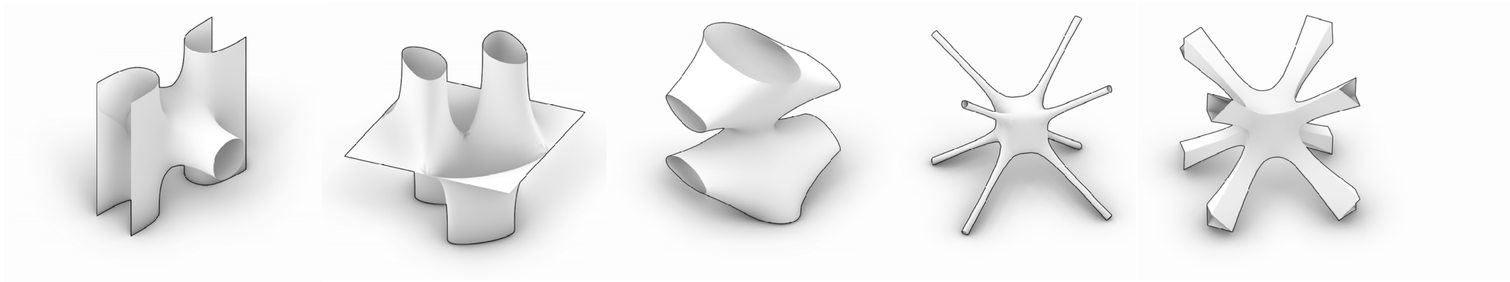


200 Lines of Code

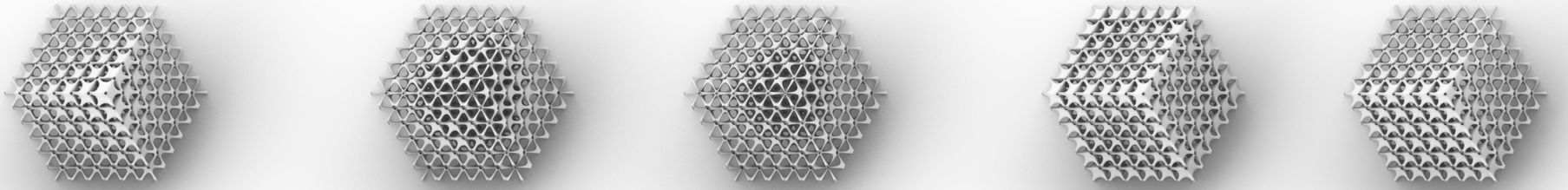
ARCH 696 LEC 01 Somerville Design Charrette

Miriam Navarrete, Obinna Ekezie, Mackenzie Garvin, Inioluwa Adedapo, Esther Ephraim-Osunde

Instructor: ***Shajay Bhooshan***

March 14 -18 2022

Contents

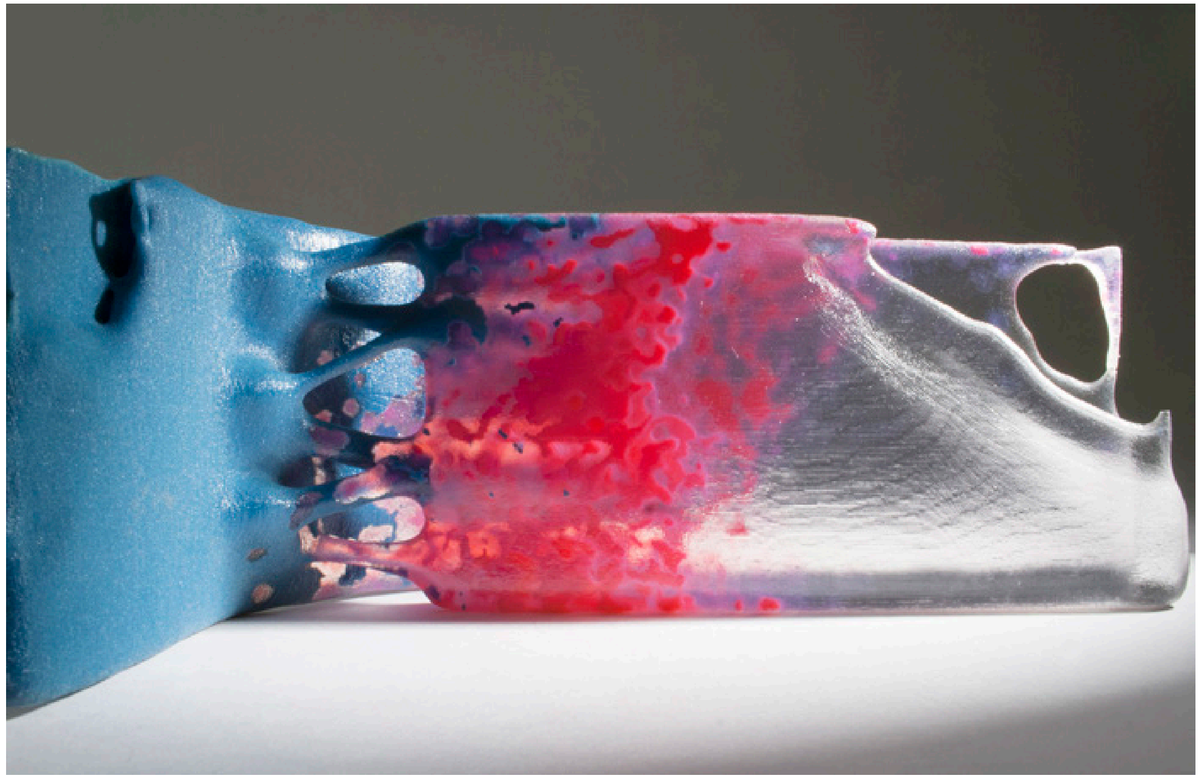


Functionally Graded Materials

https://media.springernature.com/relative-r300-703_m1050/springer-static/image/art%3A10.1038%2Fnrrheum.2016.37/MediaObjects/41584_2016_Article_BFnrrheum201637_Figa_HTML.jpg?as=webp
Credit: PHOTOTAKE Inc./Alamy

The human bone marrow

Looking at bone mineralization density distribution (BMDD), a measure of the distribution of calcium concentration across the bone matrix.



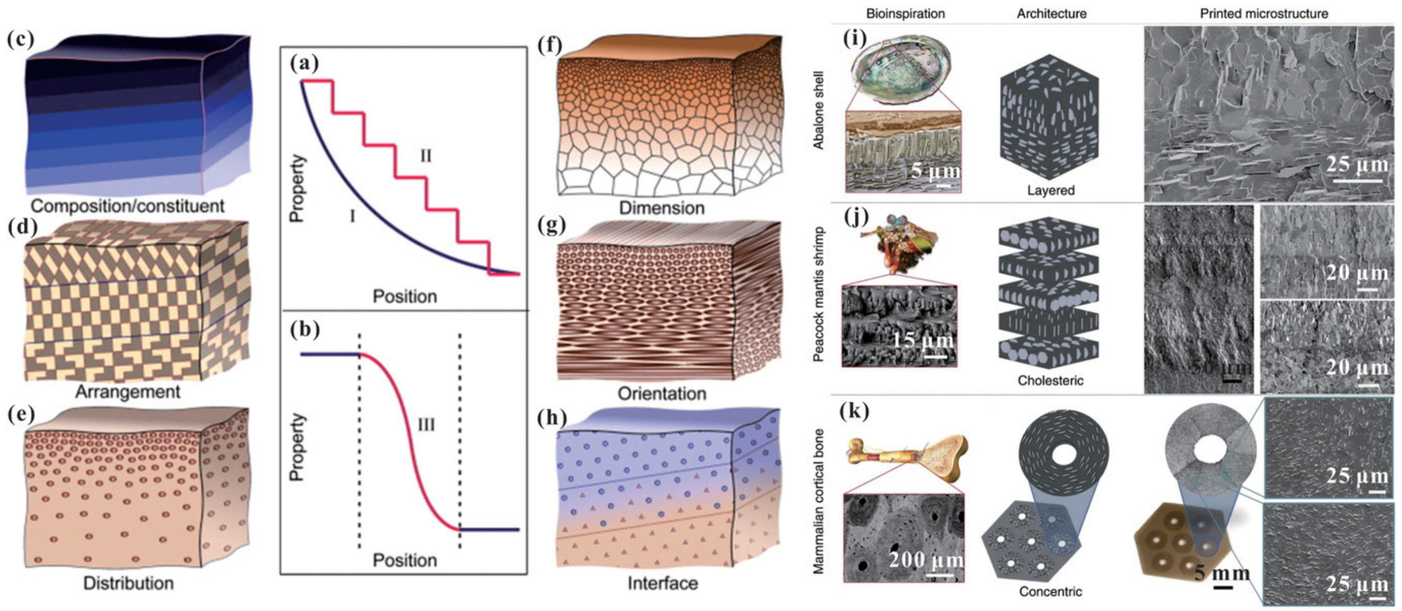
https://riba-prd-assets.azureedge.net/-/media/GatherContent/Computational-Blends/Image-One/PAR223Webjpg.jpg?rev=a90f0e62958e4a75935c5e4f5da73823&h=455&w=700&la=en&hash=36A764E5776C160B3BCFDEB774144B7E

Close up interior view of the fabricated multi-material mullion interface

Computational Blends: The Epistemology of Designing with Functionally Graded Materials (C) Kostas Grigoriadis

Functionally Graded Materials (FGMs) are materials whose ***properties are altered or changed in a manner that is extensively dependent on the material's geometric dimensions or arrangements****. With a change in one dimension or property, the characteristics also change. These materials are* ***usually composite****- that is a combination of more than one material. Thus, the geometric characteristics that evolve from this* ***combination of materials are inherently better for the purpose than the individual materials on their own****. For example, if aggregated in a way that they become denser at a point, that portion can become stronger.*

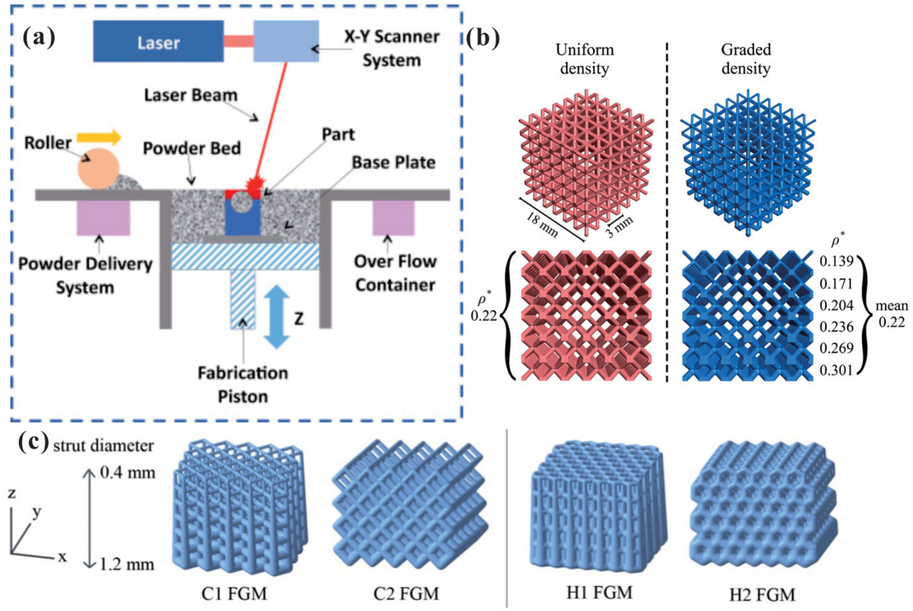
Functionally Graded Materials



https://onlinelibrary.wiley.com/cms/asset/4d0774e4-57b7-4ee2-acb1-df5fde5db140/admt201900981-fig-0013-m.jpg

Biocompatibility and Biomedical Applications / Local property profiles and basic forms of gradients in biological materials.

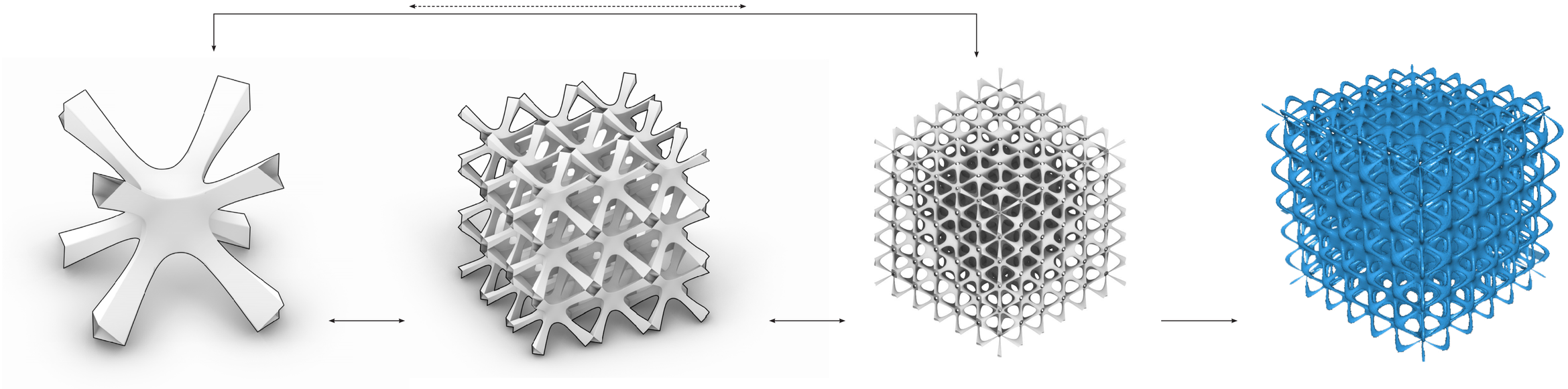
Looking at Bioligical gradients and their mechanical functions



https://onlinelibrary.wiley.com/cms/asset/fd4c6ccf-002f-4c63-b7dc-718f368779ac/admt201900981-fig-0009-m.jpg

Material Structure studies using Power Bed Fusion.

Building metal or polymer prototypes using lasers. PBF uses energy density and special structures to develop functionally graded variations. By selectively melting the powder using a laser beam at targeted regions repeatedly, single layers are stacked to form the final product.



