MOSM – Design for Additive Manufacturing

Tutorial:

Lattice Structure Design via Rhino GH Applications

Version 1.0

Spring 2024

Dr. Zhiping WANG

Lecturer at SeaTech, École d'ingénieurs de l'université de Toulon

zhiping.wang@univ-tln.fr

1. Introduction

Objective:

In this TP, we will learn how to do lattice design/filling via Crystallon plugin developed by using native Grasshopper components. With Crystallon, you can generate lattice structures within Grasshopper's design environment and combine other powerful tools in GH with lattice structure design.

Demonstration tool: Crystallon, Dendro, Weavebird, and Millipede.

Crystallon is an open source for creating lattice structures using Rhino and Grasshopper. To make the use of the capabilities of Crystallon, please consider installing the following plugins.

Dendro is a volumetric modeling plug-in in Grasshopper. It can help you wrap points, curves, and meshes as a volumetric data type, allowing you to perform various operations on these volumes. Dendro includes components for Boolean, smoothing, offsets, and morphing operations.

Weaverbird is a topological modeler that contains many of the known subdivision and transformation operators, readily usable by designers. Instead of doing the work repeatedly, or sometimes using complicated scripts, this plug-in reconstructs the shape, subdivides any mesh, even made by polylines, and helps preparing for fabrication.

(Millipede is a structural analysis and optimization component for grasshopper. It allows for very fast linear elastic analysis of frame and shell elements in 3d, 2d plate elements for in plane forces, and 3d volumetric elements. All systems can be optimized using built in topology optimization methods and have their results extracted and visualized in a variety of ways.)

Crystallon: https://www.food4rhino.com/app/crystallon.

Dendro: https://www.food4rhino.com/app/dendro.

Weavebird: http://www.giuliopiacentino.com/weaverbird/.

(Millipede: https://www.grasshopper3d.com/group/millipede.)

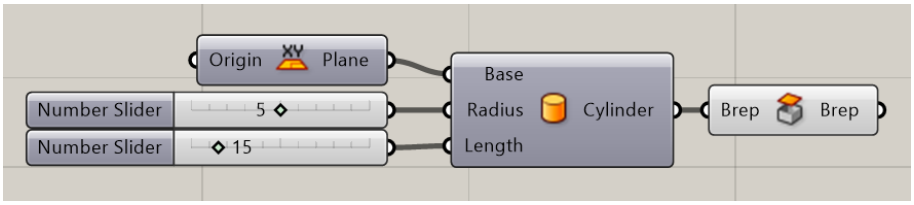
2. Tutoring example

In this tutoring example, we will create lattice structures infilling for a cylinder.

2.1. Equal size voxel infilling

Step 1: Create a cylinder geometry

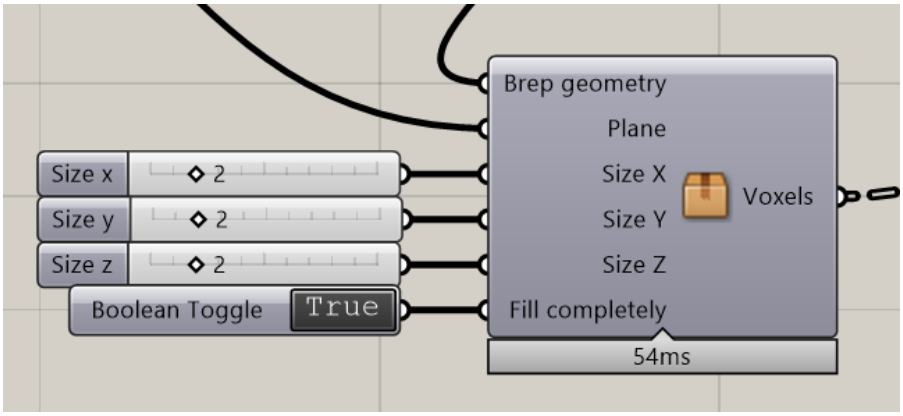
Use **Cylinder** component to generate a cylinder. (try another way to generate a cylinder.)



