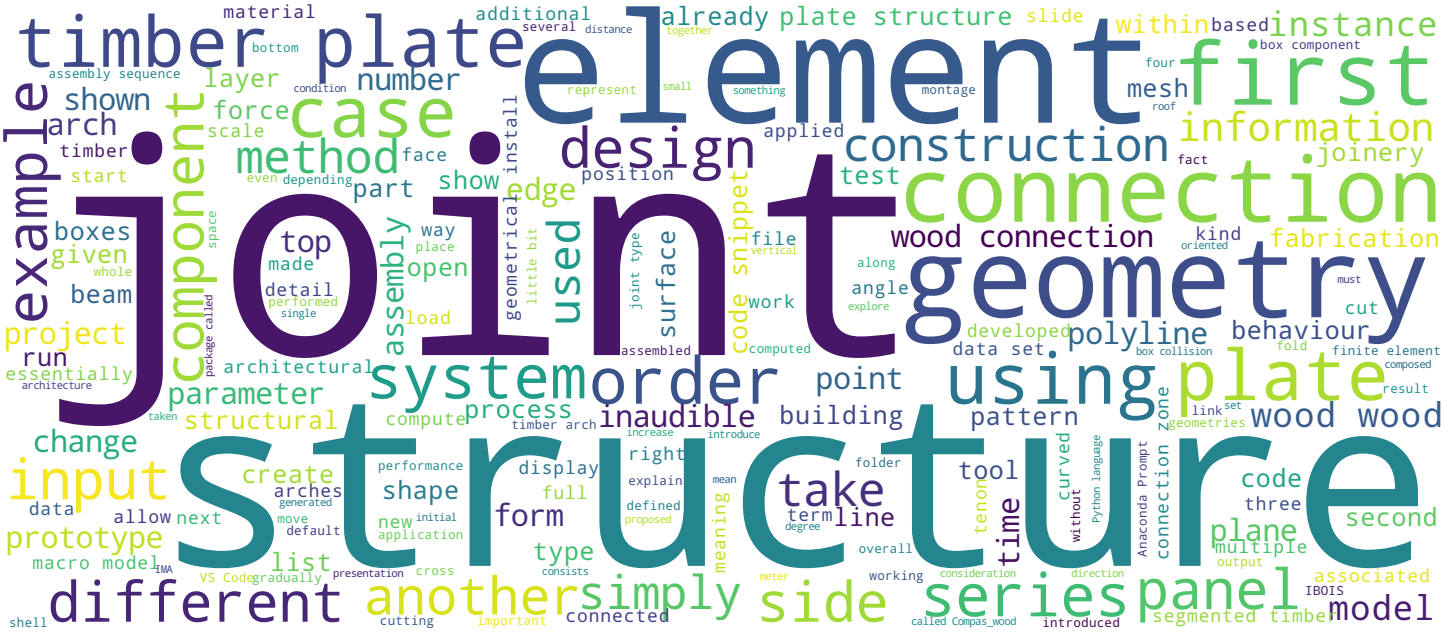
EPFL
IBOIS

Advanced Timber Plate Structural Design

Double-layer Timber Plate Shell

Structure of the chapter:

- Introduction
  - Introduction to the Double-layer Timber Plate Shell
  - Software Installation
  - Quiz
- Geometry Generation – Rhino3D, Grasshopper
  - NURBS Discretization
  - Plates
  - Joints
- Geometry Generation – Python – compas_wood
  - Code Review
  - Summary of the Method