Maya

- *Formal Explorations within 1x1x1 Voxels*
- *Iterations on combinatorial logic*

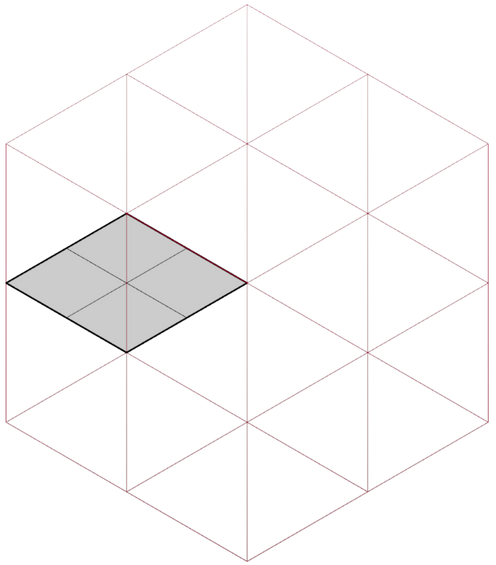
Visual Code + Alice

- *Coding voxel stacking and nodal aggreggation.*
- *Iterations on various topology towards a final developable option*

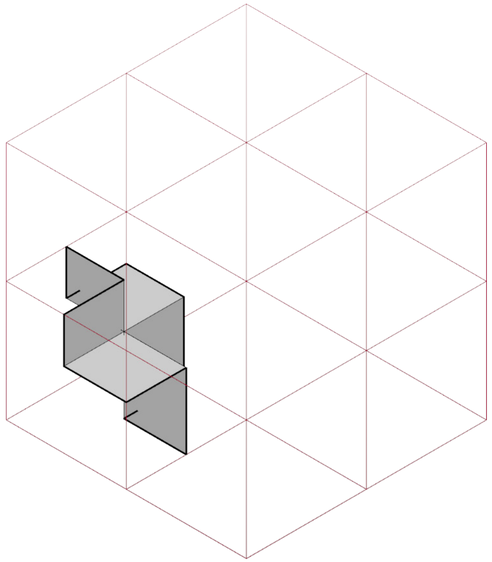
Chitubox 3DP

- *3D printing of final object*

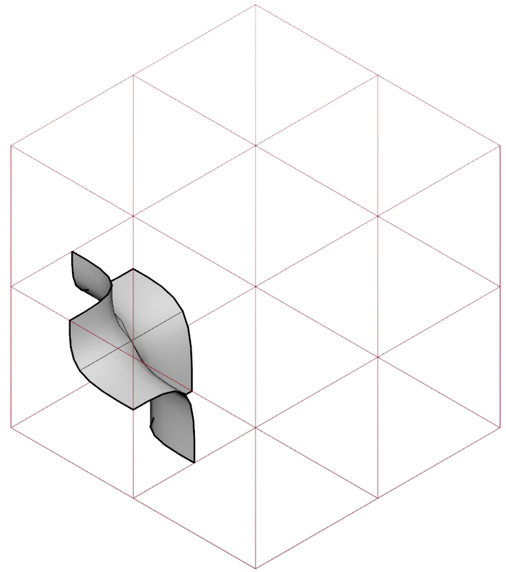
Maya Studies 1



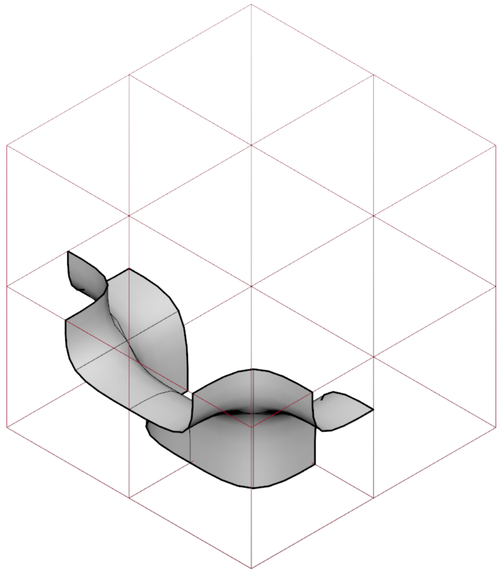
1.



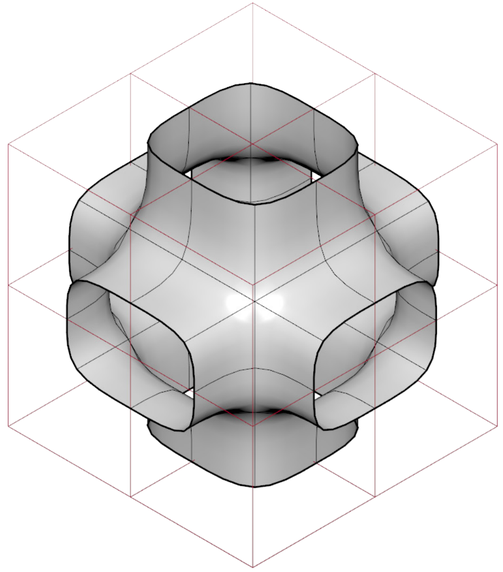
2.



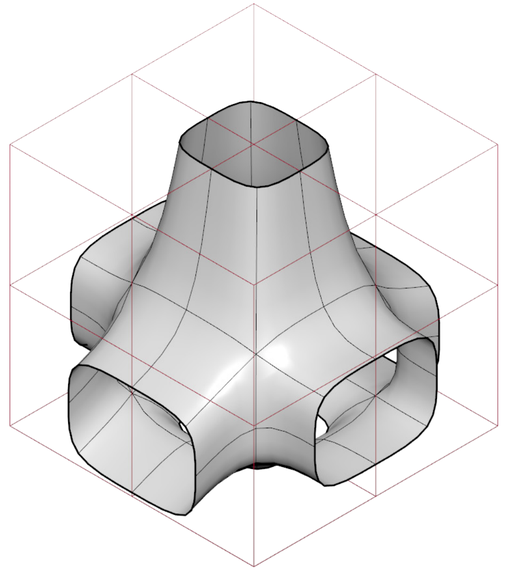
3.



4.

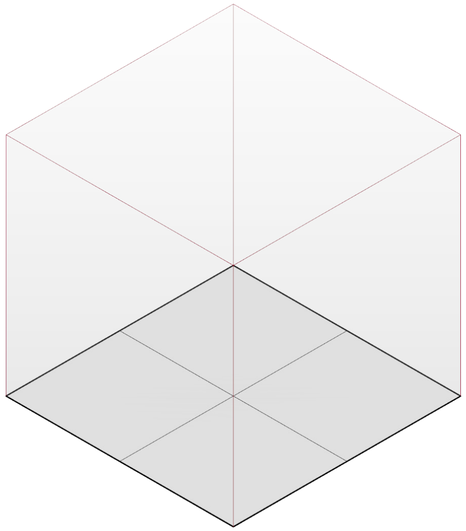


5.

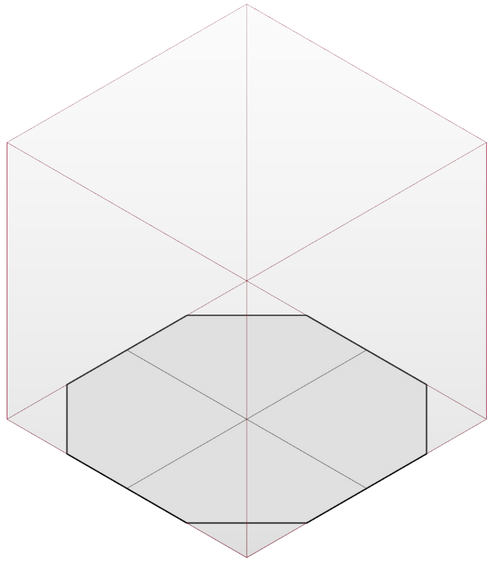


6.

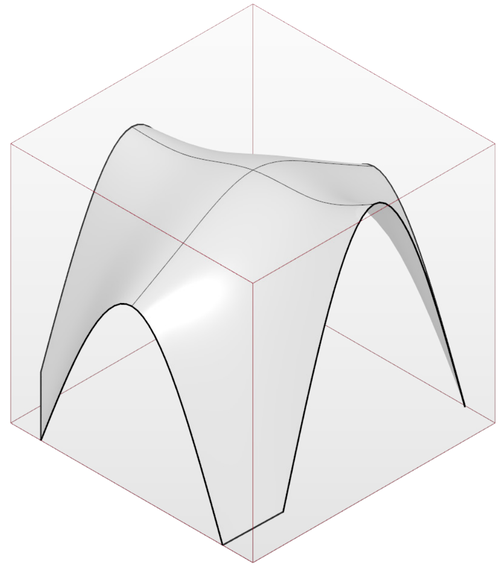
Maya Studies 2



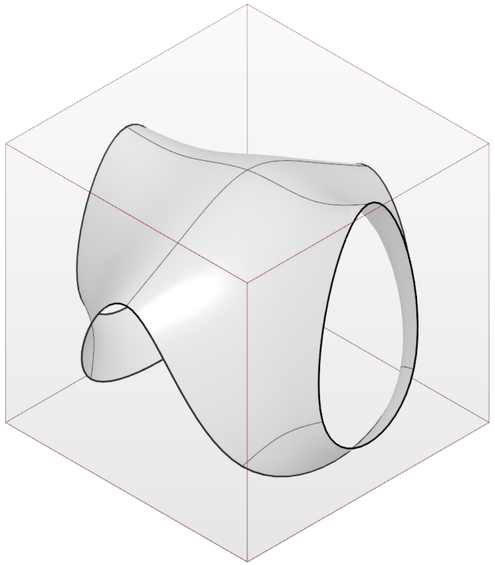
1.



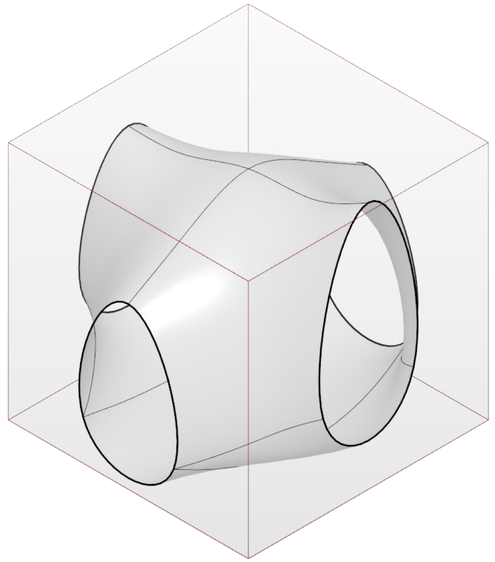
2.



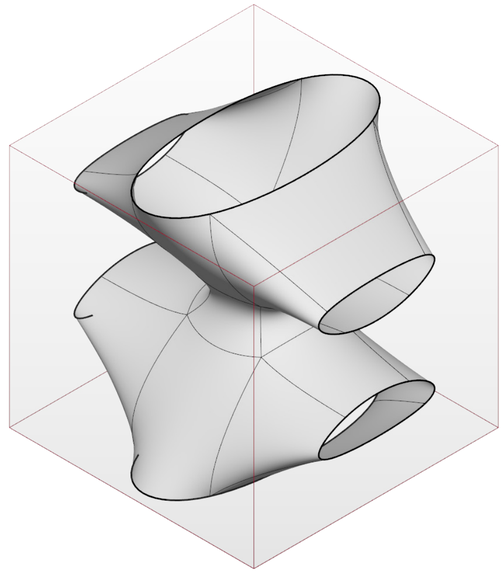
3.



4.

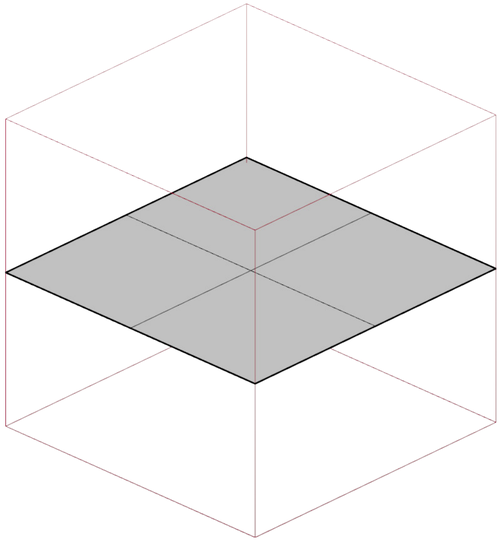


5.

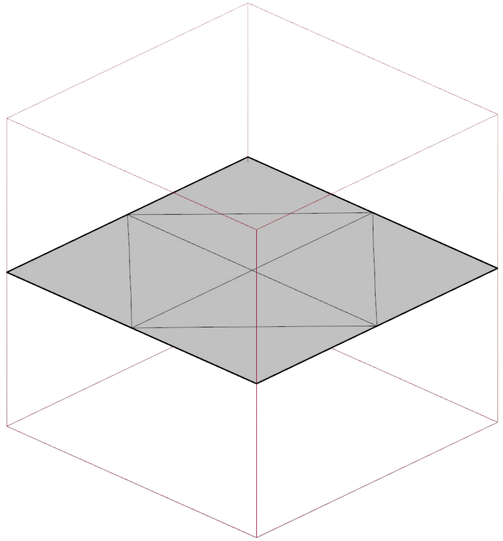


6.

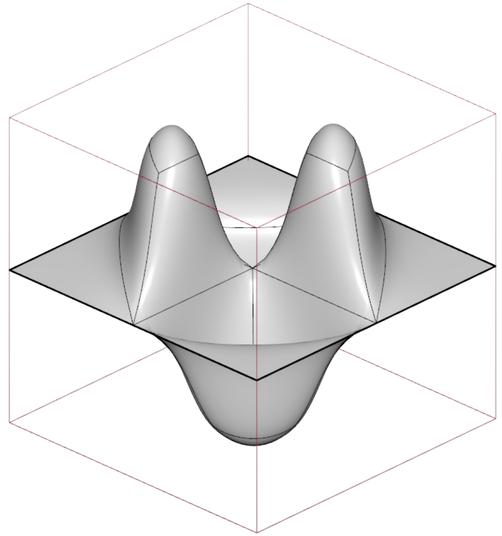
Maya Studies 3



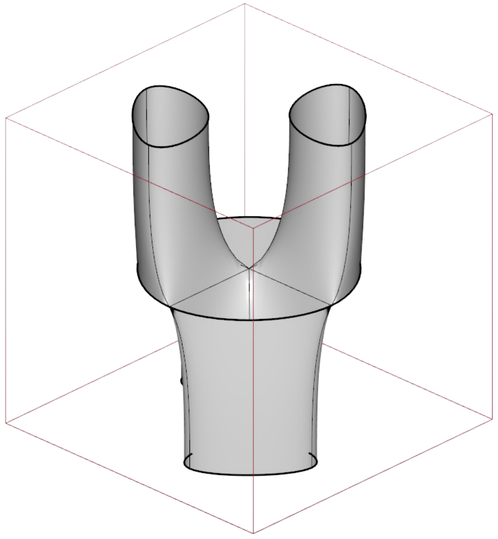
1.



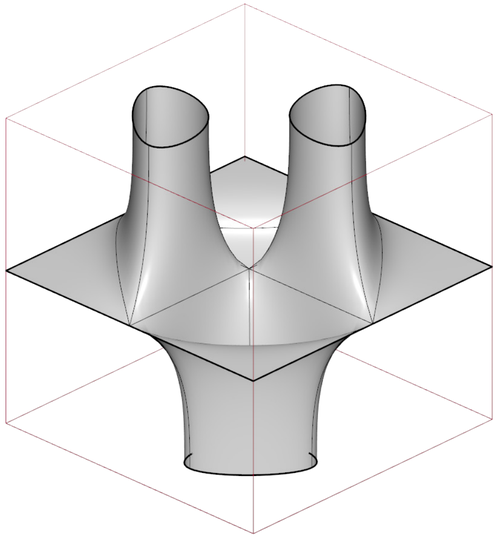
2.



3.

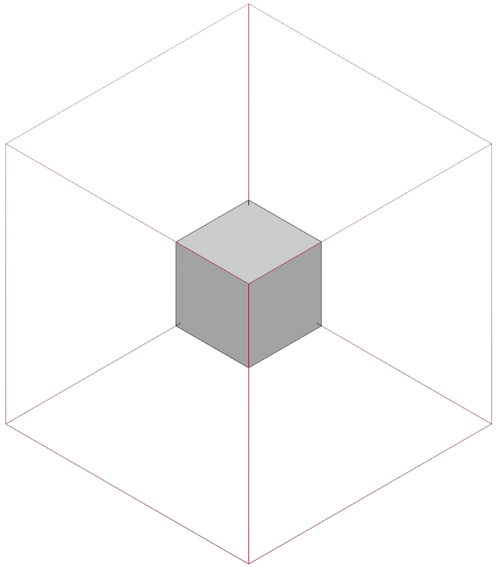


4.

