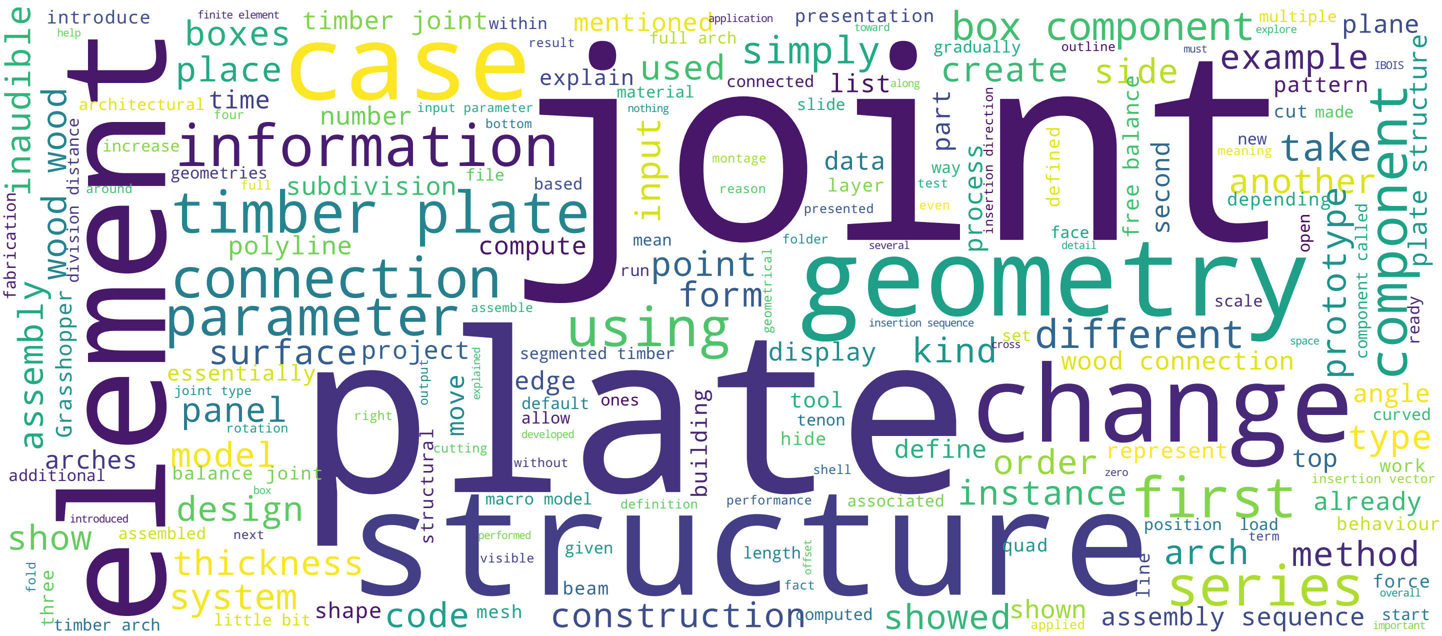
Double-layer Timber Plate Shell

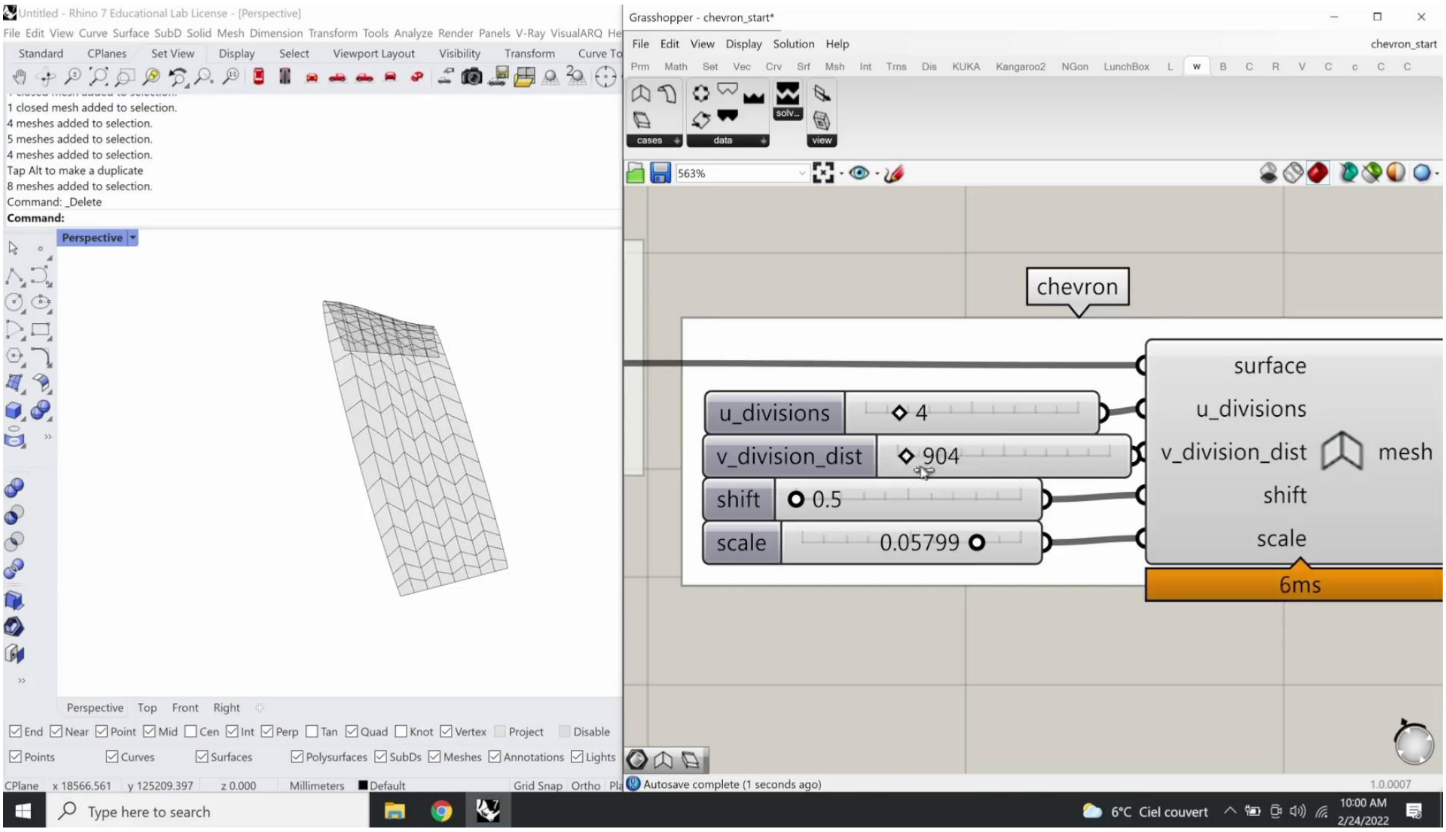
Tutorial
Rhino3D and Grasshopper
Part. 3

