

# CRYSTALLON

Lattice structure tools for Grasshopper3D

Crystallon  
*noun*

A crystal fragment dropped into a saturated solution to provide a nucleus around which the solution may crystallize: also called seed crystal.

The word "crystallography" derives from the Greek words *crystallon* "cold drop, frozen drop", and *graphein* "to write".

CRYSTALLON V2.0 -- Lattice structure tools for Grasshopper3D

Copyright (C) 2019 Aaron Porterfield

www.fequalsf.com  
info@fequalsf.com

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

## Contents (i)

1. Abstract
2. Terminology
  - a. Key words
  - b. Anatomy of a lattice structure
4. Dependencies / External Tools
3. Overview of tools
  - a. Voxelize
    1. Voxelize Distance (Vd)
    2. Voxelize Parameter (Vp)
    3. Morph Between Surfaces (MBS)
    4. Voxel Morph Value (Vmv)
    5. Voxel Morph Attractor (VMa)
    6. Mesh Offset Parameter (MOp)
    7. Mesh Offset Value (MOv)
    8. Mesh Offset Attractor (MOa)
    9. Surface Offset Value (SOv)
    10. Surface Offset Attractor (SOa)
    11. Morph Between Meshes (MBM)
    12. Conformal Rib Distance (CRd)
    13. Conformal Rib Parameter (CRp)
    14. Conformal Pipe Distance (CPd)
    15. Conformal Pipe Parameter (CPp)
  - b. Populate
    1. Cell Type (CT)
    2. Cell Selector (CS)
    3. Cell Fill (CF)
    4. Cell Shell Fill (CSF)
    5. Tween Cell Fill (TCF)
    5. Tween Shell Fill (TSF)
    6. Cell Morph Value (CMv)
    7. Cell Morph Attractor (CMA)
    8. Voronoi Fill (VF)
  - c. Modify
    1. Trim Lattice (TL)
    2. Trim Shell (TS)
    3. Remove Floating (RF)
    4. Remove Short (RS)
    5. Remove by Valence (RV)
    6. Remove Duplicate Curves (RD)
    7. Merge Curves (MC)
    9. Lattice Connections (LC)
    10. Morph Lattice to Skin (MLS)
    11. Morph Shell to Skin (MSS)

## Contents (ii)

### d. Thicken

1. Lattice Thickness Value (LTv)
2. Lattice Thickness Attractor (LTa)
3. Shell Thickness Value (STv)
4. Shell Thickness Attractor (STa)

### e. 2D

1. Lattice Hinge (LH)
2. Lattice hinge Cell Fill (LHC)
3. Lattice hinge Tween Cell Fill (LHT)
4. Lattice Hinge Shell Fill (LHCs)
5. Lattice Hinge Tween Shell Fill (LHTs)
6. Mesh Flatten (MF)
7. Mesh Cell Fill (MCF)
8. Mesh Tween Cell Fill (MTC)
9. Mesh Shell Fill (MSF)
10. Mesh Tween Shell Fill (MTF)
11. Mesh Hinge Cell - Tri (MHt)
12. Mesh Hinge Cell - Quad (MHq)

### f. Utilities

1. Lattice Topology (LT)
2. LTCX Writer (LXO)
3. LTCX Reader (LXI)
4. INP B32 (B32)
5. INP S3 (S3)
6. INP S4 (S4)
7. INP C3D6 (C3D6)
8. INP C3D8 (C3D8)
9. INP C3D4 (C3D4)

## Abstract

Crystallon is an open source project for creating lattice structures using Rhino and Grasshopper3D. It was originally developed at FATHOM ([www.studiofathom.com](http://www.studiofathom.com)) by Aaron Porterfield ([www.fequalsf.com](http://www.fequalsf.com)) as an alternative to commercially available software for lattice design. The obvious advantage is the ability to generate lattice structures within Rhino's design environment without exporting to 3rd party software, but the extended advantages include the modularity and ability to combine other powerful tools available for Grasshopper3D. The decision to release it as open source was not only to share it with the community, but to further the development through its users. We realize not all users will have the same applications and needs, so we hope to receive feedback and improvements from users with unique case studies. Each tool is left as a cluster which can be opened and modified at will, in the spirit of open source. We hope the community will continue to develop and contribute through the use of the Grasshopper forums (<https://www.grasshopper3d.com/group/crystallon>) and Github (<https://github.com/fequalsf/Crystallon>).

V1.0 -- This is the first initial release and we will begin bug fixes immediately as they are reported. Future plans are to include utilities for import and export between more robust simulation software packages which will be included in the next release. Enjoy :)

V2.0 -- This is the second release. Updates have been made to many of the "voxelize" tools to allow for parameter based divisions using the graph mapper or other parameters as well as voxelizing curves. There are additional tools for working with meshes (triangles or quads). There is a new category called "2D" for flattening a ruled surface or a doubly curved mesh surface and applying "lattice hinge" or auxetic cells to the flat pattern. Lastly the "utilities" category includes tools for import and export of "LTCX" files from nTopology, as well as export of INP geometry files for FEA packages. Enjoy :)

## Key Words

### **Voxel**

*noun*

(In computer-based modeling or graphic simulation) an array of elements of volume that constitute a notional three-dimensional space, which a representation of a three-dimensional object is divided. The word voxel originated by analogy with the word "pixel", with vo representing "volume" and el representing "element".

### **Unit Cell**

*noun*

(In crystallography) the smallest group of atoms of a substance that has the overall symmetry of a crystal of that substance, and from which the entire lattice can be built up by repetition in three dimensions.

### **Lattice**

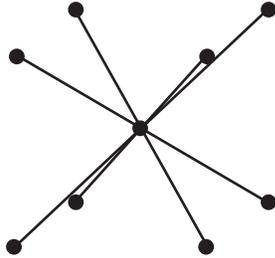
*noun*

(In material science) a regular repeated three-dimensional arrangement of atoms, ions, or molecules in a metal or other crystalline solid.

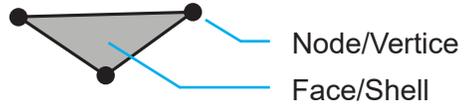
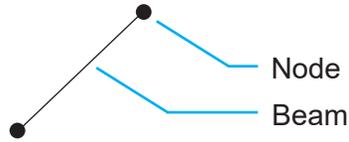
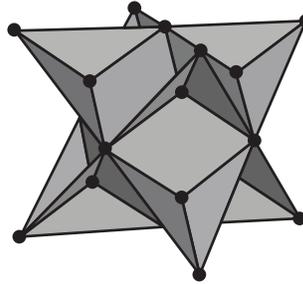
# Anatomy of a Lattice Structure

## Unit Cells

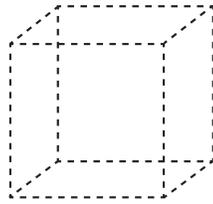
Beams (curves)



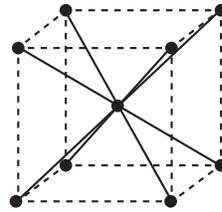
Shells (meshes)



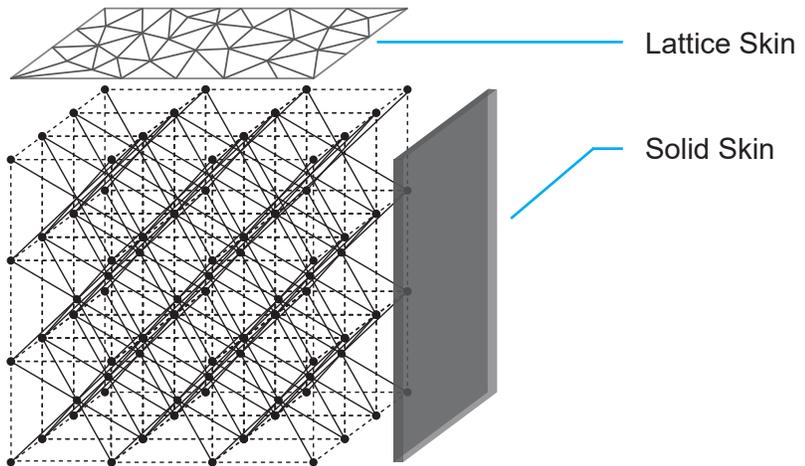
## Voxels



Voxel



Populated Voxel



Lattice

## Dependencies / External tools

Crystallon was developed to be “open source” by creating each component as open clusters using only native Grasshopper components, though, a few components do require the use of external plugins. Crystallon is also intended to be used in conjunction with other great plugins and software for things like simulation and meshing. To take full advantage of the capabilities of Crystallon, please consider installing the latest versions available of the following plugins and external software:

### **Kangaroo Physics (by Daniel Piker):**

<http://www.food4rhino.com/app/kangaroo-physics>

“Kangaroo is a Live Physics engine for interactive simulation, form-finding, optimization and constraint solving.”

### **Dendro (by ECR LABS)**

<https://www.food4rhino.com/app/dendro>

“Dendro is a volumetric modeling plug-in for Grasshopper built on top of the OpenVDB library. It provides multiple ways to wrap points, curves, and meshes as a volumetric data type, allowing you to then perform various operations on those volumes. Dendro includes components for boolean, smoothing, offsets, and morphing operations.”

### **Tetrino (by Tom Svilans)**

<https://www.food4rhino.com/app/tetrino>

“TetGen is a program to generate tetrahedral meshes of any 3D polyhedral domains. TetGen generates exact constrained Delaunay tetrahedralizations, boundary conforming Delaunay meshes, and Voronoi partitions.”

(<http://wias-berlin.de/software/index.jsp?id=TetGen&lang=1>)

### **Weaverbird (by Giulio Piacentino)**

<http://www.giuliopiacentino.com/weaverbird/>

“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 (by Panagiotis Michalatos):**

<http://www.grasshopper3d.com/group/millipede?>

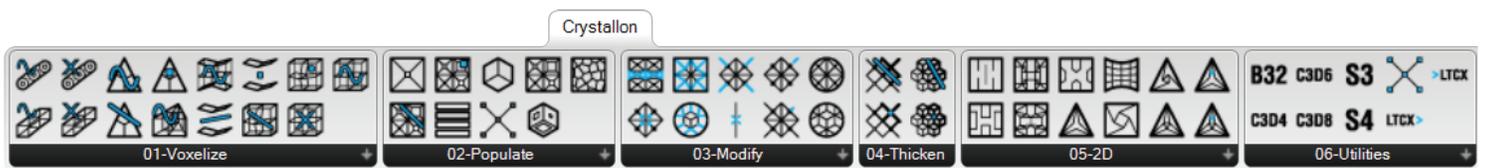
“Millipede is a structural analysis and optimization component for grasshopper. It allows for very fast linear elastic analysis of frame and shell elements in 3d, 2d plate elements for in plane forces, and 3d volumetric elements. All systems can be optimized using built in topology optimization methods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

### **Mecway**

<https://mecway.com/>

“Mecway is a comprehensive user friendly finite element analysis package for Windows with a focus on mechanical and thermal simulation such as stress analysis, vibration and heat flow. It is inexpensive and has an intuitive graphical interface for easy mesh creation and display of solutions.”

## Overview of Tools



## Voxelize

This group of tools is for generating and modifying the set of voxels which will be used to populate with unit cells to create a lattice structure.

Voxelize(d)	Vd		Surface Offset Value	SOv	
Voxelize(p)	Vp		Surface Offset Attractor	SOa	
Morph Between Surfaces	MBS		Morph Between Meshes	MBM	
Voxel Morph Value	VMv		Conformal Rib (d)	CRd	
Voxel Morph Attractor	VMa		Conformal Rib (p)	CRp	
Mesh Offset	MOp		Conformal Pipe (d)	CPd	
Mesh Offset Value	MOv		Conformal Pipe (p)	CPp	
Mesh Offset Attractor	MOa				



## Voxelize Distance (Vd)

### Description:

Fill a volume with equal sized voxels.

### Inputs:

Geometry (G) - [geometry] - Geometry to voxelize

Plane (PI) - [plane] - Base plane for voxels

Voxel Size X (X) - [number] - Size of voxels in X direction

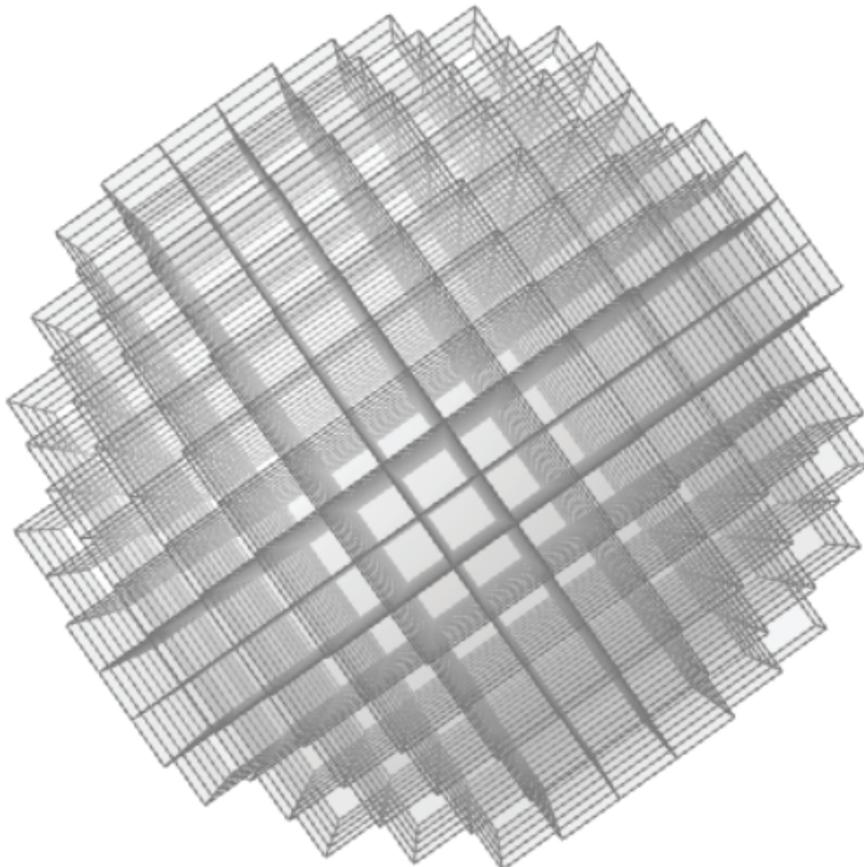
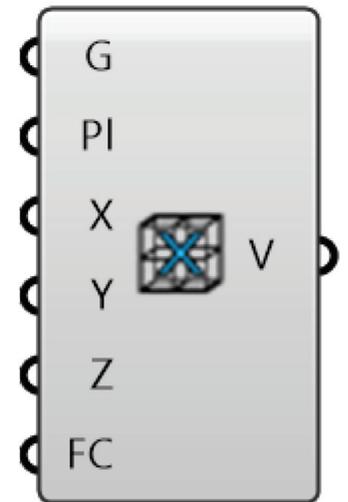
Voxel Size Y (Y) - [number] - Size of voxels in Y direction

Voxel Size Z (Z) - [number] - Size of voxels in Z direction

Fill Completely (FC) - [boolean] - If true, Voxels will fill entire volume. If false, only Voxels with their centroid within the volume will be kept.

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Voxelize Parameter (Vp)

### Description:

Fill a volume with varying sized voxels.

### Inputs:

Geometry (G) - [geometry] - Geometry to voxelize

Plane (PI) - [plane] - Base plane for voxels

X Parameter (tX) - [number] - Range of numbers (0-1) for divisions

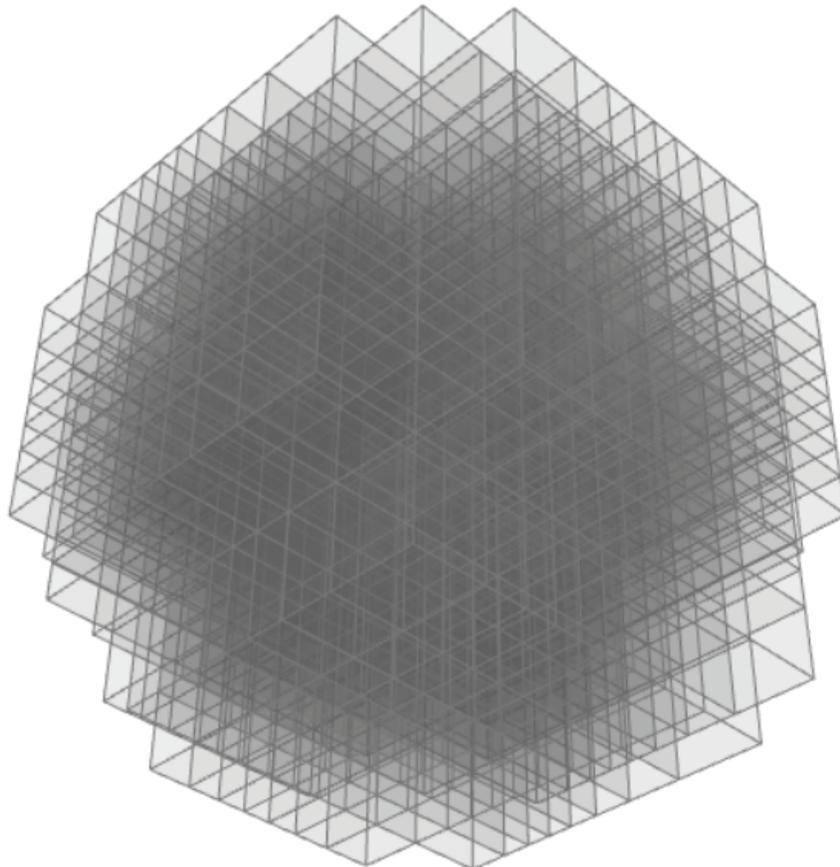
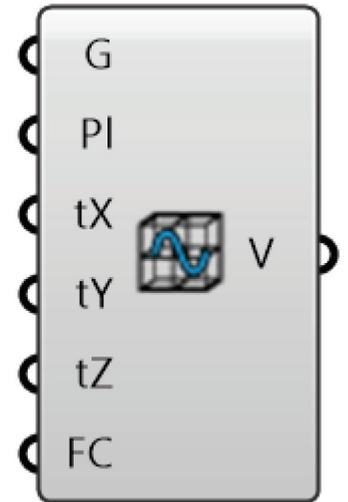
Y Parameter (tY) - [number] - Range of numbers (0-1) for divisions

Z Parameter (tZ) - [number] - Range of numbers (0-1) for divisions

Fill Completely (FC) - [boolean] - If true, Voxels will fill entire volume. If false, only Voxels with their centroid within the volume will be kept.

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Morph Between Surfaces (MBS)

### Description:

Conformal fill of voxels between two or more surfaces.

### Inputs:

Surfaces (S) - [surface] - At least 2 surfaces to morph voxels between

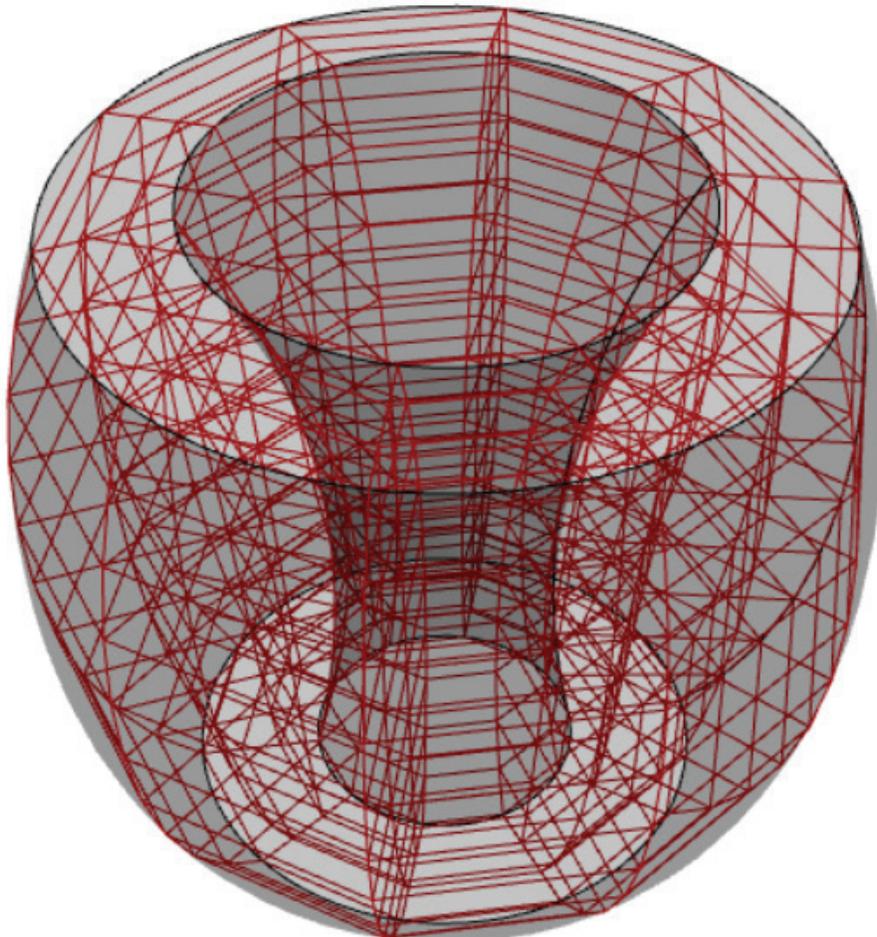
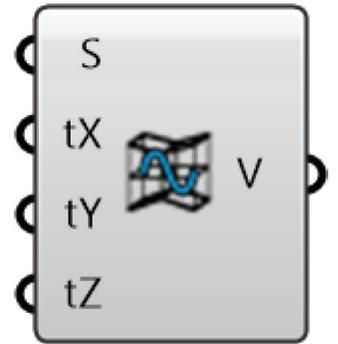
X Parameter (tX) - [number] - Range of numbers (0-1) for divisions

Y Parameter (tY) - [number] - Range of numbers (0-1) for divisions

Z Parameter (tZ) - [number] - Range of numbers (0-1) for divisions

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Voxel MorphValue (VMv)

### Description:

Morph voxels with a point cloud and corresponding values.

### Inputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Points (S) - [point] - List of points (flattened)

Values (Va) - [number] - List of values that correspond to points (flattened)

Distance Influence (DI) - [boolean] - If true the distance between the points and voxels will affect the magnitude

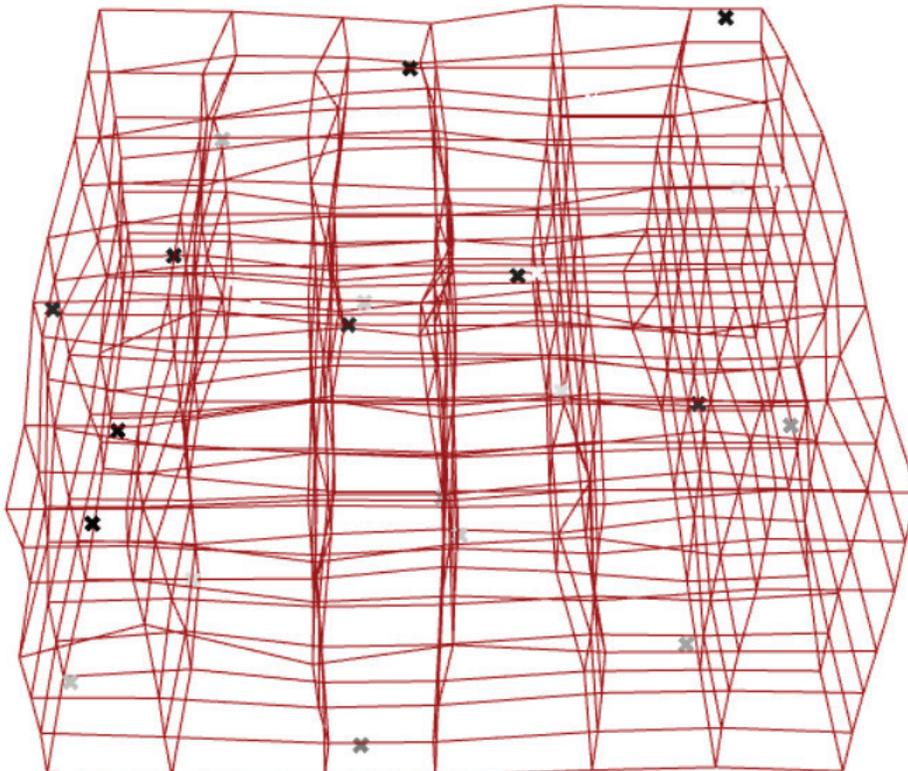
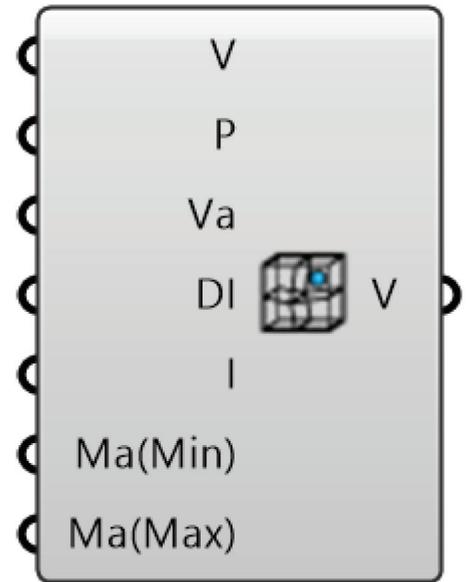
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

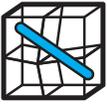
Magnitude (min) (Ma(min)) - [number] - Magnitude of the morph (minimum)

Magnitude (max) (Ma(max)) - [number] - Magnitude of the morph (maximum)

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Voxel Morph Attractor (VMa)

### Description:

Morph voxels with attractor(s).

### Inputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

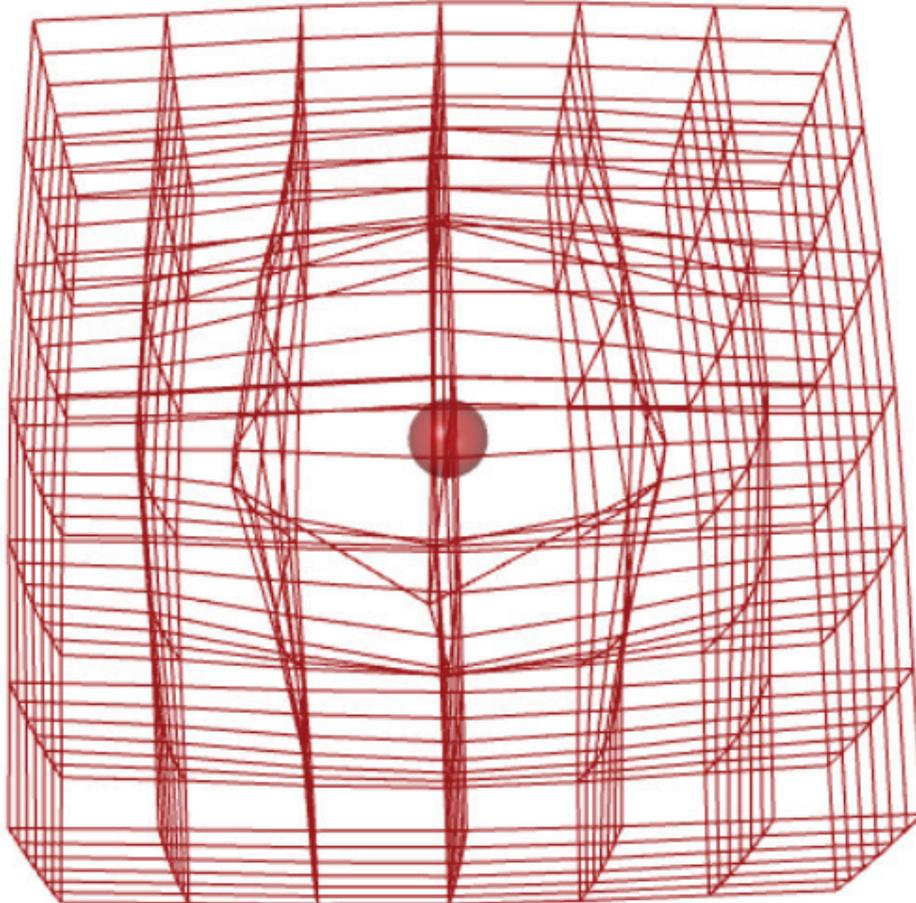
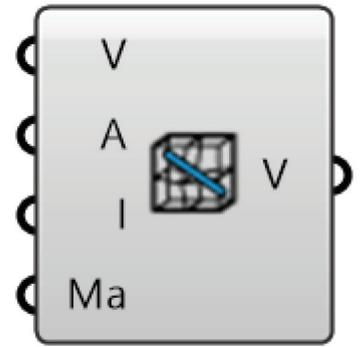
Attractor (A) - [geometry] - Any geometry to use as attractor(s)

Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (Ma) - [number] - Magnitude of the morph

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Mesh Offset Parameter (MOp)

### Description:

Offset a mesh uniformly into voxels. Quad based meshes will make a voxel per mesh face. If the mesh is triangulated, it subdivides the mesh so that each triangle becomes (3) 4-sided polygons, then offsets each point normal to the mesh surface to create a voxel on the surface.