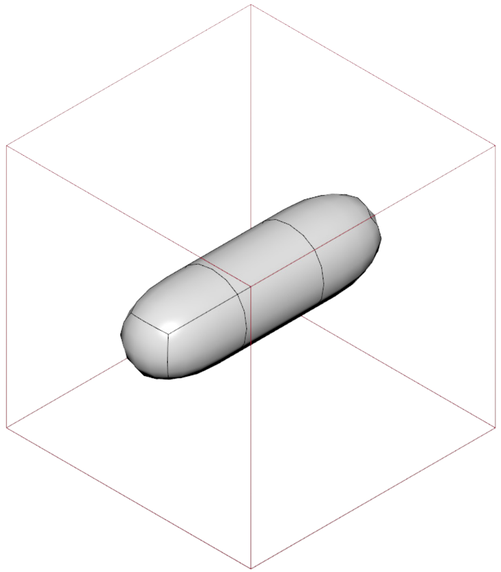


5.

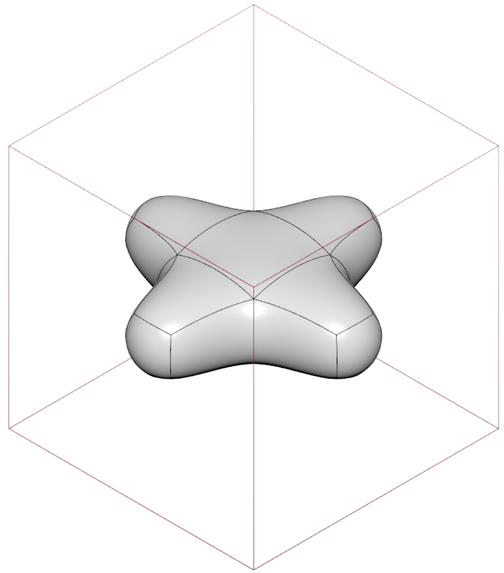
Maya Studies 4



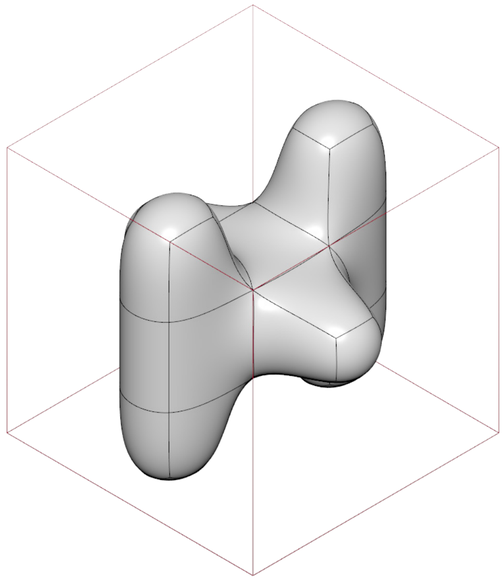
1.



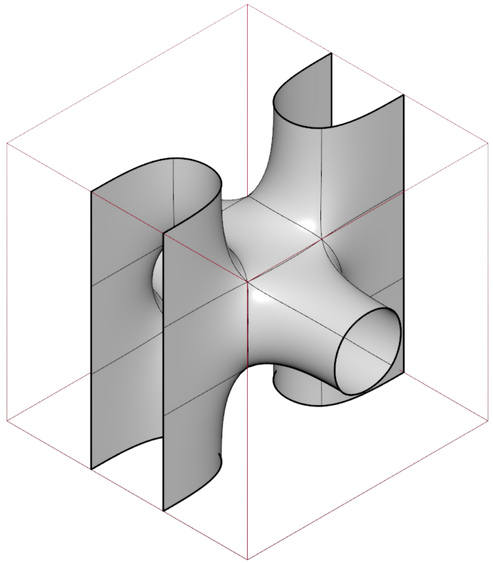
2.



3.

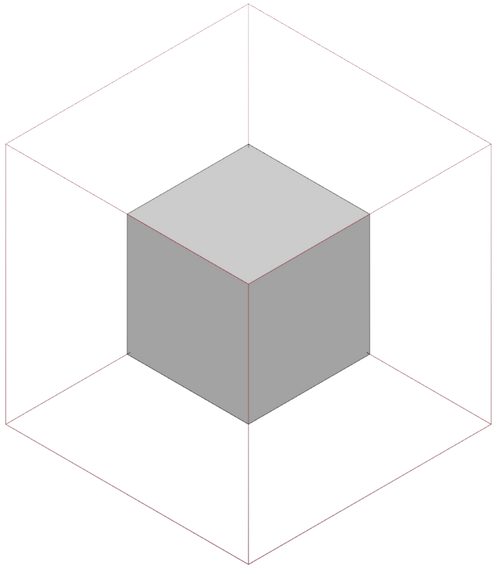


4.

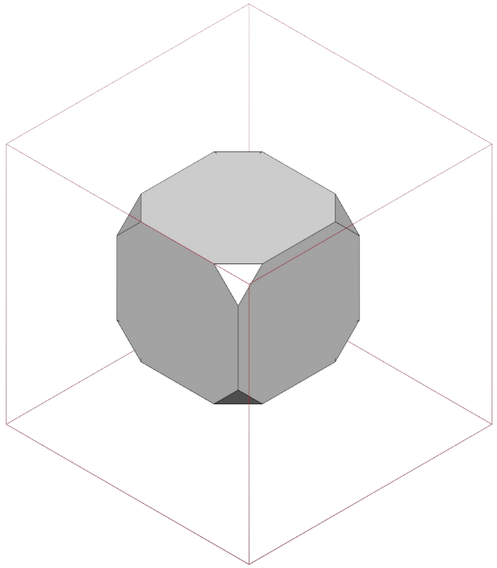


5.

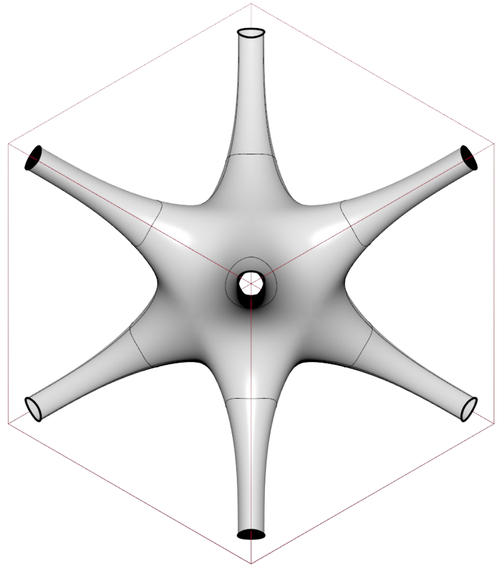
Maya Studies 5



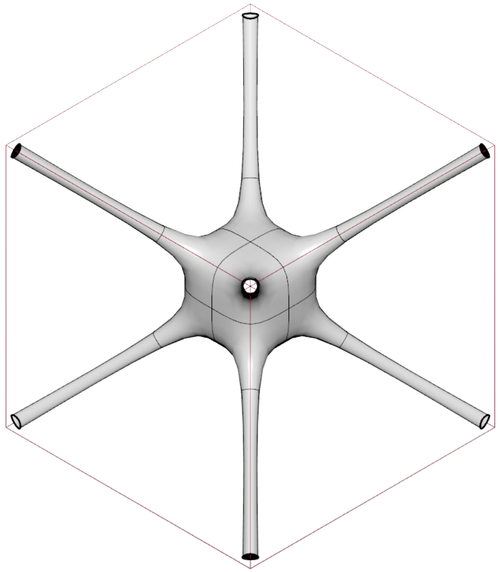
1.



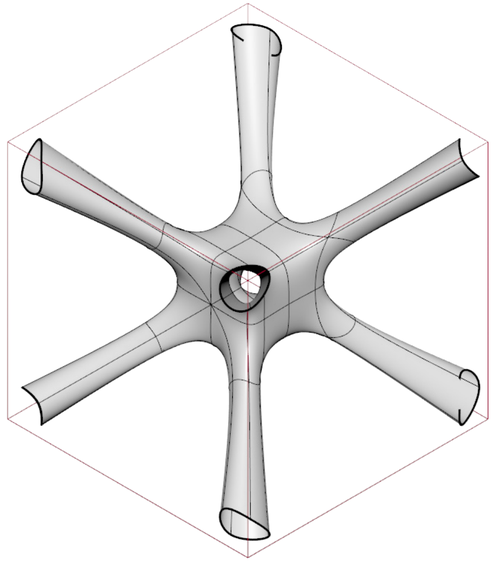
2.



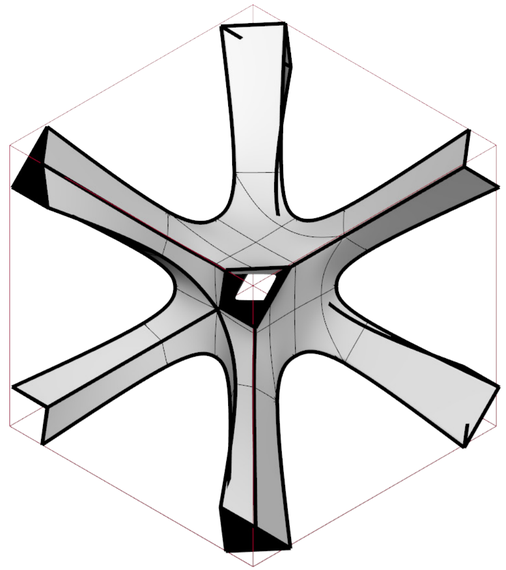
3.



4.

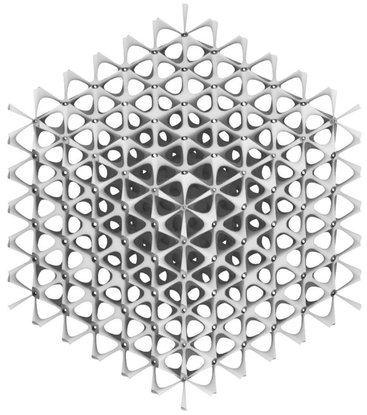


5.



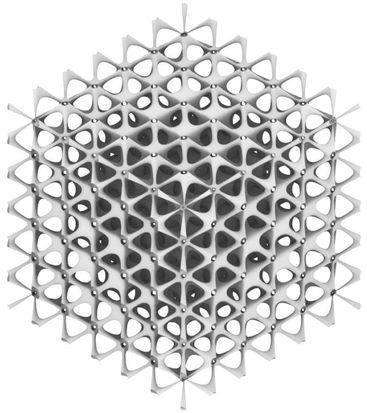
6.

Combinatorial Studies



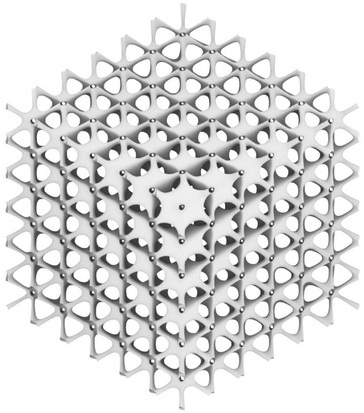
1.

Aggregating the form in a grid while creating a centre to outside gradient affecting the scaling of the central mesh vertices, slowly expanding the shape. Creating an aggregated form that is densest at the centre and becomes less dense as it expands.. .



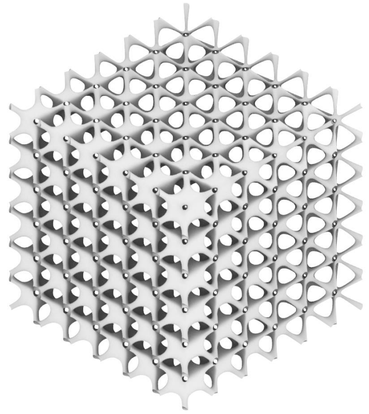
2.

Expanding on option 1 and adding a higher scale factor to the effected vertices. Creating an aggregated form that is densest at the centre and becomes less dense as it expands.



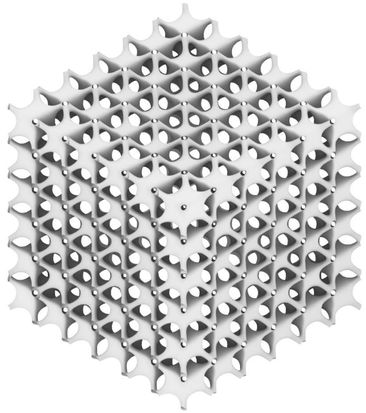
3.

Aggregating the form in a grid while creating a left to right gradient in central mesh vertices, slowly expanding the shape.



4.

Aggregating the form in a grid while creating an outside to centre gradient affecting the scaling of the central mesh vertices, slowly expanding the shape. Creating an aggregated form that is densest at the outside corners.



5.

Aggregating the form in a grid while creating a diagonal gradient affecting the scaling of the central mesh vertices, slowly expanding the shape. Creating an aggregated form that is densest at the upper corner and becomes less dense as the form is aggregated throughout the voxels.

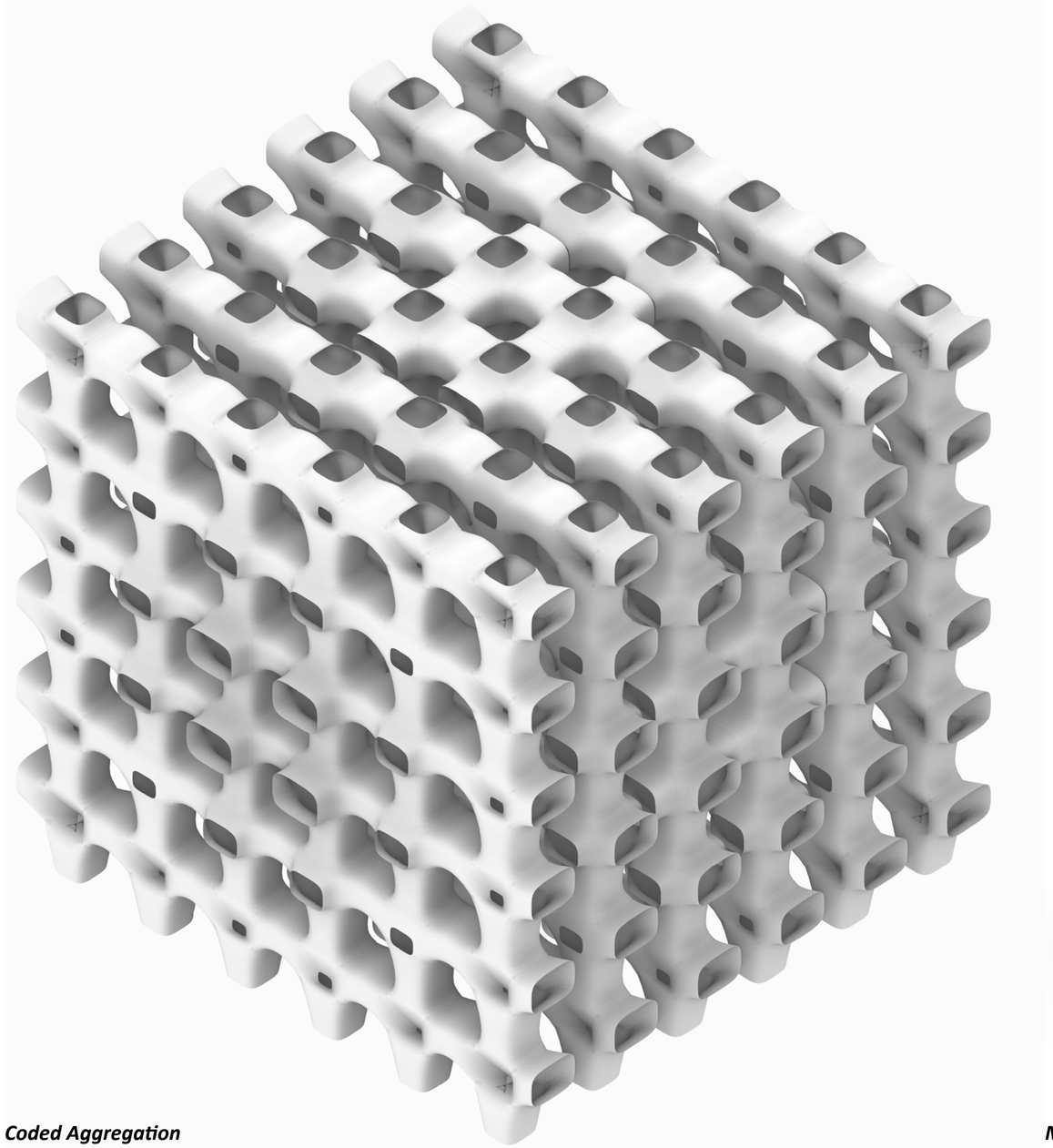
Failed Coded (C++) Aggregations 1

Intension / Process

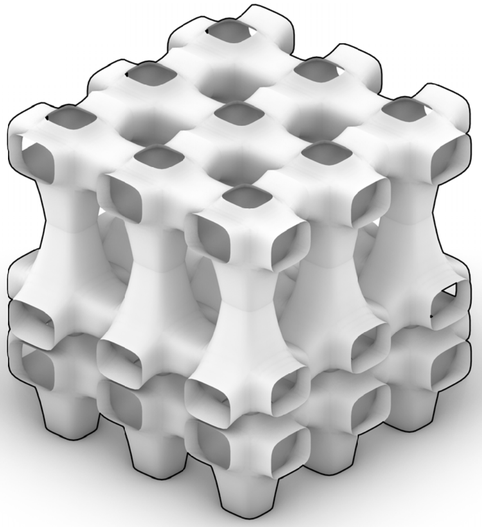
- Deploying a minimal surface aggregation.
- Add a scale component to 2 of the open faces.