Petras Vestartas, Ph.D.



Search MOOC

Video

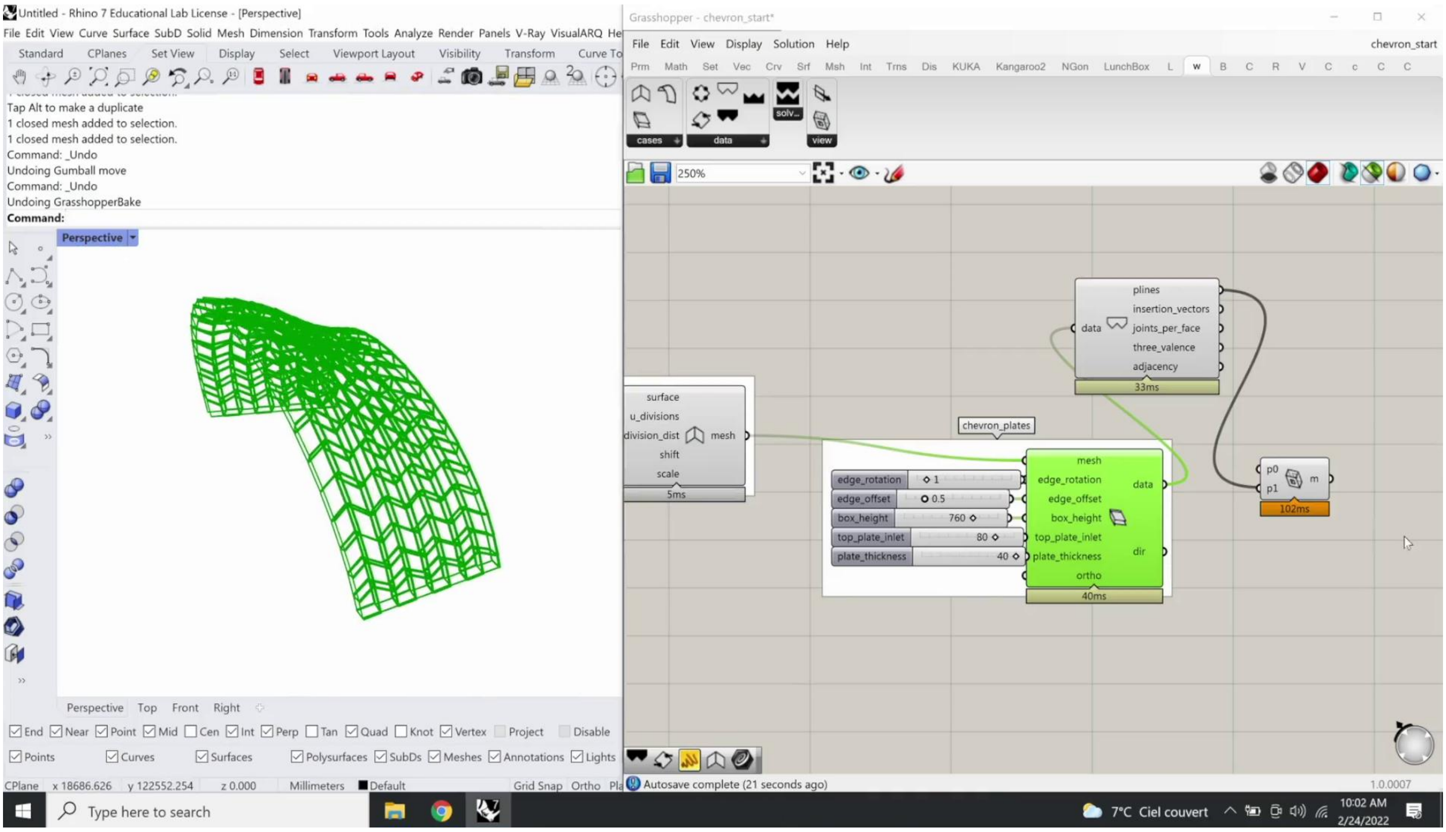


In this case, we have actually surfaces already defined coming from the initial study. There are these 33 arches, and we'll try to define the joints on one of the arch, and I'll try to explain how it is done within this workflow. First definition that has all the information, it is simply a series of surfaces, and then we can have those surfaces already in Grasshopper, so we don't need to draw them manually, so let's take the last arch. The first point, as I mentioned, during the presentation, we need to subdivide the surface into quads. For this one, we can use a component called chevron, which is a subdivision into quads that are shifted. As a helper, I also give you a certain input parameters so you don't need to guess what they are, but I'll explain you shortly what they mean. Let's hide the previous information. If you click control M, you would see the actual geometry. I will just change my display so that is a little bit more visible. We have the subdivision pattern, and we can change the number of subdivisions. We can also change division distance. you can see here that the subdivisions as mentioned in the presentation comes from two shorter ends of the arch, and then they gradually becomes merged in the middle.

