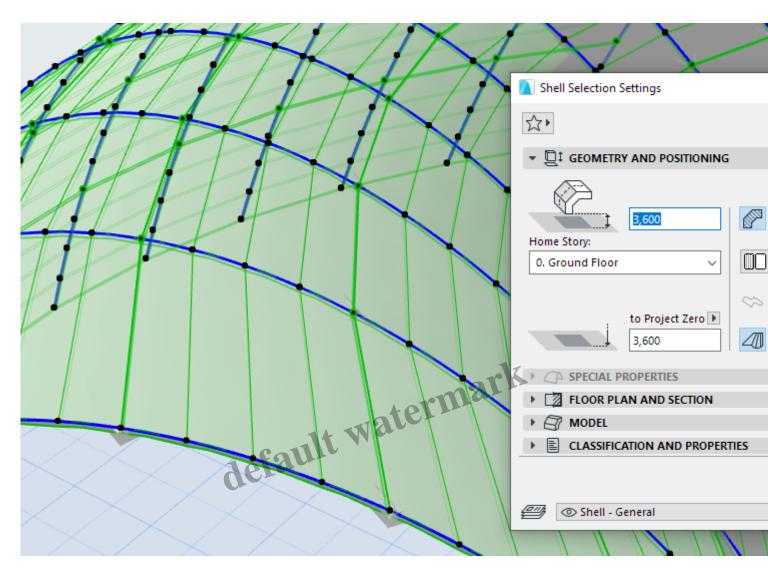


With this simple change, the shells look alright:

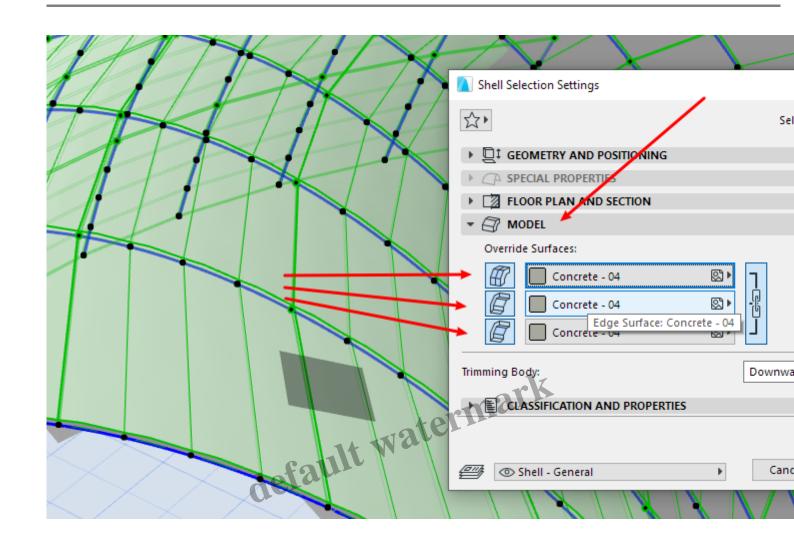


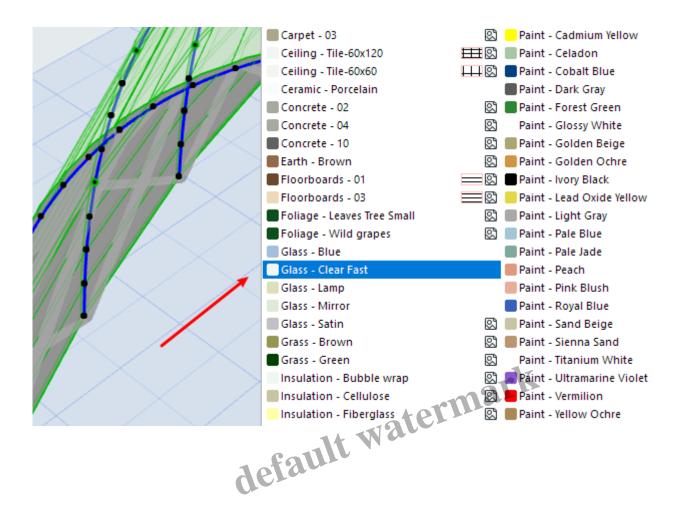
Edit ARCHICAD Shells, Part II <

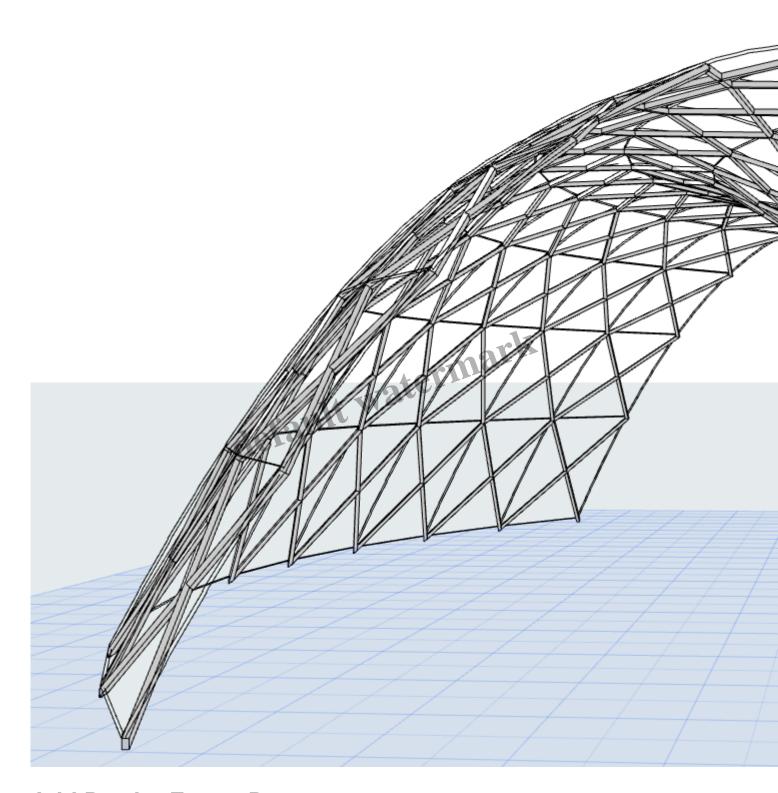
Still, I suggest changing some more things. First of all you may want to flip the shell geometry along its reference plane in order to elevate it. (Remember all the select-and-unlock-stuff.)



Then, it may be nice if the cladding is transparent. You can do this via Override Surfaces:



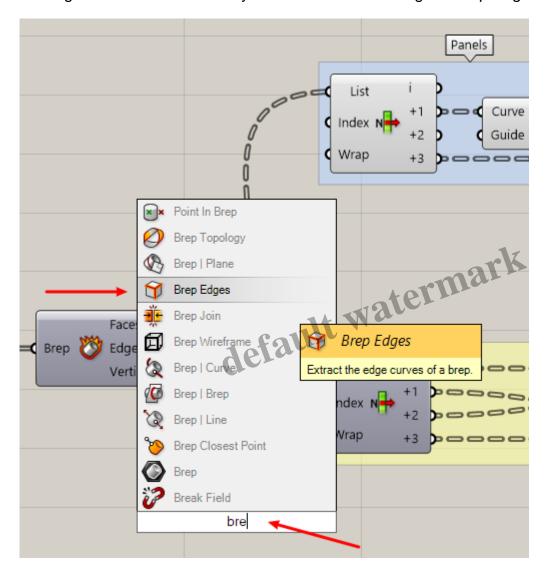




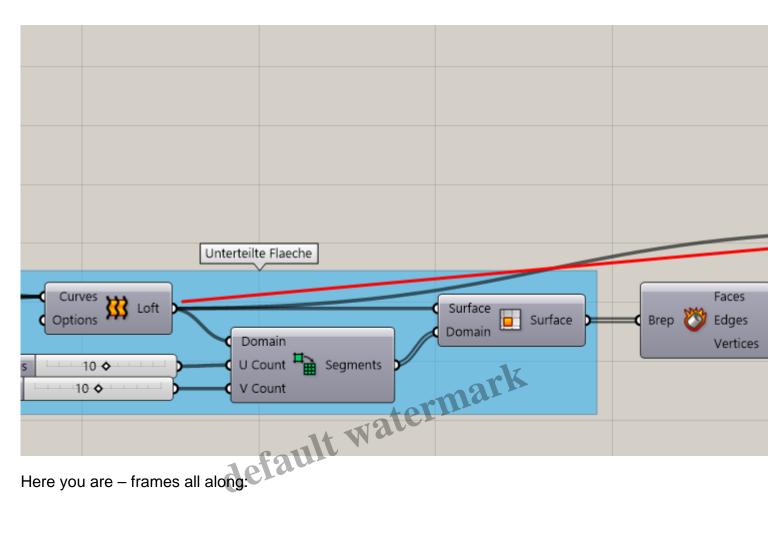
Add Border Frame Beams <

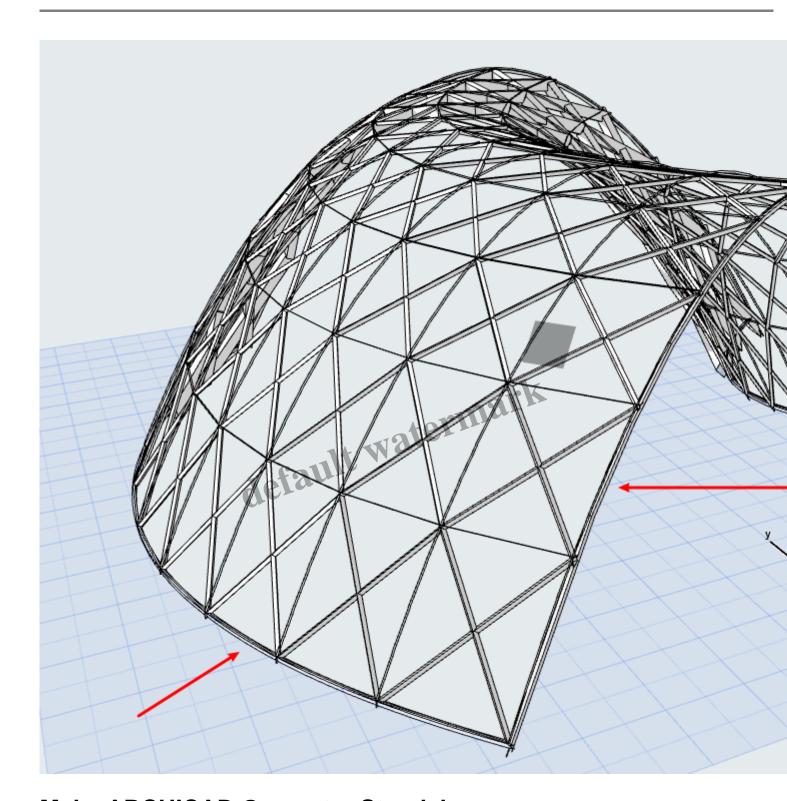
Do you miss some ARCHICAD beams along the border curves of your surface? You will have to add some more definitions inside Grasshopper.

Up to now, you have produced a Loft surface and grid lines on it – but you have not extracted its framing curves. The easiest way to retrieve them is using the *Brep Edges* component:



Be sure to connect it directly to your *original Loft surface* (not the *Isotrim surface*). Then, connect the *BrepEdges Naked* output (supplying the naked = bordering edges) to your *Beam* component's *Curve* input. (Again, press *SHIFT* so you don't lose the other inputs when doing so.)