Lessons

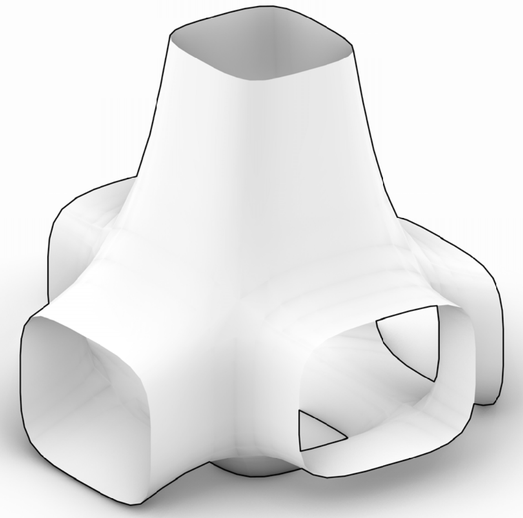
- The change in scale at the meeting faces between geometries caused anomalies in the aggregation.
- The scale factor should be applied at non-meeting faces of the geometry.



Coded Aggregation



Manual Aggregation



Voxel Topology

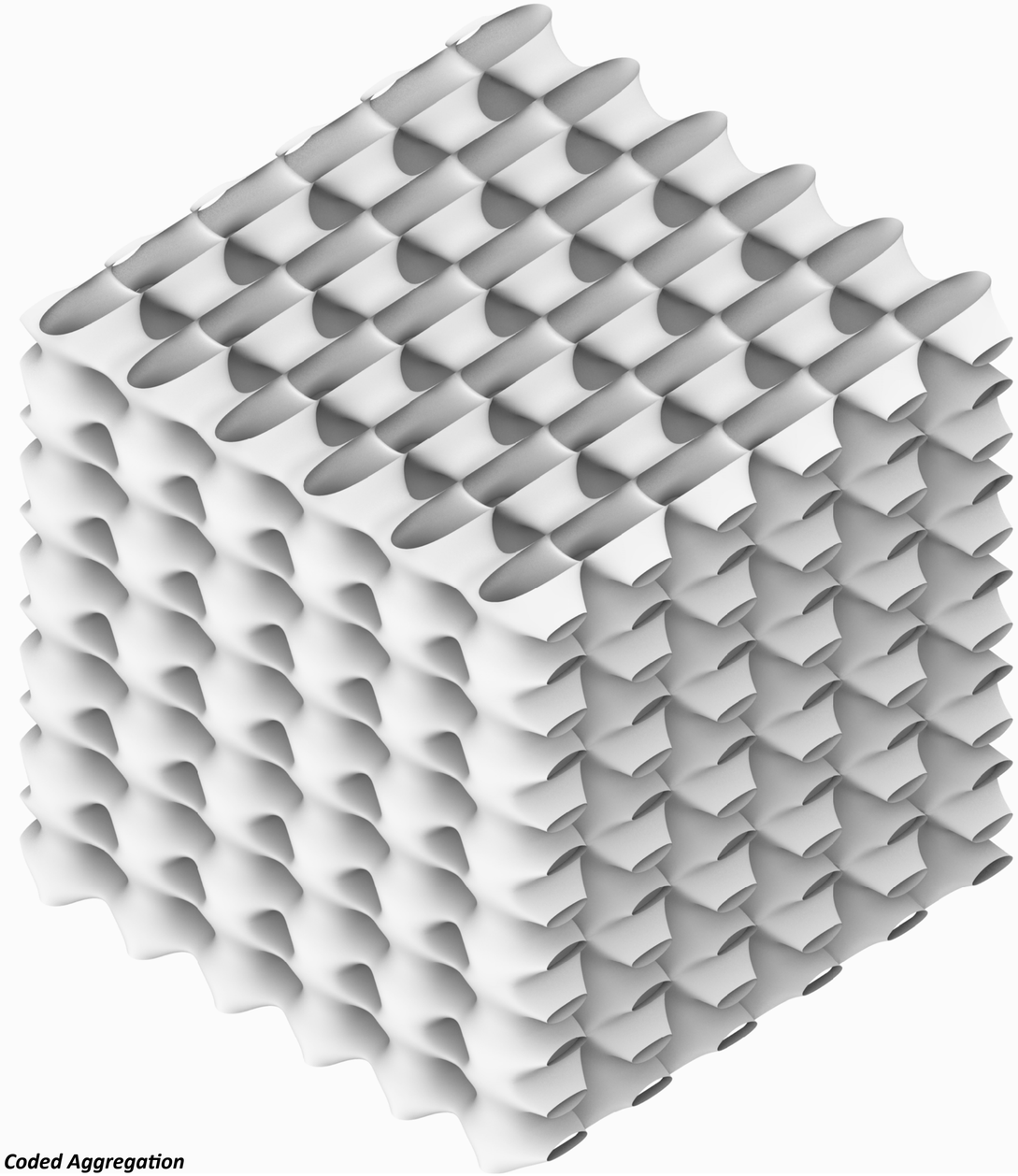
Failed Coded (C++) Aggregations 2

Intension / Process

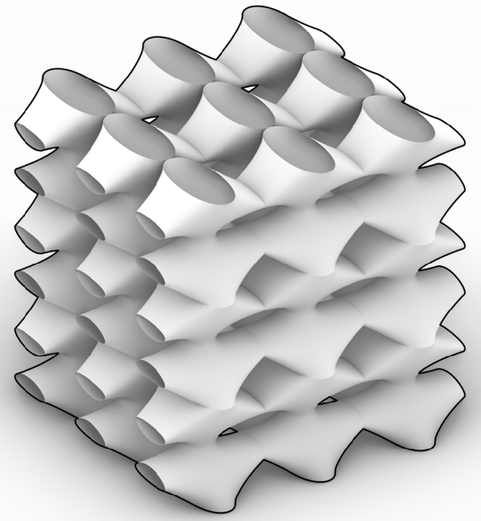
- Creating geometry using minimal surfaces.
- Creating a controlled complexity from simple geometry.

Lessons

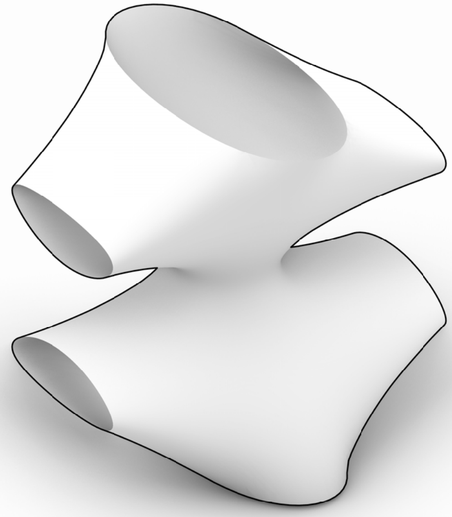
- Topology should have nodal connections at the 4 sides of the voxel to create developable/working aggregation.
- The face subdivision of the topology should have equal edge divisions.



Coded Aggregation



Manual Aggregation



Voxel Topology

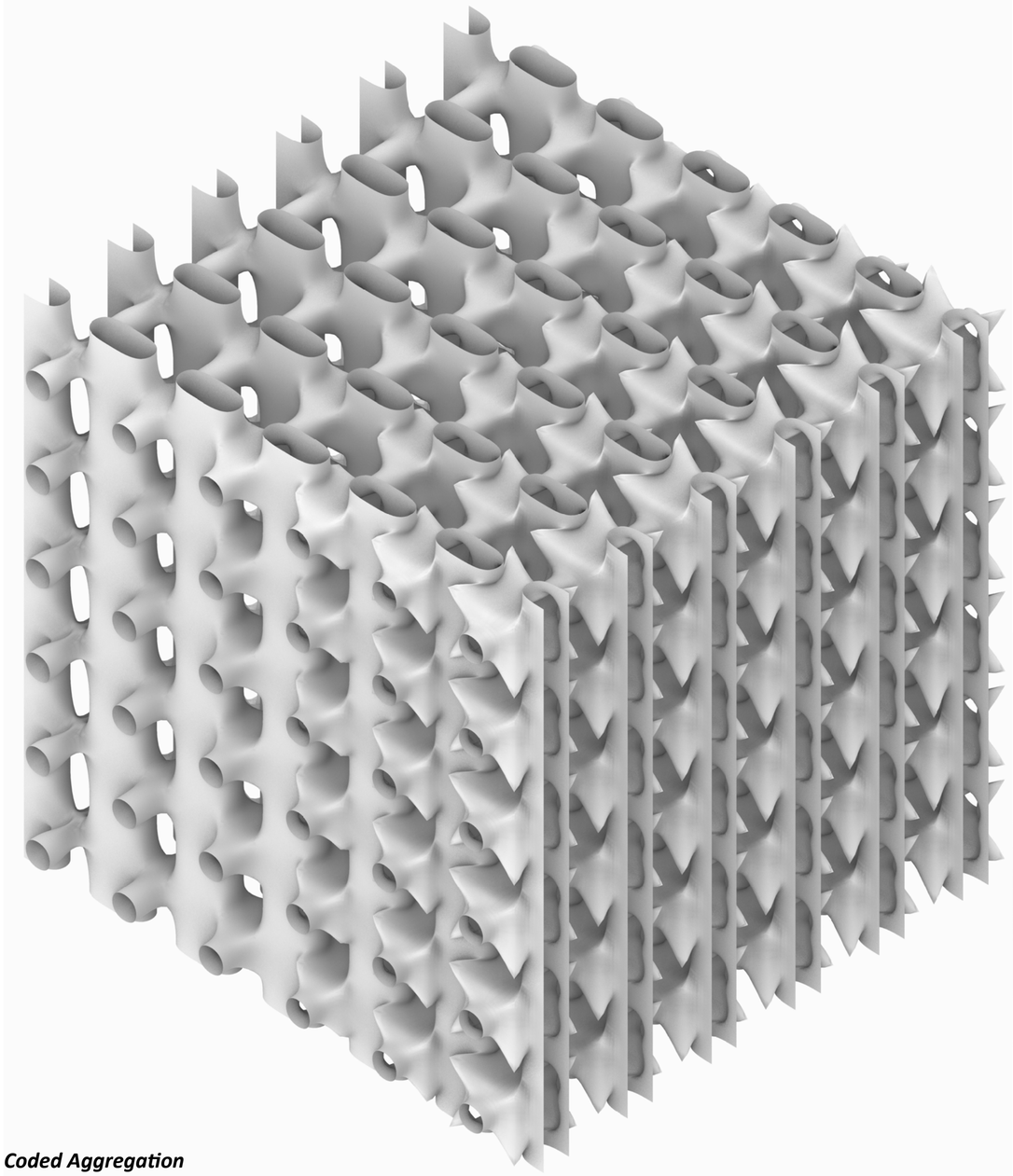
Failed Coded (C++) Aggregations 3

Intension / Process

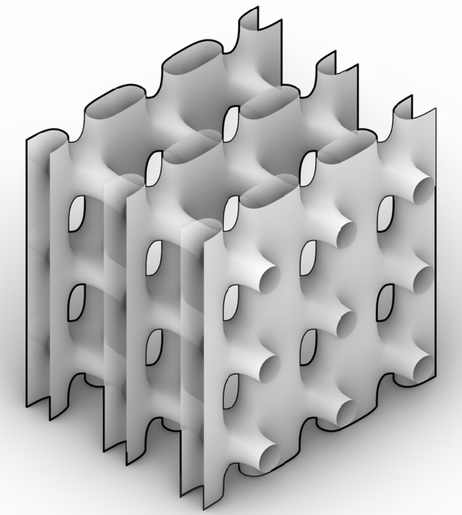
- Aggregating basic shapes to create a geometrically complex form
- Create cube to 1, 1, 1
- Scale cube to 1/3 the size
- Extrude faces of cube to form geometry
- Delete specific faces of geometry and merge vertices

Lessons

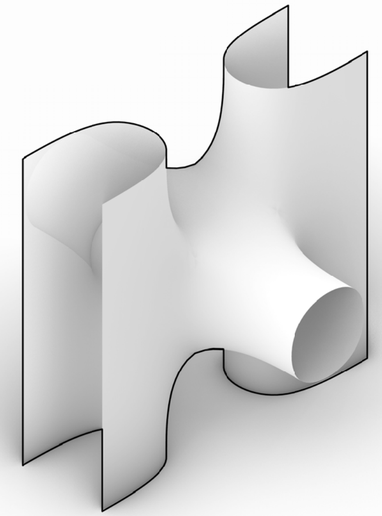
- Altering the points of specific vertices will reduce the occurence of sharp ends when taken into Alice.
- Ensure geometry is smoothened again when exported from Alice to Maya.