Step 2: Set voxel size

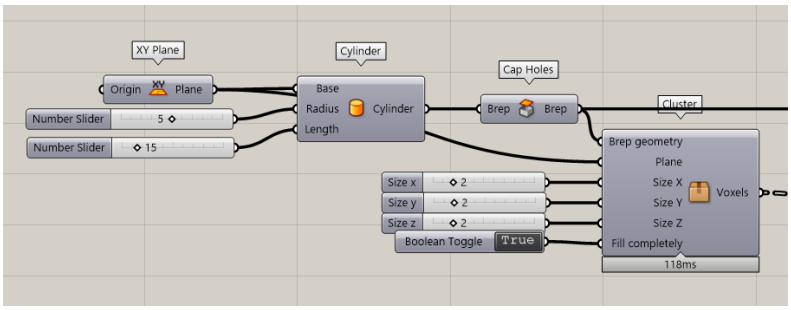
We use **Voxelize** component ***in Moodle*** to generate voxels. Here, the size of voxel is 2*2*2.

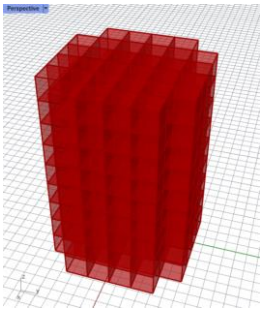
Find this component in moodle and copy it to your GH file (Use Ctrl + C).



Step 3: Generate voxels

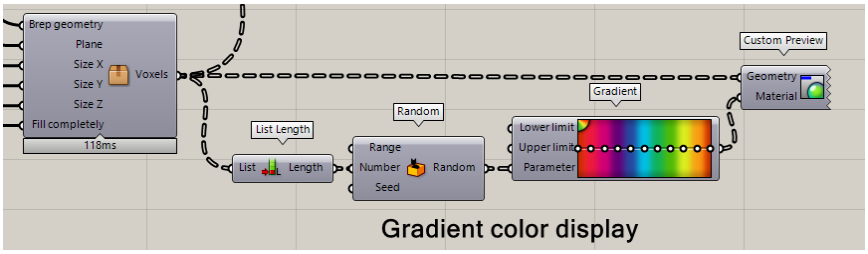
Connect closed cylinder with **Geometry**, and set **Fill Completely** as True.

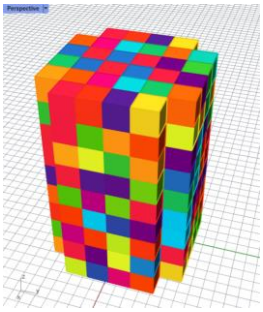




Step 4: Display voxel in random colors

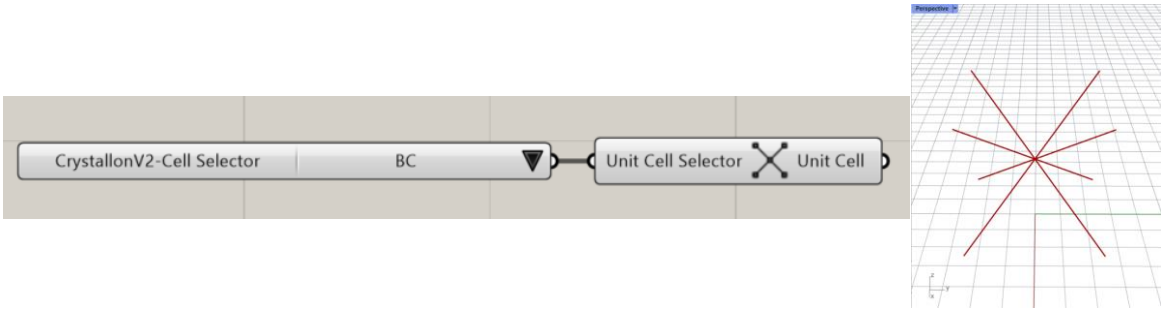
Use **Random** component to generate random values, link these random values with **Gradient** component to represent a multiple color gradient.