### Inputs:

Mesh (M) - [mesh] - Mesh to offset

Triangulated (Tri) - [boolean] - True if the mesh is triangulated. False if the mesh is quads.

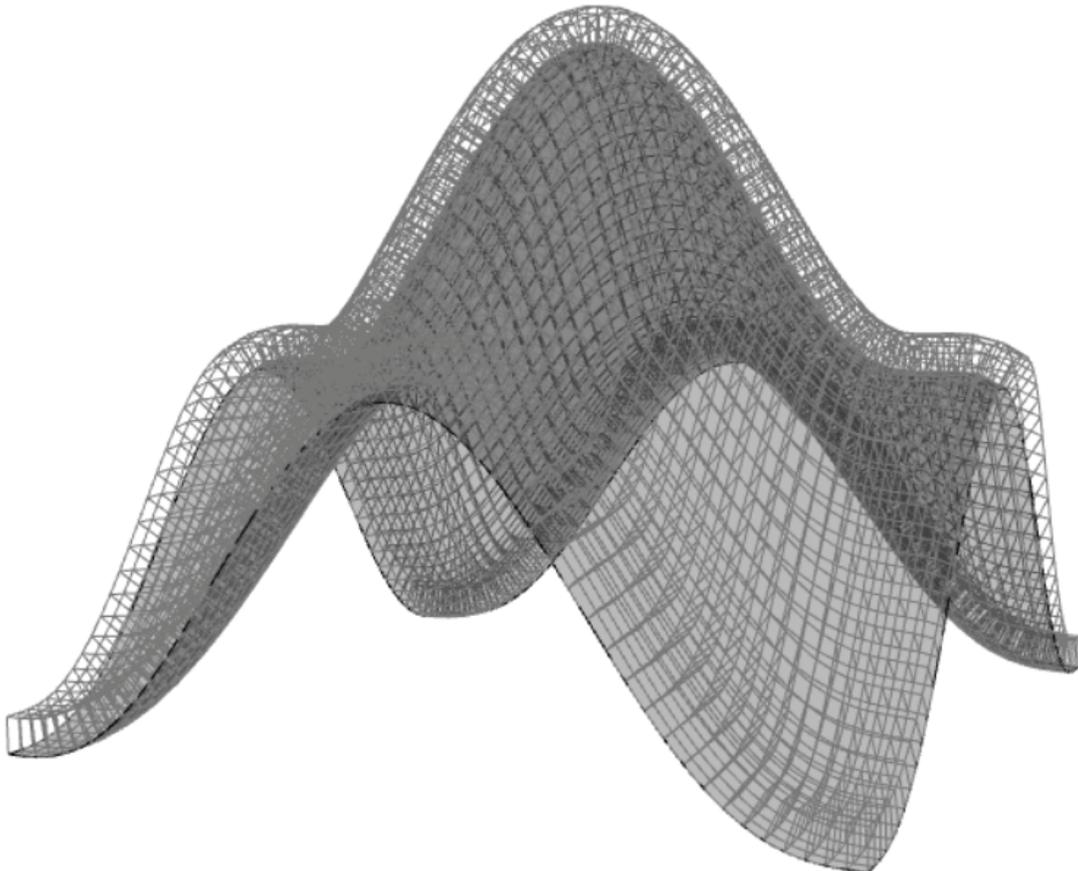
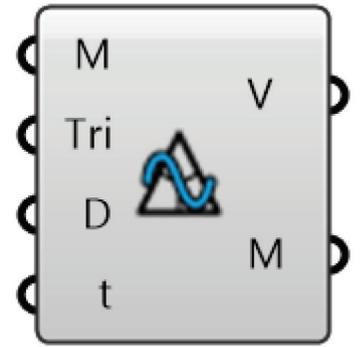
Distance (D) - [number] - Distance to offset mesh

Parameter (t) - [number] - Range of numbers (0-1) for divisions

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Quad Mesh (M) - [mesh] - Quad mesh generated from the input mesh





## Mesh Offset Value (MOv)

### Description:

Offset mesh faces into voxels with a point cloud and corresponding values.

### Inputs:

Mesh (M) - [mesh] - Mesh to offset

Triangulated (Tri) - [boolean] - True if the mesh is triangulated. False if the mesh is quads.

Points (S) - [point] - List of points (flattened)

Values (Va) - [number] - List of values that correspond to points (flattened)

Distance Influence (DI) - [boolean] - If true the distance between the points and voxels will affect the magnitude

Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - Magnitude of the offset (minimum)

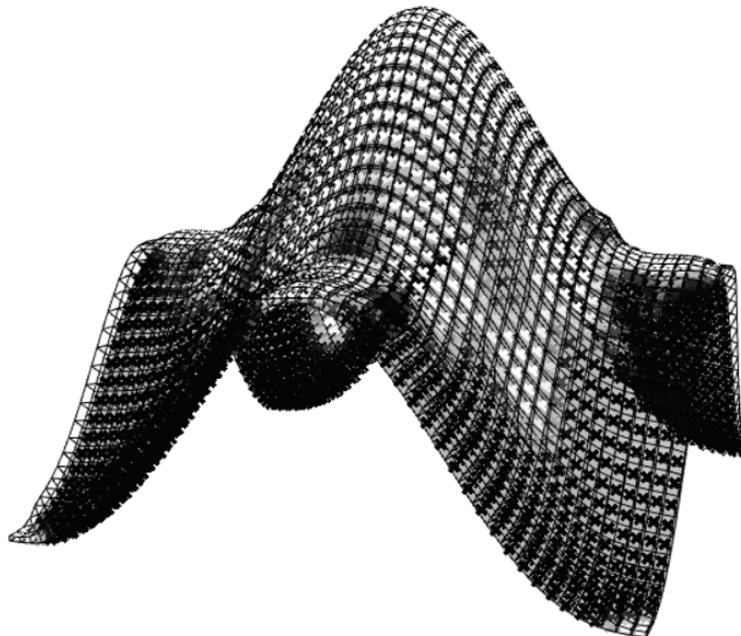
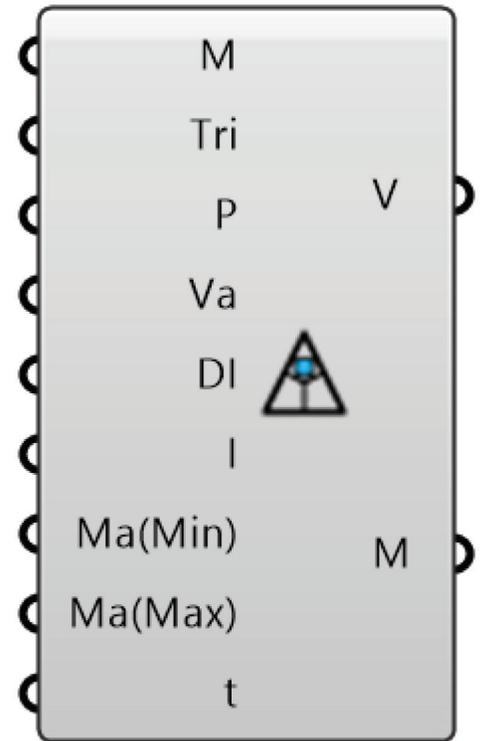
Magnitude (max) (Ma(max)) - [number] - Magnitude of the offset (maximum)

Parameter (t) - [number] - Range of numbers (0-1) for divisions

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Quad Mesh (M) - [mesh] - Quad mesh generated from the input mesh





## Mesh Offset Attractor (MOa)

### Description:

Offset mesh faces into voxels with attractor(s).

### Inputs:

Mesh (M) - [mesh] - Mesh to offset

Triangulated (Tri) - [boolean] - True if the mesh is triangulated. False if the mesh is quads.

Attractor (A) - [geometry] - Any geometry to use as attractor(s)

Lower Limit (Li) - [number] - Smoothing of the surface closest to the attractor

Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - Magnitude of the offset (minimum)

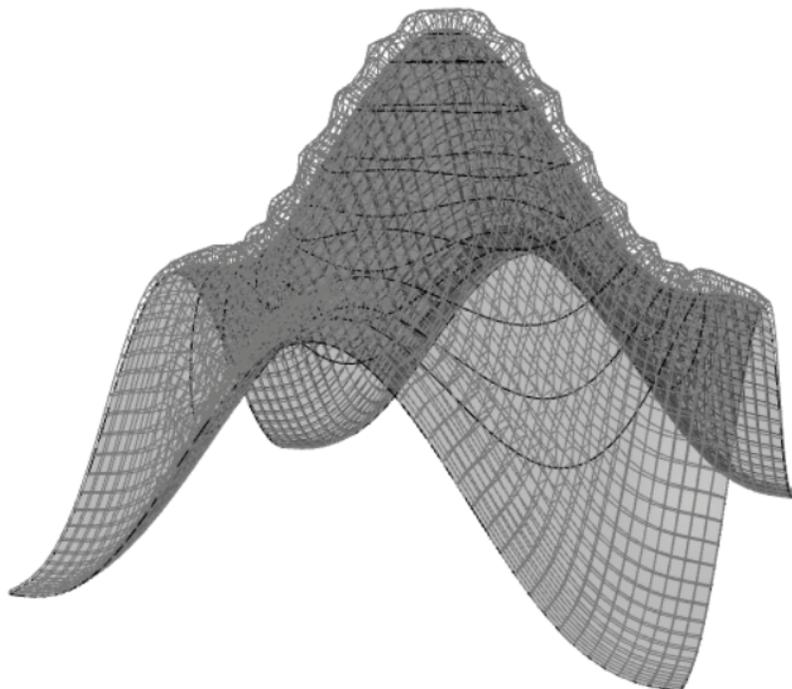
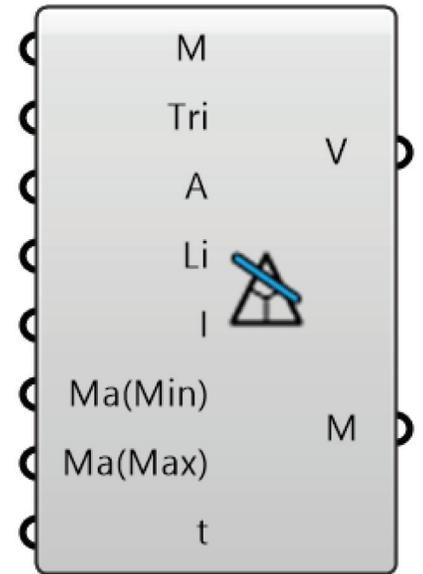
Magnitude (max) (Ma(max)) - [number] - Magnitude of the offset (maximum)

Parameter (t) - [number] - Range of numbers (0-1) for divisions

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Quad Mesh (M) - [mesh] - Quad mesh generated from the input mesh





## Surface Offset Value (SOv)

### Description:

Offset a surface with with a point cloud and corresponding values.

### Inputs:

Surface (S) - [surface] - Surface to offset

Resolution U (Re(u)) - [number] - Number of surface divisions in the U direction

Resolution V (Re(v)) - [number] - Number of surface divisions in the V direction

Points (S) - [point] - List of points (flattened)

Values (Va) - [number] - List of values that correspond to points (flattened)

Distance Influence (DI) - [boolean] - If true the distance between the points and voxels will affect the magnitude

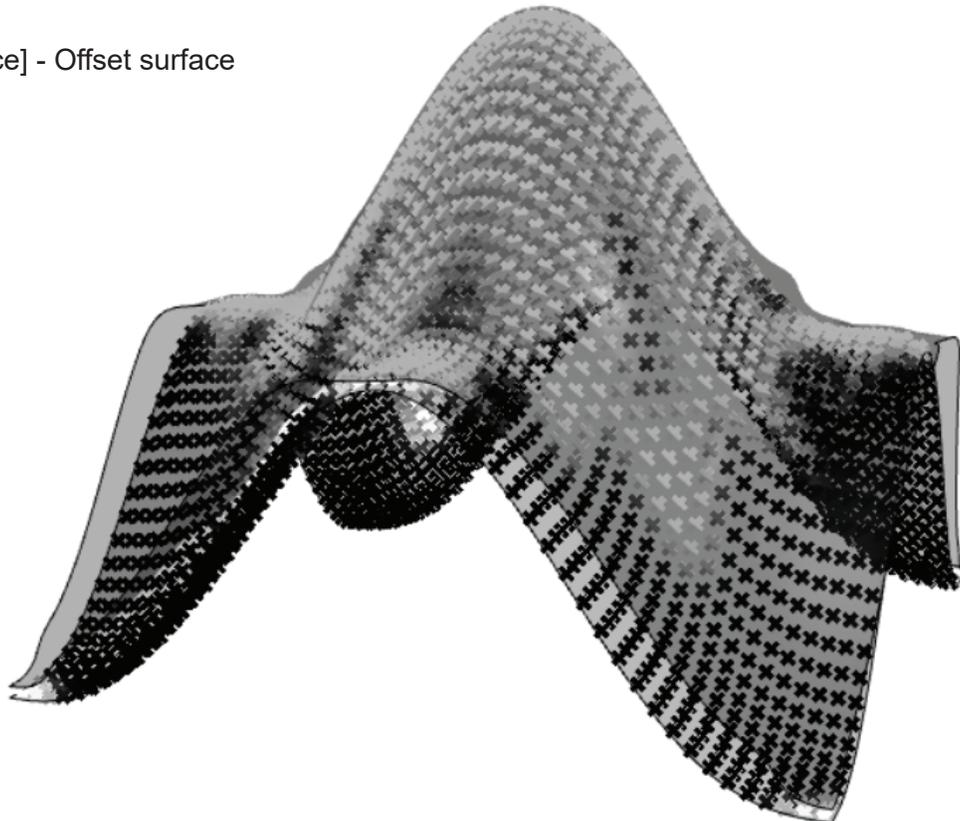
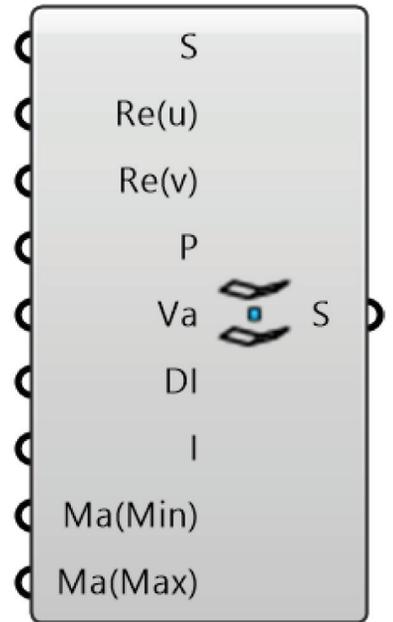
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - Magnitude of the offset (minimum)

Magnitude (max) (Ma(max)) - [number] - Magnitude of the offset (maximum)

### Outputs:

Surface (S) - [surface] - Offset surface





## Surface Offset Attractor (SOa)

### Description:

Offset a surface with attractor(s).

### Inputs:

Surface (S) - [surface] - Surface to offset

Resolution U (Re(u)) - [number] - Number of surface divisions in the U direction

Resolution V (Re(v)) - [number] - Number of surface divisions in the V direction

Attractor (A) - [geometry] - Any geometry to use as attractor(s)

Lower Limit (Li) - [number] - Smoothing of the surface closest to the attractor

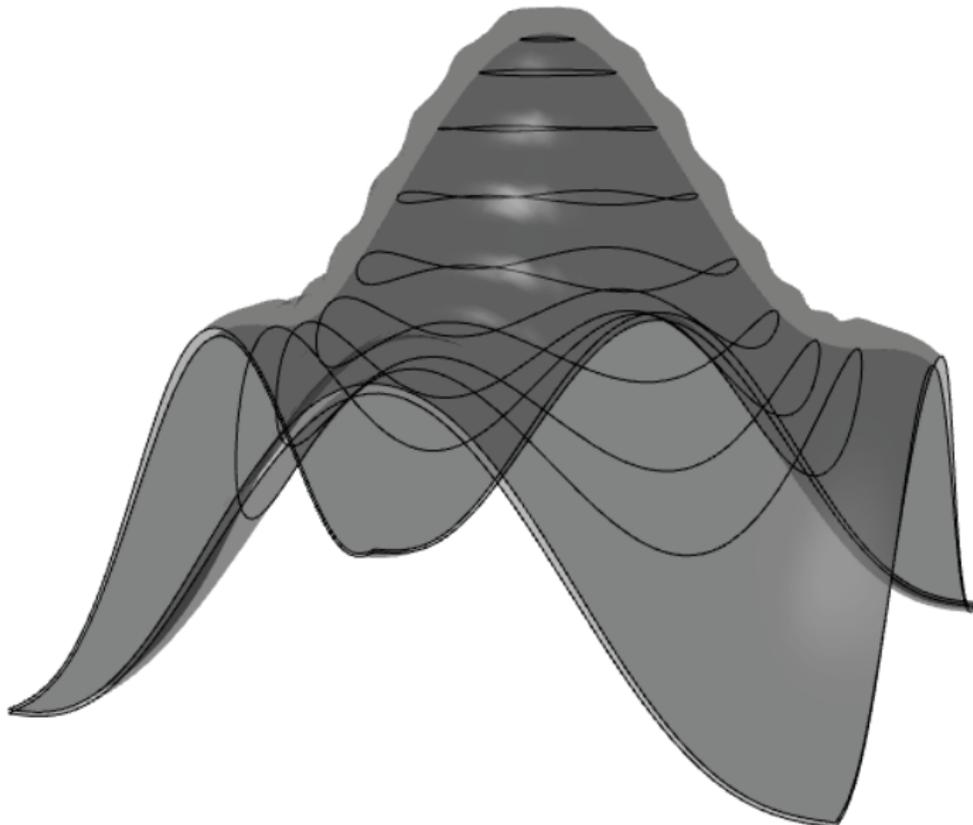
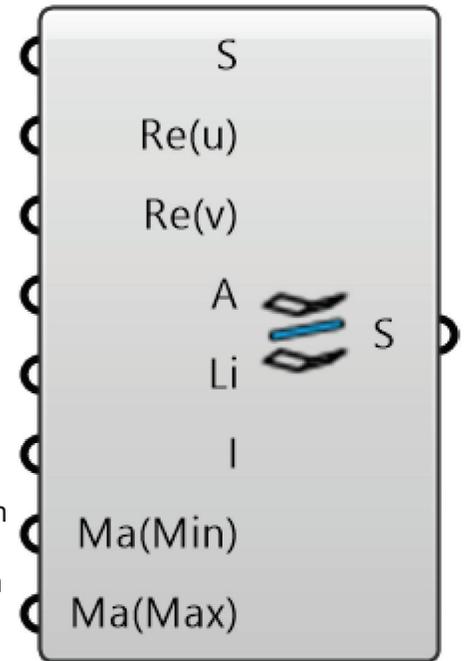
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

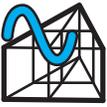
Magnitude (min) (Ma(min)) - [number] - Magnitude of the offset (minimum)

Magnitude (max) (Ma(max)) - [number] - Magnitude of the offset (maximum)

### Outputs:

Surface (S) - [surface] - Offset surface





## Morph Between Meshes (MBM)

### Description:

Create voxels between the faces of meshes with matching topology.

### Inputs:

Meshes (M) - [mesh] - Meshes to morph between

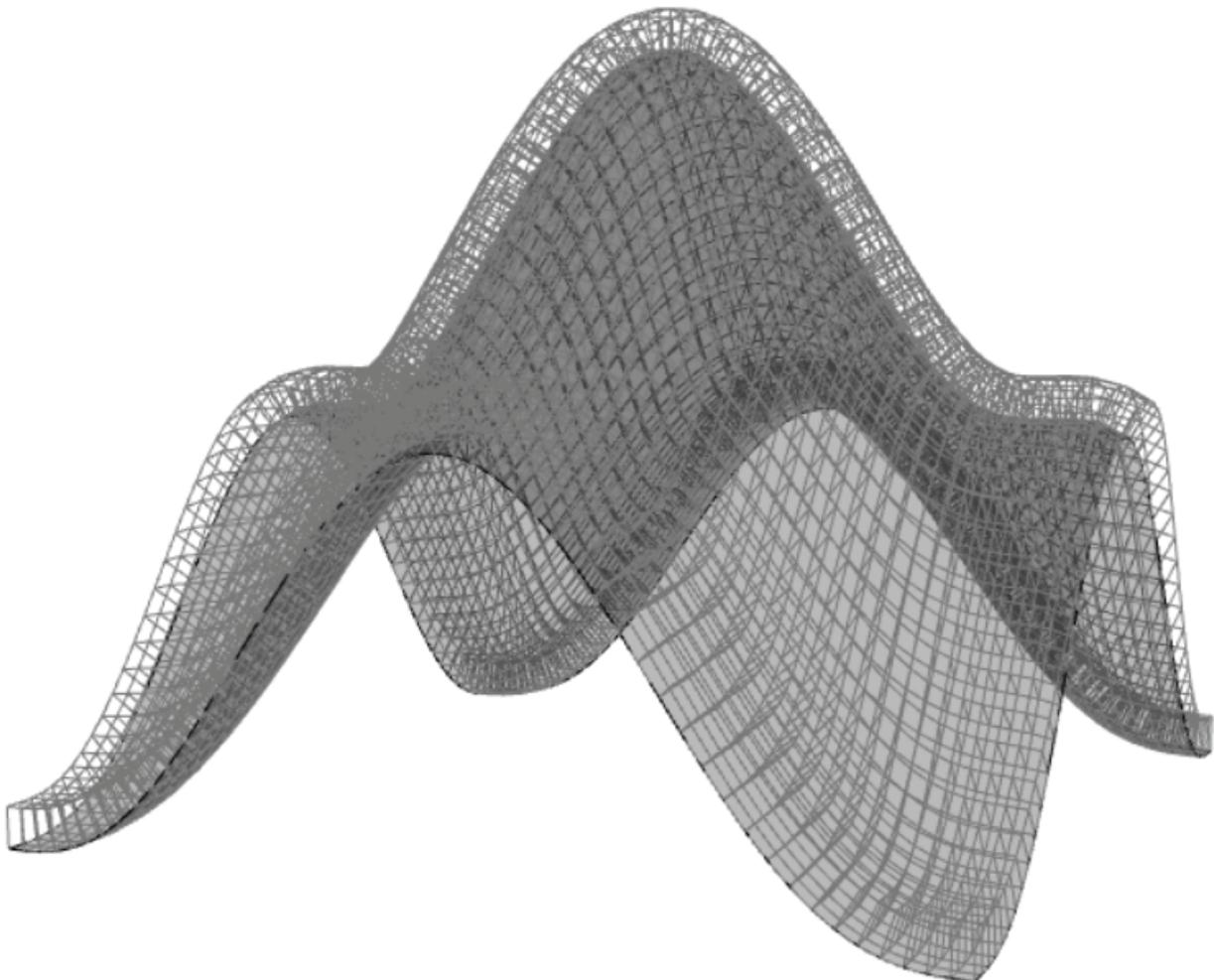
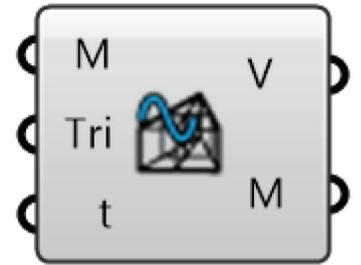
Triangulated (Tri) - [boolean] - True if the mesh is triangulated. False if the mesh is quads.

Parameter (t) - [number] - Range of numbers (0-1) for divisions

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Quad Mesh (M) - [mesh] - Quad mesh generated from the input mesh





## Conformal Rib Distance (CRd)

### Description:

Create a row of equal length voxels along a curve on a surface (mesh or brep).

### Inputs:

Geometry (G) - [geometry] - Geometry to conform to (mesh or brep)

Curve (C) - [curve] - Curve to voxelize

Periodic (P) - [boolean] - True if curve is closed

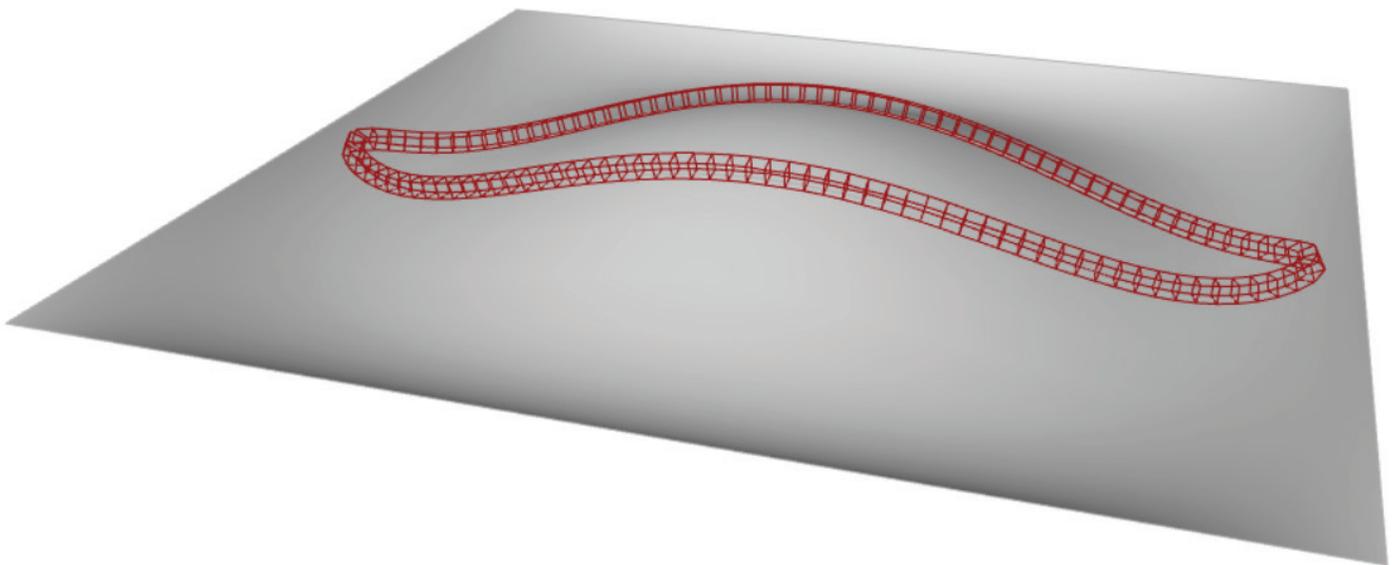
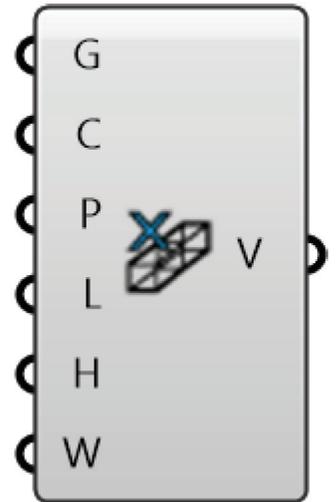
Length (L) - [number] - Length of voxels

Height (H) - [number] - Height of voxels

Width (W) - [number] - Width of voxels

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Conformal Rib Parameter (CRp)

### Description:

Create a row of varying length voxels along a curve on a surface (mesh or brep).