Coded Aggregation



Manual Aggregation



Voxel Topology

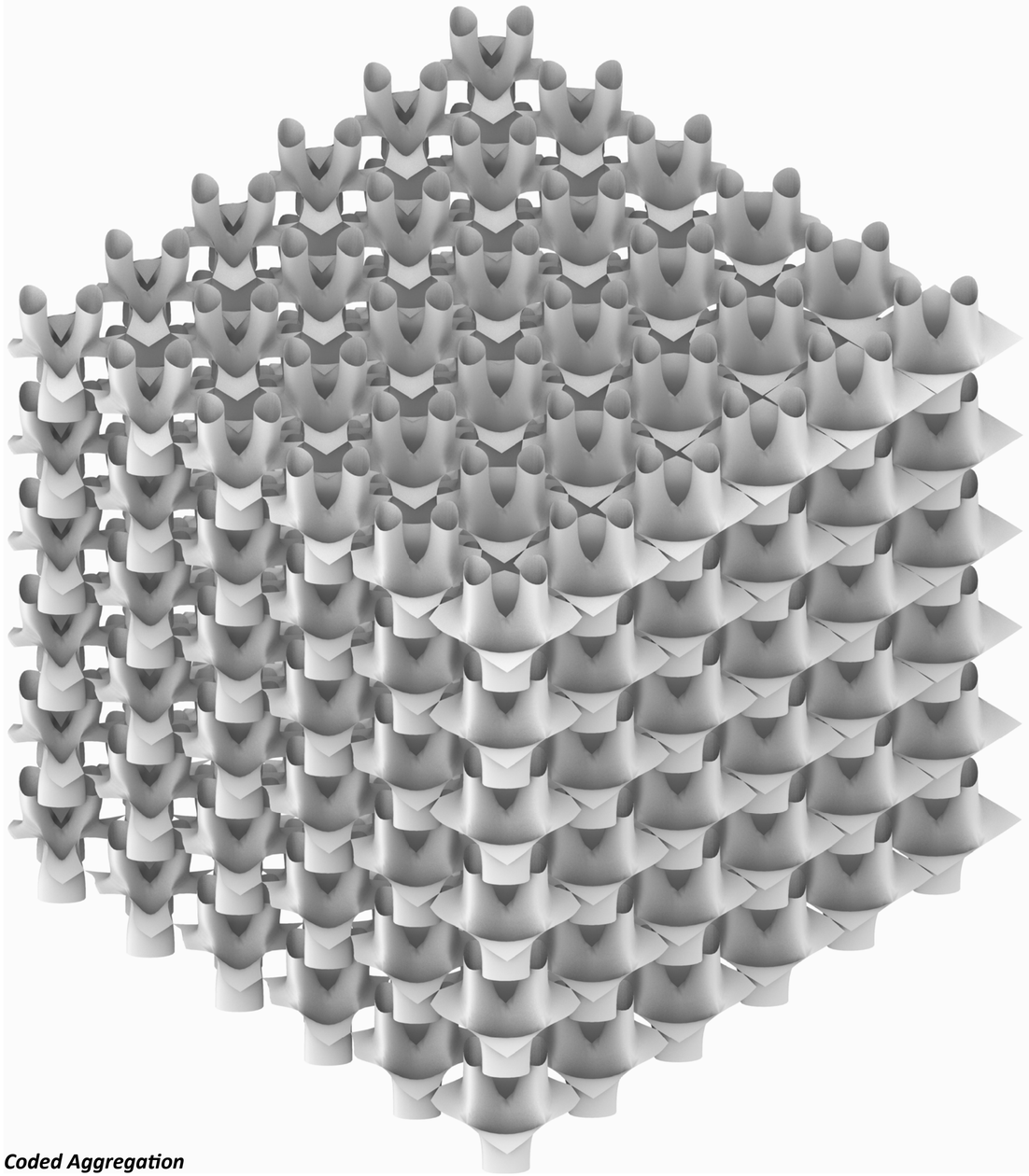
Failed Coded (C++) Aggregations 4

Intension / Process

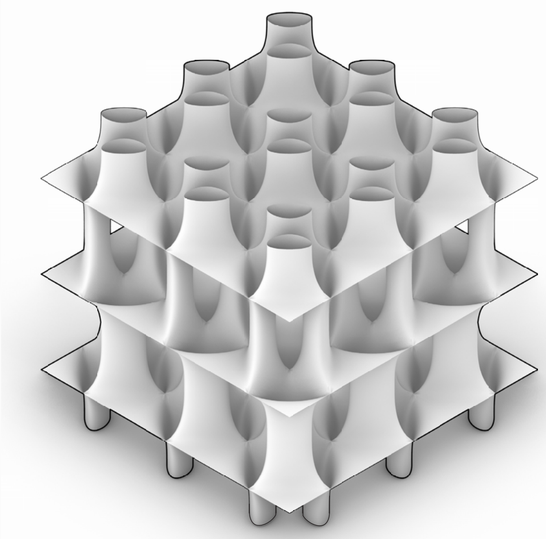
- Geometry developed from a plane extruded in the positive Z and negative Z direction.

Lessons

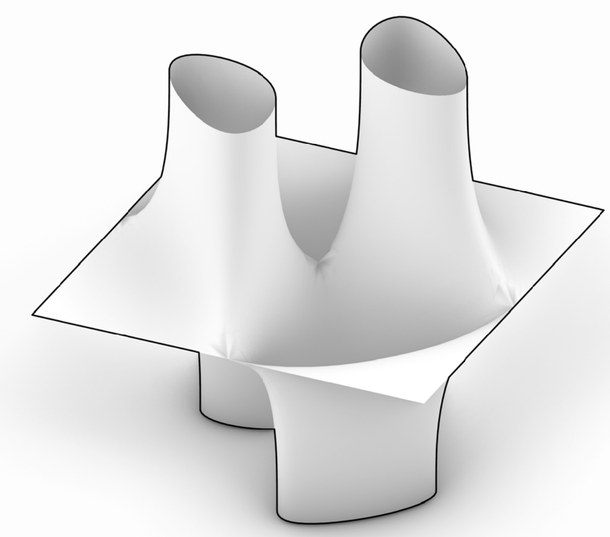
- The geometry points did not mirror correctly when the aggregate was simulated in Alice.
- This supposes that as seen in the manual aggregation done, a flipping command would have been needed to transform each alternate line.



Coded Aggregation



Manual Aggregation



Voxel Topology

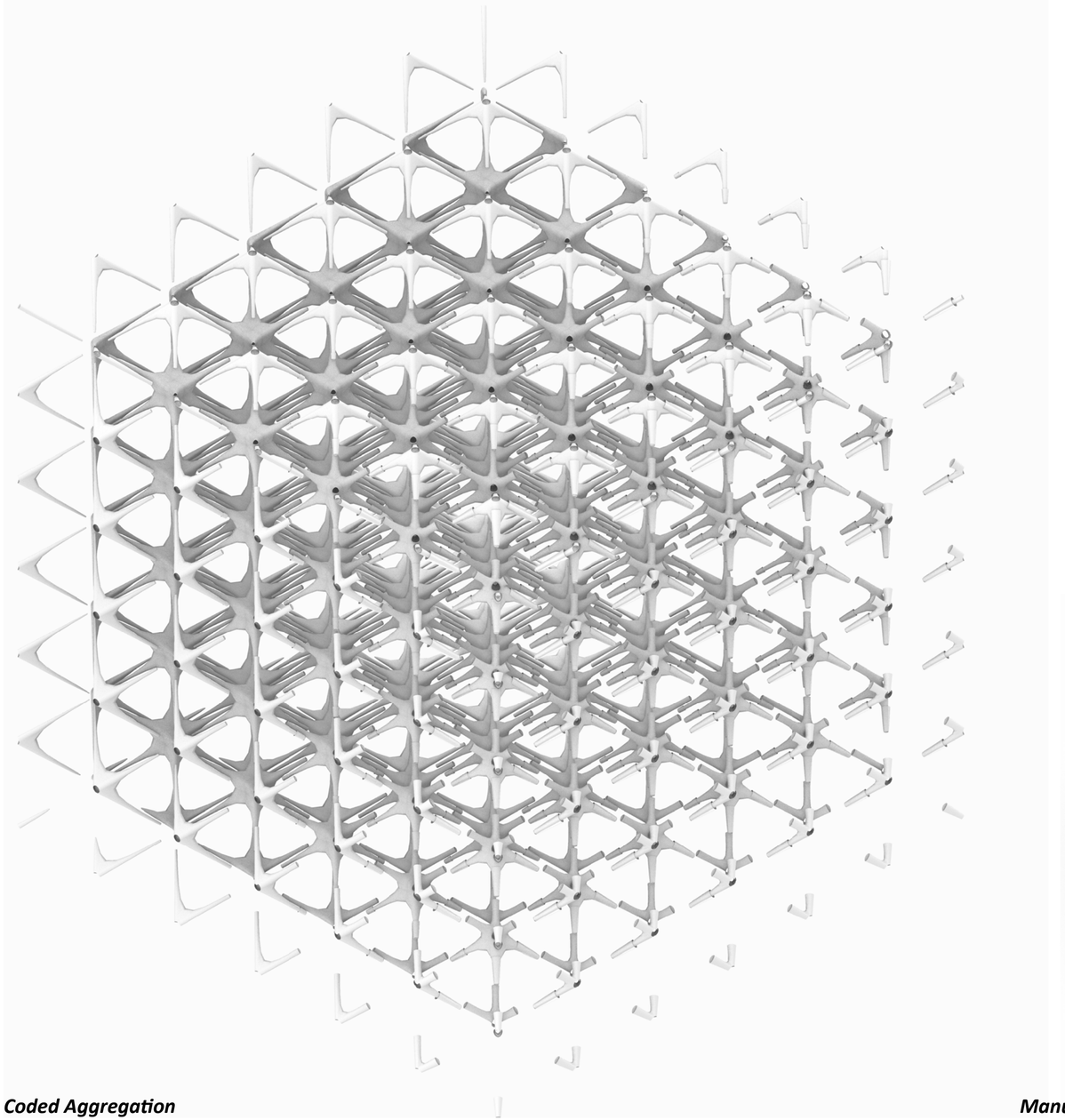
Failed Coded (C++) Aggregations 5

Intension / Process

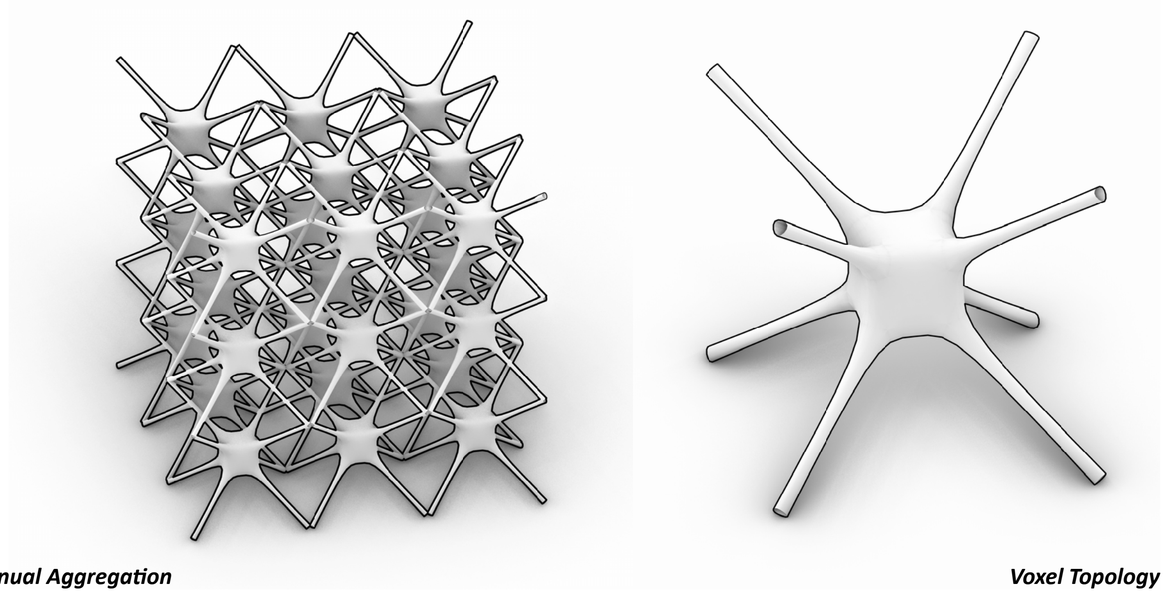
- Creating complexity form basic rules
- Developing a functinally graded material by removing material where it's not needed.

Lessons

- Resolve angular connections to make a strong connection between elements.
- The geometry was modeled as a quad mesh.