Notes

Summary

0m 15s

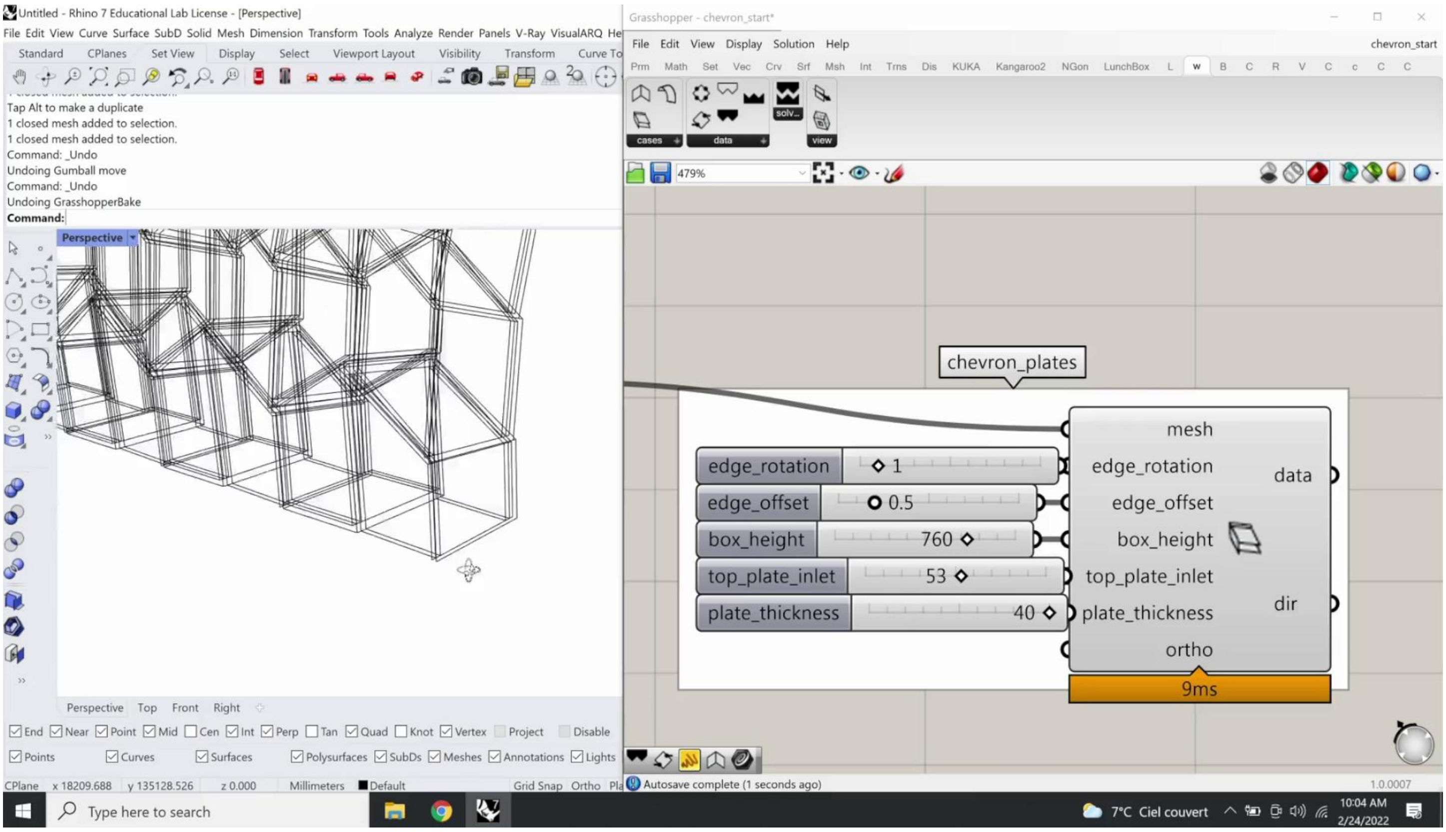


This parameter must be adjusted depending on the arch. This is the shifting formation, so you can change it depending on the value you would like to have. This is a simple quad grid, and then if you change the 0.5, this is the subdivision that was used in study case. Then finally, there are some parameters to adjust the centre point that is again depending on how big are the quads. This is the first part, simply the subdivision of the surface. Then now we need to create the plates from this. For the plates, we can use a component called chevron plates, which is again working especially with this component that has a mesh data structure, and it creates a list of plates. As you can see here, the plates don't have any joints here. Nothing. It is just a series of outlines. If I would try to bake this information, I can see that simply they are a series of plates, and we need them to introduce the third component to compute the joints, which I showed you before. This information that we talked about, the data that we defined manually, it is computed inside this component which is essentially saying what kind of joints, what kind of insertion directions, what kind of free balance joints there are.