### Inputs:

Geometry (G) - [geometry] - Geometry to conform to (mesh or brep)

Curve (C) - [curve] - Curve to voxelize

Periodic (P) - [boolean] - True if curve is closed

Division Parameter (tD) - [number] - Parameter for voxel divisions (0-1)

Height Parameter (tH) - [number] - Parameter for voxel height (0-1)

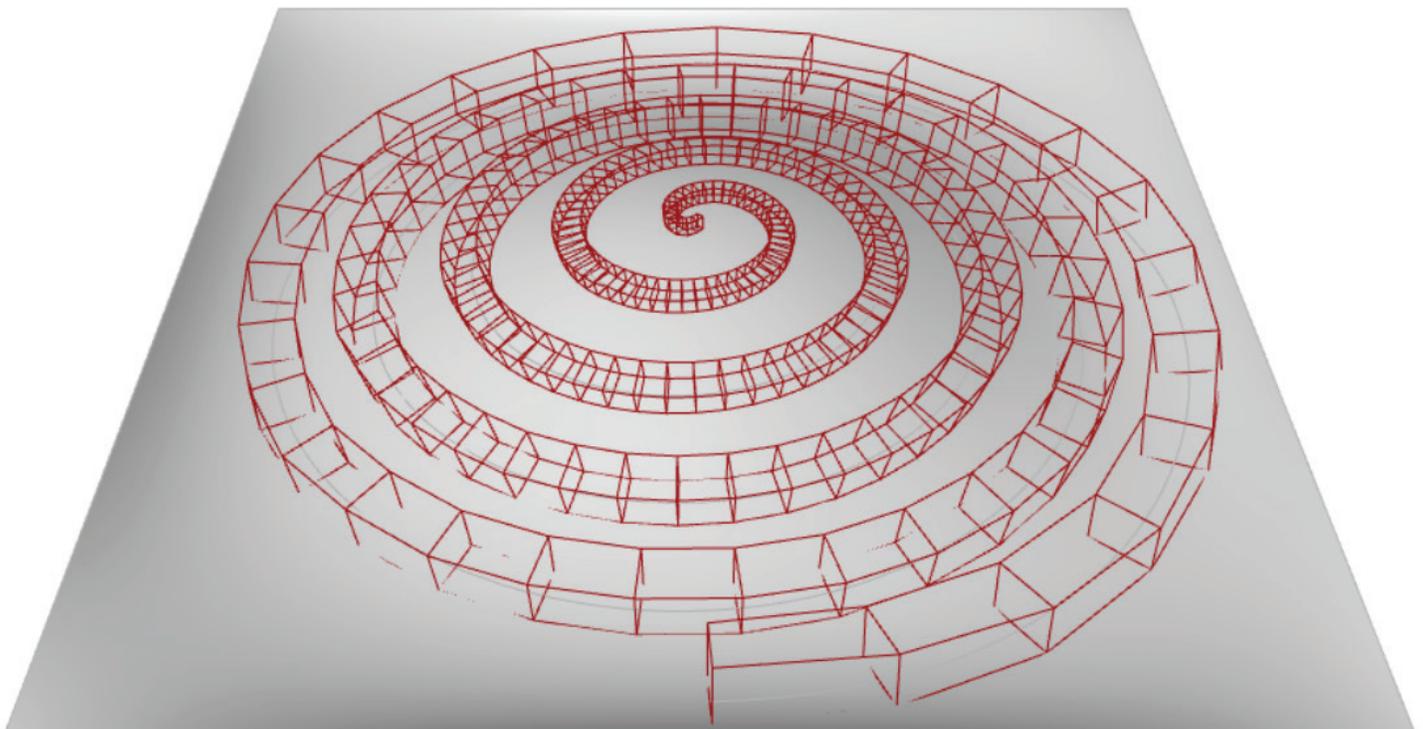
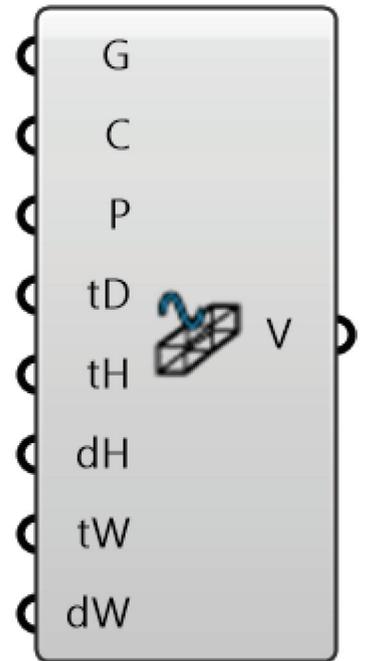
Height Domain (dH) - [domain] - Domain for voxel height

Width Parameter (tW) - [number] - Parameter for voxel width (0-1)

Width Domain (dW) - [domain] - Domain for voxel width

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Conformal Pipe Distance (CPd)

### Description:

Create a pipe of equal length voxels along a curve on a surface.

### Inputs:

Geometry (G) - [geometry] - Geometry to conform to (mesh or brep)

Curve (C) - [curve] - Curve to voxelize

Periodic (P) - [boolean] - True if curve is closed

Length (L) - [number] - Length of voxels

Distance (D) - [number] - Distance from surface

Radius (Ra) - [number] - Inner radius of pipe

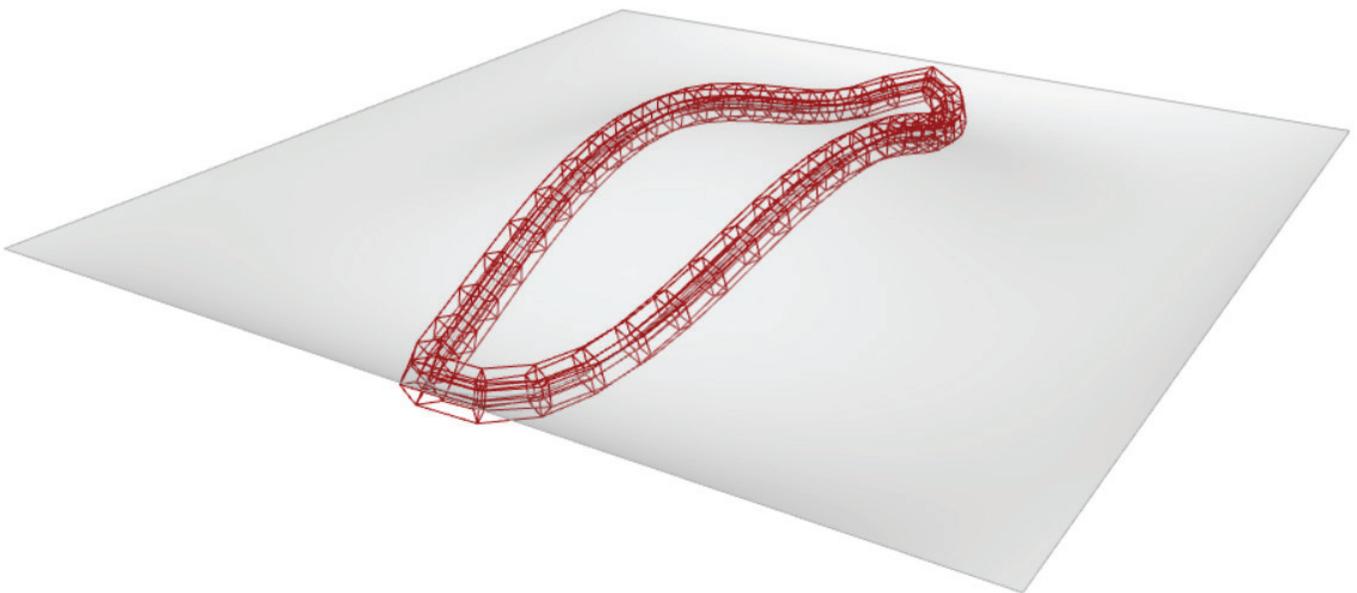
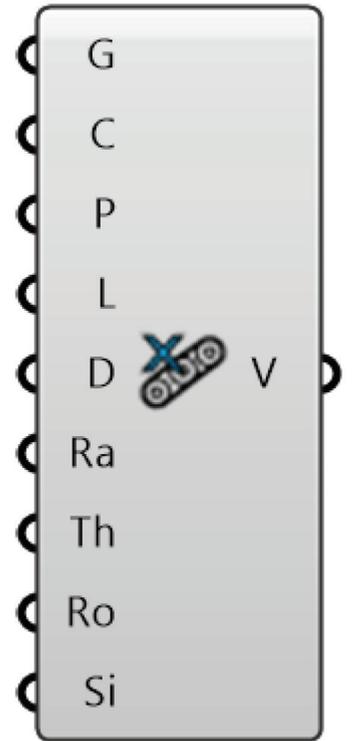
Thickness (Th) - [number] - Thickness of pipe

Rotation (Ro) - [number] - Rotation (in degrees)

Sides (Si) - [number] - Number of sides of pipe

### Outputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells





## Conformal Pipe Parameter (CPp)

### Description:

Create a pipe of varying length and size voxels along a curve on a surface.

### Inputs:

Geometry (G) - [geometry] - Geometry to conform to (mesh or brep)

Curve (C) - [curve] - Curve to voxelize

Periodic (P) - [boolean] - True if curve is closed

Division Parameter (tD) - [number] - Parameter for voxel divisions (0-1)

Distance (D) - [number] - Distance from surface

Radius Parameter (tR) - [number] - Parameter for radius (0-1)

Radius Domain (dR) - [domain] - Domain for radius

Thickness Parameter (tTh) - [number] - Parameter for thickness (0-1)

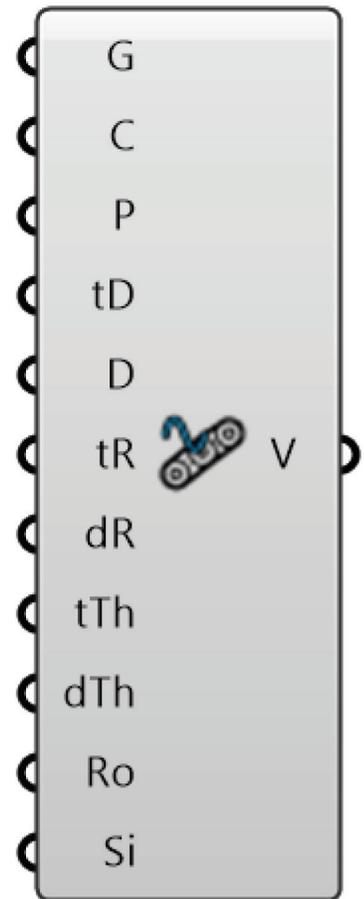
Thickness Domain (dTh) - [domain] - Domain for thickness

Rotation (Ro) - [number] - Rotation (in degrees)

Sides (Si) - [number] - Number of sides of pipe

### Outputs:

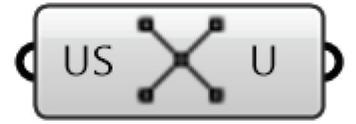
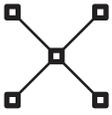
Voxels (V) - [twistedbox] - Voxels to be filled with unit cells



## Populate

This group of tools is for populating voxels with specific unit cells (beams or shells) or an entire volume with a voronoi (stochastic) lattice.

Cell Type	CT	
Cell Selector	CS	
Cell Fill	CF	
Cell Shell Fill	CSF	
Tween Cell Fill	TCF	
Tween Shell Fill	TSF	
Cell Morph Value	CMV	
Cell Morph Attractor	CMA	
Voronoi Fill	VF	



## Cell Type (CT)

### Description:

A collection of pre-made unit cells. These can also be used as reference to create your own. The data tree format is shown below.

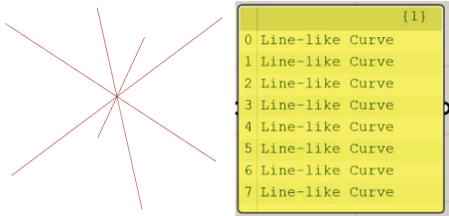
### Inputs:

Unit Cell Selector (US) - [geometry] - Unit cell selection

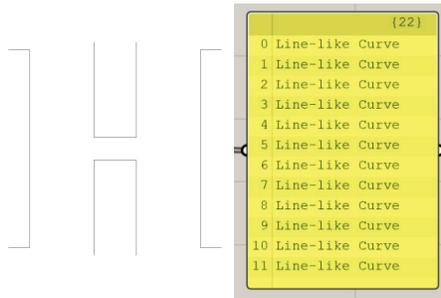
### Outputs:

Unit Cell (U) - [geometry] - Single unit cell to populate voxels

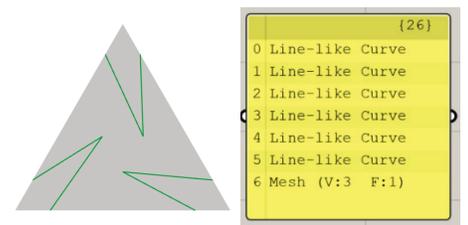
### Lattice Cell



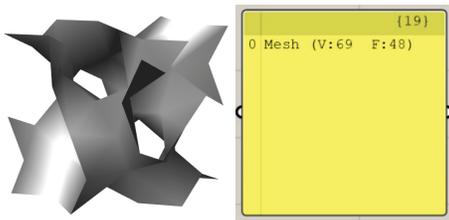
### Lattice Hinge Cell



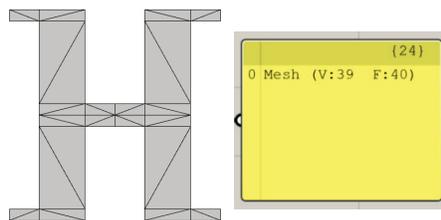
### Mesh Cell



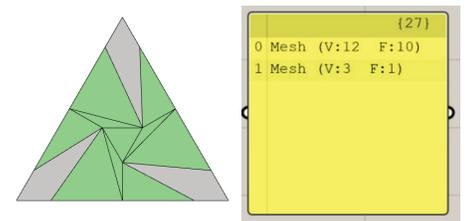
### Lattice Shell



### Lattice Hinge Shell



### Mesh Shell





### Cell Selector (CS)

#### Description:

Choose preset unit cell using the drop down menu.

#### Outputs:

Unit Cell Selector (US) - [curves / mesh] - Unit cell selection





## Cell Fill (CF)

### Description:

Fill voxels with unit cells. Use the Unit Cell Selector and Cell Type to choose from preset cells or create your own.

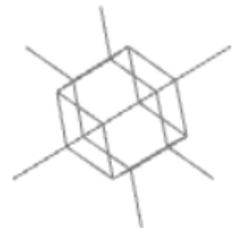
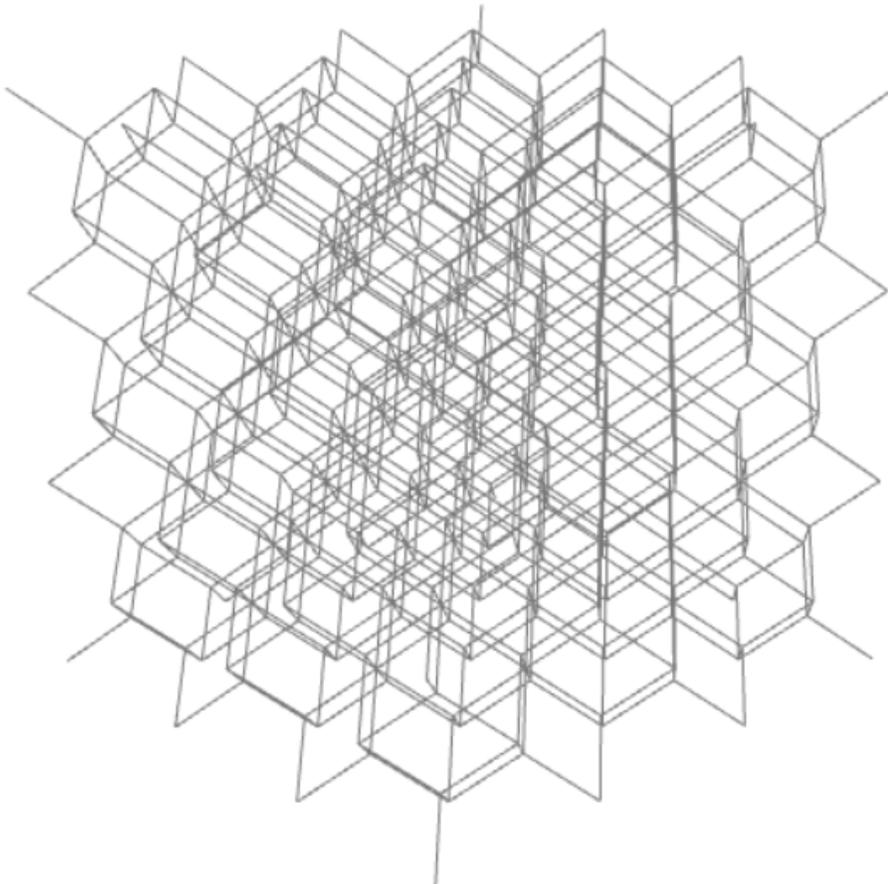
### Inputs:

Unit Cell (U) - [curves / mesh] - Single unit cell to populate voxels

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

### Outputs:

Lattice (L) - [curves] - List of curves that make up the lattice





## Cell Shell Fill (CSF)

### Description:

Fill voxels with shell unit cells. Use the Unit Cell Selector and Cell Type to choose from preset cells or create your own.

### Inputs:

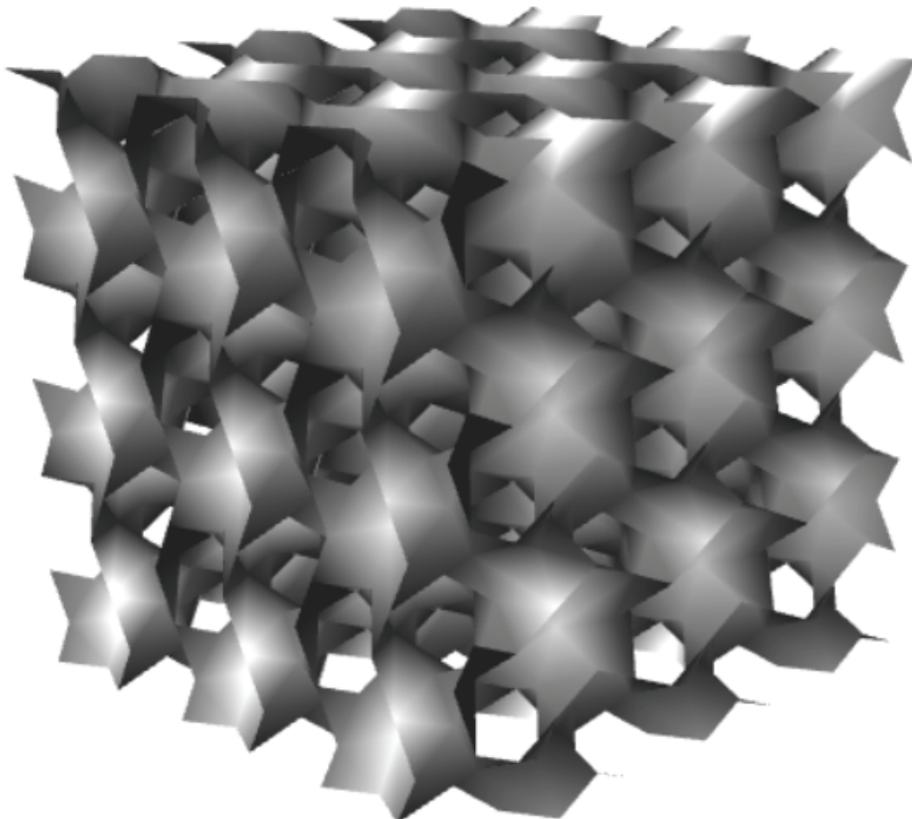
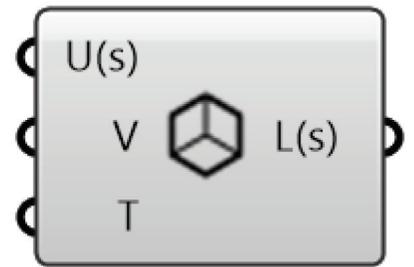
Unit Cell (shell) (U(s)) - [mesh] - Single unit cell to populate voxels

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Tolerance (T) - [number] - Tolerance for merging faces and vertices (~0.1 - 0.001)

### Outputs:

Lattice (shell) (L(s)) - [mesh] - Joined mesh that makes up the lattice





## Tween Cell Fill (TCF)

### Description:

Fill voxels with morphed lattice unit cells. Use one of the 'Cell Morph' tools to create values for each voxel then use two 'tween' type unit cells to morph between. Values range from 0 (unit cell 1) to 1 (unit cell 2). Unit cells must have the same topology (same list of curves or nodes).

### Inputs:

Unit Cell 1 (U) - [curves] - Single unit cell to populate voxels

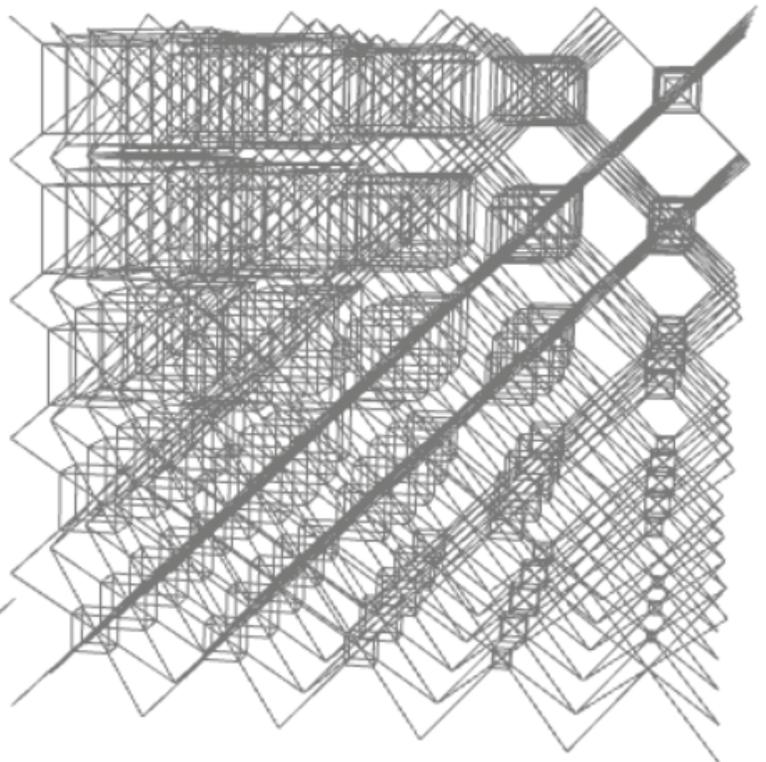
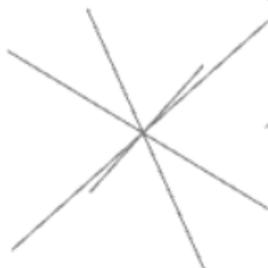
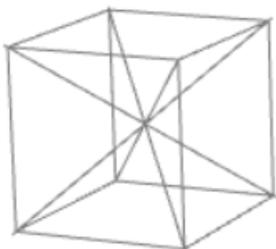
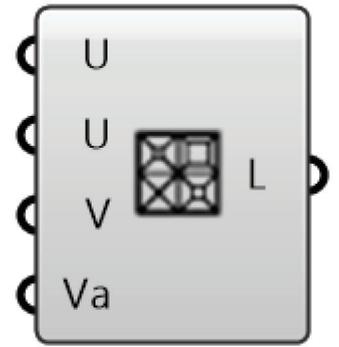
Unit Cell 2 (U) - [curves] - Single unit cell to populate voxels

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Values (Va) - [number] - List of values from 0-1 (flattened) corresponding to each voxel

### Outputs:

Lattice (L) - [curves] - List of curves that make up the lattice





## Tween Shell Fill (TSF)

### Description:

Fill voxels with morphed shell unit cells. Use one of the 'Cell Morph' tools to create values for each voxel then use two 'tween' type unit cells to morph between. Values range from 0 (unit cell 1) to 1 (unit cell 2). Unit cells must have the same topology (same list of curves or nodes).

### Inputs:

Unit Cell 1 (U(s)1) - [mesh] - Single unit cell to populate voxels

Unit Cell 2 (U(s)2) - [mesh] - Single unit cell to populate voxels

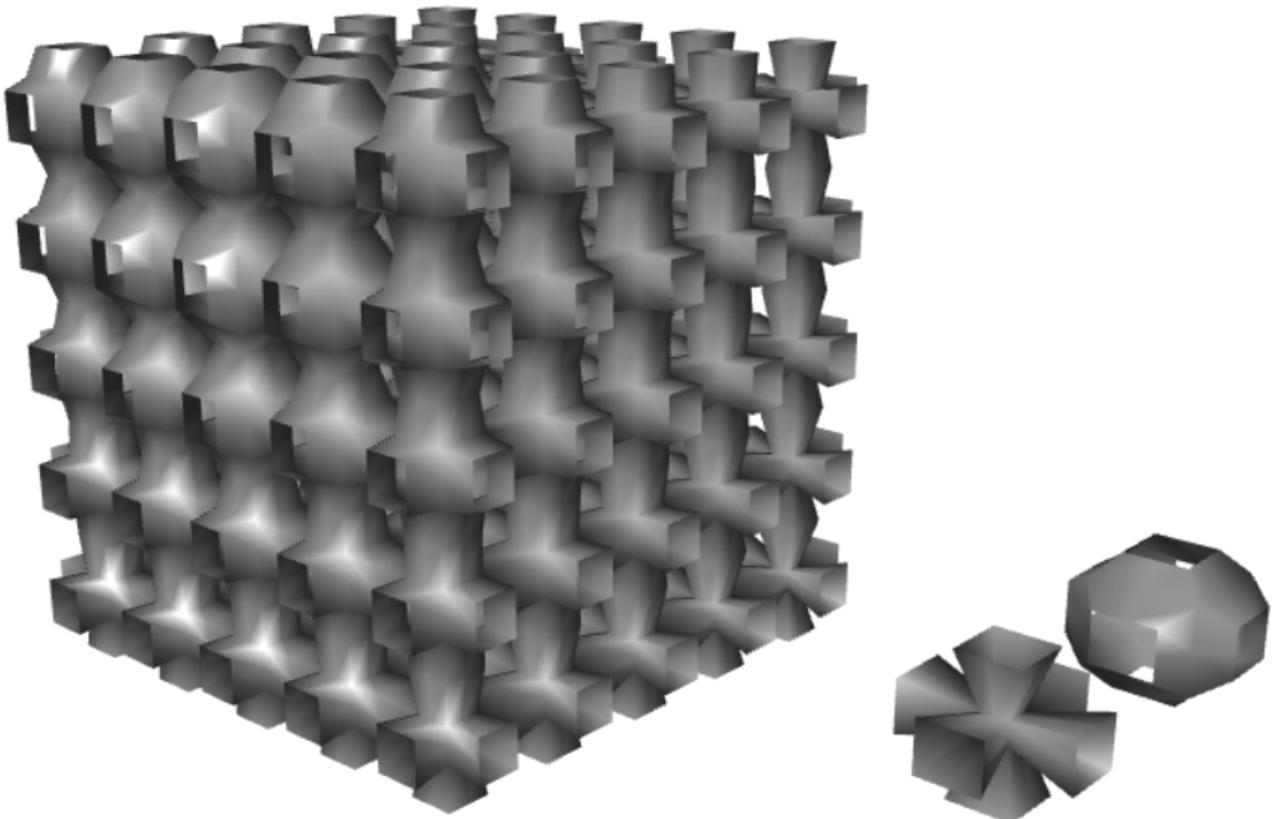
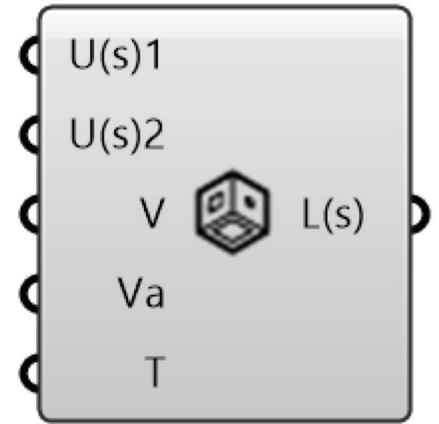
Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

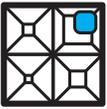
Values (Va) - [number] - List of values from 0-1 (flattened) corresponding to each voxel

Tolerance (T) - [number] - Tolerance for merging faces and vertices (~0.1 - 0.001)

### Outputs:

Lattice (shell) (L(s)) - [mesh] - Joined mesh that makes up the lattice





## Cell Morph Value (CMv)

### Description:

Morph unit cells with a point cloud and corresponding values.

### Inputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Points (S) - [point] - List of points (flattened)

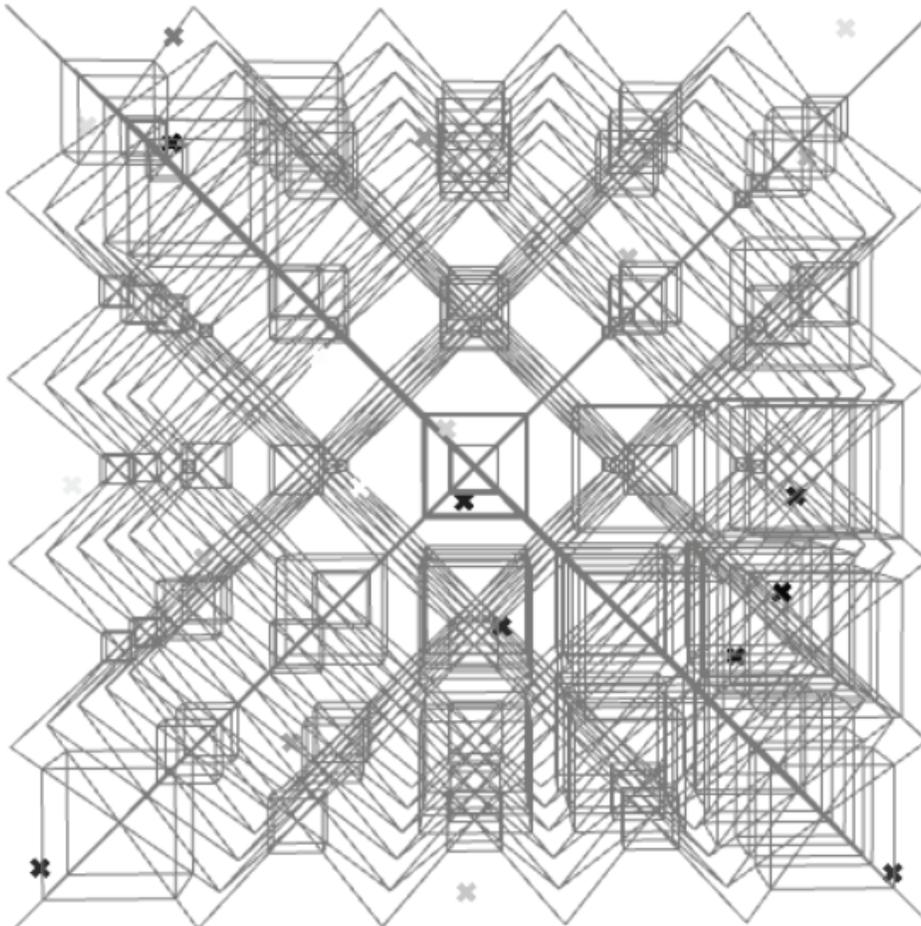
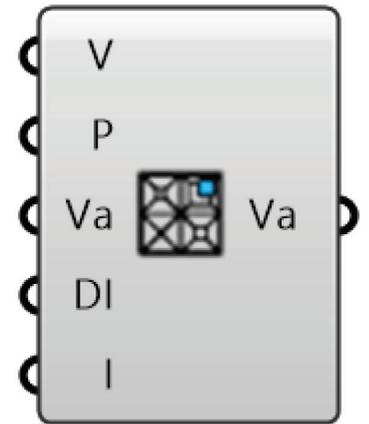
Values (Va) - [number] - List of values that correspond to points (flattened)

Distance Influence (DI) - [boolean] - If true the distance between the points and voxels will affect the magnitude

Influence (I) - [number] - Smoothed range of attractor influence (0-1)

### Outputs:

Values (Va) - [number] - List of values from 0-1 corresponding to each voxel





## Cell Morph Attractor (CMA)

### Description:

Morph unit cells with attractor(s).

### Inputs:

Voxels (V) - [twistedbox] - Voxels to be filled with unit cells

Attractor (A) - [geometry] - Any geometry to use as attractor(s)

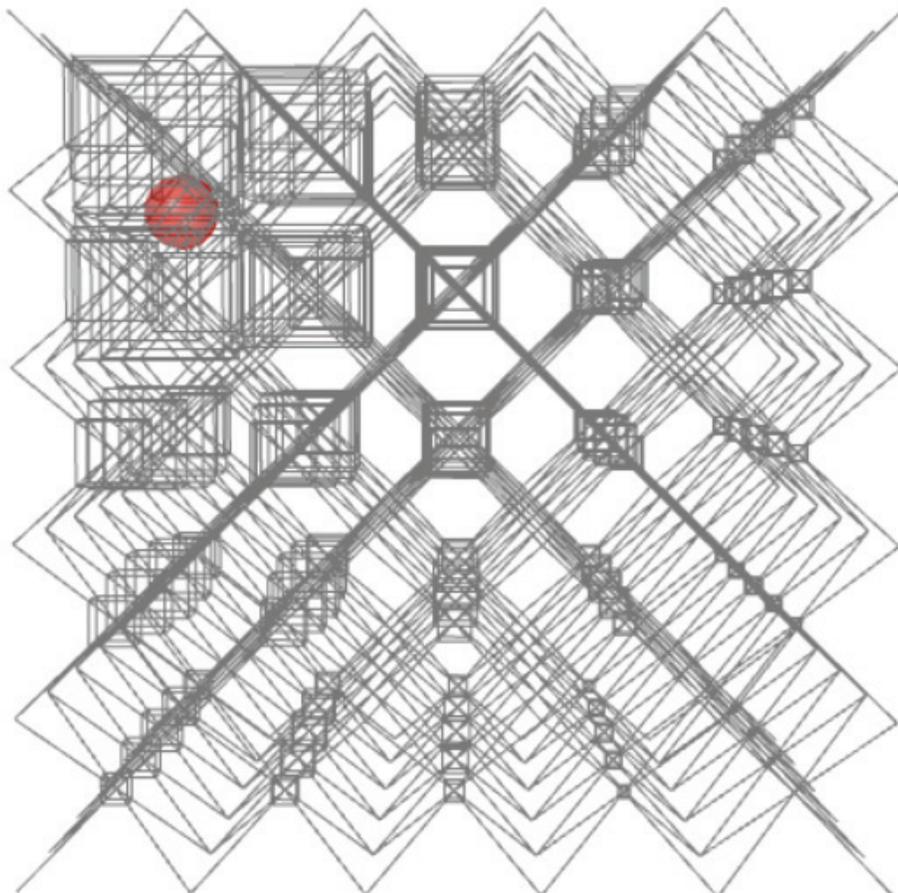
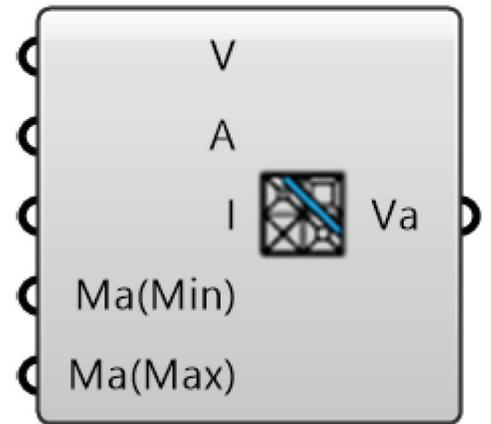
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - Magnitude of the morph (minimum)

Magnitude (min) (Ma(min)) - [number] - Magnitude of the morph (minimum)

### Outputs:

Values (Va) - [number] - List of values from 0-1 corresponding to each voxel





## Voronoi Fill

### Description:

Fill a volume with randomized voronoi cells. Use attractors to vary the density.

### Inputs:

Geometry (G) - [geometry] - Geometry to fill

Number of points (N) - [number] - The initial number of random points to fill the volume

Use Attractor (UA) - [boolean] - If true, points will be filtered by distance to attractor(s)

Attractor (A) - [geometry] - Any geometry to use as attractor(s)

Magnitude (Ma) - [number] - Magnitude of the random filtering

Number of filtered points (N) - [number] - The number of points to filter

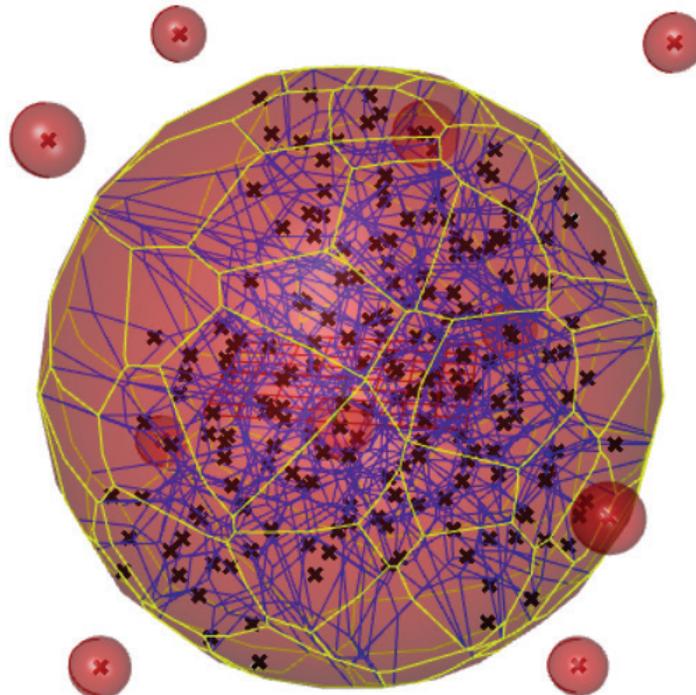
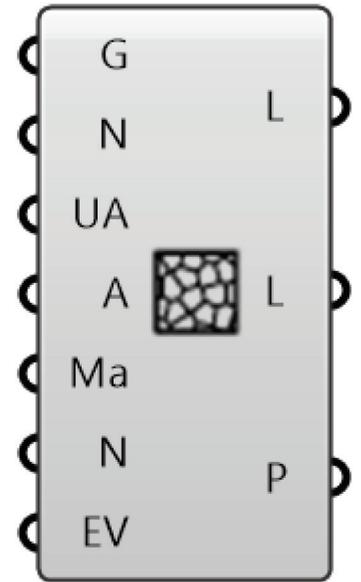
Enable Voronoi (EV) - [boolean] - If true, voronoi cells will be generated (can be very slow)

### Outputs:

Voronoi Lattice (L) - [curves] - List of curves that make up the voronoi lattice

Voronoi Lattice Skin (L) - [curves] - List of curves that make up the voronoi skin

Preview Points (P) - [points] - Preview of points that will make up the voronoi lattice



## Modify

This group of tools is for the “clean up” of the populated lattice by trimming to a specific volume, removing unwanted beams, and connecting exterior beams to a “skin” or another lattice.

Trim lattice	TL	
Trim Shell	TS	
Remove Floating	RF	
Remove Short	RS	
Remove by Valence	RV	
Remove Duplicate Curves	RD	
Merge Curves	MC	
Lattice Connections	LC	
Morph Lattice to Skin	MLS	
Morph Shell to Skin	MSS	



## Trim Lattice

### Description:

Trim lattice with a closed volume (mesh or brep). This will trim the lattice curves to within the boundary of a closed volume.

### Inputs:

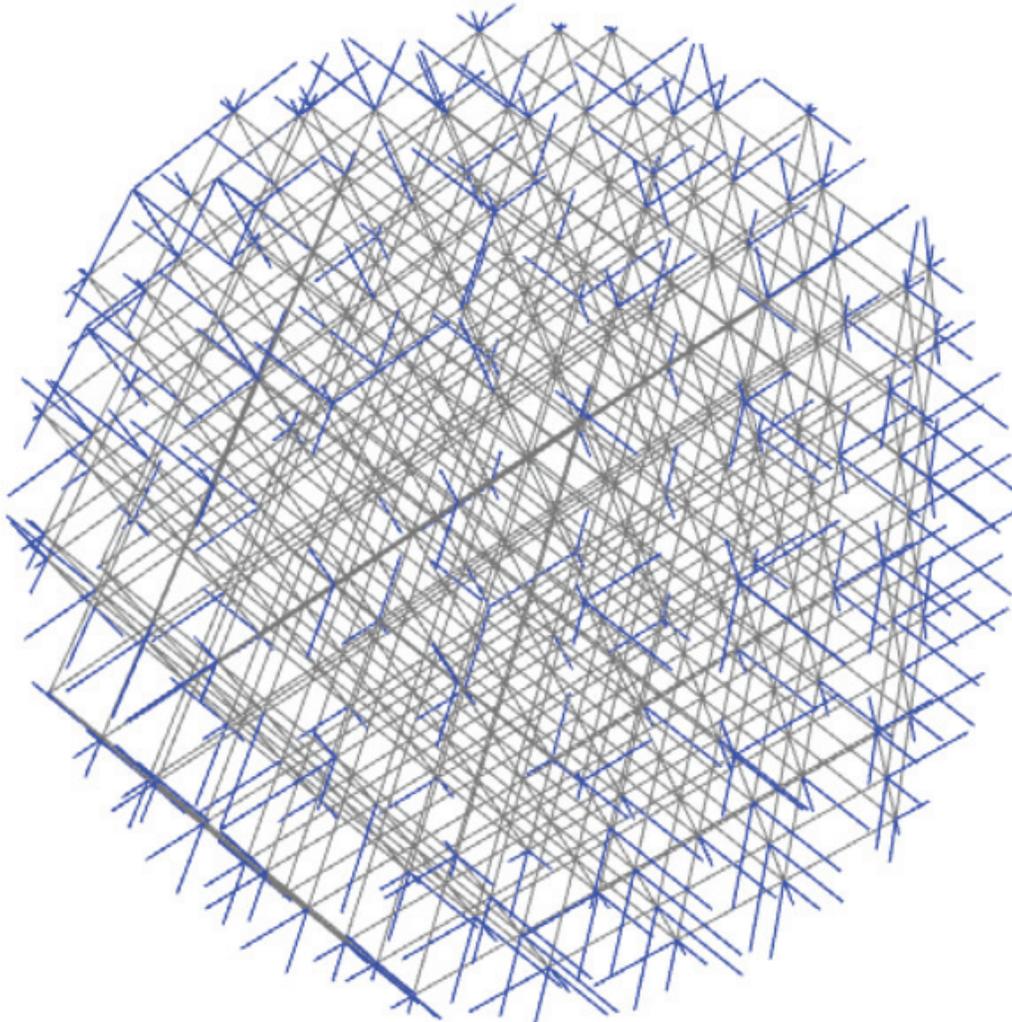
Geometry (G) - [brep / mesh] - Geometry to trim with (closed brep or mesh)

Lattice (L) - [curves] - Lattice curves to trim

### Outputs:

Trimmed Lattice (L) - [curves] - Lattice curves that have been trimmed

Untrimmed Lattice (L) - [curves] - Lattice curves that have not been trimmed (Typically removes curves with valence 1)





## Trim Shell

### Description:

Trim shell lattice with a closed volume (mesh or brep). This will trim a shell by removing the mesh vertices that are outside of the boundary of a closed mesh or brep. Subdividing the mesh of the shell to add more vertices before trimming will make a more accurate trim but will take longer.

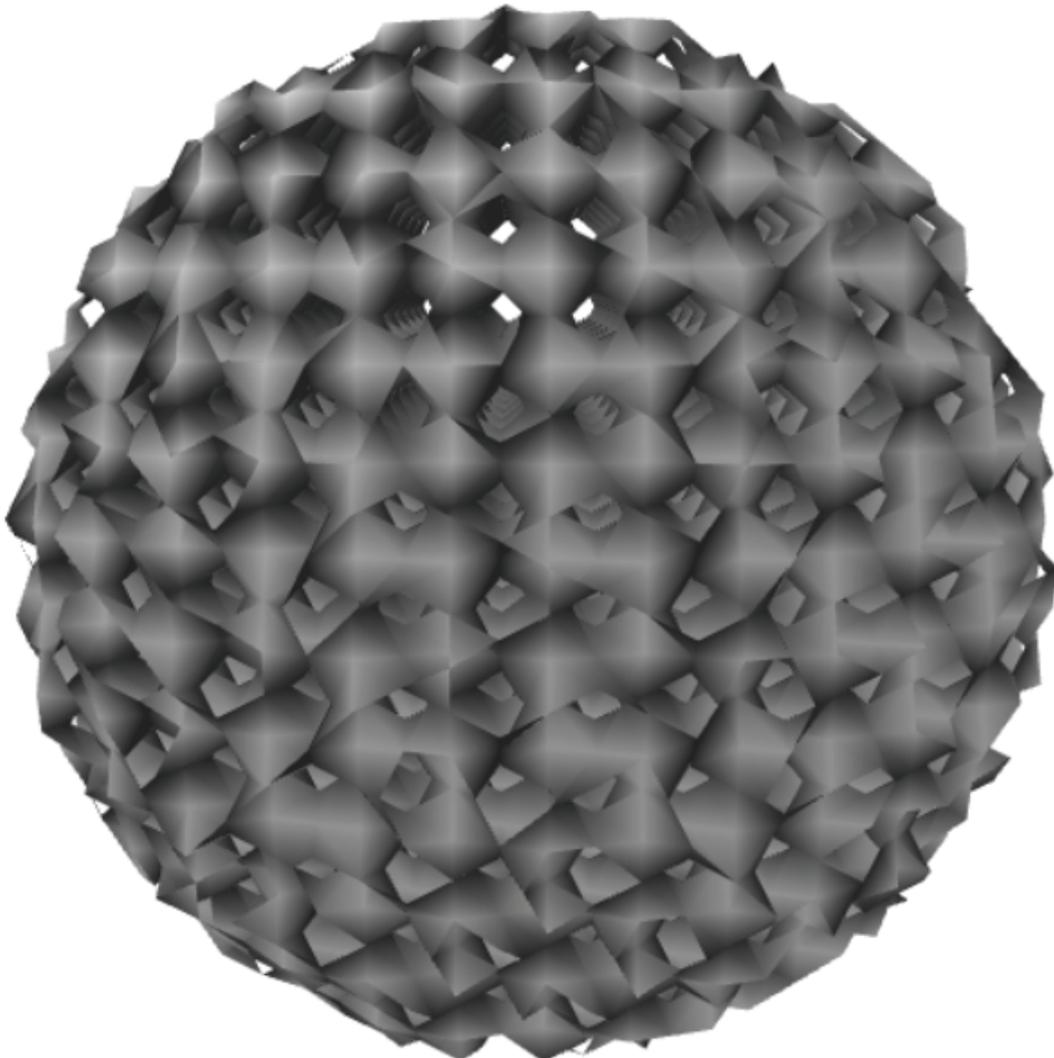
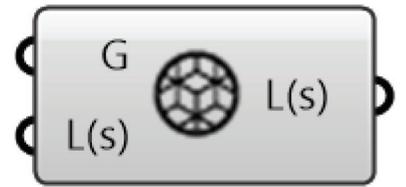
### Inputs:

Lattice (shell) (L(s)) - [mesh] - Lattice shell to trim

Geometry (G) - [brep / mesh] - Geometry to trim with (closed brep or mesh)

### Outputs:

Lattice (shell) (L(s)) - [mesh] - Trimmed lattice shell





## Remove Floating

### Description:

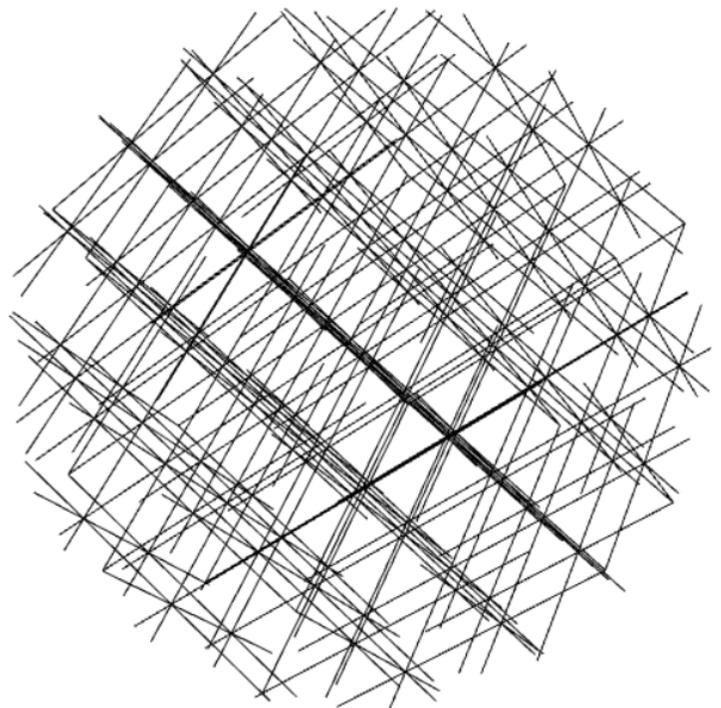
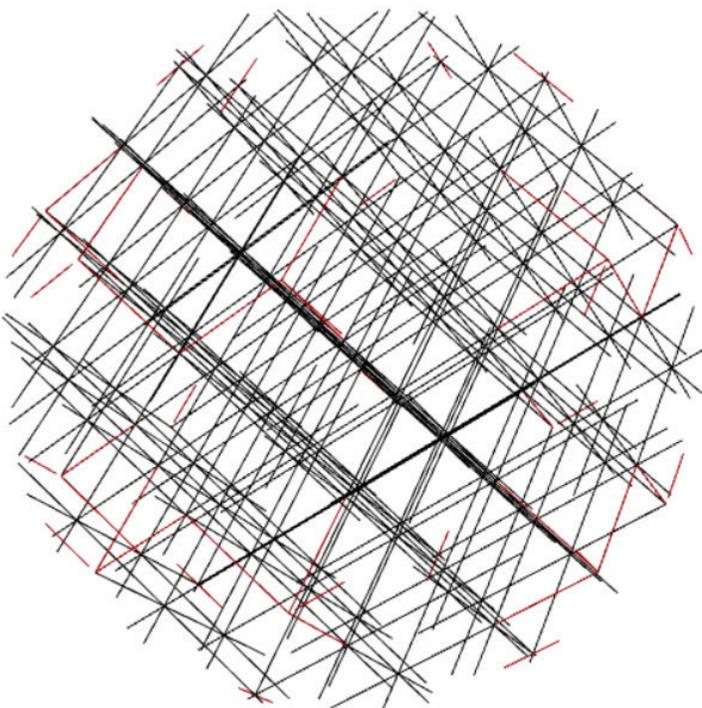
Remove disconnected curves. This will remove curves that have no connections to other curves. To reduce time, use this tool on only the 'Trimmed Lattice' output of the Trim Lattice tool.

### Inputs:

Lattice (L) - [curves] - Lattice curves to filter

### Outputs:

Lattice (L) - [curves] - Filtered lattice curves





## Remove Short

### Description:

Remove curves by length. To reduce time, use this tool on only the 'Trimmed Lattice' output of the Trim Lattice tool.

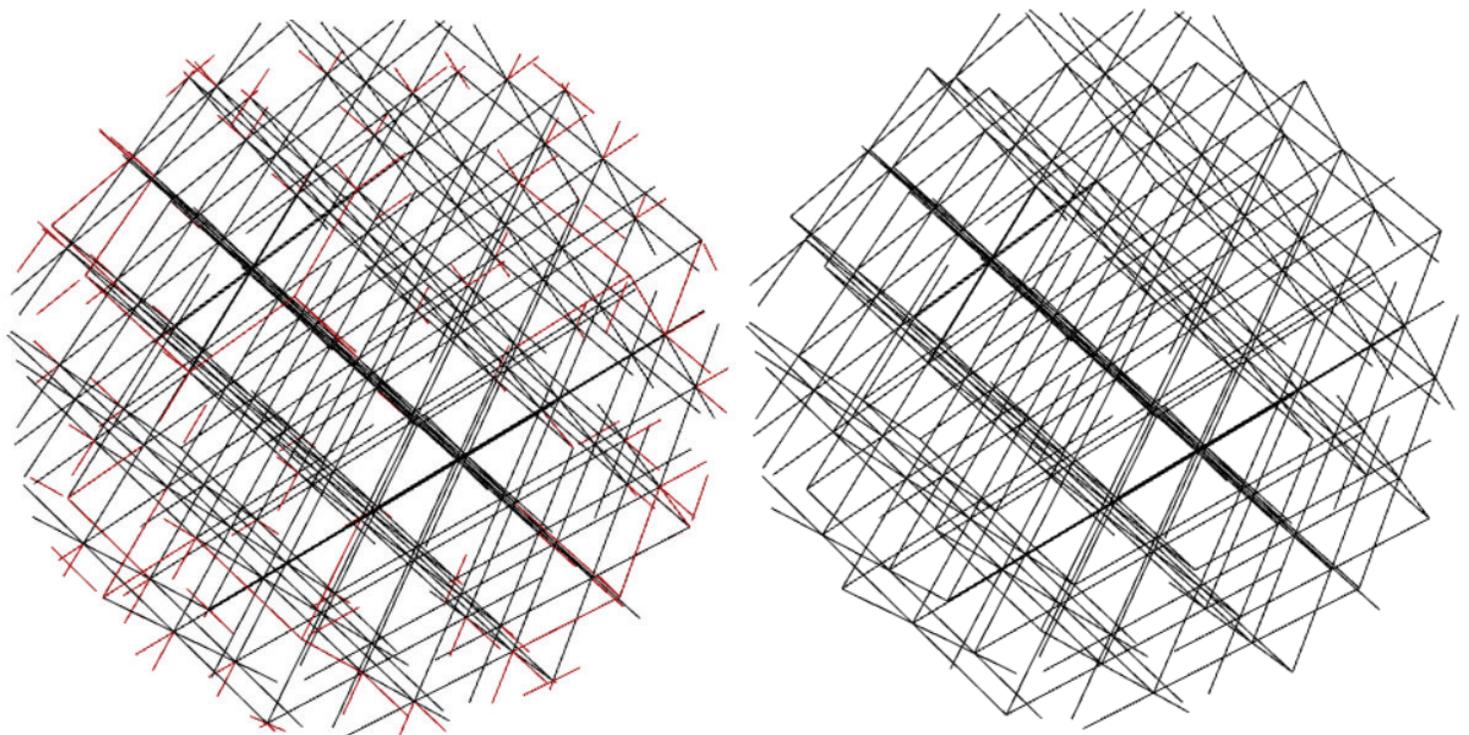
### Inputs:

Lattice (L) - [curves] - Lattice curves to filter

Length (N) - [number] - Minimum length of lattice curves

### Outputs:

Lattice (L) - [curves] - Filtered lattice curves





## Remove by Valence

### Description:

Remove curves by number of connections (this can be very slow, sorry). If only removing valence 1, the 'Untrimmed Lattice' output of Trim Lattice will have the same result.

### Inputs:

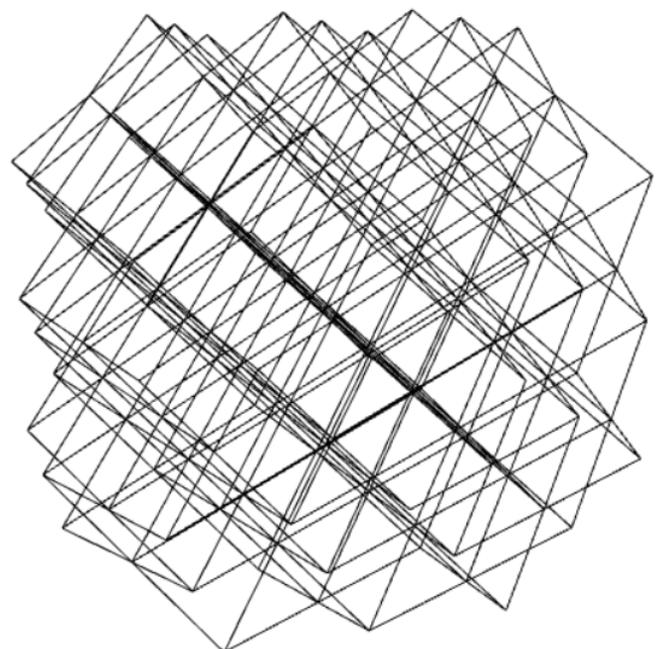
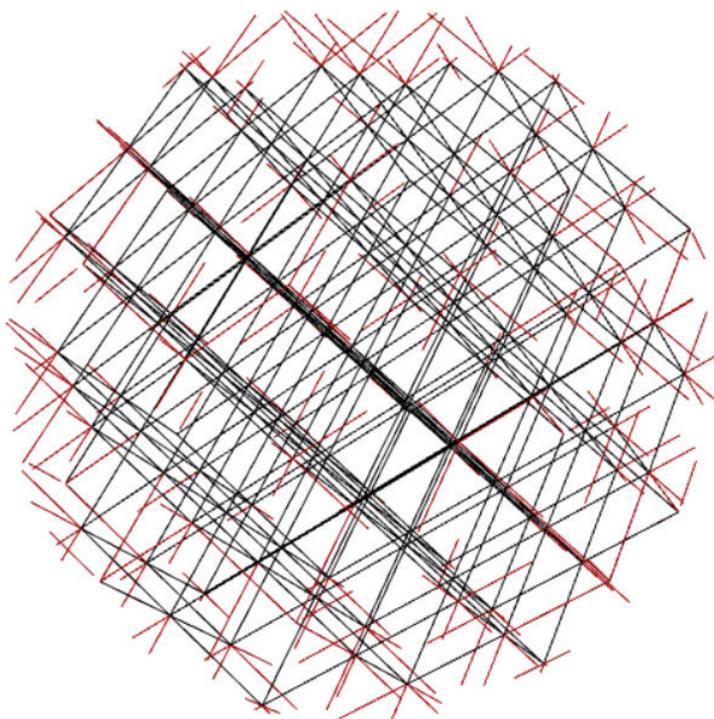
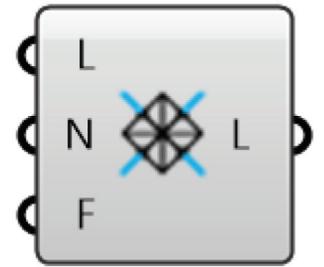
Lattice (L) - [curves] - Lattice curves to filter

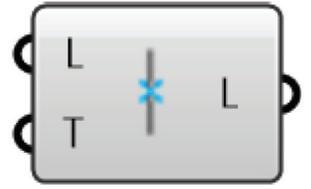
Valence Number (N) - [number] - Minimum number of connections each curve has

Flip (F) - [boolean] - If true, curves under valence number are removed. If false, curves over valence number are removed

### Outputs:

Lattice (L) - [curves] - Filtered lattice curves





## Remove Duplicate Curves

### Description:

Remove curves that share the same midpoint. This is a simple and fast method of removing duplicates by searching for common midpoints rather than common endpoints. Since all lattice curves should connect end-to-end, none should have common midpoints unless they are duplicates.