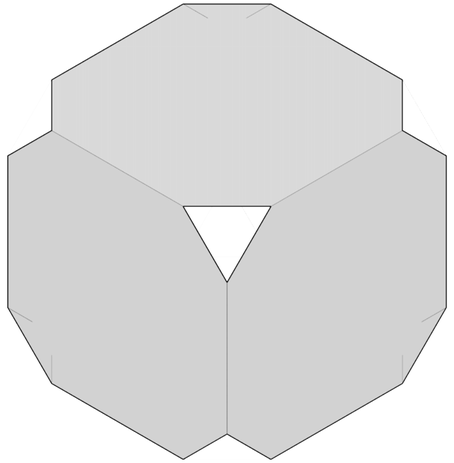
Coded Aggregation



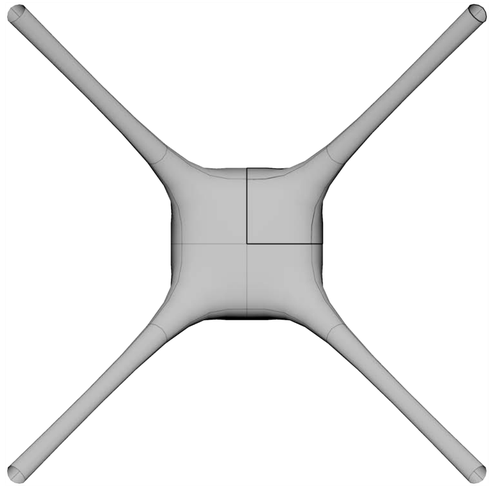
Manual Aggregation

Voxel Topology

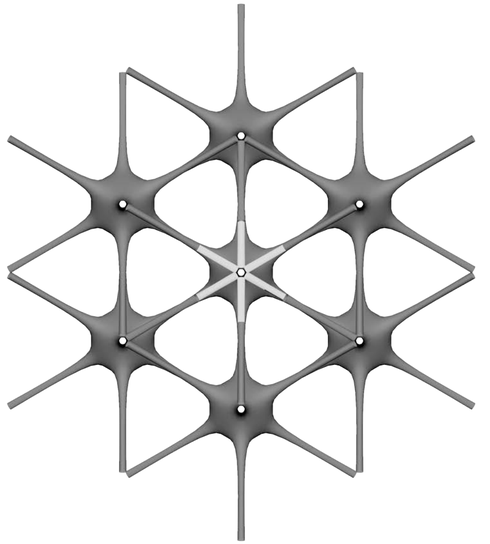
Connection Resolution



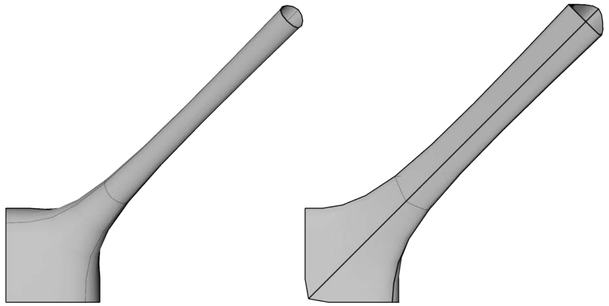
3 sided connection nodes



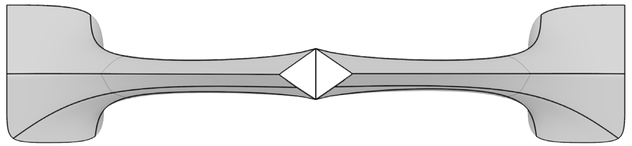
3 sided failed connection



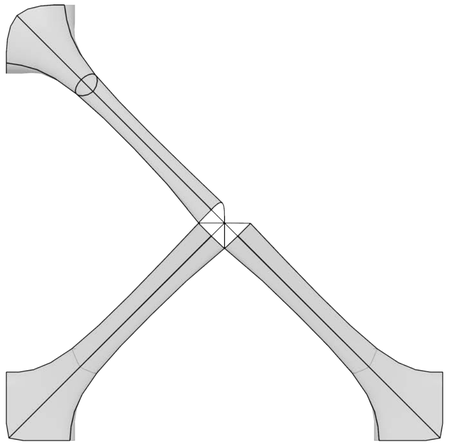
Add an extra edge to nodes



Conneciton on Y direction



Connection on Z direction



Nodal combination check before aggregating final voxel topology

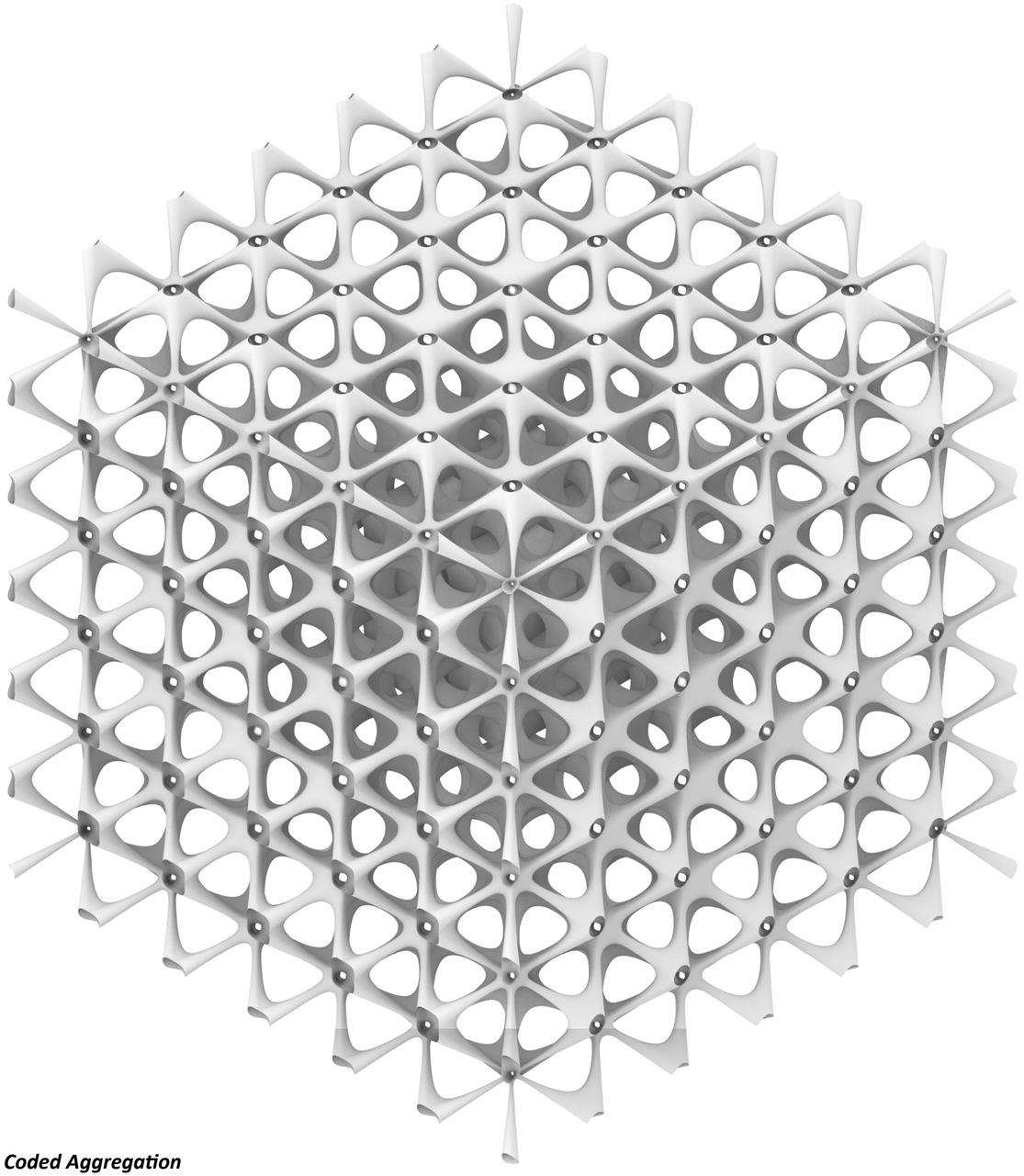
Coded (C++) Aggregations _1

Intensions / Process

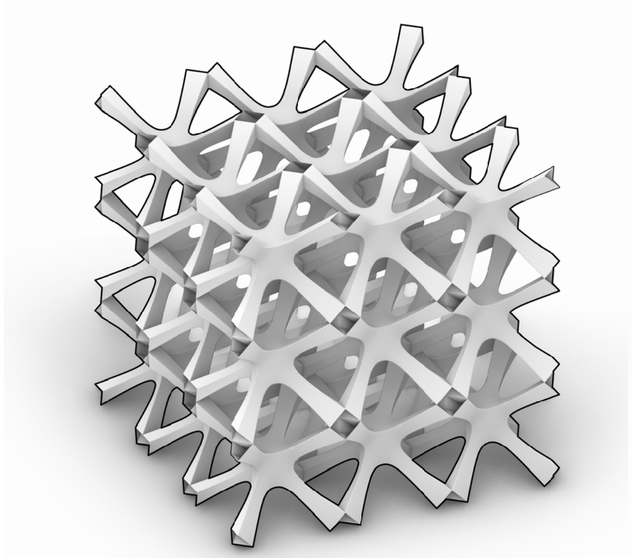
- Creating complexity from basic rules
- Geometry connected at the edge of the voxels

Lessons

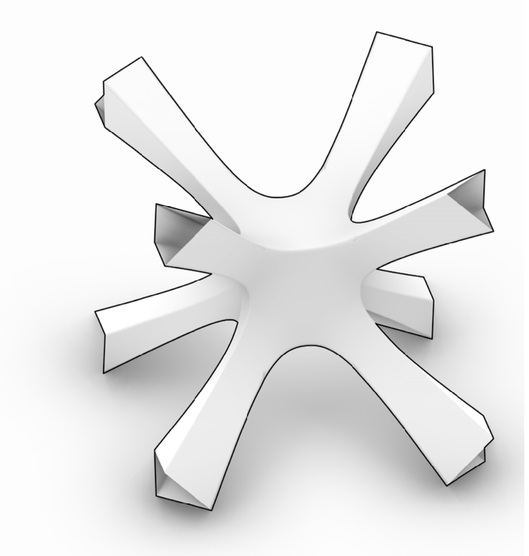
-Resolve angular connection to make a strong connection between elements.
-The geometry was modeled as a quad mesh.