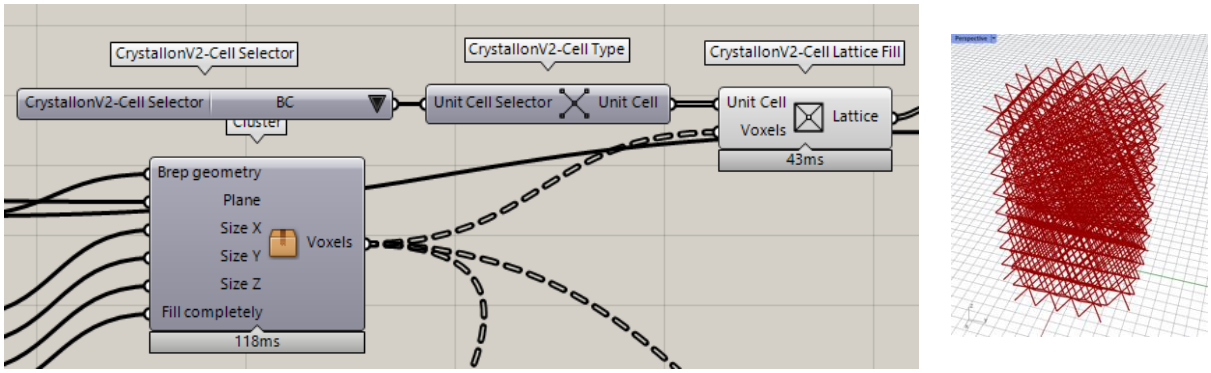
Step 5: Select a lattice type

Use **Cell Selector** and **Cell Type** to complete lattice unit cell selection. Here, we choose BC lattice unit cell.



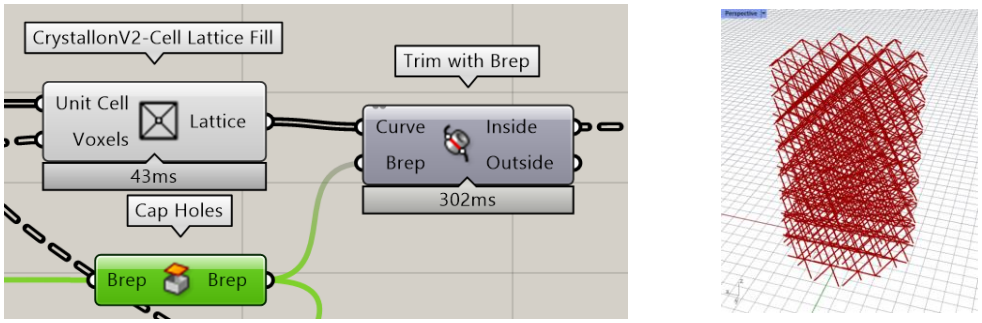
Step 6: Populate lattice unit cell into voxels

Use **Cell Lattice Fill** component to connect **unit cell** and **voxels**



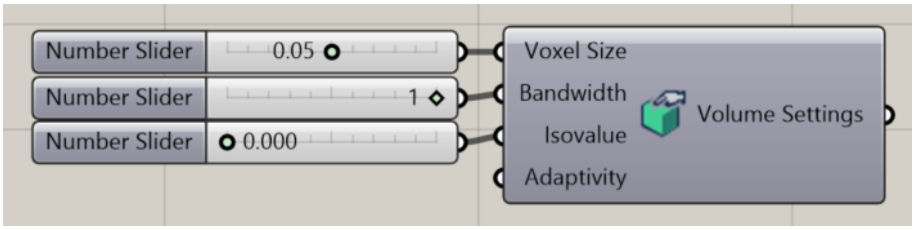
Step 7: Trim lattices

Use **Trim with Brep** component to trim the lattices.