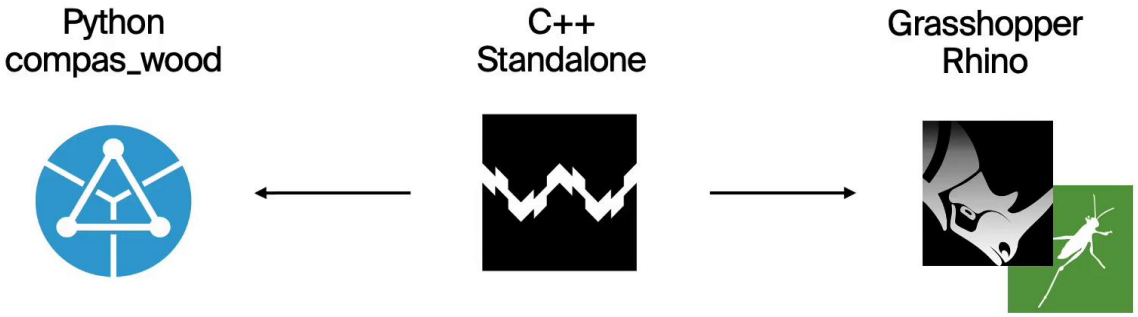
Search MOOC

Video

EPFL

Compas_wood

Petras Vestartas



Hello. This is the last part of the presentation, Double-layered Timber Plate Shell. The presentation is composed from two parts. That is a code review and a summary of the method. This will be, again, the practical example. But this time, we are going to use a Python language and a package called Compas_wood. In order to explain this generation method, regarding the software part, is that the joinery itself is running on a standalone C++ project, and then it is wrapped for the Rhino Grasshopper library that you just saw in the last lecture. And then there is another method. I propose a library of Compas_wood in order to run this standalone code also in the Python language.