Coded Aggregation



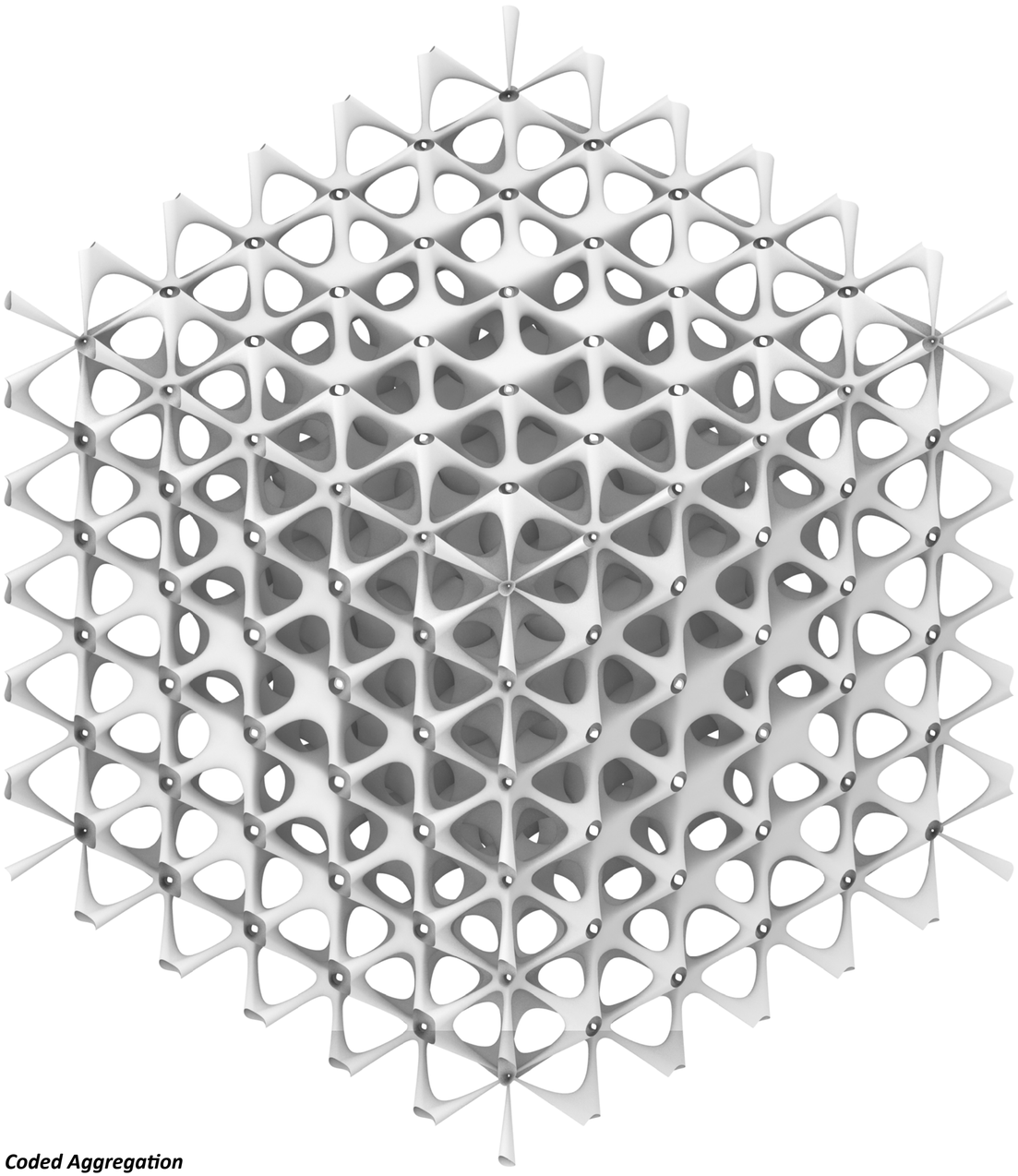
Manual Aggregation



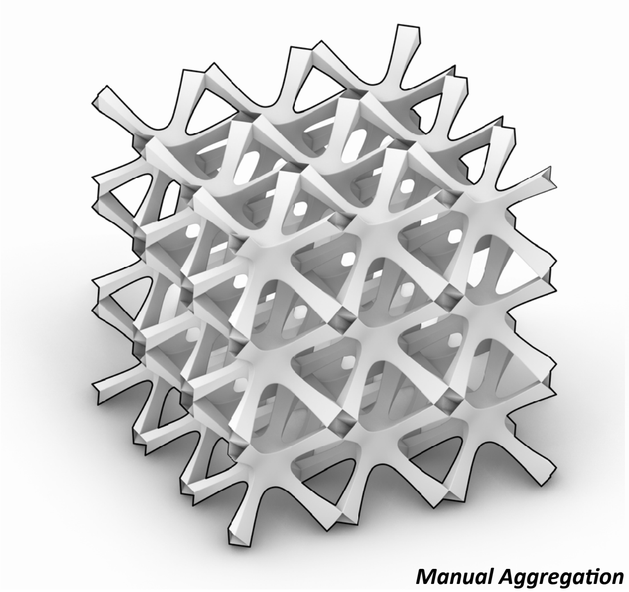
Voxel Topology

Coded (C++) Aggregations 2

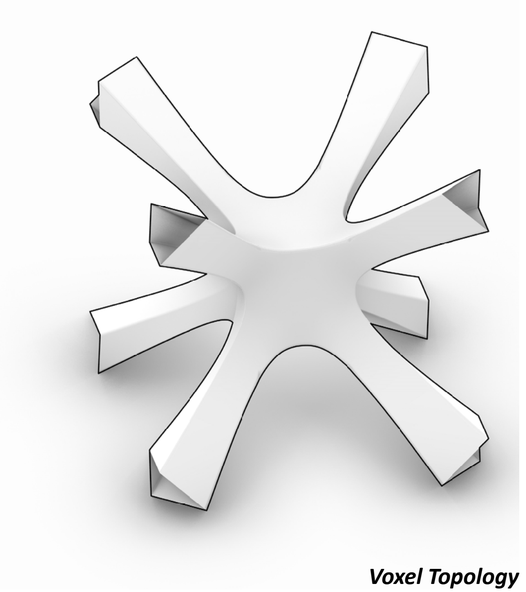
Preferred Iteration



Coded Aggregation

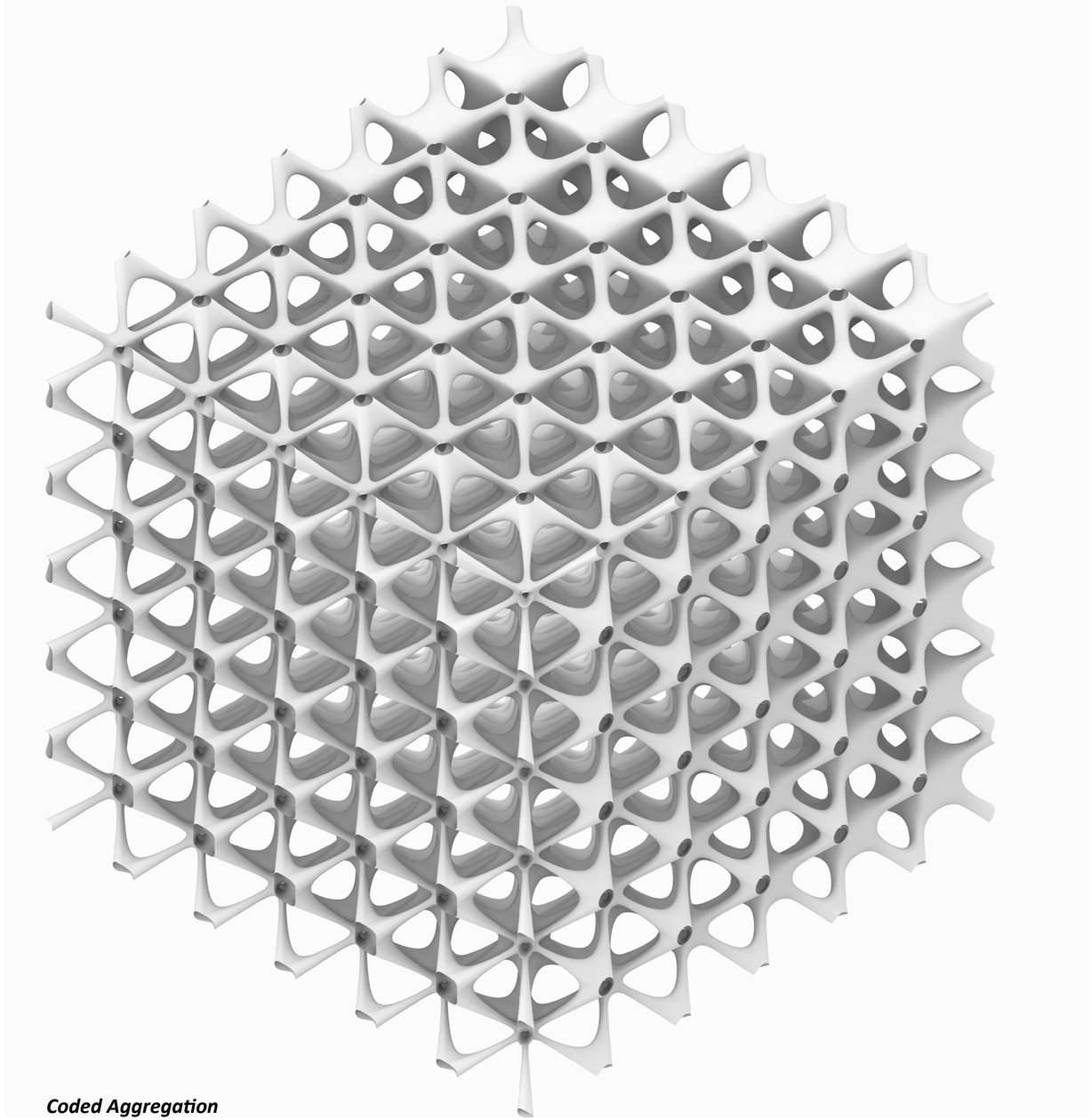


Manual Aggregation

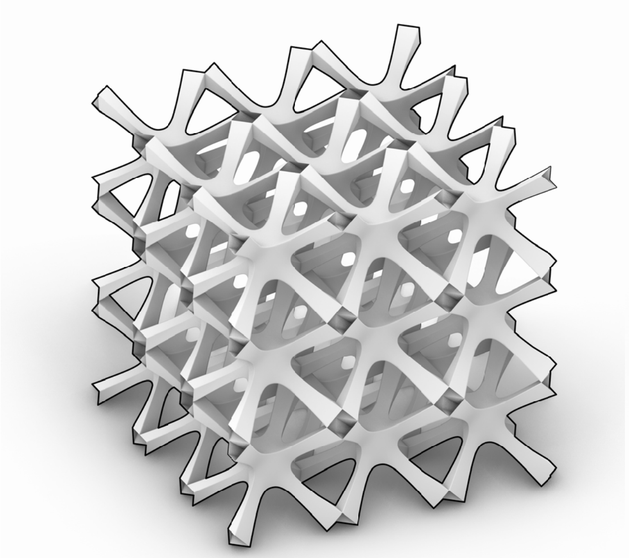


Voxel Topology

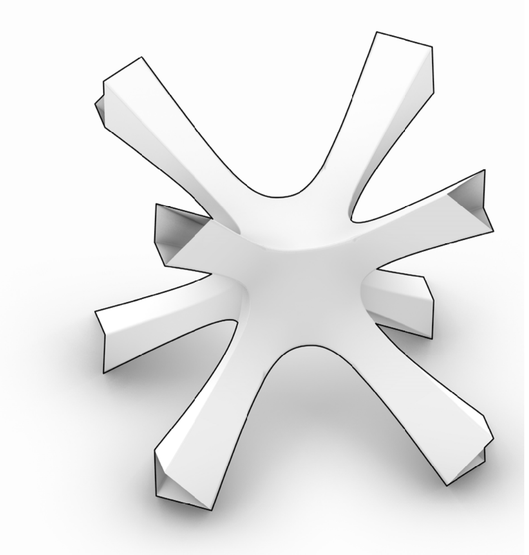
Coded (C++) Aggregations 3 _ NE



Coded Aggregation

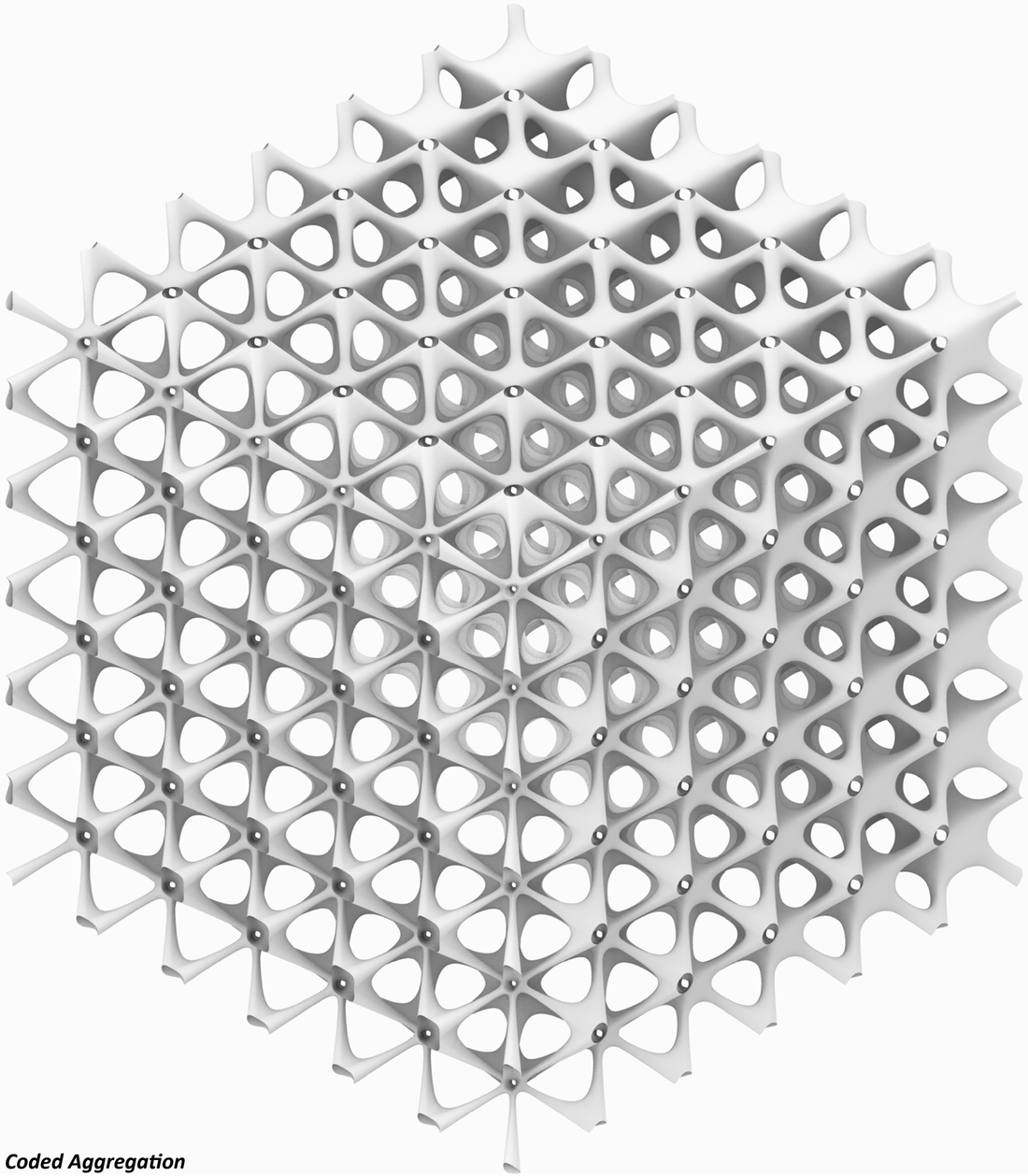


Manual Aggregation

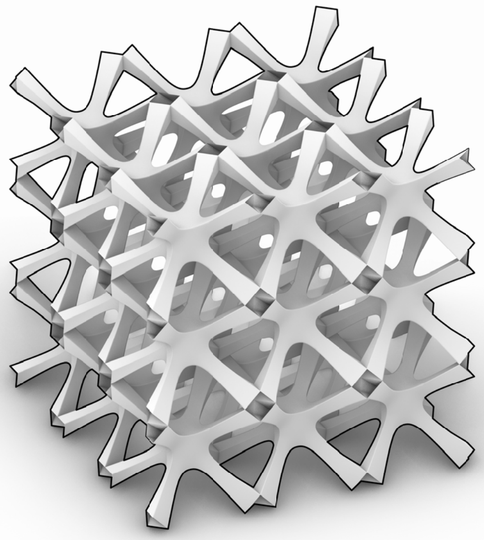


Voxel Topology

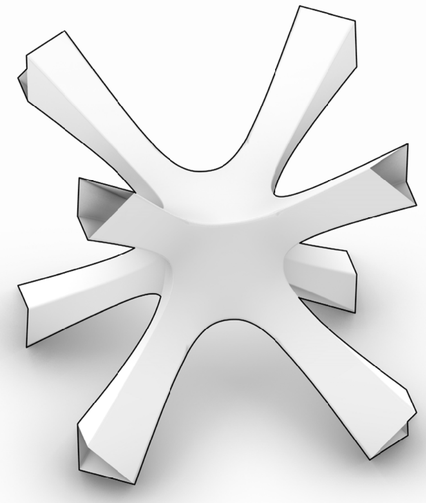
Coded (C++) Aggregations 3_SW



Coded Aggregation

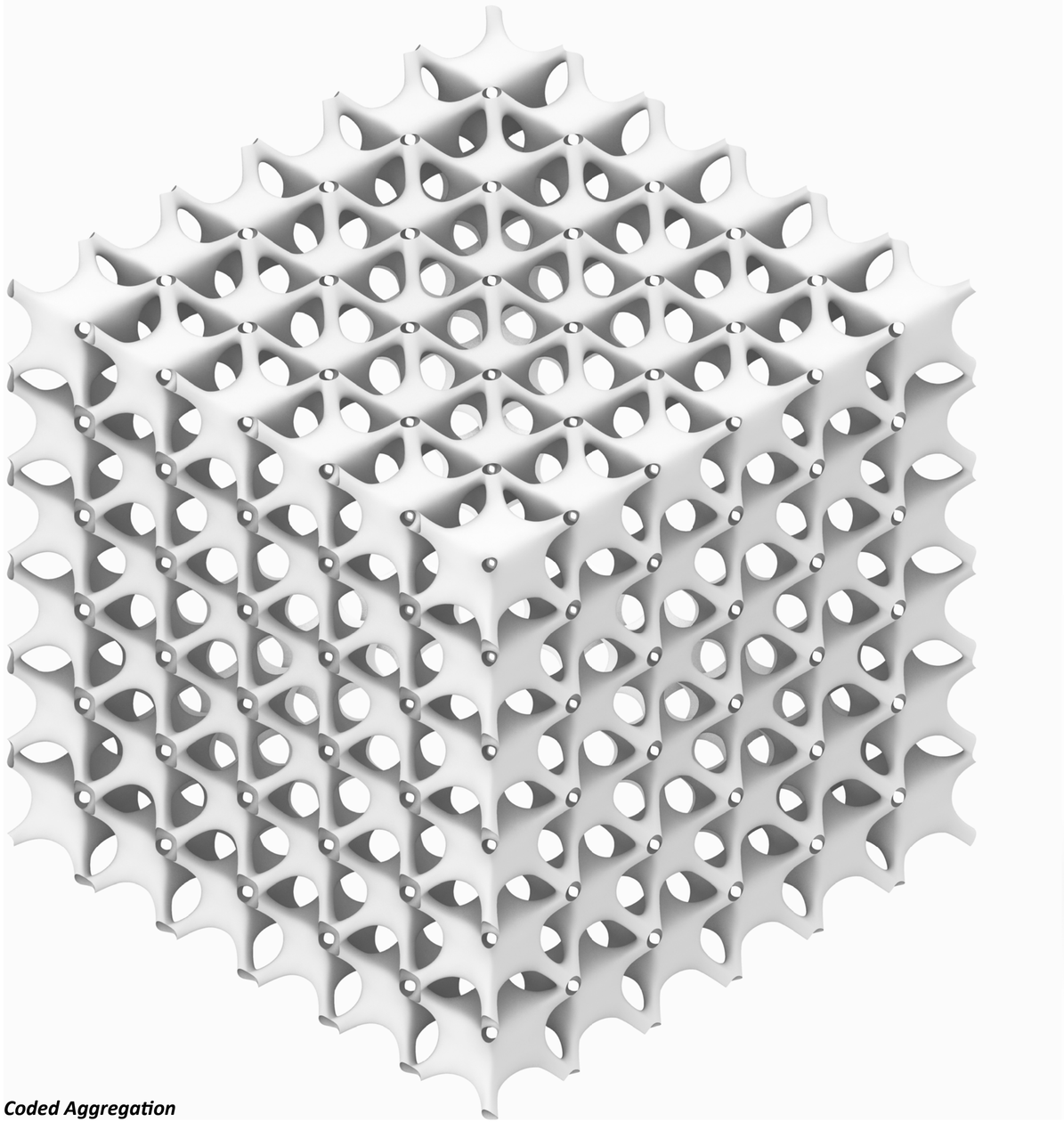


Manual Aggregation

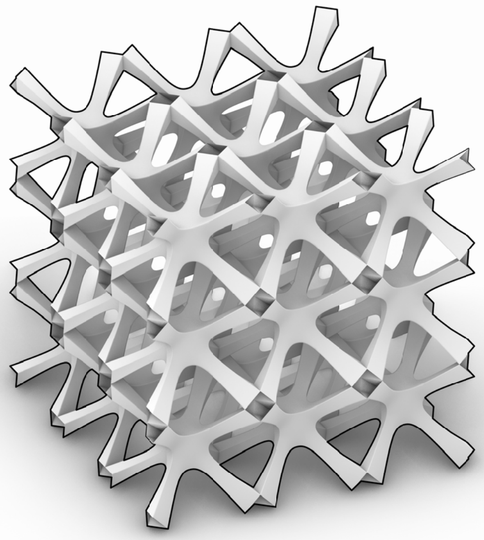


Voxel Topology

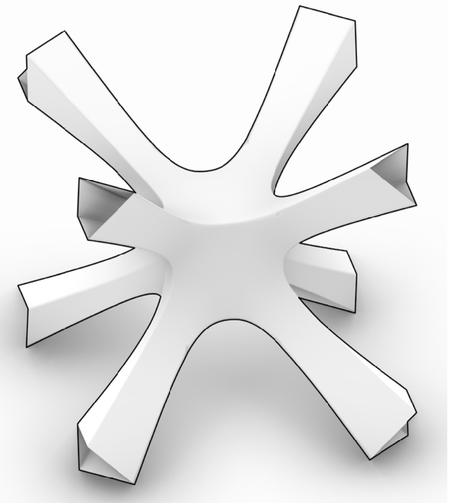
Coded (C++) Aggregations 4