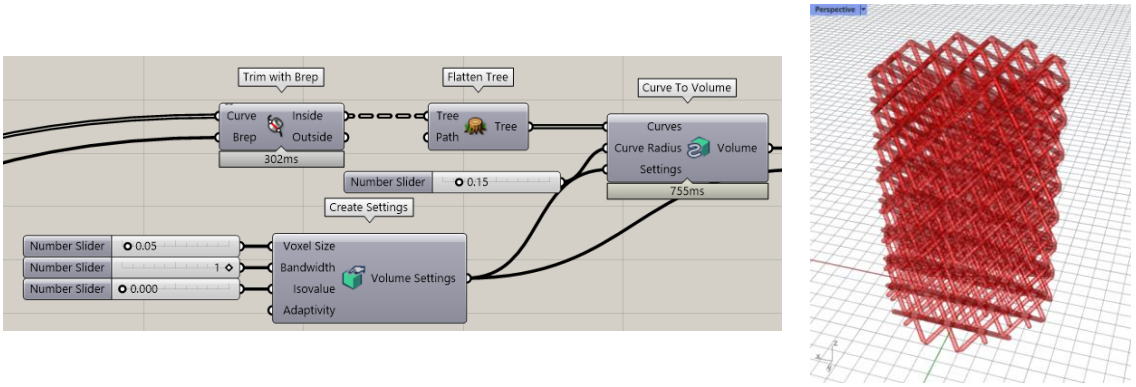
Step 8: Volume setting (Dendro)

Use Dendro plugin to generate volume. First, we need to define the **Volume Setting**.



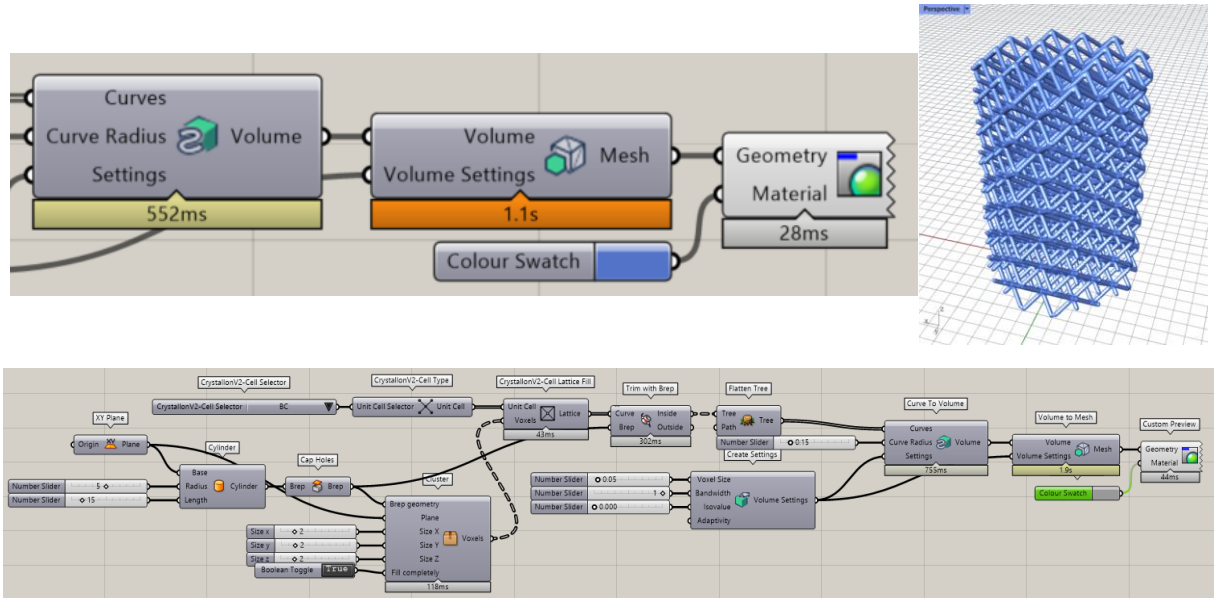
Step 9: Lattice volume generation

Connect Trimmed and untrimmed Lattice to **Curves** and connect **Volume Setting** to **Settings**. Set **Curve Radius** as 0.15mm.