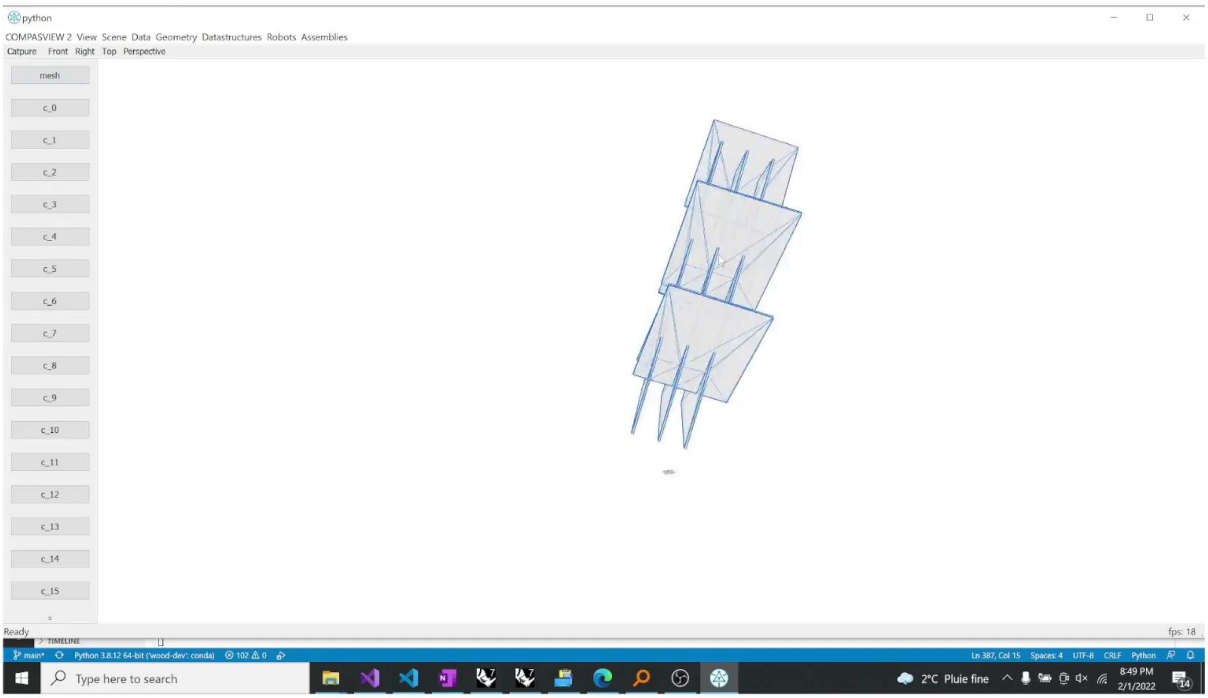
Notes

Summary

0m 04s

Compas_wood

Petras Vestartas



This is a small demonstration how the code was developed. It was actually input of series of different projects, of different data sets, in order to test where the code breaks for debugging reasons. You can see different, essentially, examples that you can actually explore within the Python library. This lecture will be again going to have the same example for the sake of consistency. We are going to explore the segmented timber arch of these two joint types and free balance joint.

Notes

Summary

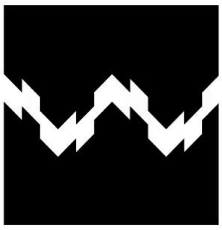
1m 00s

Software Installation

Petras Vestartas

Requirements:

- ☐ VSCode
https://code.visualstudio.com/download
- ■ Anaconda prompt
https://docs.anaconda.com/anaconda/install/windows/
- ■ compas_wood
https://ibois-epfl.github.io/compas_wood/latest/installation.html



Let's move to the installation page. We will need three parts. We will use VS Code as the code editor for the Python. Then we use Anaconda Prompt to install the packages. And then we're going to install the Python package called Compas_wood using the Anaconda Prompt.

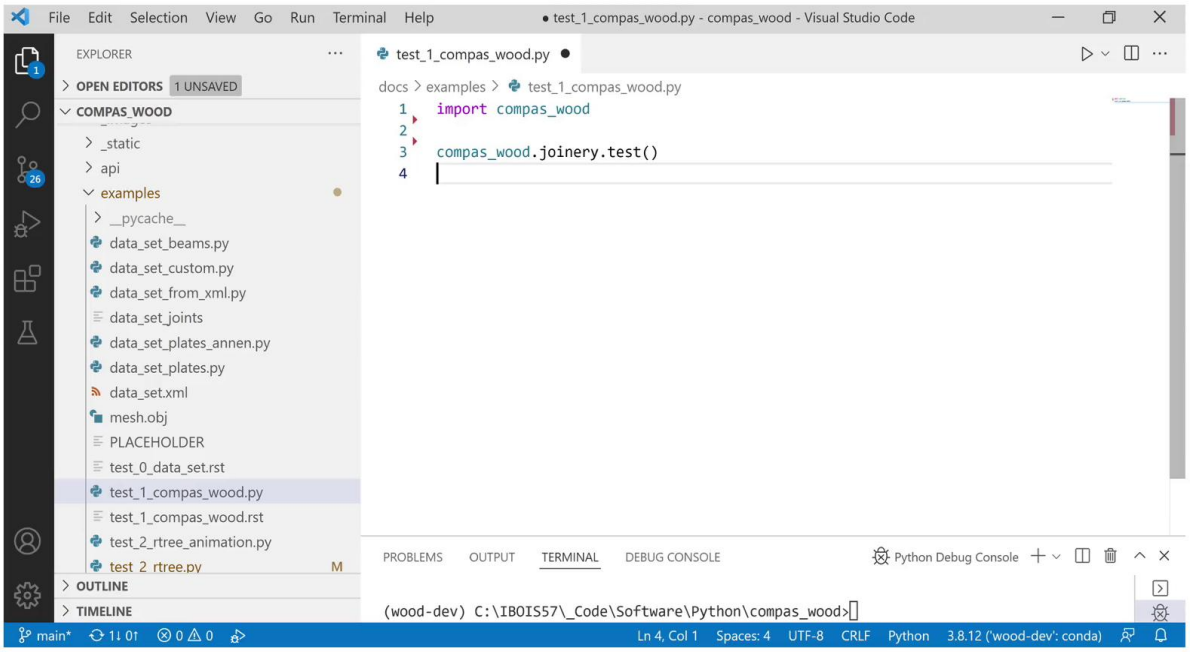
Notes

Summary

1m 49s

1_compas_wood

Petras Vestartas



That are different cases for Windows and Mac users regarding the installation. Once you go to the documentation page, please follow the links below and simply follow the instructions. Once you open the VS Code, you must choose the correct environment in the bottom part, either on your right or left side. For Windows users, the default terminal must be Command Prompt as seen in the second step.