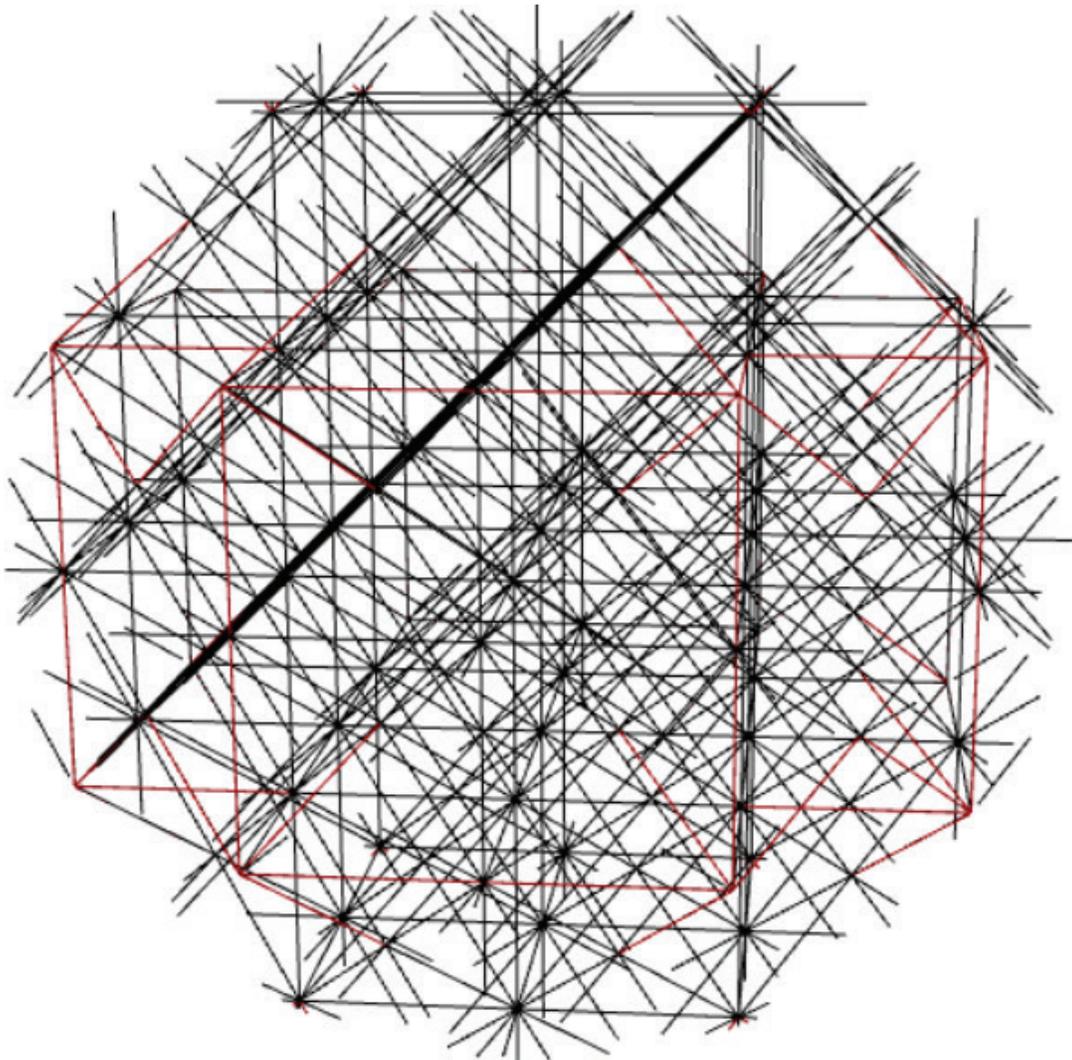
### Inputs:

Lattice (L) - [curves] - Lattice curves to filter

Tolerance (T) - [number] - Tolerance for merging curves (distance between midpoints  $\sim 0.1 - 0.001$ )

### Outputs:

Lattice (L) - [curves] - Filtered lattice curves





## Merge Curves

### Description:

Merge endpoints of curves by distance.

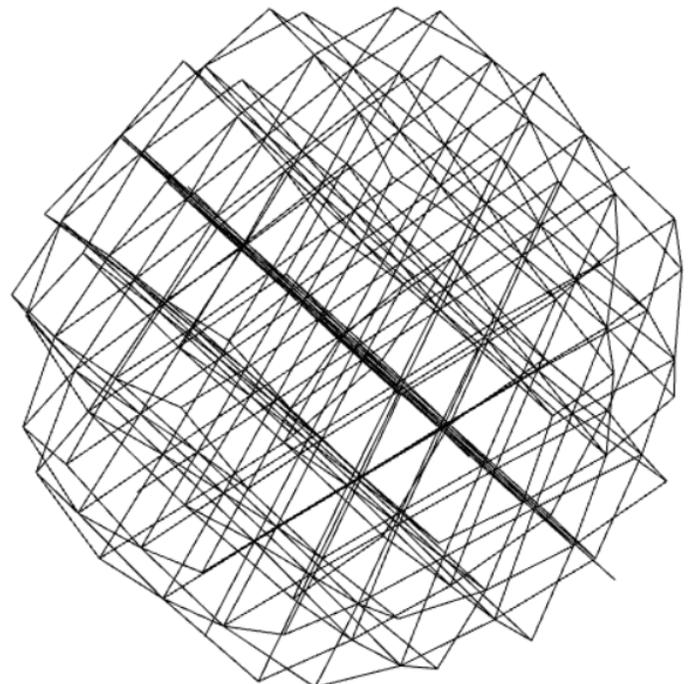
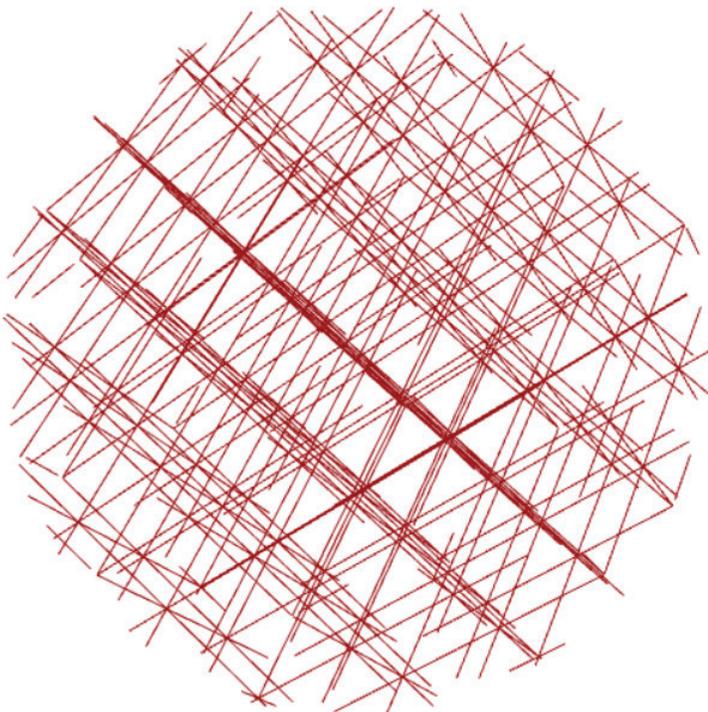
### Inputs:

Lattice (L) - [curves] - Lattice curves to merge

Max Distance (D) - [number] - Maximum distance to merge

### Outputs:

Lattice (L) - [curves] - Merged lattice curves





## Lattice connections

### Description:

Connect two adjacent lattices by adding struts between the closest nodes.

### Inputs:

Lattice A (L) - [curves] - First lattice to make connections between

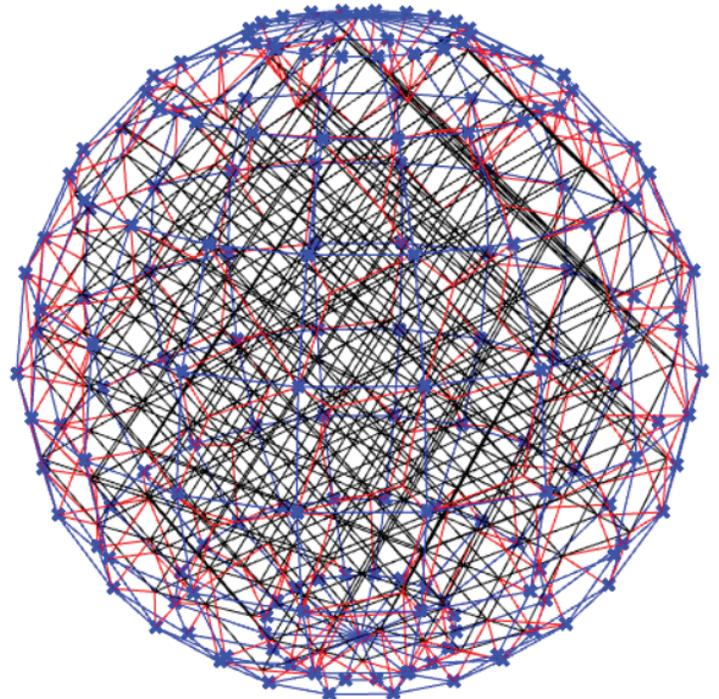
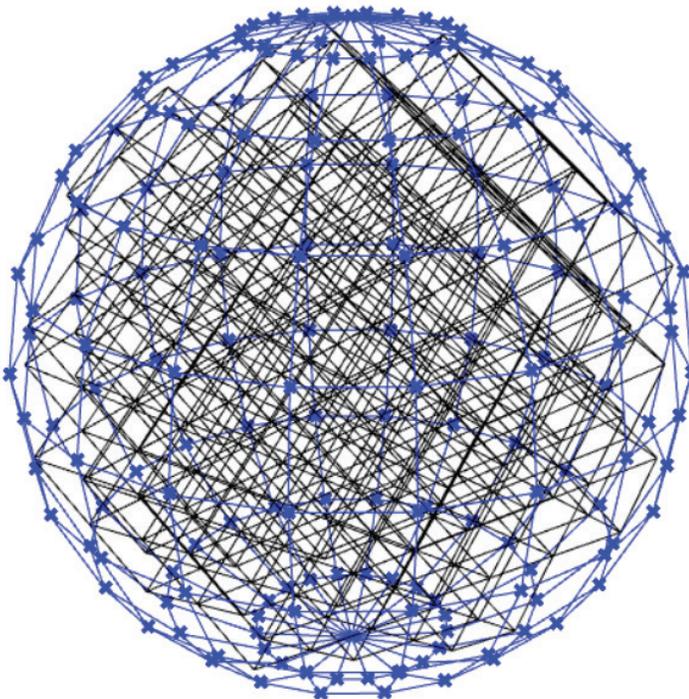
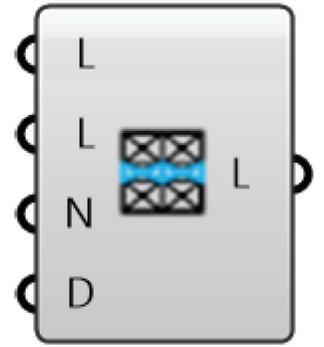
Lattice B (L) - [curves] - Second lattice to make connections between

Number of Connections (N) - [number] - Number of connections to make from each node

Distance (D) - [number] - Maximum distance for each connection

### Outputs:

Lattice (L) - [curves] - Lattice connections





## Morph Lattice to Skin

### Description:

Morph lattice to connect to skin. This will move the nodes of the lattice within a set distance from the skin to the nearest node of the skin.

### Inputs:

Lattice (L) - [curves] - Lattice to morph to skin

Geometry (G) - [geometry] - Geometry to morph to (usually the geometry (mesh or brep) that makes the skin)

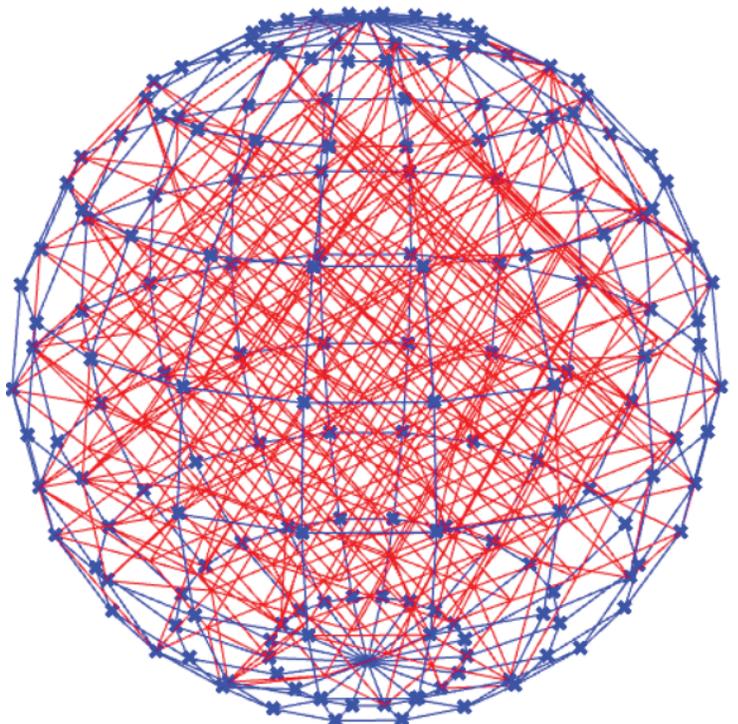
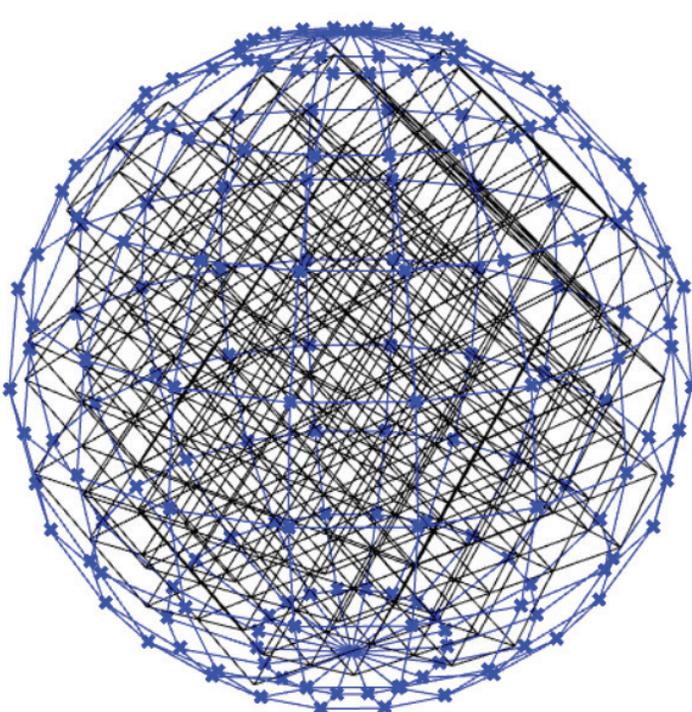
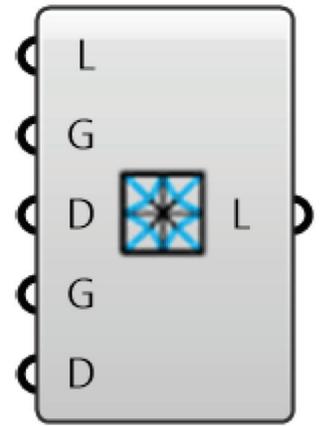
Distance (D) - [number] - Maximum distance from geometry

Geometry (G) - [geometry] - Geometry to morph to (usually the nodes or vertices of the skin)

Distance (D) - [number] - Maximum distance to morph

### Outputs:

Lattice (L) - [curves] - Morphed lattice





## Morph Shell to Skin

### Description:

Morph shell to connect to skin. This will move vertices of the mesh shell within a set distance from the skin to the closest point on the skin. Subdividing the mesh of the shell to add more vertices will have a smoother result but will take longer.

### Inputs:

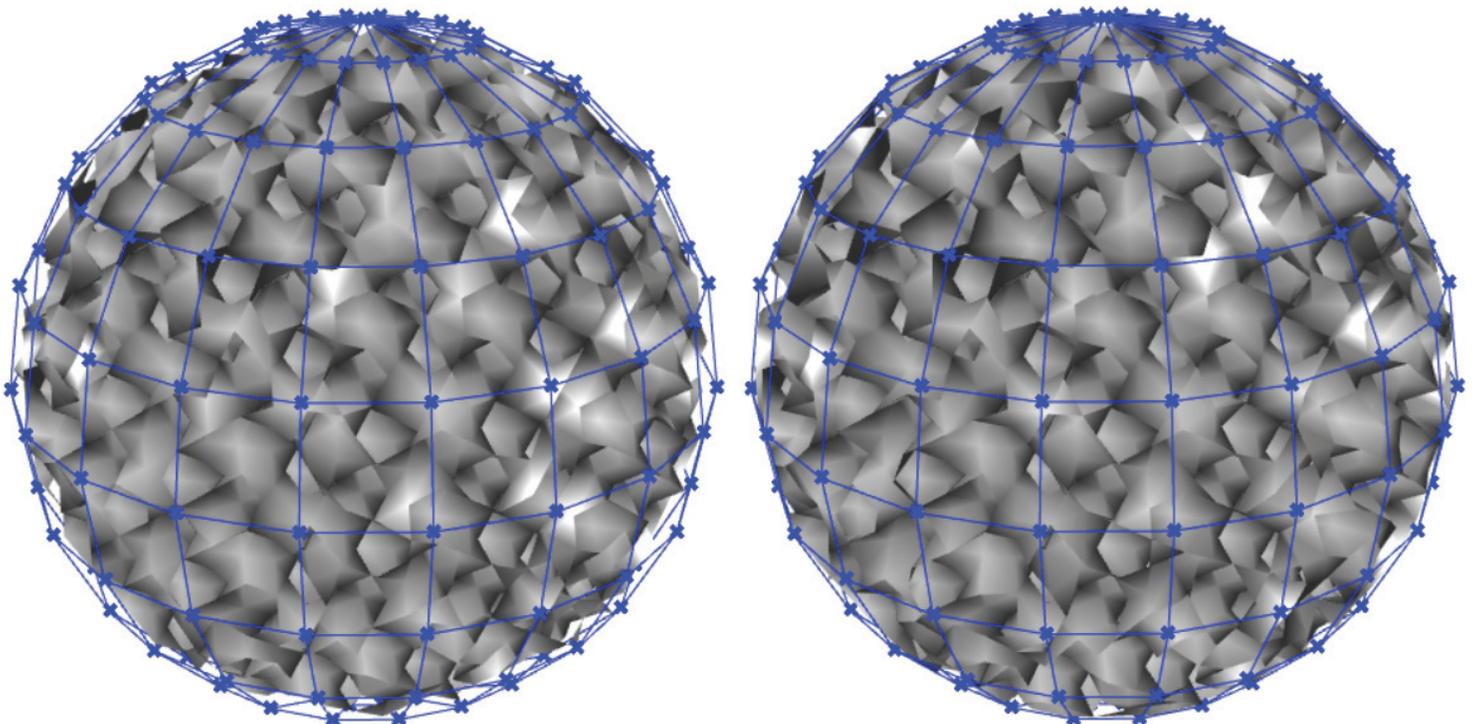
Lattice (shell) (L(s)) - [mesh] - Lattice shell to morph

Geometry (G) - [geometry] - Geometry to morph to

Distance (D) - [number] - Maximum distance from geometry

### Outputs:

Lattice (shell) (L(s)) - [mesh] - Morphed lattice shell



## Thicken

This group of tools is for applying a thickness value to the lattice which can be used for simulation or generating a solid mesh for manufacturing.

Lattice Thickness Value

LTV



Lattice Thickness Attractor

LTA



Shell Thickness Value

STV



Shell Thickness Attractor

STA





## Lattice Thickness Value

### Description:

Apply thickness values to lattice curves with a point cloud and corresponding values. The output will be one value for each curve.

### Inputs:

Lattice (L) - [curves] - Lattice to thicken

Points (S) - [point] - List of points (flattened)

Values (Va) - [number] - List of values that correspond to points (flattened)

Distance Influence (DI) - [boolean] - If true the distance between the points and voxels will affect the magnitude

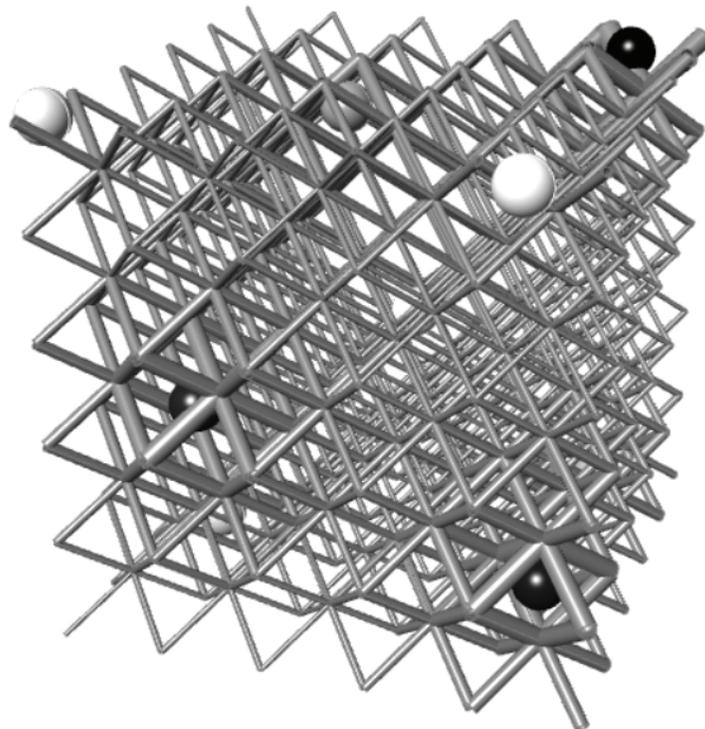
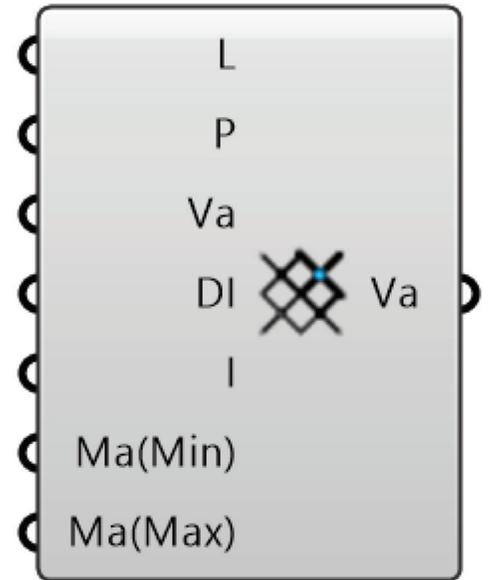
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - Lattice beam thickness (minimum)

Magnitude (max) (Ma(max)) - [number] - Lattice beam thickness (maximum)

### Outputs:

Values (Va) - [number] - List of beam thickness values corresponding to each lattice curve





## Lattice Thickness Attractor

### Description:

Apply thickness values to lattice curves with attractor(s). The output will be one value for each curve.

### Inputs:

Lattice (L) - [curves] - Lattice to thicken

Attractor (A) - [geometry] - Any geometry to use as attractor(s)

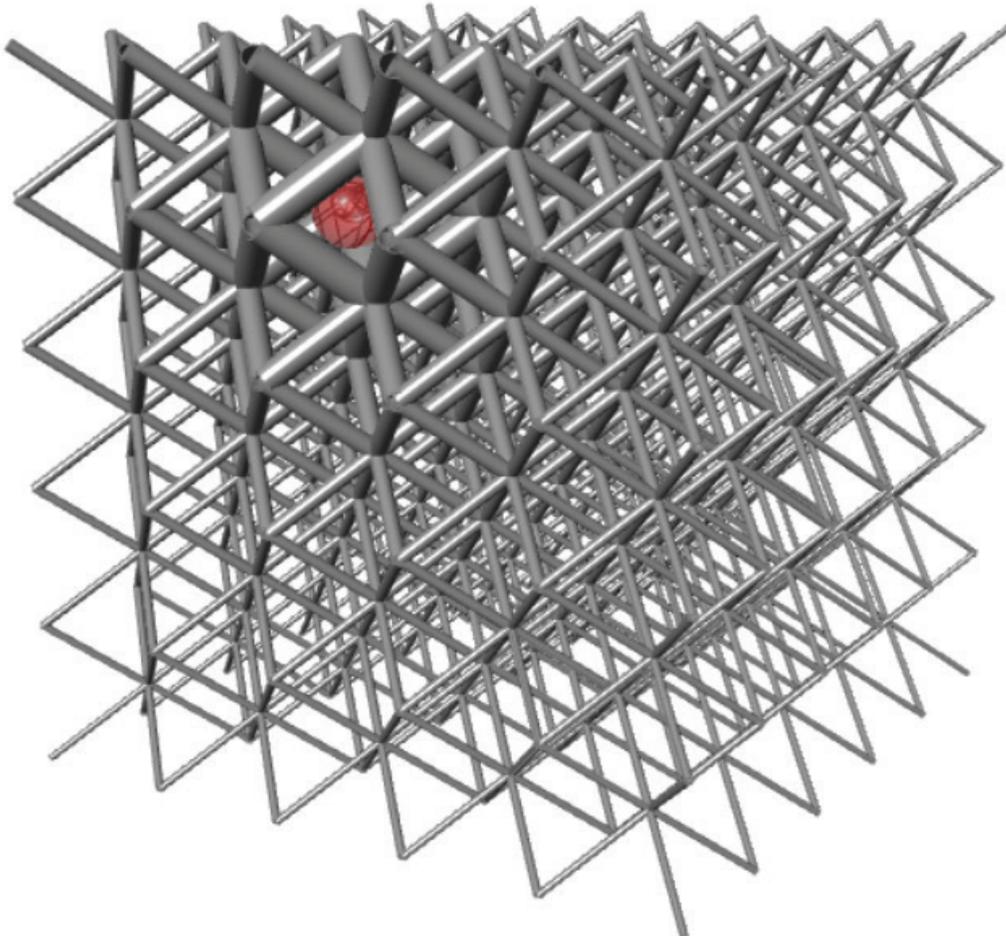
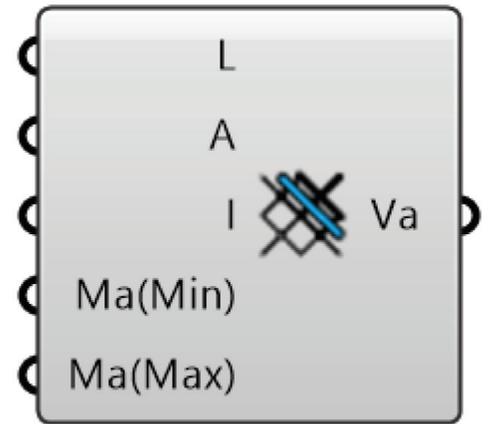
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - Lattice beam thickness (minimum)

Magnitude (max) (Ma(max)) - [number] - Lattice beam thickness (maximum)

### Outputs:

Values (Va) - [number] - List of beam thickness values corresponding to each lattice curve





## Shell Thickness Value

### Description:

Apply thickness values to shell vertices with a point cloud and corresponding values. The output will be one value for each vertice of the mesh (useful for Weaverbird's 'Mesh Thicken' tool).