Notes

Summary

2m 08s

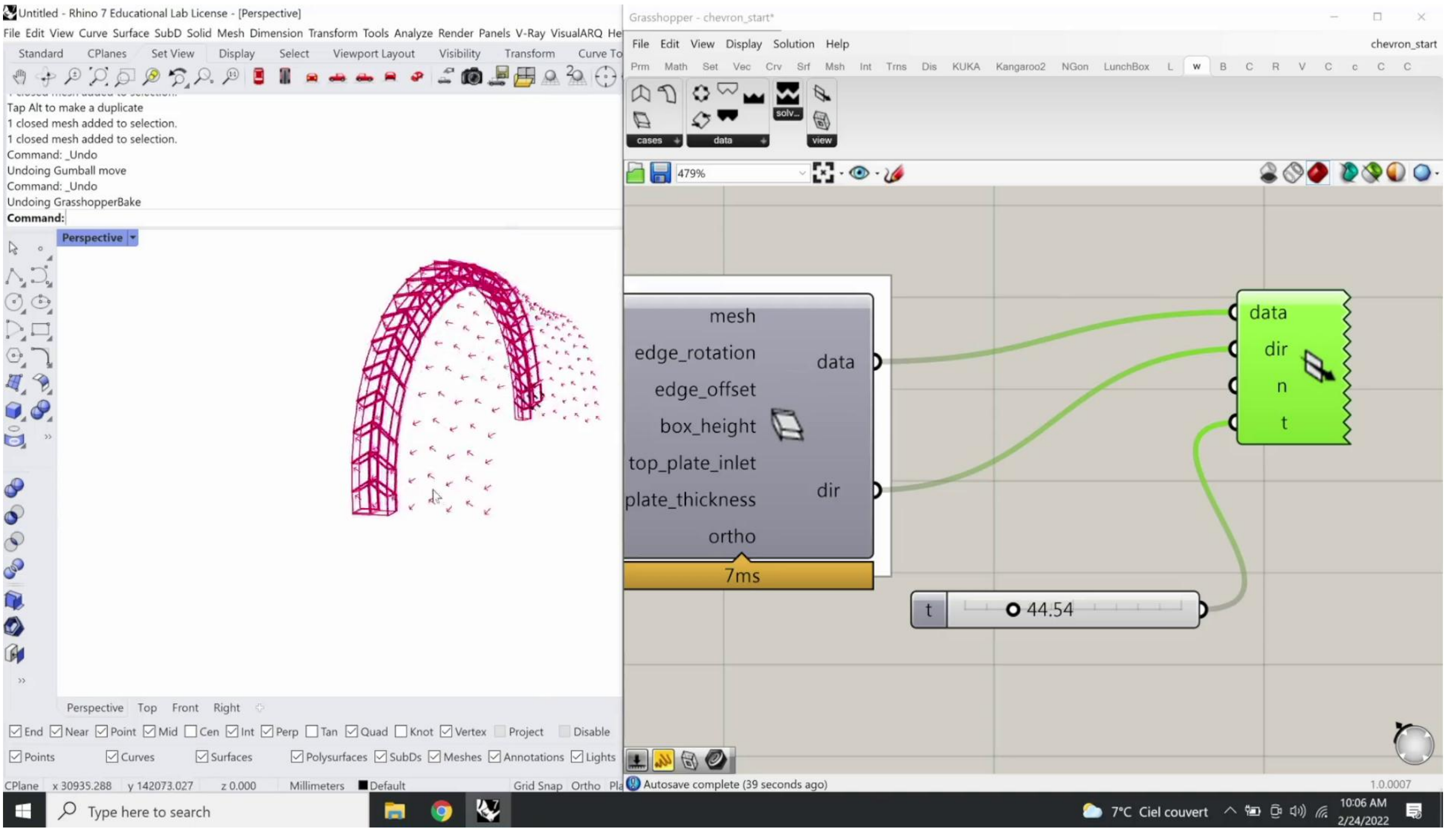


This information is already computed in this component. But before computing the actual joints, I want to explain what are the input parameters because it helps you to understand a bit better how this project was defined. The edge rotation, let's try to zoom in little bit better. The rotation component is trying to rotate one side of quad. You can see that there is a rotation. The reason to do that, at the moment, there are thickness of timber plates, they actually physically intersect. It's not really physically possible to compute these kind of joints because there could be some strange intersection areas. There is a small trick to introduce a gap within each vertex of the mesh by simply rotating longitudinal edge of mesh edges. Then another one, we need to actually move in opposite directions the shorter side of the mesh edges. We could actually acquire this little gap using rotation and translation on two opposite directions. Then we also have the thickness, how this thickness is defined. There's a parameter of box height. This is the box height. Then there is a box inlet. As you can see here, there is a certain amount of offsets.

Notes

Summary

4m 00s