Notes

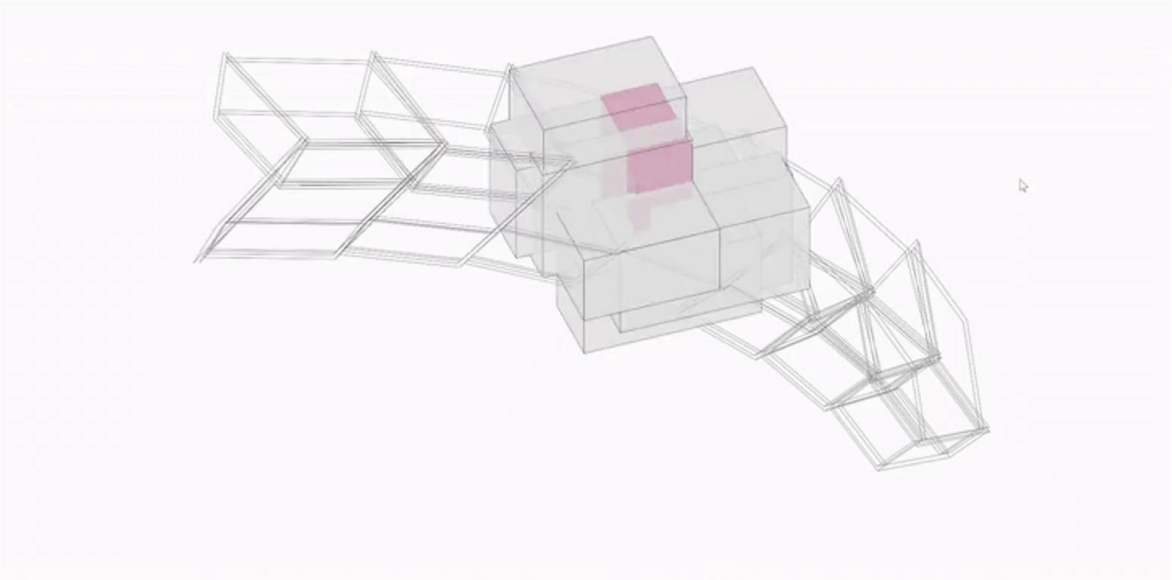
Summary

2m 08s

EPFL IBOIS

2_rtree

Petras Vestartas



Advanced Timber Plate Structural Design

First, we try if this package is actually working on your computer, then we'll try the first simple examples, for instance how to compute Rtree search.