### Inputs:

Lattice (shell) (L(s)) - [mesh] - Lattice to thicken

Points (S) - [point] - List of points (flattened)

Values (Va) - [number] - List of values that correspond to points (flattened)

Distance Influence (DI) - [boolean] - If true the distance between the points and voxels will affect the magnitude

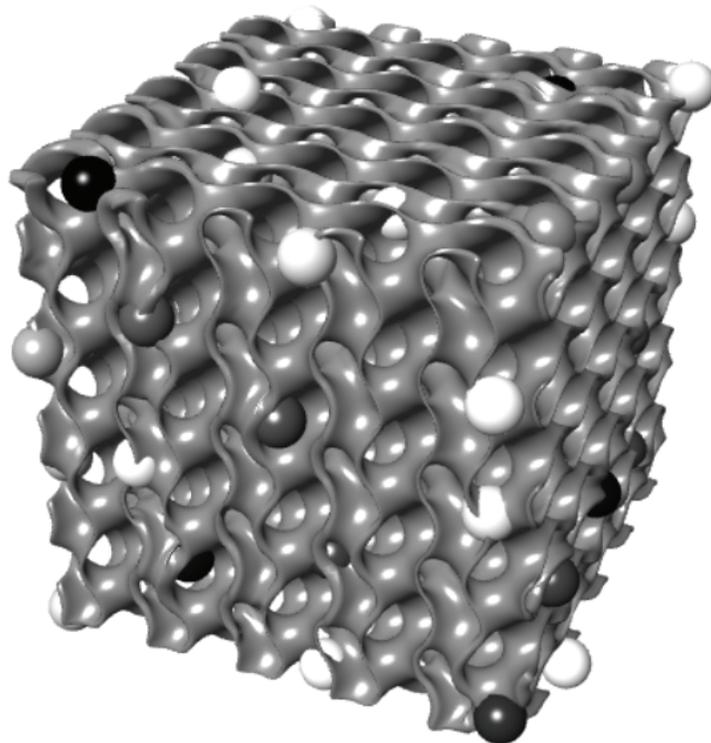
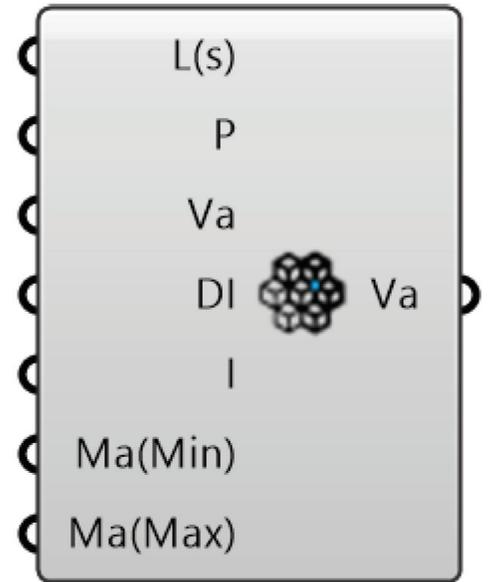
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - shell thickness (minimum)

Magnitude (max) (Ma(max)) - [number] - shell thickness (maximum)

### Outputs:

Values (Va) - [number] - List of thickness values corresponding to each shell vertice





## Shell Thickness Attractor

### Description:

Apply thickness values to shell vertices with attractor(s). The output will be one value for each vertice of the mesh (useful for Weaverbird's 'Mesh Thicken' tool).

### Inputs:

Lattice (shell) (L(s)) - [mesh] - Lattice to thicken

Attractor (A) - [geometry] - Any geometry to use as attractor(s)

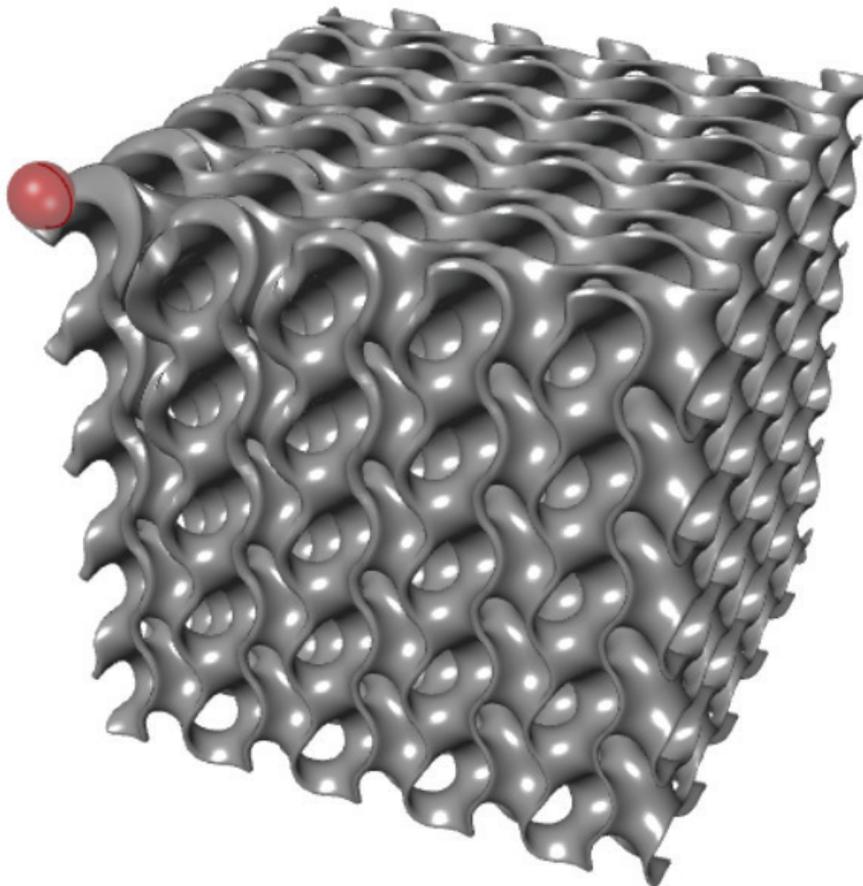
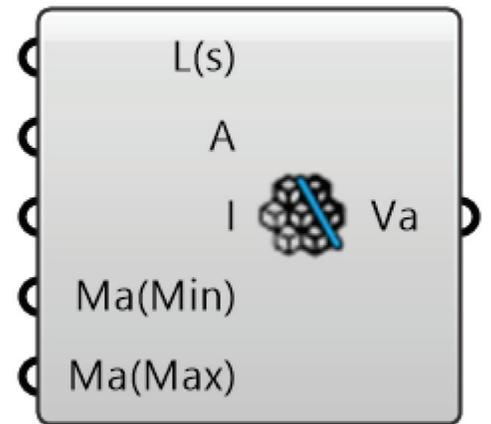
Influence (I) - [number] - Smoothed range of attractor influence (0-1)

Magnitude (min) (Ma(min)) - [number] - Lattice shell thickness (minimum)

Magnitude (max) (Ma(max)) - [number] - Lattice shell thickness (maximum)

### Outputs:

Values (Va) - [number] - List of beam thickness values corresponding to each shell vertice



## 2D

This group of tools is (a work in progress) for generating 2D geometry from a 3D shape. The lattice hinge tools are for creating relief cuts on a rigid material for bending. The mesh flatten tools are for use with plastic or flexible materials that allow for compound curvature.

Lattice Hinge	LH		Mesh Cell Fill	MCF	
Lattice Hinge Cell Fill	LHC		Mesh Tween Cell Fill	MTC	
Lattice Hinge Tween Cell Fill	LHT		Mesh Shell Fill	MSF	
Lattice Hinge Shell Fill	LHC(s)		Mesh Tween Shell Fill	MTF	
Lattice Hinge Tween Cell Fill	LHT(s)		Mesh Hinge Cell (T)	MHt	
Mesh Flatten	MF		Mesh Hinge Cell (Q)	MHq	



## Lattice Hinge (LH)

### Description:

Create a ruled surface between two curves and unroll to apply kerf bending patterns.  
Disclaimer: This is a work in progress and not fully tested with real materials (requires Kangaroo <https://www.food4rhino.com/app/kangaroo-physics>).

### Inputs:

Curve 1 (C1) - [curve] - Curve to loft between

Curve 2 (C2) - [curve] - Curve to loft between

U Divisions (U) - [number] - Divisions in the U direction

V Divisions (V) - [number] - Divisions in the V direction

Influence (I) - [number] - Influence from surface curvature (0-1)

Voxel Height (H) - [number] - Height of voxels on surface

### Outputs:

Voxels (rolled) (Vr) - [twisted box] - Voxels of the rolled surface (for preview)

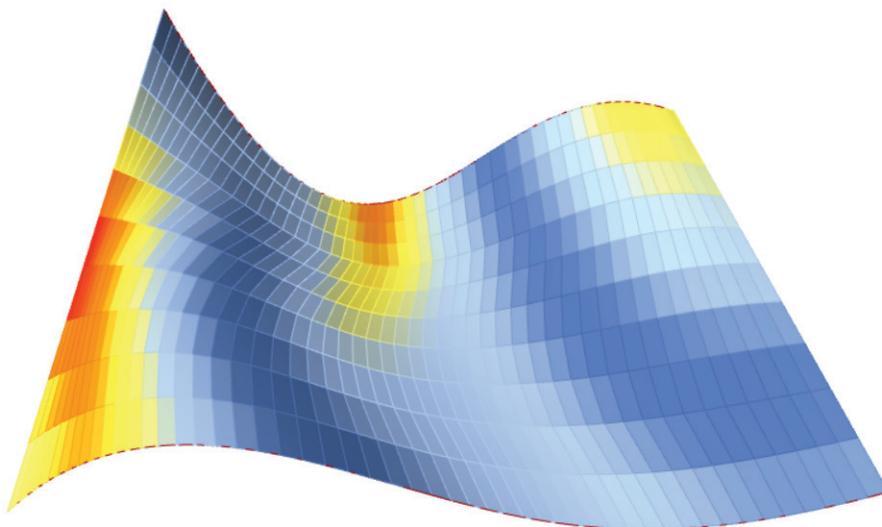
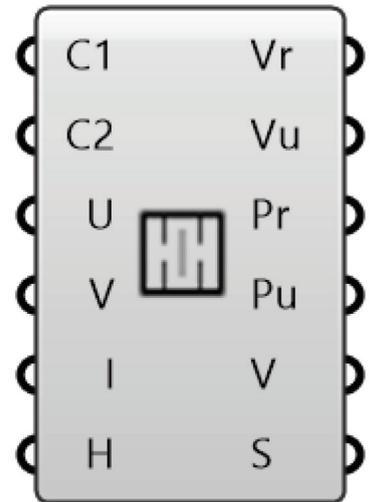
Voxels (unrolled) (Vu) - [twisted box] - Voxels of the unrolled surface (for mapping)

Panels (rolled) (Pr) - [surface] - Panels of the rolled surface (for preview)

Panels (unrolled) (Pu) - [surface] - Panels of the unrolled surface (for mapping)

Panel Values (V) - [number] - Values measuring curvature (0-1)

Unrolled Surface (S) - [surface] - The unrolled surface





## Lattice Hinge Cell Fill (LHC)

### Description:

Populate lattice hinge panels with a curve based unit cell

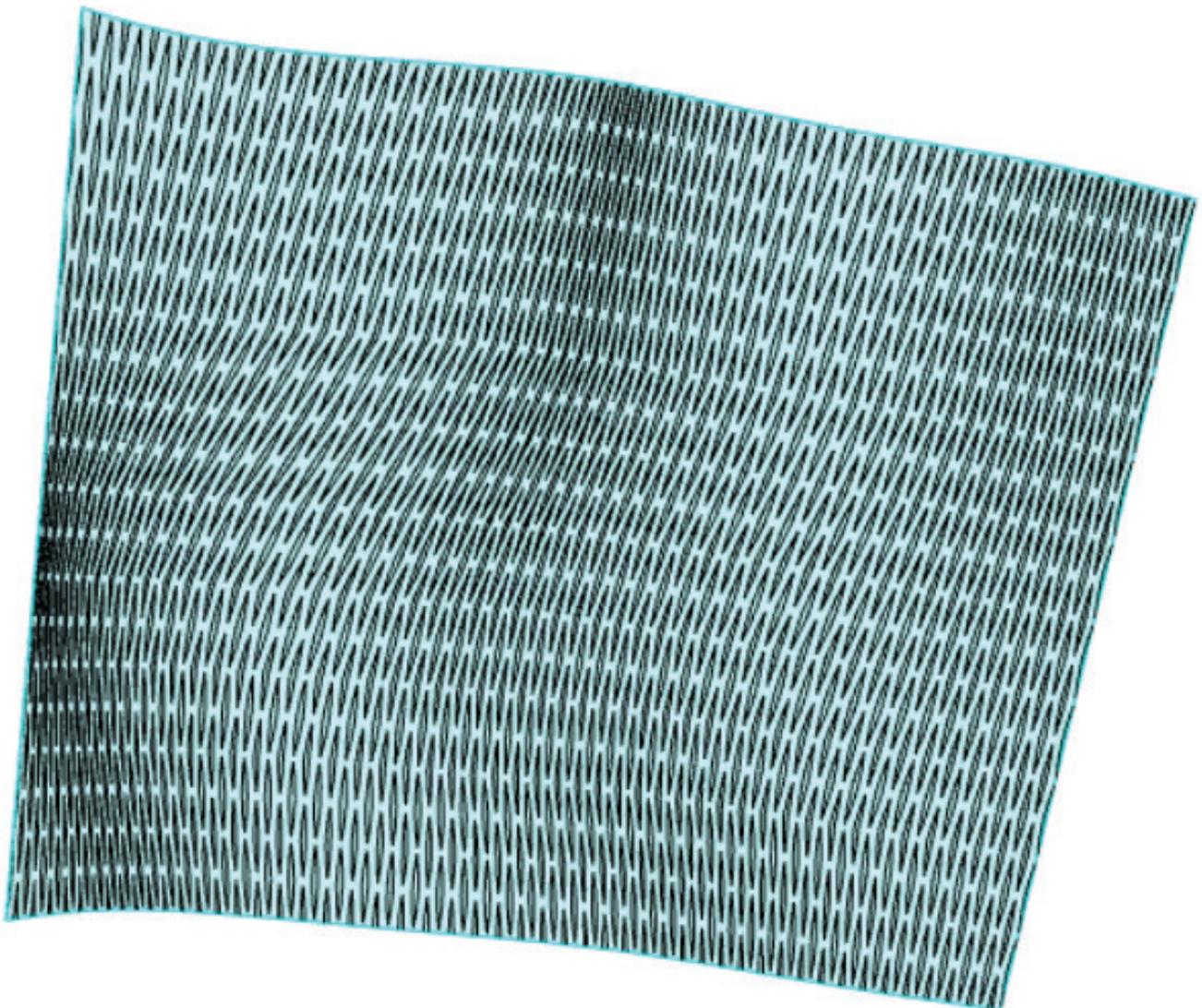
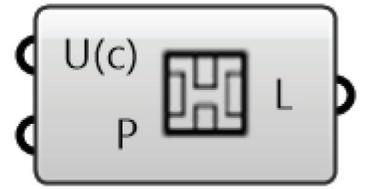
### Inputs:

Unit cell ( $U(c)$ ) - [curves] - Unit cell to populate lattice hinge panels

Panels ( $P$ ) - [surface] - Panels to be populated with unit cells

### Outputs:

Lattice Hinge ( $L$ ) - [curve] - List of curves that make up the lattice hinge





## Lattice Hinge Tween Cel Fill (LHT)

### Description:

Tween between lattice hinge panels with curve based unit cells

### Inputs:

Unit cell 1 (U(c)) - [curve] - Unit cell to populate lattice hinge panels

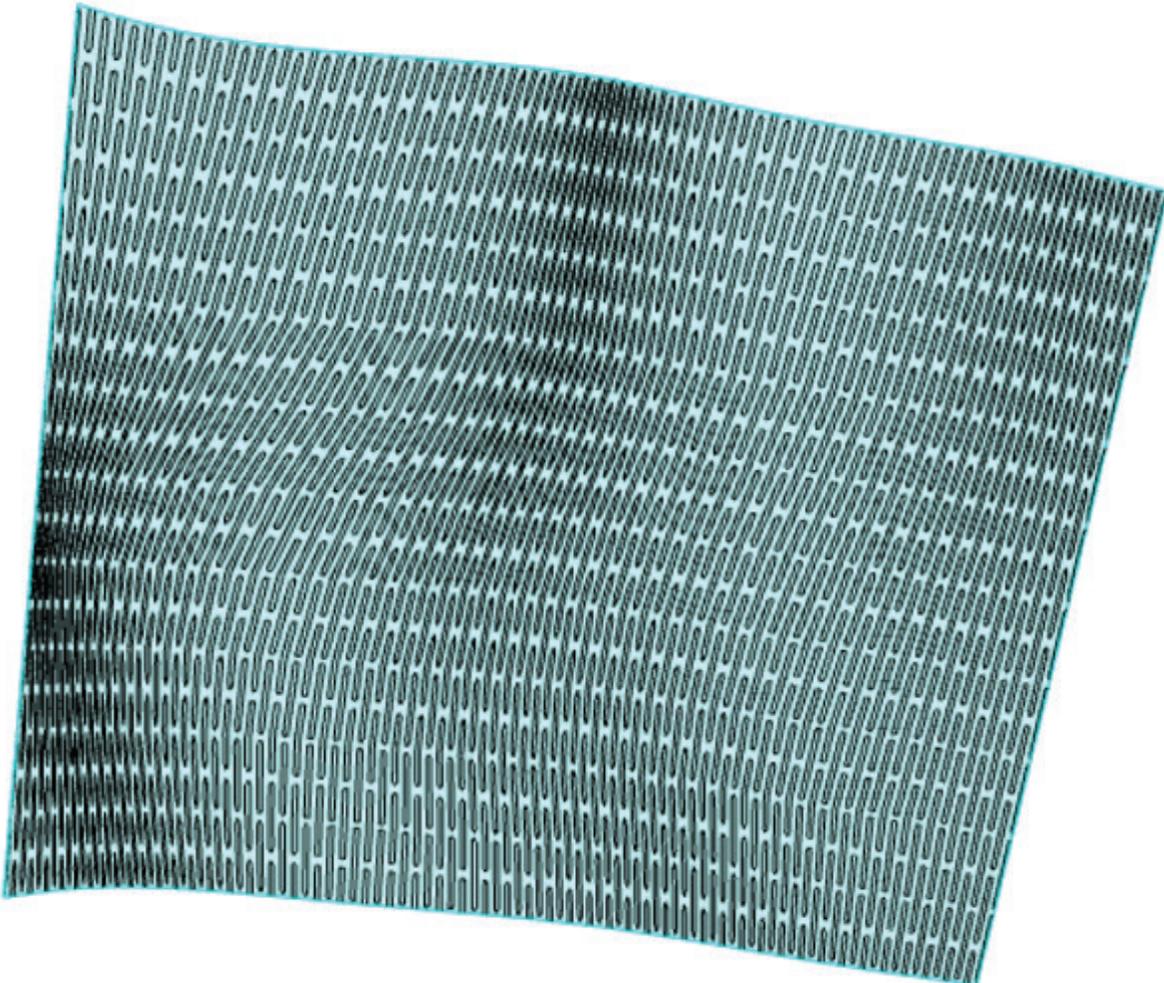
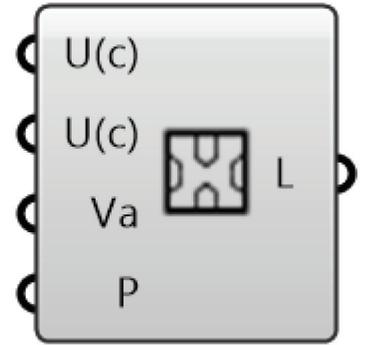
Unit cell 2 (U(c)) - [curve] - Unit cell to populate lattice hinge panels

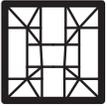
Value (Va) - [number] - Value from 0-1 corresponding to each panel

Panels (P) - [surface] - Panels to be populated with unit cells

### Outputs:

Lattice Hinge (L) - [curve] - List of curves that make up the lattice hinge





## Lattice Hinge Shell Fill (LHS)

### Description:

Populate lattice hinge panels with a shell based unit cell

### Inputs:

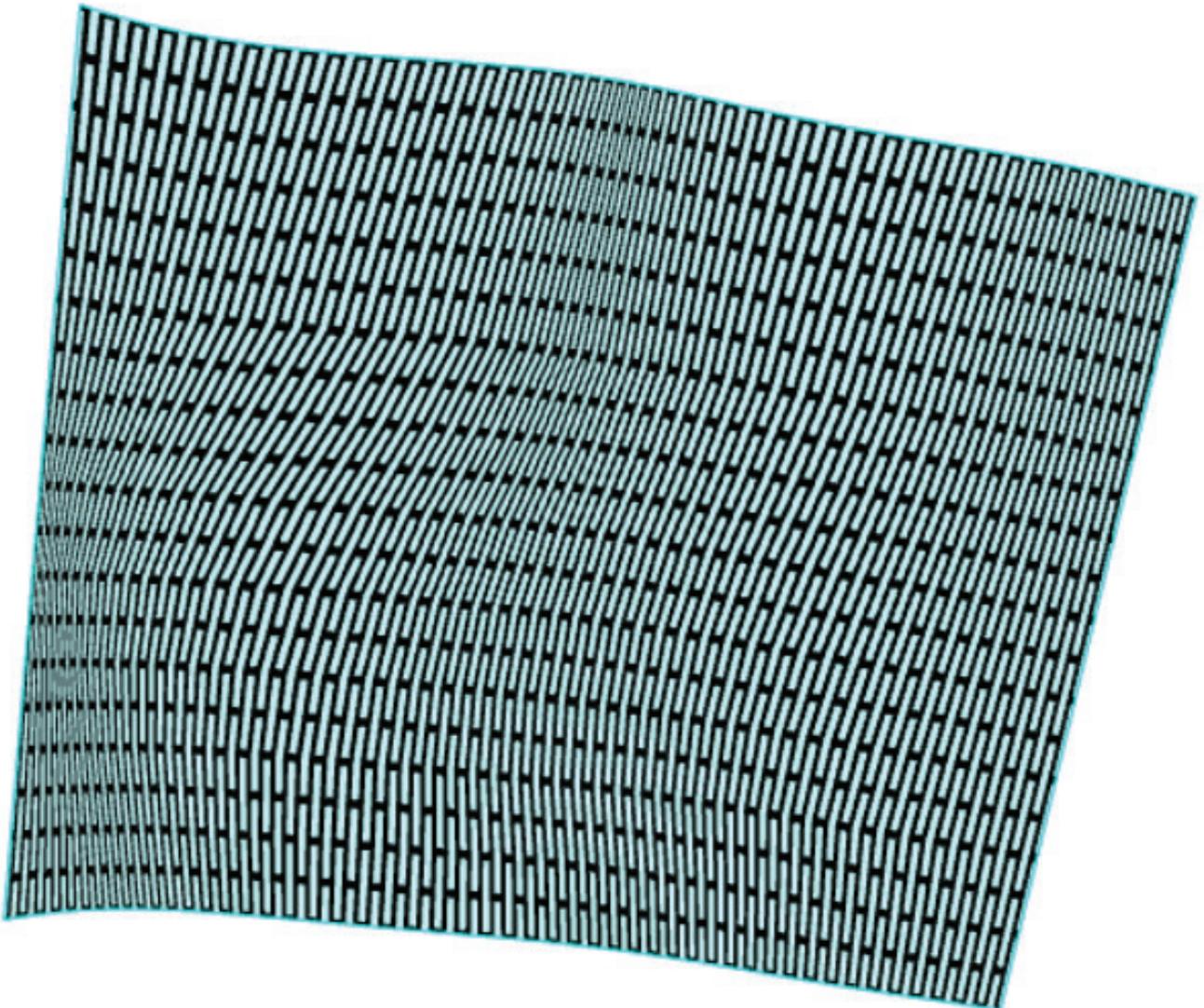
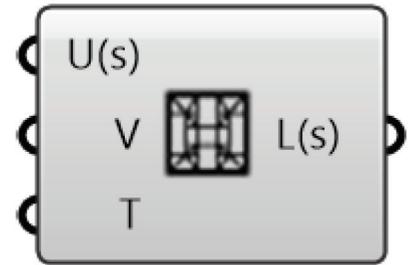
Unit cell (shell) ( $U(s)$ ) - [mesh] - Unit cell to populate lattice hinge panels

Voxels ( $V$ ) - [twisted box] - Voxels to be populated with unit cells

Tolerance ( $T$ ) - [number] - Tolerance for merging faces and vertices ( $\sim 0.1 - 0.001$ )

### Outputs:

Lattice Hinge (shell) ( $L(s)$ ) - [mesh] - Joined mesh that make up the lattice hinge





## Lattice Hinge Tween Shell Fill (LHTs)

### Description:

Tween between lattice hinge panels with shell based unit cells

### Inputs:

Unit cell 1 (shell) (U(s)1) - [mesh] - Unit cell to populate lattice hinge panels

Unit cell 2 (shell) (U(s)2) - [mesh] - Unit cell to populate lattice hinge panels

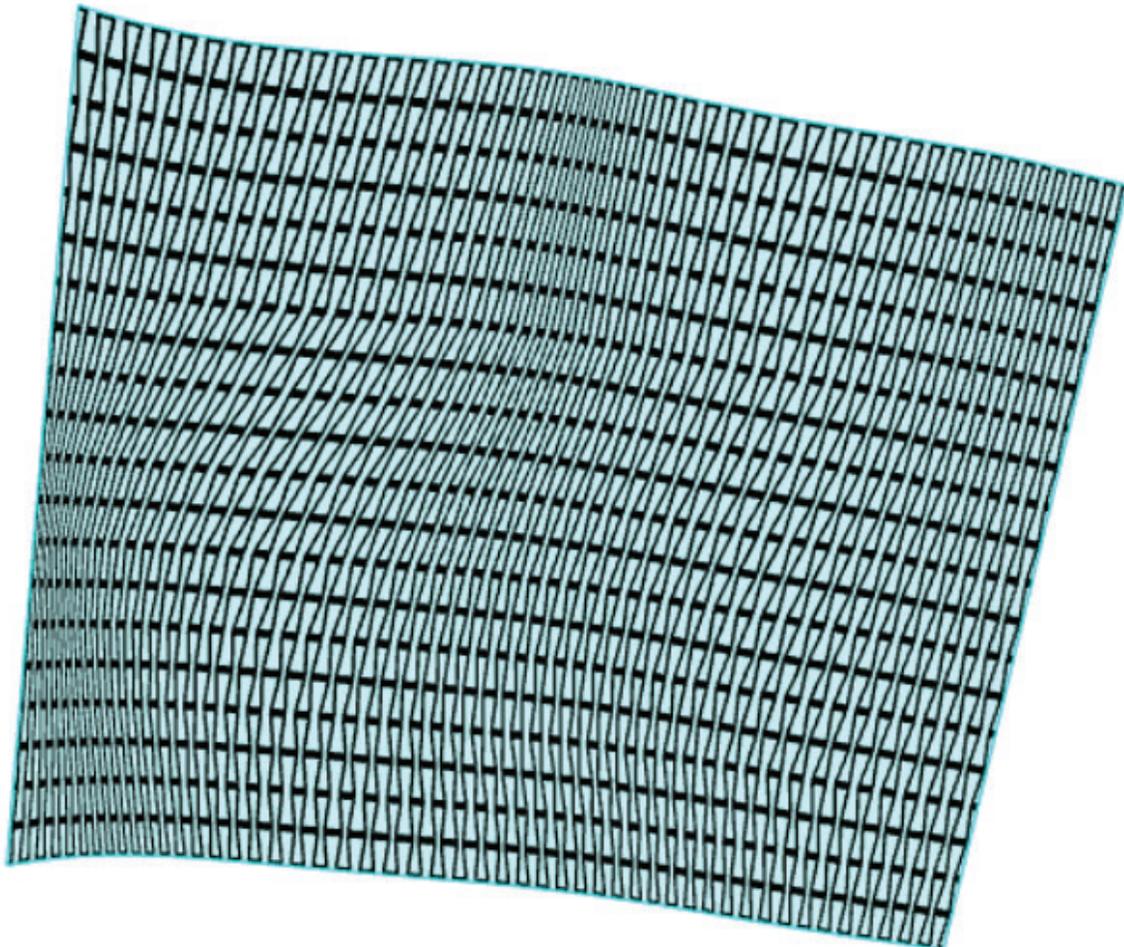
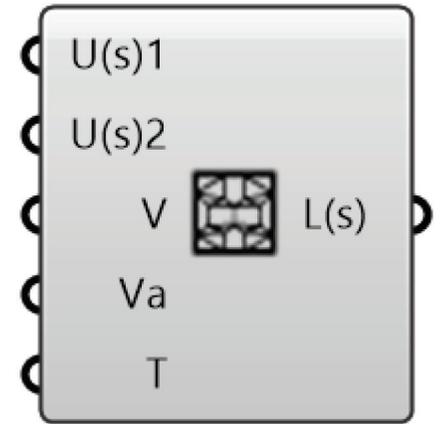
Voxels (V) - [twisted box] - Voxels to be populated with unit cells

Value (Va) - [number] - Value from 0-1 corresponding to each voxel

Tolerance (T) - [number] - Tolerance for merging faces and vertices (~0.1 - 0.001)

### Outputs:

Lattice Hinge (shell) (L(s)) - [mesh] - Joined mesh that make up the lattice hinge





## Mesh Flatten (MF)

### Description:

Flatten a mesh surface and measure the strain per mesh face (requires Kangaroo2 <https://www.food4rhino.com/app/kangaroo-physics>). Disclaimer: This is a work in progress and has not been fully tested and validated.