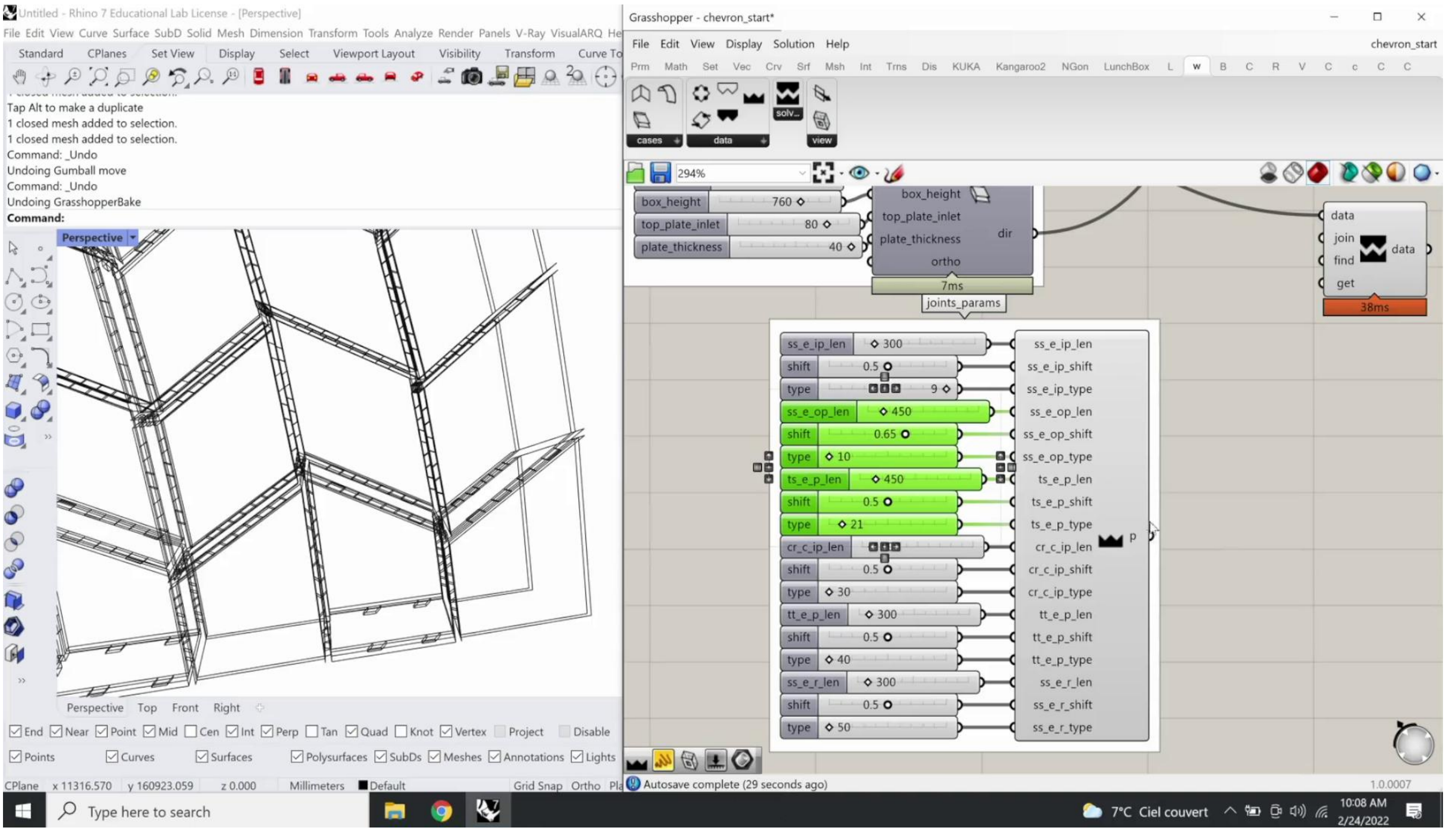


If I would change this parameter, this was the thickness of the plate of the whole shell, and this is the 80 centimetre inlet. If I would do some station like this, there will be actually no place to cut the joint. That's why we use at least small little distance to have a thickness of the plate, just sufficient in this case. Then we also have the plate thickness, which represents the 4 centimetre overhead timber plate, the LVL panel. Another important subject here is the assembly sequence. To explore the assembly sequence, I can use a component called the assembly, which represents how those elements were assembled. If I put data to data and there there, and then, for instance, I would try to hide the previous information, and then I mentioned that there are around 900 plates, but there are around 300 to 200 boxes, you can see here that the assembly is computed is based on the zigzagging pattern. It's important to mention that first we go in one row, then we actually not starting from the same point. We flip and we start from the upper end, and then gradually, we assemble the full arch. This component also shows... It shows the insertion sequence based on each box component.