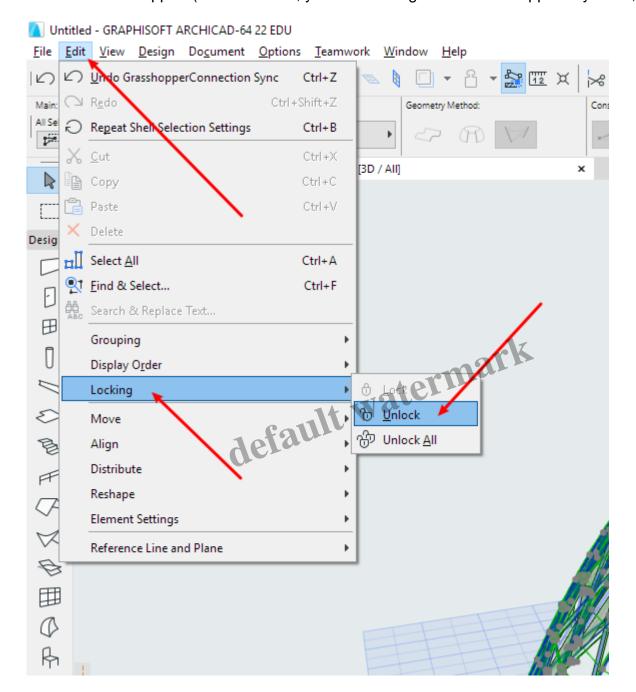


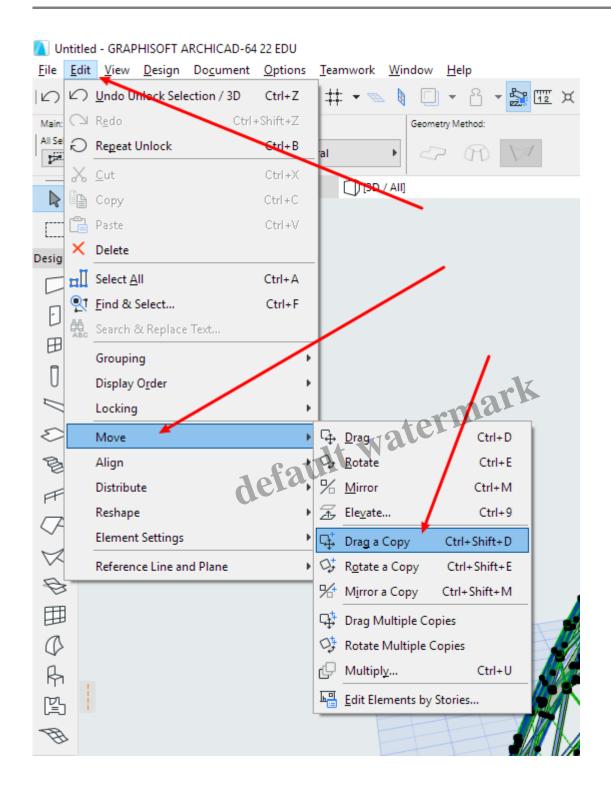
Make ARCHICAD Geometry Standalone ≤

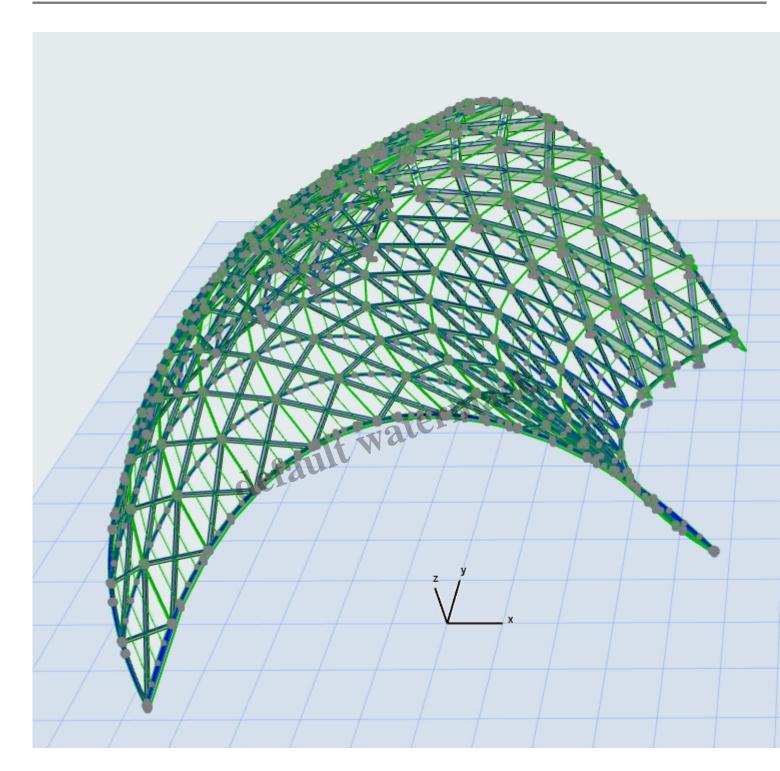
Up to now, all your ARCHICAD geometry is dependant on its Grasshopper definition. To proceed, you may want to make your new freeform structure standalone in ARCHICAD – thus making it independant from Grasshopper.

To do this, simply select it completely, unlock it and copy-drag it elsewhere. This copy will no longer be

linked to Grasshopper. (And of course, you can't change it via Grasshopper any more, too.)







Parametric Playground <

Turning your Grasshopper definition into standalone geometry in ARCHICAD allows you to work on alternative structures in Grasshopper:

- Change the NURBS curves in Rhino
- Change the *U-V Subdivison* in Grasshopper

Roundup and Links <

With Rhino3D Grasshopper you can enhance ARCHICAD's modeling capabilities. As you can observe, Grasshopper plays the main role in this play. To make the most of this modeling freedom it is a good idea to work on a profound understanding of how parametric definitions in Grasshopper work.

Basic information on Grashopper can be found here.

Grasshopper-ARCHICAD-connection: See here for more information.

© 2018 / Horst Sondermann / All Rights reserved

Category

- 1. Archicad
- 2. Rhino/Grasshoppper

Tags

- 1. BIM Model
- 2. Parametric Modeling

Date Created
December 2018
Author

hsondermanncom