### Inputs:

Mesh (M) - [mesh] - Mesh to flatten

Plane (PI) - [plane] - Plane to flatten mesh to

Threshold (Th) - [number] - Threshold for Kangaroo solver (~1e-7)

Max iterations (I) - [number] - Maximum iterations for the Kangaroo solver

### Outputs:

Iterations (I) - [number] - Iterations of the kangaroo solver

Flattened mesh (M) - [mesh] - The flattened mesh

Mesh faces (Mf) - [mesh] - Faces of the flattened mesh (exploded into separate meshes)

Ratio cP (cP) - [number] - Ratio of strain of mesh edges

cP Lines (LcP) - [line] - Lines for preview

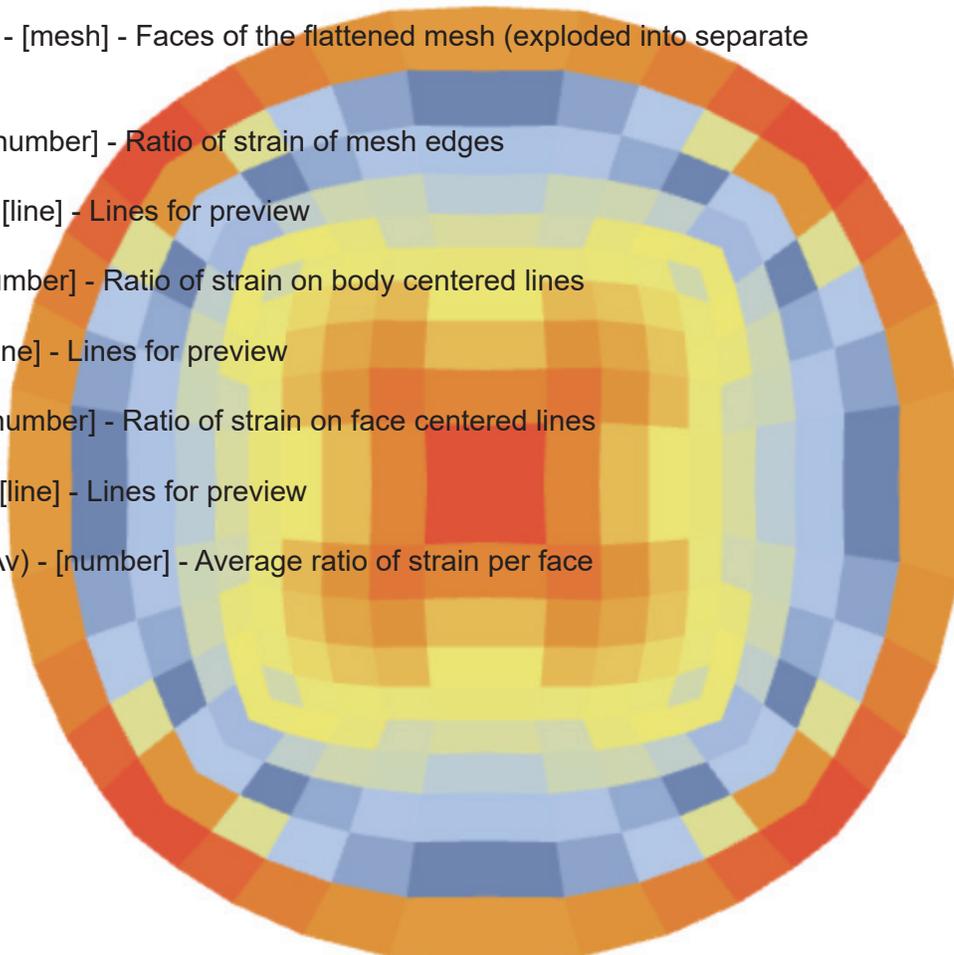
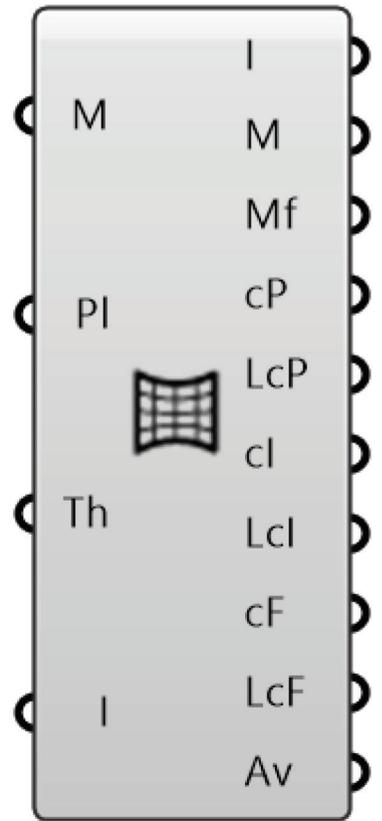
Ratio cl (cl) - [number] - Ratio of strain on body centered lines

cl Lines (Lcl) - [line] - Lines for preview

Ratio cF (cF) - [number] - Ratio of strain on face centered lines

cF Lines (LcF) - [line] - Lines for preview

Ratio average (Av) - [number] - Average ratio of strain per face





## Mesh Cell Fill (MCF)

### Description:

Map lattice unit cells to mesh faces

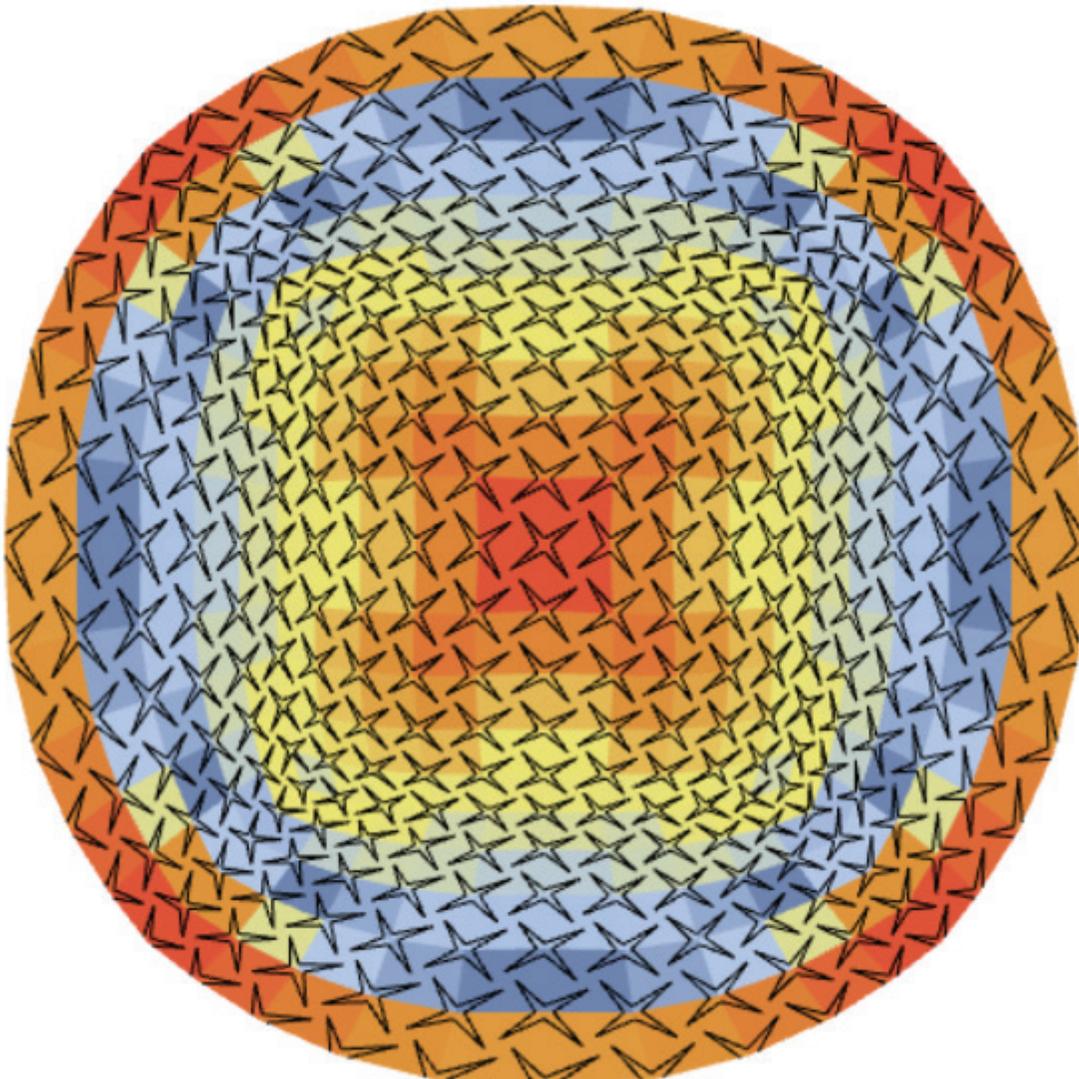
### Inputs:

Mesh cell (Mc) - [geometry] - Lattice cell unit to map to mesh faces

Mesh faces - (Mf) - [mesh] - Mesh face (exploded as single meshes)

### Outputs:

Lattice Hinge (L) - [curve] - Joined curves that make up the lattice hinge





## Mesh Tween Cell Fill (MTC)

### Description:

Tween between lattice unit cells on mesh faces

### Inputs:

Mesh cell 1 (Mc1) - [geometry] - Lattice cell unit to map to mesh faces

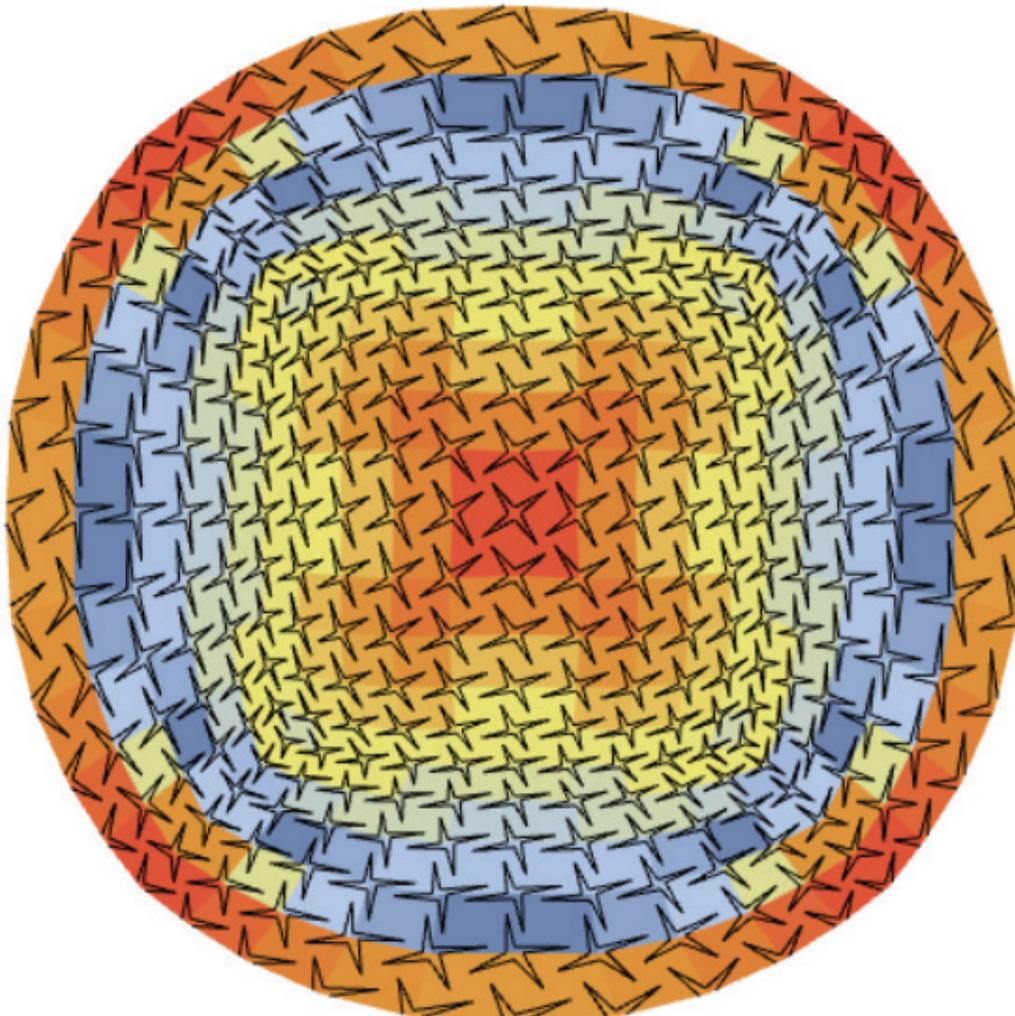
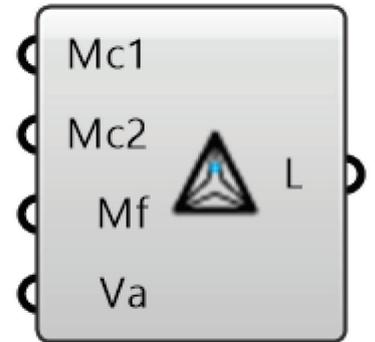
Mesh cell 2 (Mc2) - [geometry] - Lattice cell unit to map to mesh faces

Mesh faces - (Mf) - [mesh] - Mesh face (exploded as single meshes)

Value (Va) - [number] - Value from 0-1 corresponding to each mesh face

### Outputs:

Lattice Hinge (L) - [curve] - Joined curves that make up the lattice hinge





## Mesh Shell Fill (MSF)

### Description:

Map shell lattice unit cells to mesh faces

### Inputs:

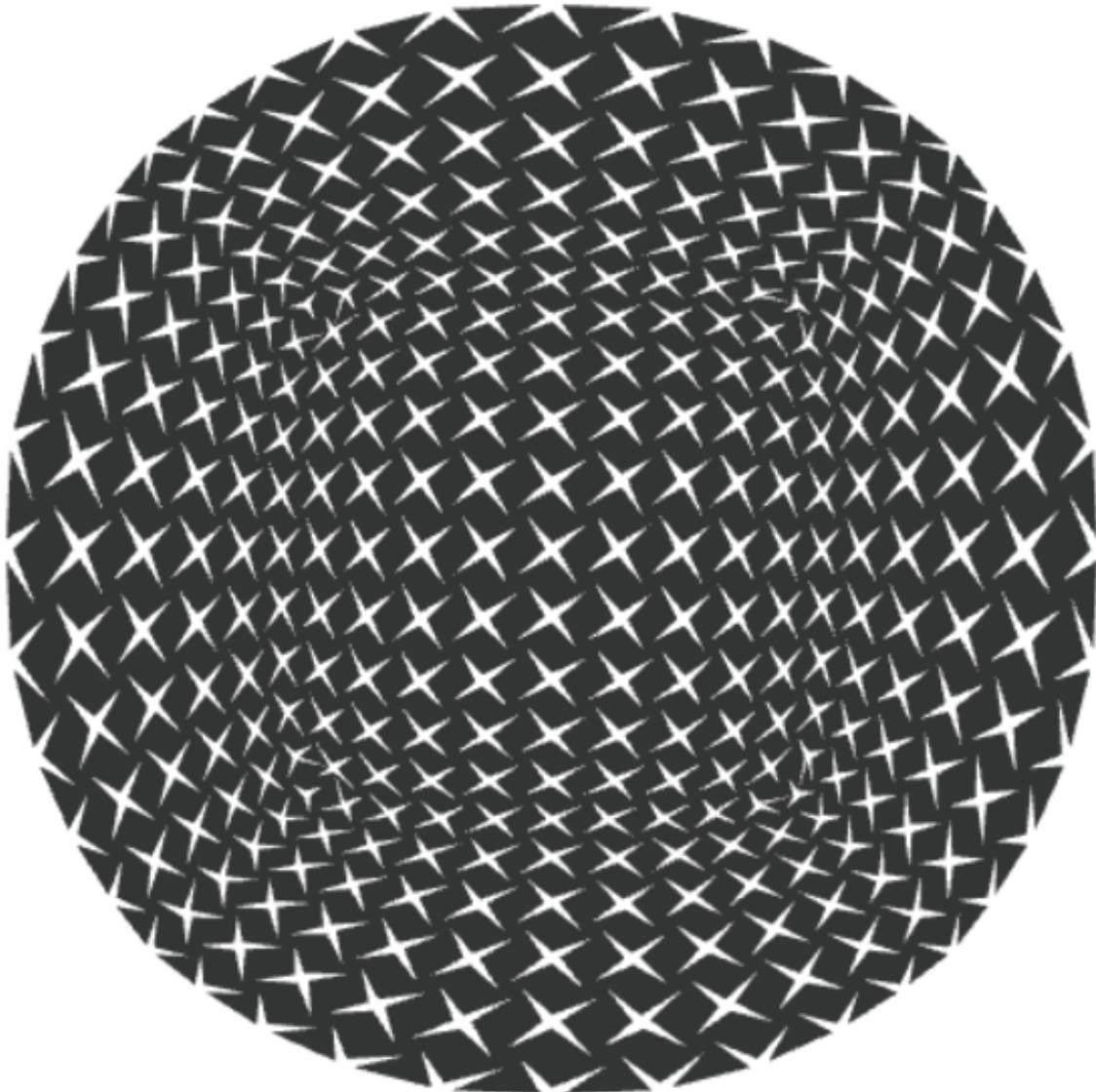
Mesh shell (Ms) - [geometry] - Lattice shell unit to map to mesh faces

Mesh faces - (Mf) - [mesh] - Mesh face (exploded as single meshes)

Tolerance (T) - [number] - Tolerance for merging faces and vertices (~0.1 - 0.001)

### Outputs:

Lattice Hinge (shell) (Ls) - [mesh] - Joined mesh that make up the lattice hinge





## Mesh Tween Shell Fill (MTF)

### Description:

Tween between mesh shell unit cells on mesh faces

### Inputs:

Mesh shell 1 (Ms1) - [geometry] - Lattice shell unit to map to mesh faces

Mesh shell 2 (Ms2) - [geometry] - Lattice shell unit to map to mesh faces

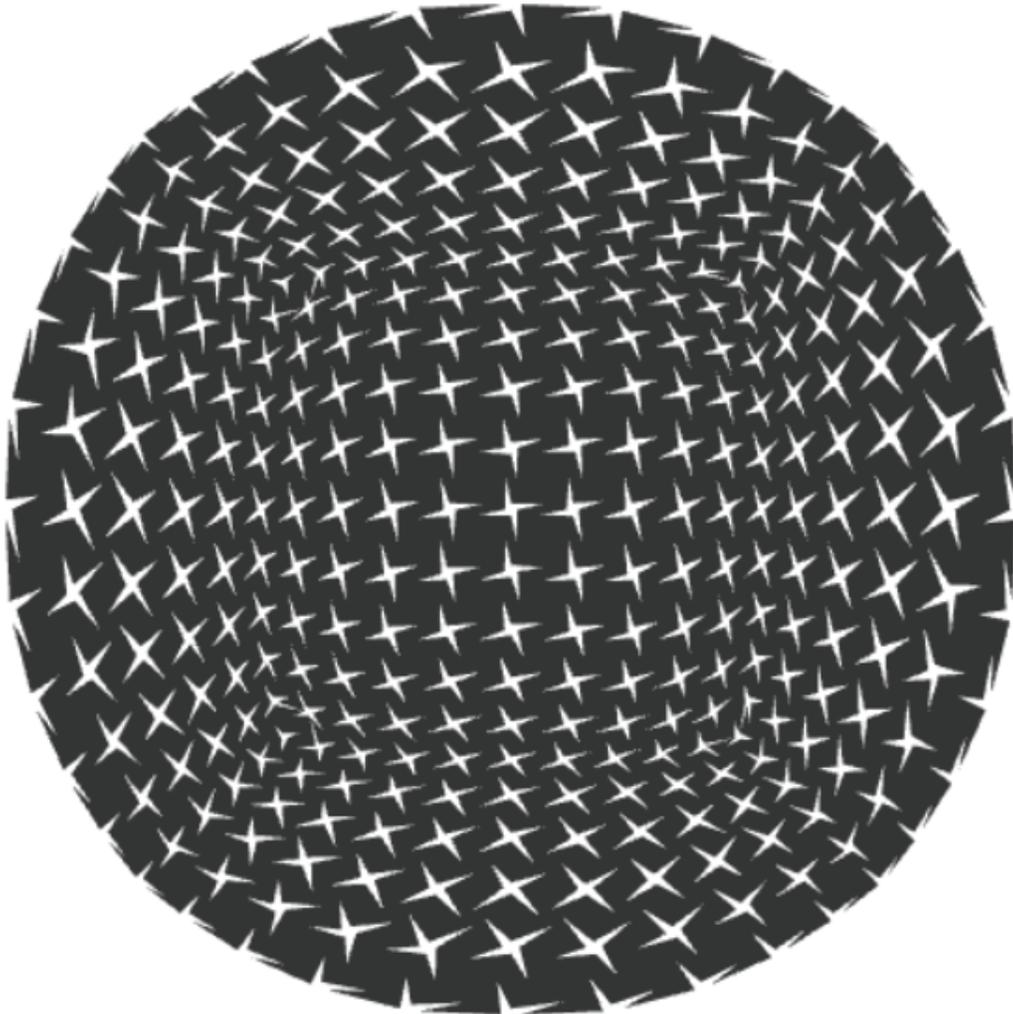
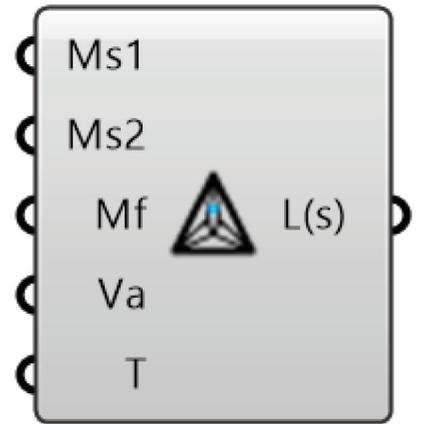
Mesh faces - (Mf) - [mesh] - Mesh face (exploded as single meshes)

Value (Va) - [number] - Value from 0-1 corresponding to each mesh face

Tolerance (T) - [number] - Tolerance for merging faces and vertices (~0.1 - 0.001)

### Outputs:

Lattice Hinge (shell) (Ls) - [mesh] - Joined mesh that make up the lattice hinge





## Mesh Hinge Cell - Tri (Mht)

### Description:

Map a triangle based parametric cell with two domains to mesh faces. (This is an example of how to use parameters from the “Mesh Flatten” component).

### Inputs:

Parameter 1a (t1a) - [number] - Value 0-1

Parameter 1b (t1b) - [number] - Value 0-1

Parameter 1c (t1c) - [number] - Value 0-1

Parameter 2a (t2a) - [number] - Value 0-1

Parameter 2b (t2b) - [number] - Value 0-1

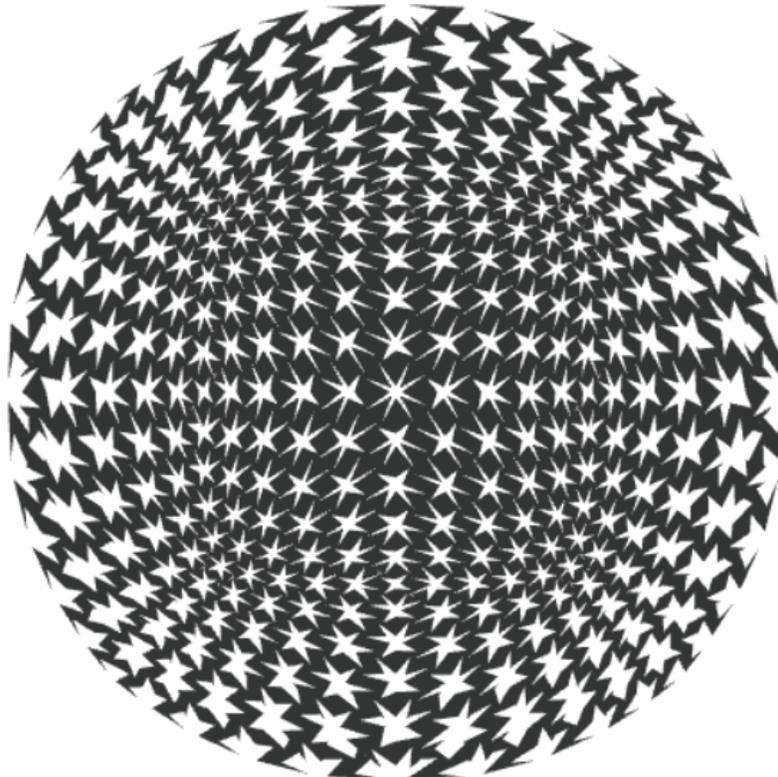
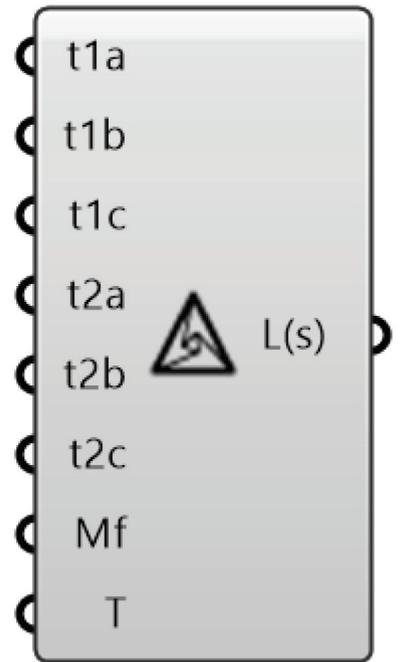
Parameter 2c (t2c) - [number] - Value 0-1

Mesh faces - (Mf) - [mesh] - Mesh face (exploded as single meshes)

Tolerance (T) - [number] - Tolerance for merging faces and vertices (~0.1 - 0.001)

### Outputs:

Lattice Hinge (shell) (Ls) - [mesh] - Joined mesh that make up the lattice hinge





## Mesh Hinge Cell - Quad (MHq)

### Description:

Map a quad based parametric cell with two domains to mesh faces. (This is an example of how to use parameters from the "Mesh Flatten" component).

### Inputs:

Parameter 1a (t1a) - [number] - Value 0-1

Parameter 1b (t1b) - [number] - Value 0-1

Parameter 1c (t1c) - [number] - Value 0-1

Parameter 1d (t1d) - [number] - Value 0-1

Parameter 2a (t2a) - [number] - Value 0-1

Parameter 2b (t2b) - [number] - Value 0-1

Parameter 2c (t2c) - [number] - Value 0-1

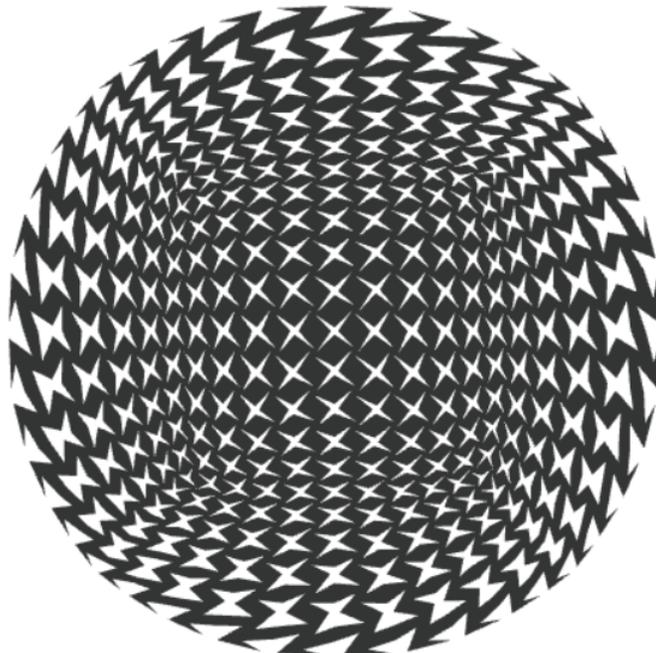
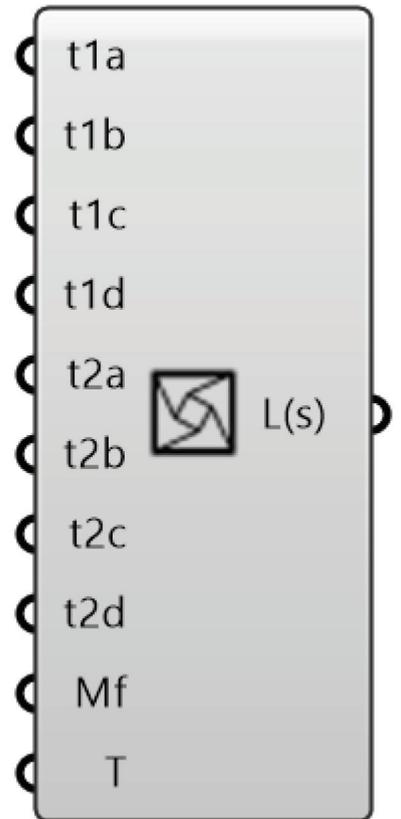
Parameter 2d (t2d) - [number] - Value 0-1

Mesh faces - (Mf) - [mesh] - Mesh face (exploded as single meshes)

Tolerance (T) - [number] - Tolerance for merging faces and vertices (~0.1 - 0.001)

### Outputs:

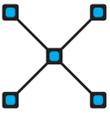
Lattice Hinge (shell) (Ls) - [mesh] - Joined mesh that make up the lattice hinge



## Utilities

This group of tools is for the organization, import and export of lattice geometry to other software tools.