Notes

Summary

5m 43s

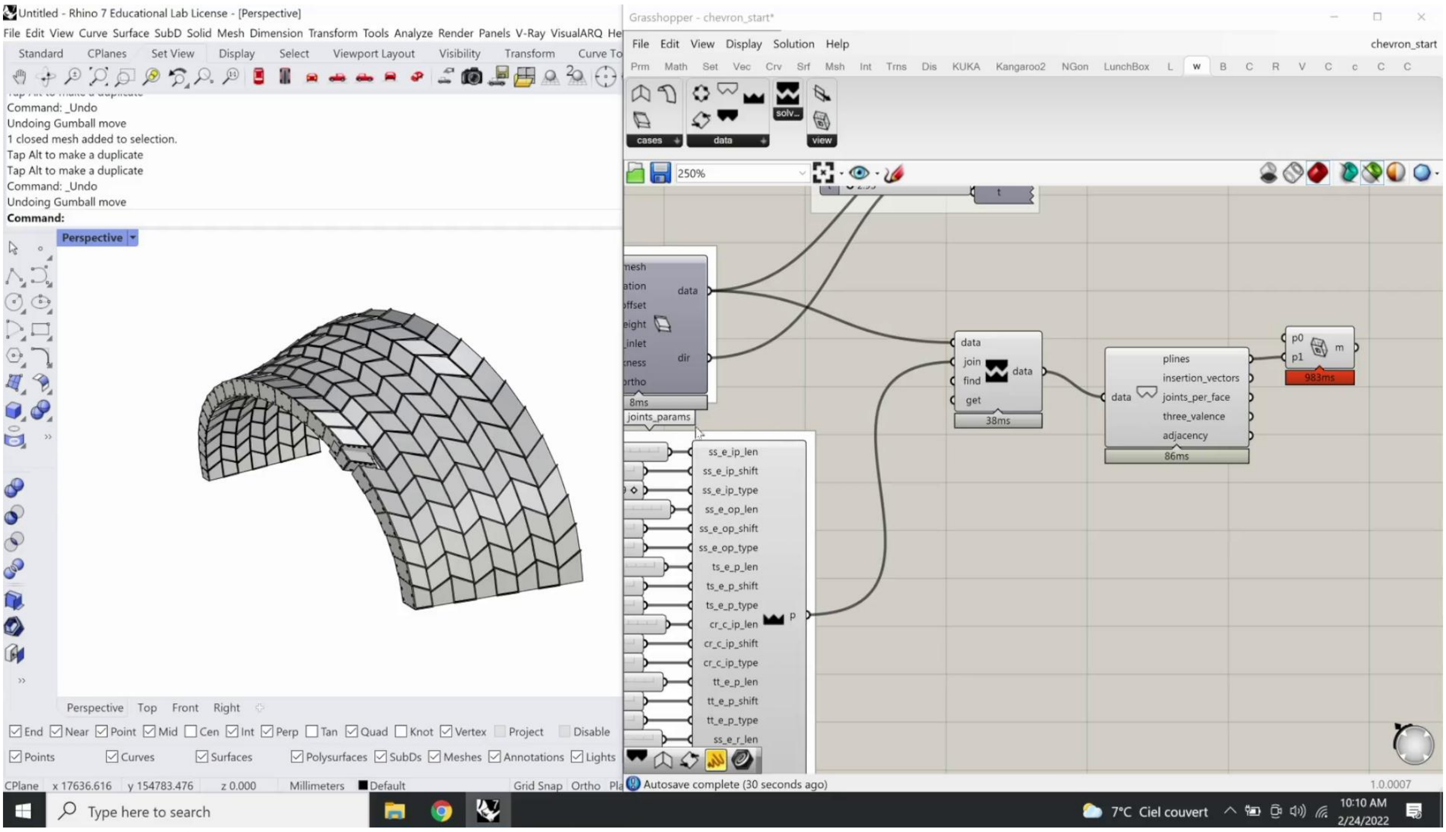


If I try to really go towards the beginning, you see here that there is the box direction. If I would move the slider, see that the boxes are inserted incrementally using one insertion direction per box. Also already this component within this data. Because you can access the information from the C# or Python libraries that I'll describe in the next session. You can define the insertion vectors based on the code. You also can define the zero, which is no joint, 10, which is docktail, and 20 for the telemortiz joints. Once we have this data, you can actually compute the timber joints. This will be next step that we're going to do. I'm going to hide this information, and then I'm going to place the component that we already know called solver. You're going to place data inside data, and by default this information is correctly computed in case these parameters will change or are different. You can use all these parameters that actually defines the scale, defines the joint, and so on. In this specific case, we only are using these ones. We are also using these ones and nothing else. Only two type of joints, then the type of 10 and type of 21.

