

HOW VIRTUAL BECOMES REAL

Fabricate a gridshell structure with horizontal holed strips (F01)

Creative team

Alberto Pugnale and Stanislav Roudavski

Content developed by

Alberto Pugnale, Alessandro Liuti and Louis Gadd

*Supported by the faculty
“Strategic Initiative Funding”*



msd

**Melbourne
School of Design**

FACULTY OF
ARCHITECTURE,
BUILDING AND
PLANNING

www.msd.unimelb.edu.au

0:00 SLIDE - Intro

- In this tutorial you are going to fabricate the Diagrid generated in tutorial G01 with horizontal holed strips

0:09 SLIDE - Example projects

- This method is particularly effective when you have to realise large scale models, with more than two streams of laths that intersect at the nodes
- It is also helpful if you are planning to post-form a flat grid of laths to realise a timber gridshell model. In this case, you are likely to have at least five streams of laths, four for the main structural grid, and one for the bracing elements
- You can join the laths with small screws. That's an expensive solution which provides an elegant outcome. For cheaper models, small pins and cable ties are more suitable materials
- Now, let's start!

0:50 GRASSHOPPER

- The Diagrid generated in tutorial G01 outputs four streams of segments and respective vertices, so that data is already organised for easy fabrication
- You need to select one stream at a time, so now decide which stream you'd like to fabricate first
 - o For example, select the main members in direction 1
- Use the **Scale** component to adopt a fabrication scale
For instance, you can make a model at 1:10

1:20 SLIDE – Material thickness

- Now, you have to determine if the strips belong to the upper or the lower layer of the gridshell. For the top layer, the strips will have to be slightly stretched to take into account the thickness of the fabrication material. Vice versa for the bottom layer, the strips will have to shrink. In this tutorial, you can assume the top layer is selected

1:40 GRASSHOPPER

In Grasshopper, get the surface you have initially used to discretise the grid. This will serve as a reference to either stretch or shrink the strips for fabrication

- First, **Scale** the surface to the selected fabrication scale – in this case, 1:10
- Then, use **EvaluateSurface** to get the surface normal vectors. You now need to refer them to the polyline vertices
 - o To do so, use **SurfaceClosestPoints** to get the uv parametric coordinates of the vertices, which are now referred to the surface
 - o Feed these uv coordinates into **EvaluateSurface**
 - o Always use **VectorDisplay** to double-check if the vectors have been drawn correctly. Feed the *Surface points* into *Anchor points* and the *Normals* into *Vector to preview*
- Now use **Move** to shift the polyline vertices to the top layer
 - o Remember that you are drawing in metres. Therefore, to determine the right strip thickness at the fabrication scale, you need to divide the actual sheet size, for instance 1mm, by 1000
 - o Now multiply the resulting value by the normal vectors
 - o And feed the output into **Move**
- Use **Polyline** to transform the sets of vertices into sets of continuous polylines, then **Explode** the output to obtain a nested list of lines

Now you have to find the list of vectors, which are perpendicular to the surface normals and the polyline segments. These vectors will be tangent to the starting surface, and will be used to define the orientation of the strip width

- To do so, use the **CrossProduct** component:
 - o Input the polyline segments and the normal vectors
 - o Then, use again **VectorDisplay** to check if the tangent vectors have been drawn correctly
- If that's the case, **Multiply** this vector list by the strip width, for instance 10mm divided again by 1000
 - o Now feed this value into a **Move** component, and **Divide** it by 2 with an expression
 - o As the input geometry, feed the shifted vertices of the strips
 - o You are now ready to draw the first edge of the strips. Use **Polyline** to do so
 - o Repeat the procedure, but inverting the vector direction first, for the second edge of the strips
- Now **Merge** the edges, and extend them with an **ExtendCurve** component
 - o Feed the strip width, divided by two, into the L0 and L1 inputs. The curves will be extended at both ends so that, further on, the holes can be cut with an adequate margin
 - o Finally, **Loft** the edges to get the strip surfaces

5:44 GH – Unroll strips with holes

- Make sure you have installed the *FabTools* plugin for Grasshopper, version 0.95
- Use the **Unroll** component to develop the strips
 - o You can also feed additional geometries to unroll. In this case, input the list of the polyline vertices, as they will be used to define the centre of the holes
- Connect the unrolled Breps to a **BrepJoin** component. This will join the strip surfaces
 - o Now use **Arrange** to distribute the strips on the XY plane. With this component you can also fit the strips to a defined frame
- You can now use **BrepEdges** to get the strip outlines. Remember to take the naked edge curves only
- You now want to cut the holes in the strips. To do so:
 - o Use a **CircleCNR**. This will draw the holes on the strips
 - o Use the **Transform** component to get the central points of the circles. This component will move the polyline vertices from their original position to the unrolled strips. Connect first the transform data from **Arrange**. Then, feed the unrolled points
 - o Now select a value for the circle radius. If you are going to connect the laths with small screws, let's say with 3mm shaft gauge, use a 1.5mm radius
- To our knowledge, *FabTools* may unroll some of your geometries wrongly when the reference strips are too small. For instance, if you discretise a starting surface of 5x5 square metres into strips, the issue might arise.

7:35 GH – Strip labels

- Before moving to the actual fabrication, you still need to label the strips and make them recognisable for assembly
- You can use the **3DTextAdvanced** component to generate the labels
 - o You need to provide label positions and texts as inputs
- To define the label positions:
 - o Get a **DeconstructBrep** component and connect it to the unrolled geometries which are already arranged on the XY plane

- Use a **ListItem** to select the first surface of each strip, and feed the output into an **Area** component. Now, feed the *Centroid* into **3DTextAdvanced**
- To define the label texts:
 - Get a **Series** component, and use **ListLength** to calculate the number of labels you will need. You can measure, for instance, the length of the *Centroid* list. This will return the list of progressive labelling numbers
 - Now use **Concatenate** to add a prefix, such as “T”, which reminds you that the strips are belonging to the top layer
 - Feed the **Concatenate** output into **3DTextAdvanced**
- You can adjust the label size, for instance, to half of the strip height
- You can finally use a **ValueList** component to change the text justification

9:19 SLIDE

- This is the end of the tutorial. In the support master file, you will find the refined GH definition, which is linked to the other definitions of tutorials G01, G01.1 and G01.2:
 - To activate or deactivate this modulus, use the switch on the top left (**ValueList**)
 - The components highlighted in green are placeholders of the Diagrid input geometry
 - The components highlighted in pink allow you to define the model scale, strip width, thickness and position (either top layer or bottom layer), hole radius and label alignment
 - The first group in the middle generates the actual strips, which are then unrolled by the second group of components. In the second group, the strip holes are also defined
 - The labels are instead added by the last group of components, on the bottom right
 - The components on the bottom left will finally allow you to arrange the unrolled beams on a rectangular sheet for fabrication

10:15 END OF TUTORIAL F01