Lattice Topology	LT	
LTCX Writer	LXO	<a href="#">&gt;LTCX</a>
LTCX Parser	LXI	<a href="#">LTCX&gt;</a>
INP B32	B32	<b>B32</b>
INP S3	S3	<b>S3</b>
INP S4	S4	<b>S4</b>
INP C3D6	C3D6	<b>C3D6</b>
INP C3D8	C3D8	<b>C3D8</b>
INP C3D4	C3D4	<b>C3D4</b>



## Lattice Topology (LT)

### Description:

Creates a clean graph from a lattice with unique nodes and their connections

### Inputs:

Lattice (L) - [curves] - Lattice curves

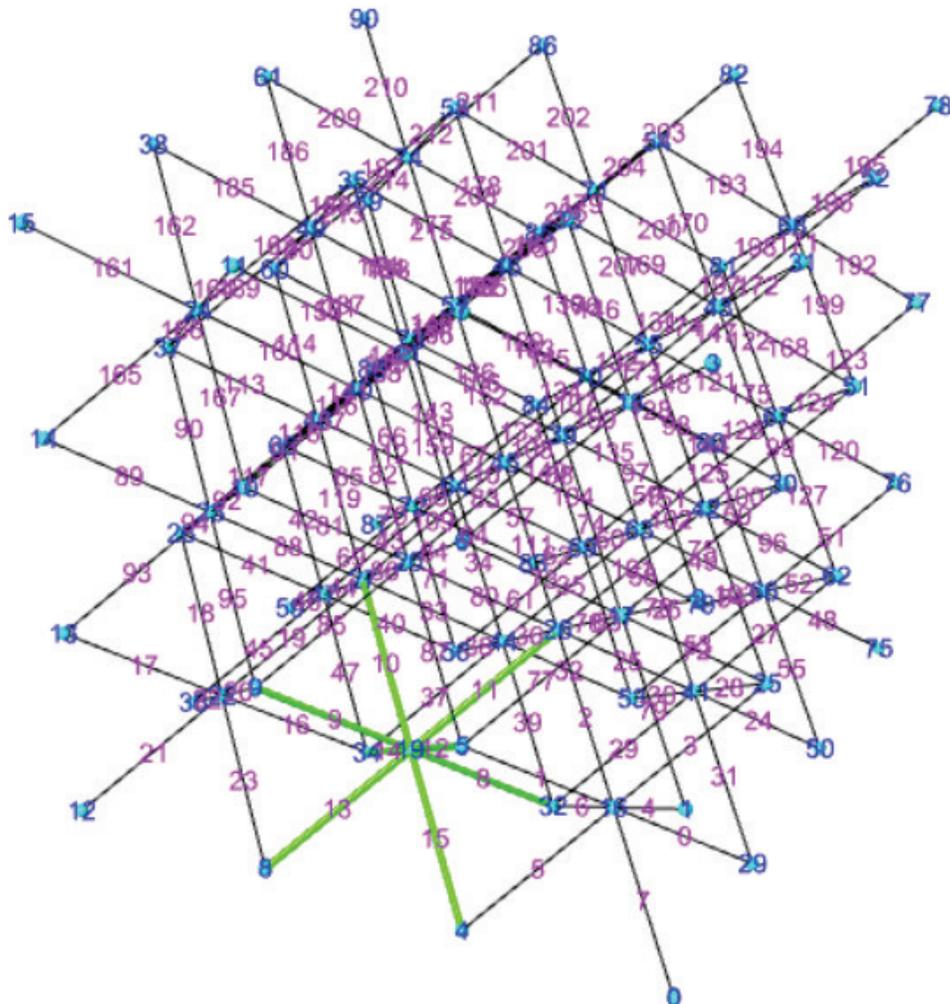
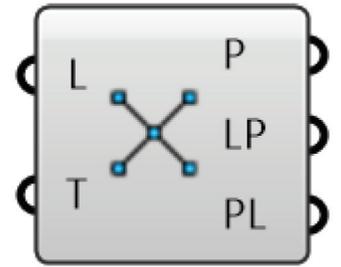
Tolerance (T) - [number] - Tolerance to merge points

### Outputs:

Nodes (P) - [point] - sorted unique points in the network

Line-Point Connectons (LP) - [number] - Index of points that make up each beam

Point-Line Connections (PL) - [number] - For each point, lists the index of the lines connected to it



## >LTCX

### LTCX Writer (LXO)

#### Description:

Creates a LTCX file format from a beam based lattice

#### Inputs:

Graph ID (ID) - [text] - id, unique (within the file) integer

Name (Name) - [text] - File name

Units (Units) - [text] - Units (in/mm/cm/m/ft)

Type (Tye) - [text] - Cross section (rnd)

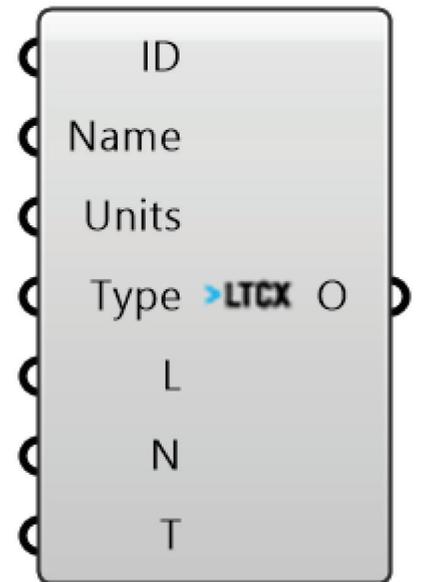
Lattice Curves (L) - [curve] - Lattice curves

Node Radius (N) - [number] - List of beam radii that will be averaged to each node

Tolerance (T) - [number] - Tolerance to merge points

#### Outputs:

Output (O) - [text] - output text (copy and paste contents to a text editor and save with \*.ltx extension)



```
{0}
0 <?xml version="1.0" encoding="UTF-8"?>
  <!--Generator: Grasshopper , LTCX export module , Aaron Porterfield. -->
1 <graph id="001" name="example" units="mm" type="rnd">
2 <nodegroup>
3     <node id="0" x="-10" y="-10" z="0" r="0.095"/>
4     <node id="1" x="-8" y="-8" z="2" r="0.099"/>
5     <node id="2" x="-10" y="-6" z="0" r="0.099"/>
6     <node id="3" x="-8" y="-4" z="2" r="0.106"/>
7     <node id="4" x="-10" y="-2" z="0" r="0.103"/>
8     <node id="5" x="-8" y="0" z="2" r="0.11"/>
9     <node id="6" x="-10" y="2" z="0" r="0.104"/>
10    <node id="7" x="-8" y="4" z="2" r="0.108"/>
11    <node id="8" x="-10" y="10" z="0" r="0.097"/>
12    <node id="9" x="-10" y="6" z="0" r="0.101"/>
13    <node id="10" x="-8" y="8" z="2" r="0.102"/>
14    <node id="11" x="-6" y="-10" z="0" r="0.095"/>
15    <node id="12" x="-4" y="-8" z="2" r="0.098"/>
16    <node id="13" x="-6" y="-6" z="0" r="0.099"/>
```

# LTCX >

## LTCX Reader (LXI)

### Description:

Reads a LTCX file and outputs beams and radii

### Inputs:

File path (L) - [text] - File path of .ltx file

Enable beams (E) - [boolean] - Enable the generation of beam curves

### Outputs:

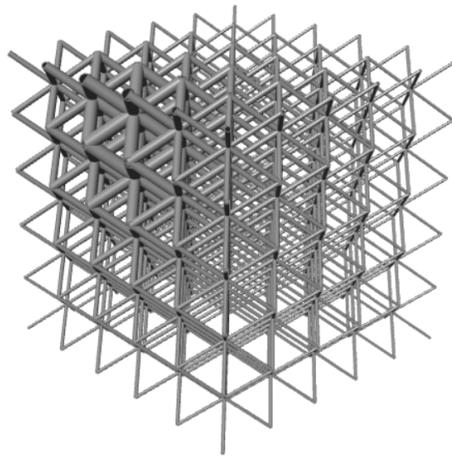
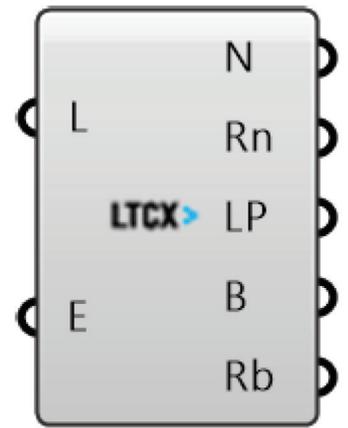
Nodes (P) - [point] - Nodes in the network

Node radius (Rn) - [number] - a list of radii for each node

Line-point connection (LP) - [number] - index of points that make up each beam

Beam curves (B) - [curve] - If enabled, the beams as curves

Beam Radius (Rb) - [number] - The radii of each beam (average of two nodes)



```
{0;0}
0 <?xml version="1.0" encoding="UTF-8"?>
1 <!--Generator: Grasshopper , LTCX export module , Aaron Porterfield. -->
2 <graph id="001" name="example" units="mm" type="rnd">
3 <nodegroup>
4     <node id="0" x="-10" y="-10" z="0" r="0.095"/>
5     <node id="1" x="-8" y="-8" z="2" r="0.099"/>
6     <node id="2" x="-10" y="-6" z="0" r="0.099"/>
7     <node id="3" x="-8" y="-4" z="2" r="0.106"/>
8     <node id="4" x="-10" y="-2" z="0" r="0.103"/>
9     <node id="5" x="-8" y="0" z="2" r="0.11"/>
10    <node id="6" x="-10" y="2" z="0" r="0.104"/>
11    <node id="7" x="-8" y="4" z="2" r="0.108"/>
12    <node id="8" x="-10" y="10" z="0" r="0.097"/>
13    <node id="9" x="-10" y="6" z="0" r="0.101"/>
14    <node id="10" x="-8" y="8" z="2" r="0.102"/>
15    <node id="11" x="-6" y="-10" z="0" r="0.095"/>
16    <node id="12" x="-4" y="-8" z="2" r="0.099"/>
```

# B32

## INP B32 (B32)

### Description:

Create a B32 INP file from a lattice

### Inputs:

Filename (F) - [text] - Name of file (also NSET, ELSET, etc)

Lattice Curves (L) - [curve] - Lattice curves

Tolerance (T) - [number] - Tolerance to merge points

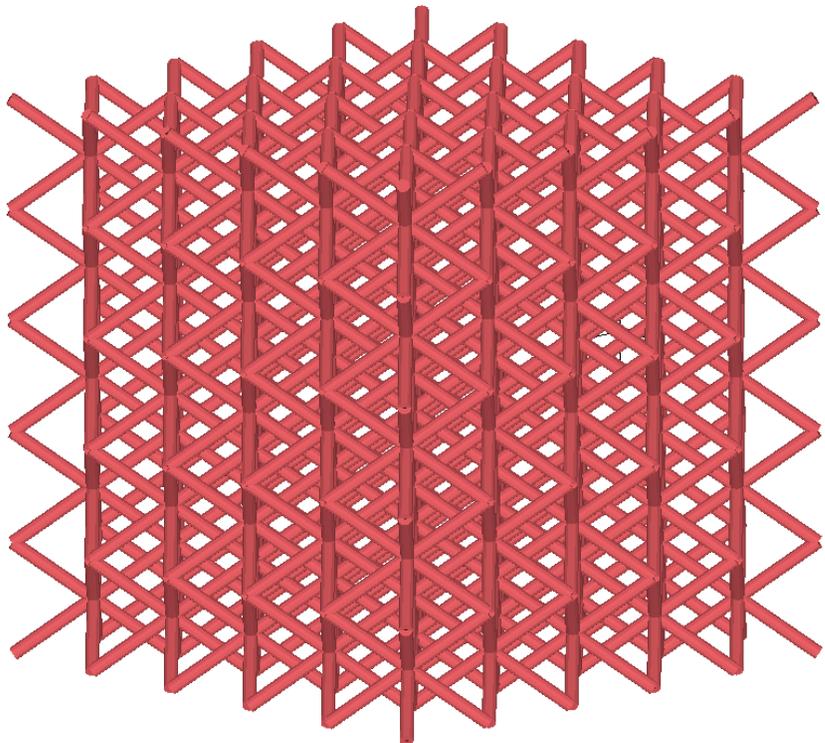
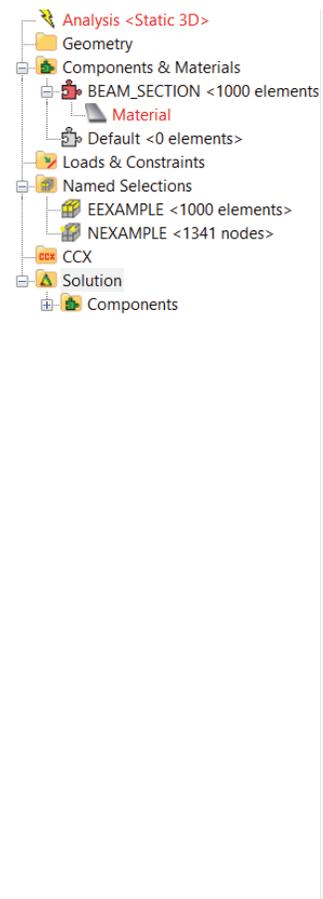
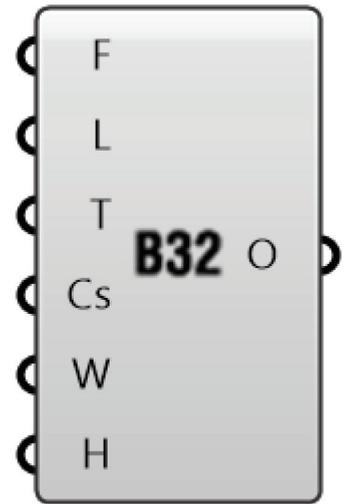
Cross Section (Cs) - [text] - True if CIRC, false if RECT

Width (W) - [number] - Width of cross section

Height (H) - [number] - Height of cross section

### Outputs:

Output (O) - [text] - output text (copy and paste contents to a text editor and save with \*.inp extension)



# S3

## INP S3 (S3)

### Description:

Create a S3 INP file from a triangle based shell.

### Inputs:

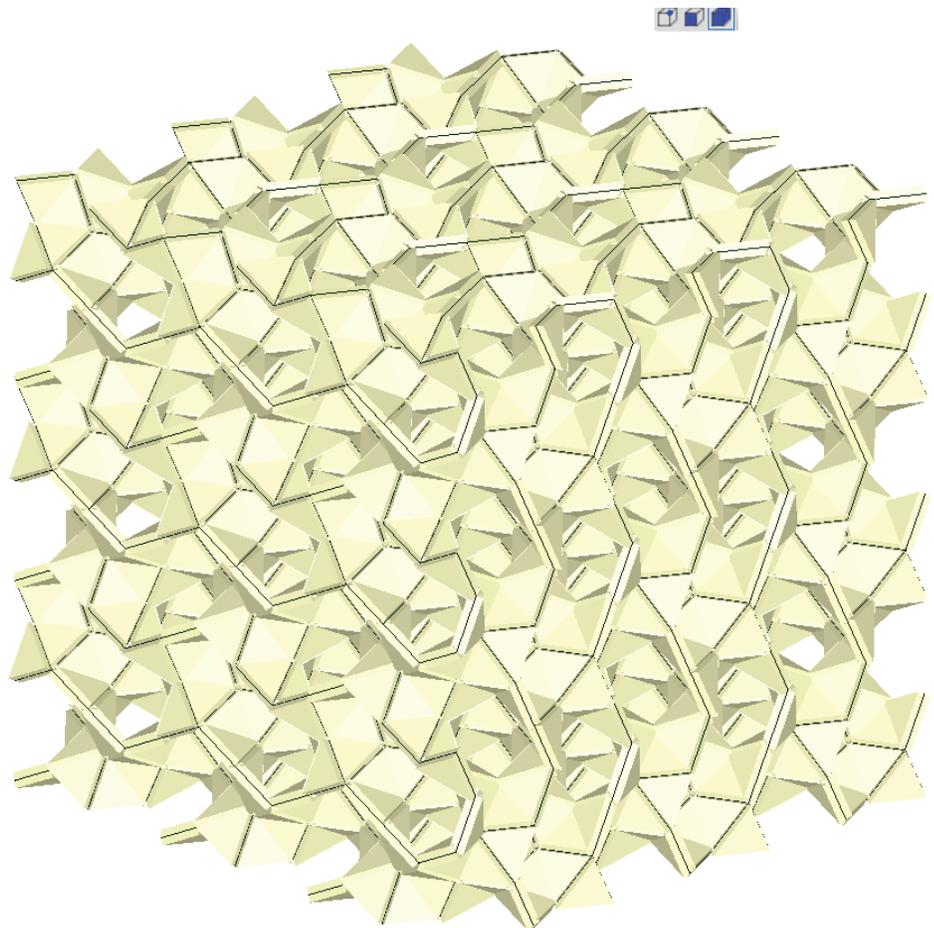
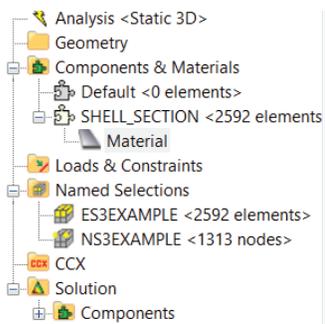
Filename (F) - [text] - Name of file (also NSET, ELSET, etc)

Lattice shell (L(s)) - [mesh] - Lattice shell (as joined mesh)

Thickness (T) - [number] - Thickness of cross section

### Outputs:

Output (O) - [text] - output text (copy and paste contents to a text editor and save with \*.inp extension)



# S4

## INP S4 (S4)

### Description:

Create a S4 INP file from a quad based shell.

### Inputs:

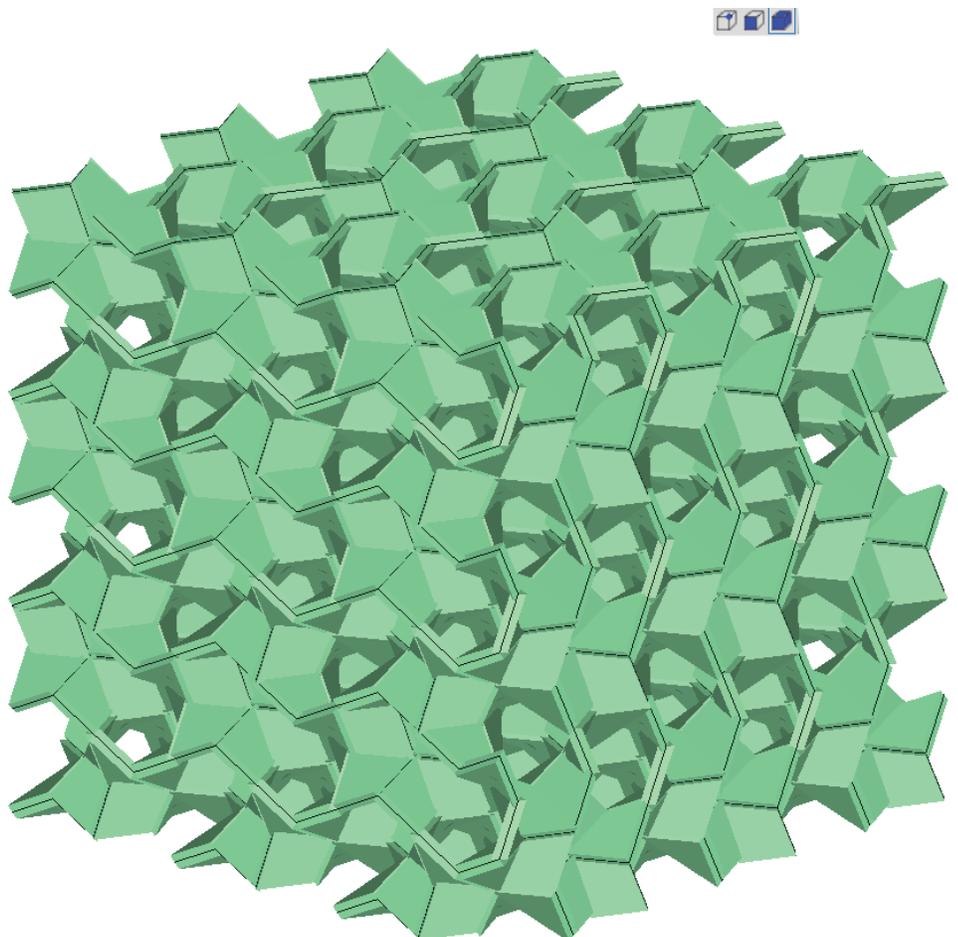
Filename (F) - [text] - Name of file (also NSET, ELSET, etc)

Lattice shell (L(s)) - [mesh] - Lattice shell (as joined mesh)

Thickness (T) - [number] - Thickness of cross section

### Outputs:

Output (O) - [text] - output text (copy and paste contents to a text editor and save with \*.inp extension)



# C3D6

## INP C3D6 (C3D6)

### Description:

Creates a C3D6 INP file from a triangulated mesh.

### Inputs:

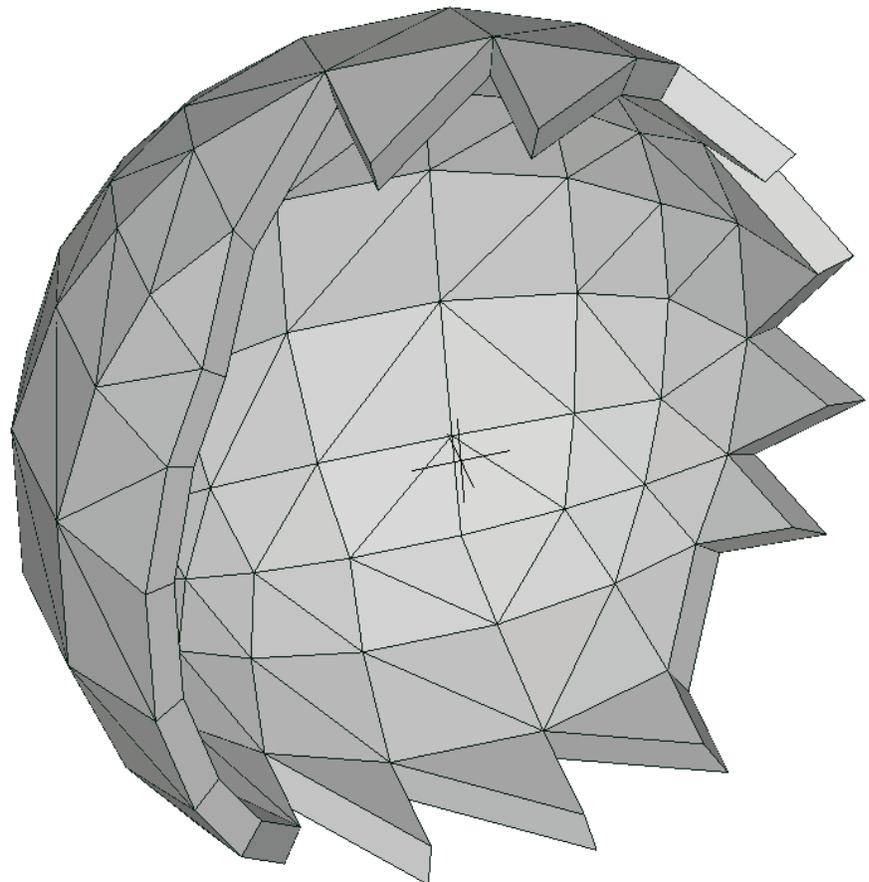
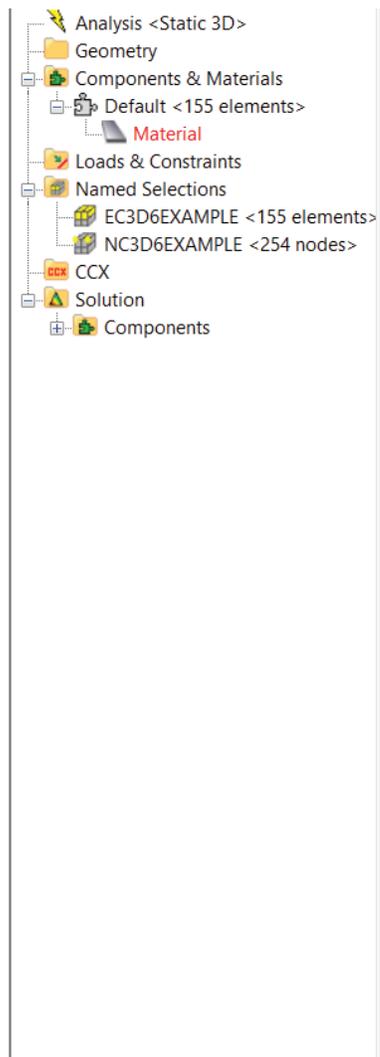
Filename (F) - [text] - Name of file (also NSET, ELSET, etc)

Lattice shell (L(s)) - [mesh] - Lattice shell (as joined mesh)

Thickness (T) - [number] - Thickness of cross section (offset)

### Outputs:

Output (O) - [text] - output text (copy and paste contents to a text editor and save with \*.inp extension)



# C3D8

## INP C3D8 (C3D8)

### Description:

Creates a C3D8 INP file from a quad mesh.

### Inputs:

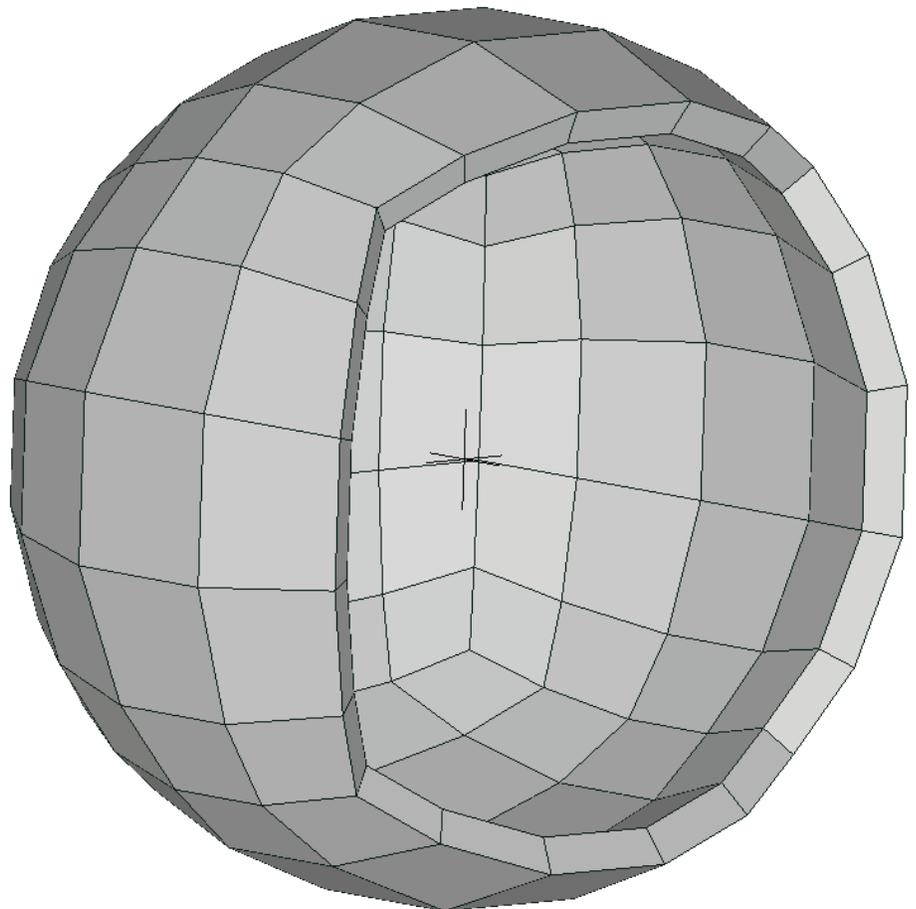
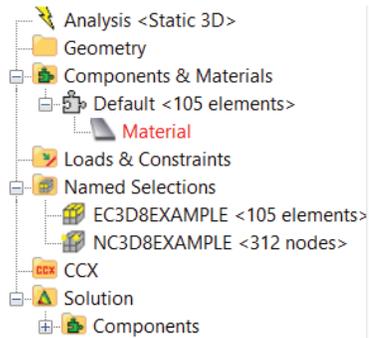
Filename (F) - [text] - Name of file (also NSET, ELSET, etc)

Lattice shell (L(s)) - [mesh] - Lattice shell (as joined mesh)

Thickness (T) - [number] - Thickness of cross section (offset)

### Outputs:

Output (O) - [text] - output text (copy and paste contents to a text editor and save with \*.inp extension)



# C3D4

## INP C3D4 (C3D4)

### Description:

Creates a C3D4 INP file from a tetrahedral mesh (requires Tetrino  
<https://www.food4rhino.com/app/tetrino>)

### Inputs:

Filename (F) - [text] - Name of file (also NSET, ELSET, etc)

Indices (I) - [number] - Indices of tetrahedrons ("I" output of Tetgen)

Points (P) - [point] - Points of tetrahedral mesh ("P" output of Tetgen)

### Outputs:

Output (O) - [text] - output text (copy and paste contents to a text editor and save with \*.inp extension)