Step 10: Lattice mesh generation and display

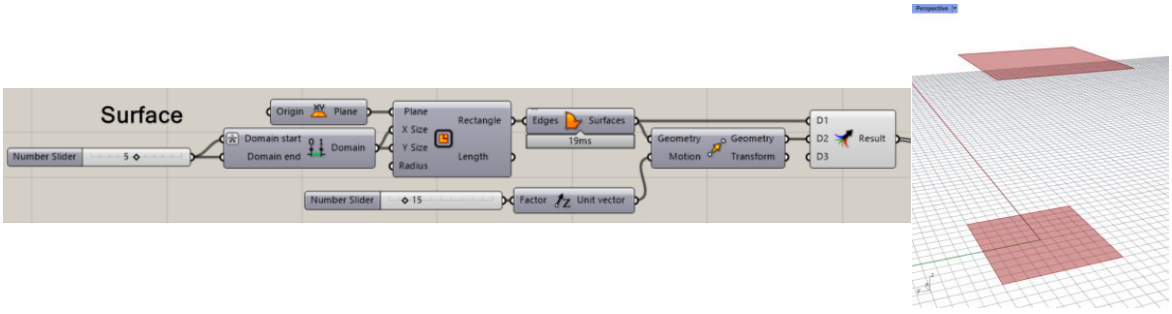
Volume to Mesh component is used to convert **Volume** to **Mesh**.



2.2. Conformal lattice generation

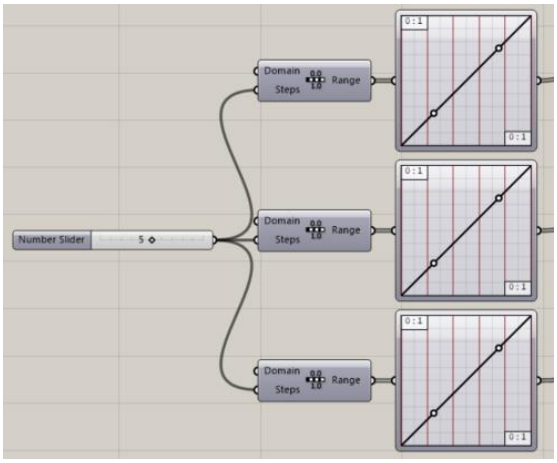
Step 1: Create two surfaces

Use **Rectangle** and **Move** components to create two surfaces.



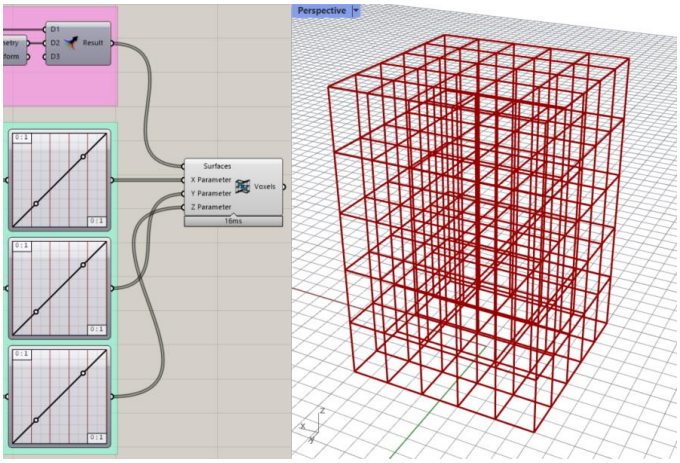
Step 2: Parameter definition

Use **Range** and **Graph Mapper** components to define X, Y and Z parameters of **Morph Between Surfaces** component.