Notes

Summary

7m 32s

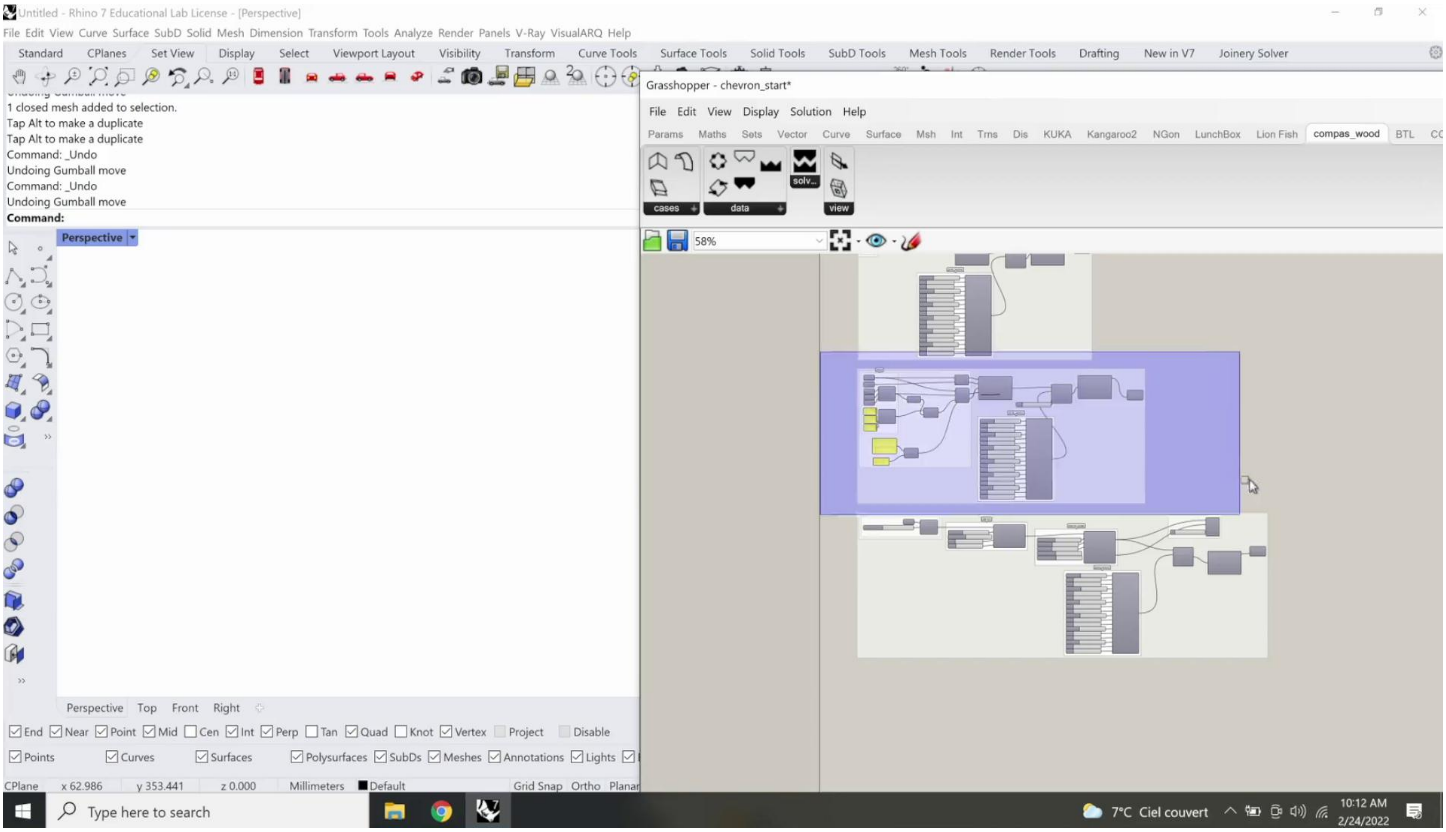


If I would place the P parameter to the joint parameter, you can actually see that we can change the division distance of the docktails, we can change the angle, and then we can also change the same length for the top side connection, which we can increase the subdivision number. Those free balance joints are sold automatically because we gave the sequence of free balance joints already in this component. This works quite fast. We can also change the arch number. You can see here that we can just move those arch numbers because those surfaces, as mentioned in the presentation, are simply two-dimensional grids added to the three-dimensional surfaces. But these are geometries that are self-similar, so the definition will work in all of them. Then finally, once this arch is ready, we can measure the geometry, which can take a bit more time. But this is only for display. It takes around one second. Then if we take this geometry, we can see that all the arches have already the timber joints in place with the necessary assembly sequence. That's it. It is ready-to-go definition for a segmented timber arch, and I hope I explained the workflow for this segmented timber arch.