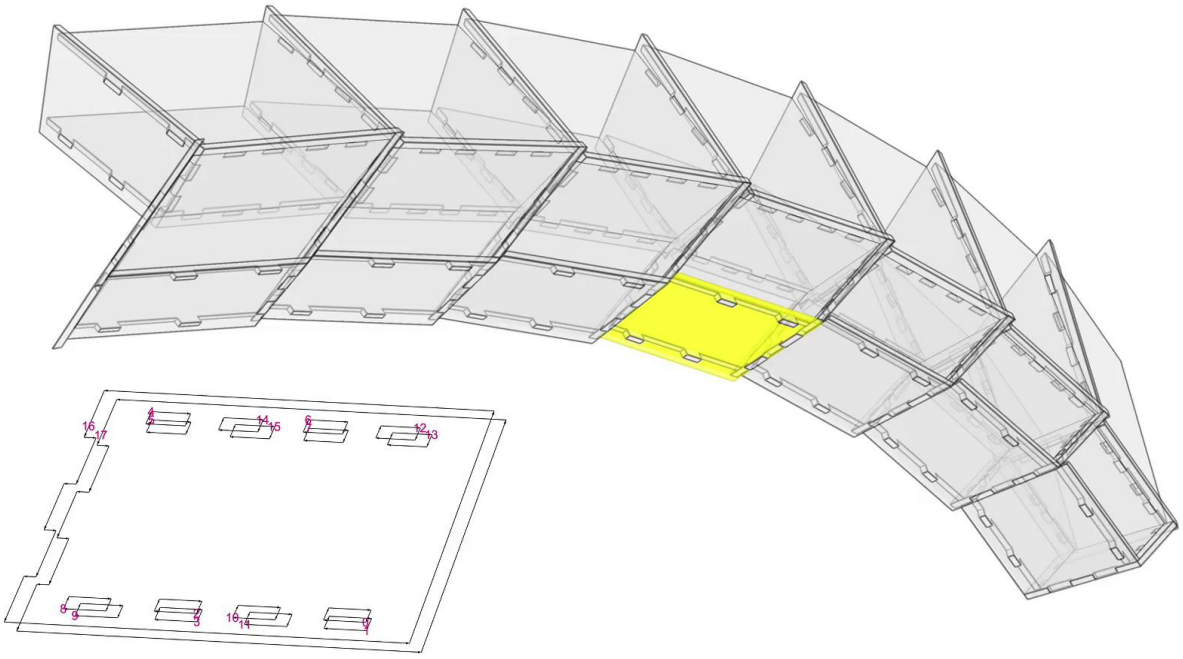
Notes

Summary

2m 35s

5_closed_mesh_from_polylines

Petras Vestartas



It's essentially a method how to find connection zones or essentially neighbours between adjacent plates, because you're not using any additional graph, we're just using a collision method to speed up the search. We're using essentially two subtypes. One is the box-box collision and another one is actually object-aligned box collision. This is a code snippet how to do it, and we will go in more detail. Then we'll try to detect the connection zones. They are actually coloured depending on what kind of connections are possible, depending on side or on top side or other conditions how joints can be generated, so it gives a little bit of user information. This is again the code snippet. Then we will compute the joinery for segmented timber arch. This is also the code snippet. Lastly, we will mesh the geometry for display reasons.

Notes

Summary

2m 45s

5_closed_mesh_from_polylines

Petras Vestartas

Advanced Timber Plate Structural Design

```python
# compas_wood
from compas_wood.joinery import get_connection_zones
from compas_wood.joinery import closed_mesh_from_polylines
import data_set_plates

# viewer
from compas_wood.viewer_helpers import display



def test_connection_detection():

    # Parameters
    division_length = 300
    joint_parameters = [
        division_length,
        0.5,
        9,
        division_length * 1.5,
        0.65,
        10,
        division_length * 1.5,
        0.5,
        21,
        division_length,
        0.95,
        30,
        division_length,
        0.95,
        40,
        division_length,
        0.95,
        50,
    ]
```

```python
    # Generate connections
    result = get_connection_zones(
        data_set_plates.annen_small_polylines(),
        data_set_plates.annen_small_edge_directions(),
        data_set_plates.annen_small_edge_joints(),
        data_set_plates.annen_small_three_valance_element_indices_and_instruction(),
        None,
        joint_parameters,
    )

    # Mesh Polylines
    meshes = []
    for i in range(len(result)):
        mesh_result = closed_mesh_from_polylines(result[i])
        meshes.append(mesh_result)

    # Display via Compas_View2
    display(None, None, meshes, 0.01, 2, 0)

    # output
    return result


# ==============================================================================
# call the compas_wood methods
# ==============================================================================
test_connection_detection()
```

This is again the code snippet. Let's start the actual assignment, installation procedure and the coding part.

Notes

Summary

3m 49s