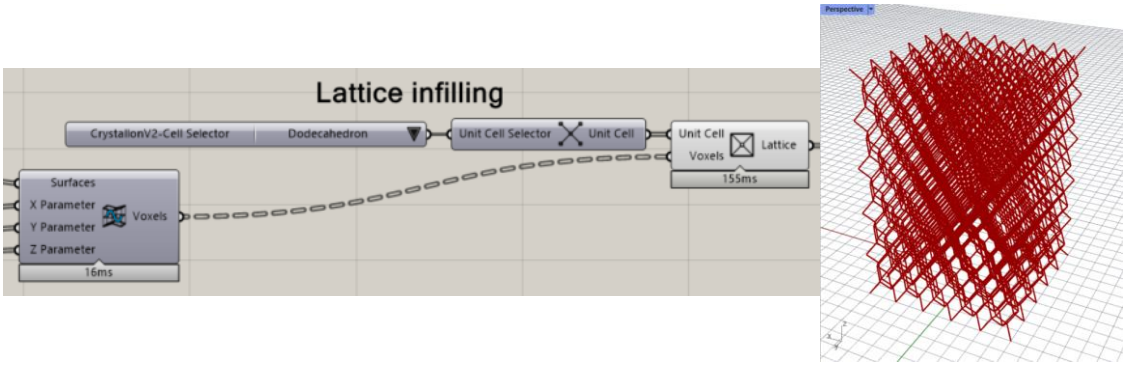


Step 3: Conformal voxel generation

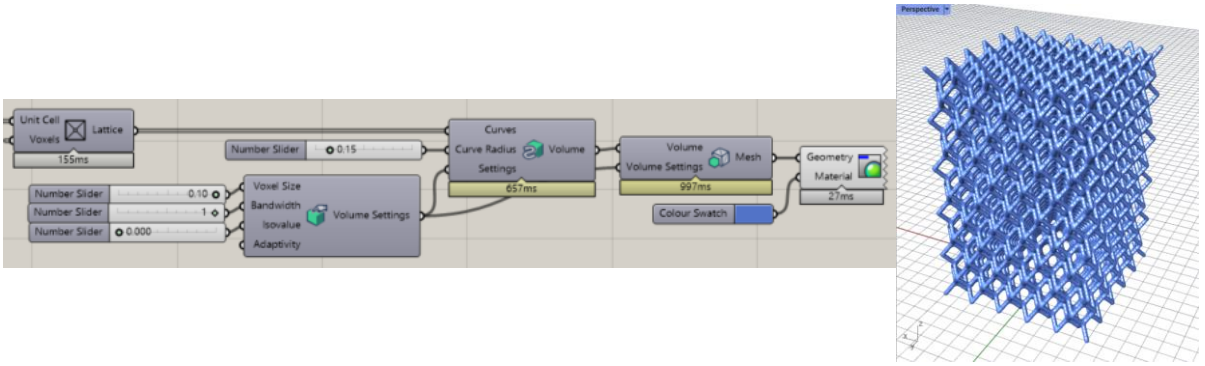
Use **Morph Between Surfaces** components to create conformal voxels. Connect surfaces and parameters with the components.



Step 4: Populate lattice unit cell (Dodecahedron) within voxels



Step 5: Lattice structure generation


Unit Cell
Voxels
Lattice
155ms
Number Slider
0.15
Curves
Curve Radius
Settings
Volume
657ms
Volume
Volume Settings
Mesh
997ms
Geometry
Material
27ms
Number Slider
0.10
Number Slider
1
Number Slider
0.000
Voxel Size
Bandwidth
Isovalue
Adaptivity
Volume Settings
Colour Swatch
Perspective