

HOW VIRTUAL BECOMES REAL

Post-forming a timber gridshell
with Kangaroo (G_02.1)

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 - Cover

- In this tutorial, you are going to simulate post-forming by means of Kangaroo Live Physics, using the grid of laths you have already generated in tutorial G02
- Now, let's start

0:12 GRASSHOPPER

- In tutorial G02, you have parametrised two sets of grid laths in such a way that the two polyline directions, X and Y, can now be accessed separately
- For the moment, let's concentrate on the first set of polylines, oriented along the Y direction
- First, use **Explode** to get the polyline vertices
- Then, get Kangaroo **Bend** to define the lath bending. This component requires three points, or in our case, three lists of points as inputs

0:43 Slide - Bending

- This diagram describes how the **Bend** component wants you to input the three lists of points
- Let's consider the first vertex of each polyline, with index 0, as *Point 1*. *Point 2* will be its nearest neighbour, that is the second vertex in the list with index 1. In the same way, *Point 3* will be the second nearest neighbour, that is the third in the list with index 2
- In this way, Kangaroo transforms the points into particles and is able to simulate post-forming by calculating the "bending resistance acting on an angle defined by 3 points"

1:16 GRASSHOPPER

- To perform this in Grasshopper:
 - o **Shift** the list of vertices of the exploded polylines. Make sure the *Wrap* input is set to False
 - o Apply another **Shift** to the output list, and set again *Wrap* to False
 - o Now use the **ShortestList** component to trim the ends of the three lists according to "the shortest length amongst them"
 - o Finally feed *A* into *Point1*, *B* into *Point2* and *C* into *Point3*
 - o As the *BendStrength*, input, for instance, 800
- Now get a **SpringFromLine** component to transform the polyline segments into spring elements, or better into elastic connections between the points, or particles
 - o You also need to provide a *Stiffness* value, for instance 2000, and a *Rest Length*. In this case, directly feed the length of the segments
- You can repeat this procedure for the second set of laths, in direction X
- You now need to define the anchor points of the Particle-spring system before running the simulation. For the purpose of this exercise, you can consider the gridshell as supported at the four small edges you got through trimming:
 - o Get the **EndPoint**s of the polylines in both directions. Now get **SetIntersection**, and feed the start and end points of the first stream into A. Then, connect the start and end points of the second stream of polylines into B. Now flatten the inputs to get the gridshell edge points as output

3:12 Slide – Post-forming

- In practice, the anchor points are pushed towards the centre of the structure to post-form the laths, while vertical point loads are applied to some specific nodes with the aid of timber posts

3:24 GRASSHOPPER

- To simulate this in Grasshopper, you can **Scale** the anchor points by a factor using a **Slider** that ranges between 0 and 1
- To define the centre of scaling, feed the initial trimming curve into an **Area** component. Then, use the *Centroid* point
- You now have to set a vertical lifting force, which simulates the contribution of the posts but also makes sure the gridshell deforms towards the right direction
 - o Get a **UnaryForce** and feed a **UnitZ** vector as force input
 - o Before feeding the two lists of lath vertices into *Point*, use **RemoveDuplicatePts** to get rid of the coincident vertices. Remember to flatten the input
- You are now ready to run the simulation with the aid of **KangarooPhysics**
 - o Feed *ForceObjects* with the **SpringFromLine**, **Bend** and **UnaryForce** components, and remember to flatten the inputs
 - o Now feed the **Scale** output into *AnchorPoints*
 - o Connect the exploded polyline segments to *Geometry* and flatten the input
 - o Get a **BooleanToggle**, set it to True, and feed it into *SimulationReset*
 - o Now connect a Timer to **KangarooPhysics**, and set the solution refresh interval to 20ms
- To run the simulation, click on the **Toggle** first, and then play with the scaling factor **Slider** of the anchor points

5:50 Slides – Final code

- This is the end of the tutorial. In the support master file, you will find the refined Grasshopper definition, which is linked to the code of tutorial G02:
 - o To components highlighted in green are placeholders of the input geometry. They contain the lath polylines in the X and Y directions, as well as the grid trimming curve
 - o This group of components transforms the polyline segments into springs, the polyline vertices into particles and it also performs bending
 - o With the aid of this **Scale** component, you can bring the anchor points nearer to the centre of the gridshell footprint
 - o This **UnaryForce** defines vertical point loads which help the structure deformation
 - o These two **Sliders** highlighted in pink allow you to define the lath stiffness and bend strength
 - o The components on the top right finally perform the gridshell post-forming

6:45 END OF TUTORIAL G02.1