Notes

Summary

9m 15s

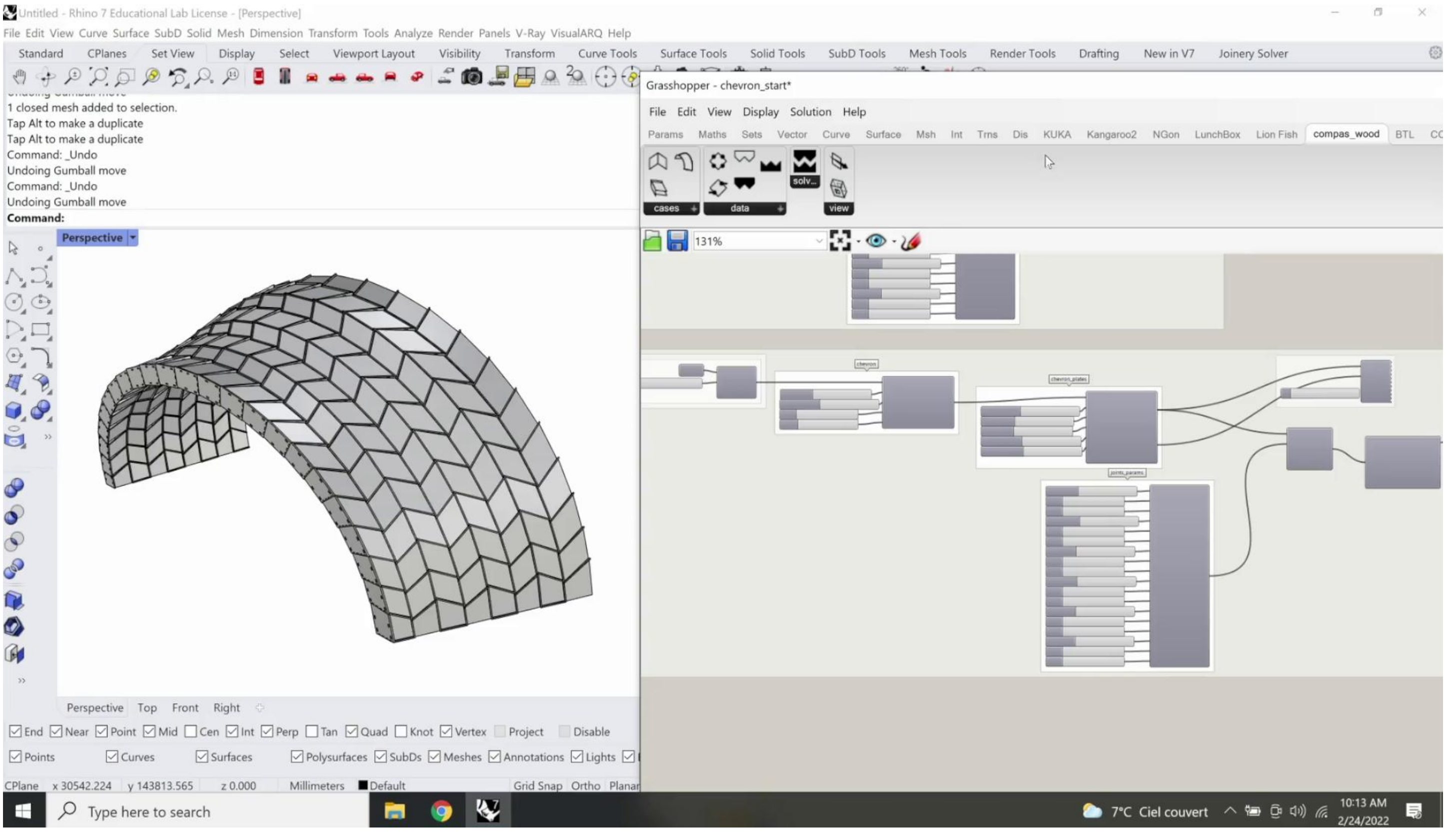


I showed you the basic definitions of box components in a very abstract manner. Then there is a bit more detailed example of automation of this segemented timber arch, which is defined here. This session is ending at this point. For the next lecture, we actually will go more deeper into understanding how the joiner solver works, how the timber joints can be generated within the code. Because at this point, the Grasshopper was used mainly as an interface, where the different codes were wrapped into these blocks of code and gradually outputted within the given information. We covered in total, for example, cases. To reiterate, I will just go through all of them. I showed you how to create simple box component, box component corner with the insertion sequence. I also showed you how to create one box component depending on the joint types, depending on the division distance. Then I showed you the free balance joint that was also presented in the presentation that we need to align this edge, so one box component. Then finally, I showed you the full arch, how the full arch was actually made, including all the timber joints and also the information regarding the longitudinal edge that is actually flat because this input ortho is set to true.

Notes

Summary

11m 17s



Yes, that's basically it. It's two type of joints creating one box component, and then those box components create the arches and then the full shell is essentially a series of arches within self-similar surfaces along the 23 north surfaces.

Notes

Summary

13m 07s