Coded Aggregation

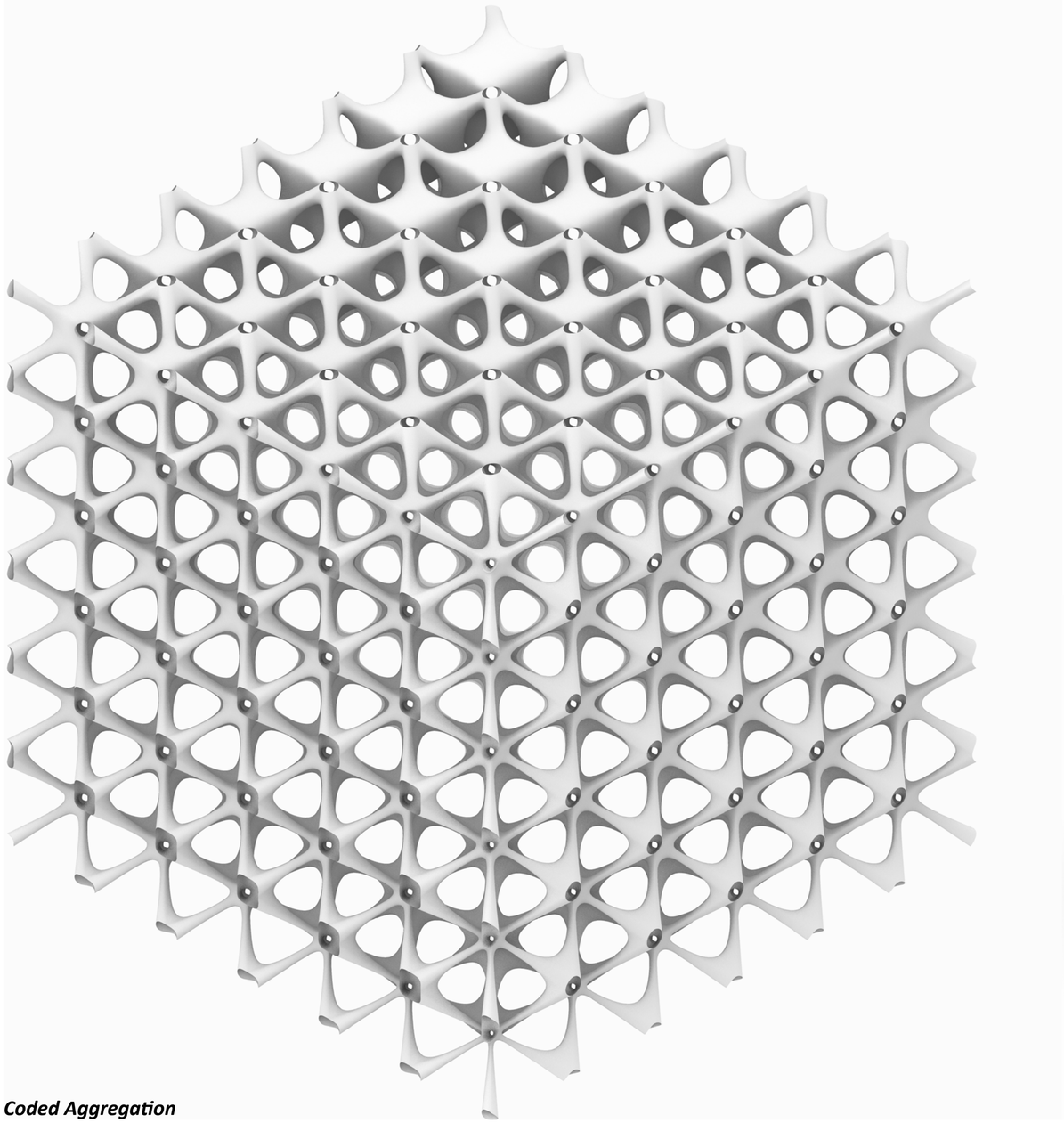


Manual Aggregation

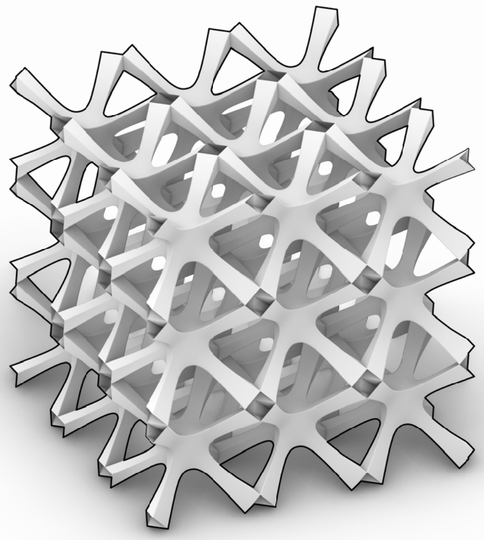


Voxel Topology

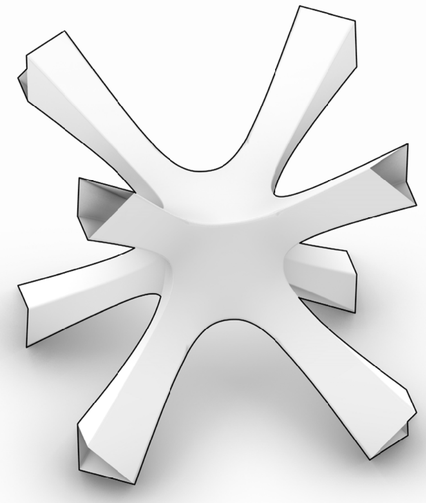
Coded (C++) Aggregations 5



Coded Aggregation

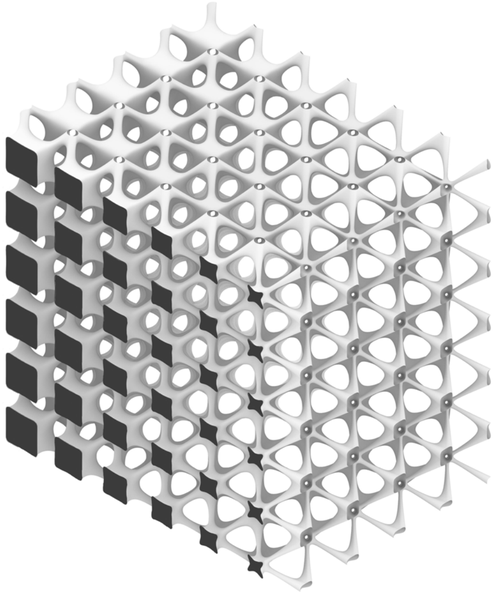


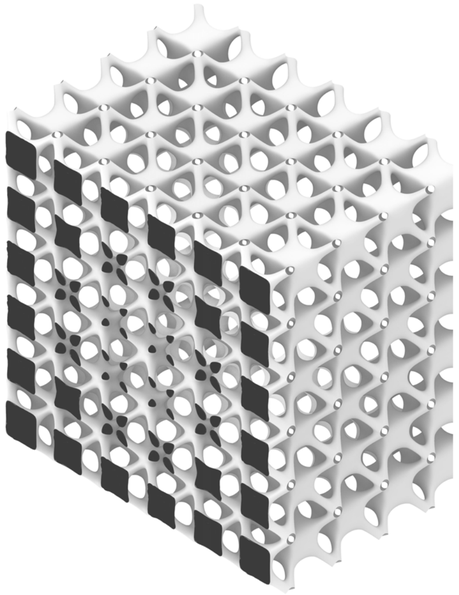
Manual Aggregation

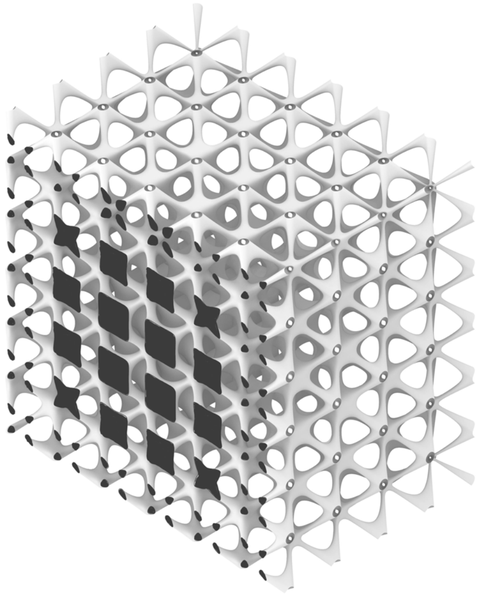


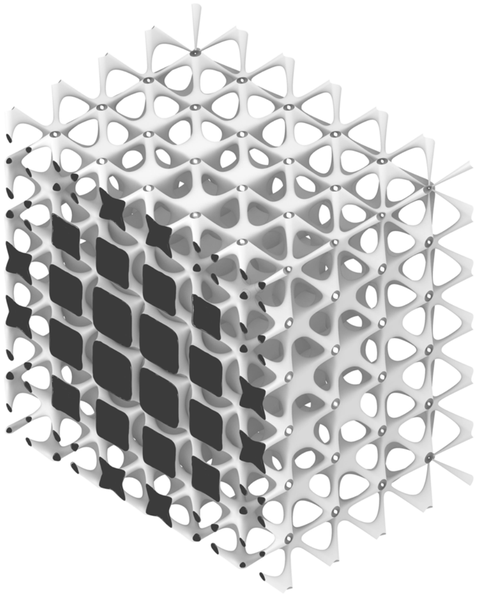
Voxel Topology

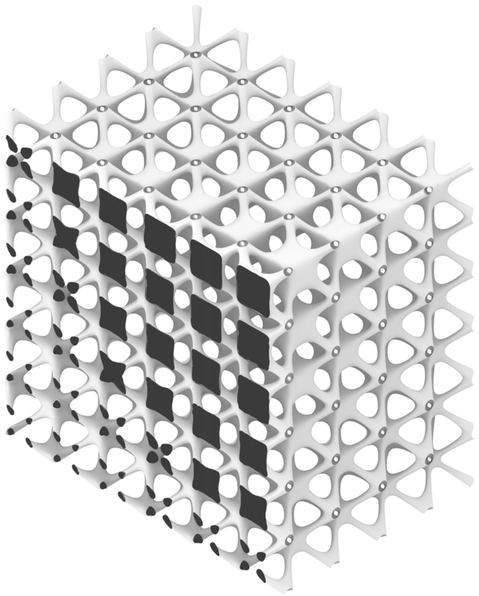
Sectional Qualities



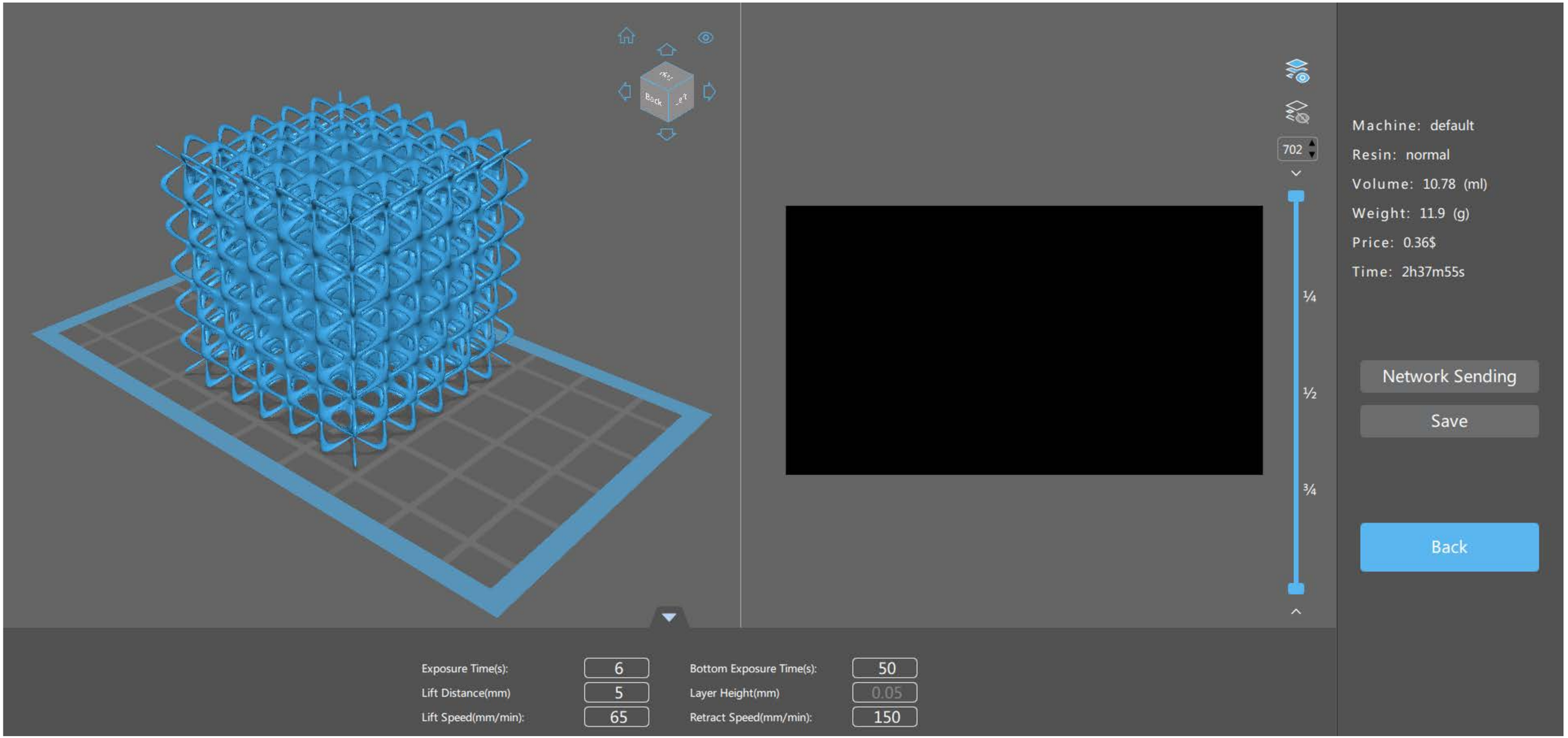




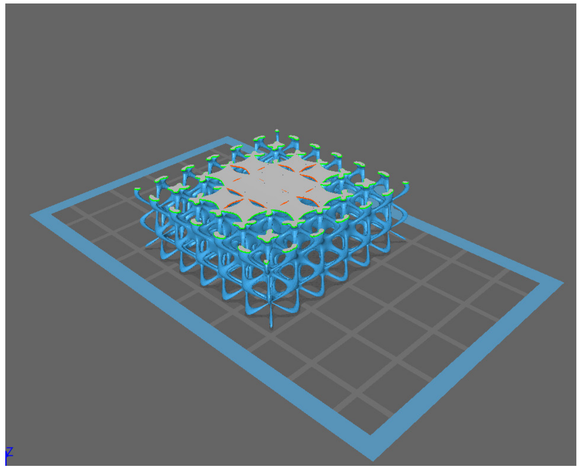




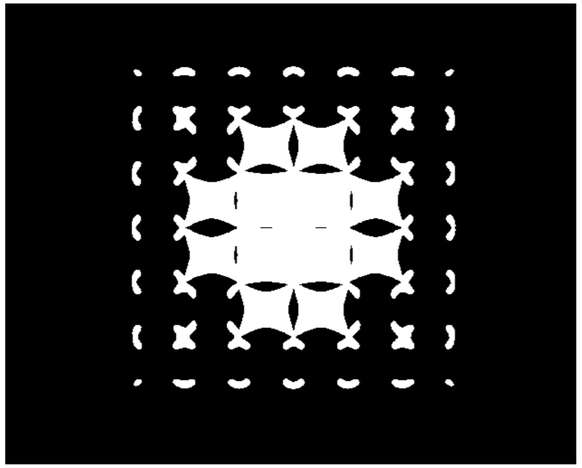
Final 3d printing page indicating print time and other information



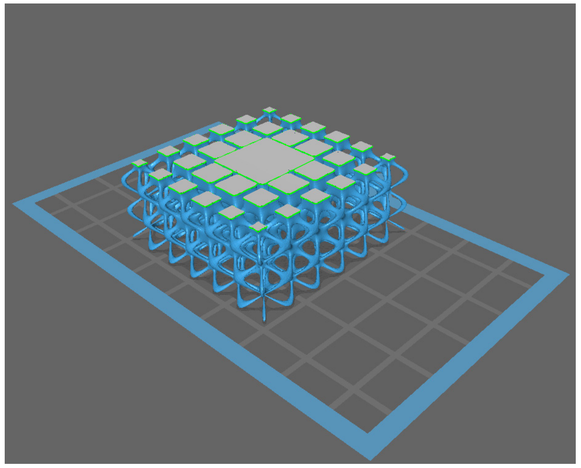
3D Printing Logistics



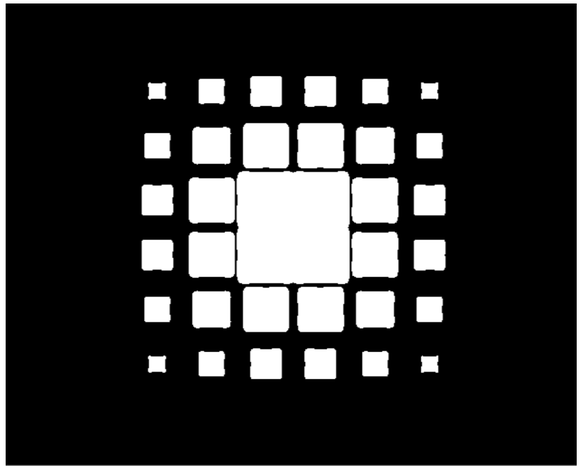
250 Model



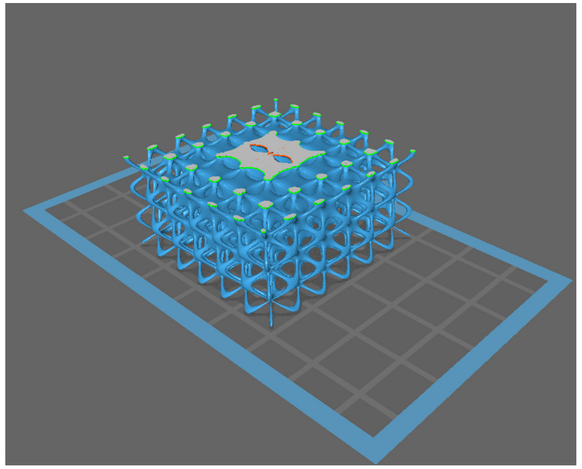
250 Sliced



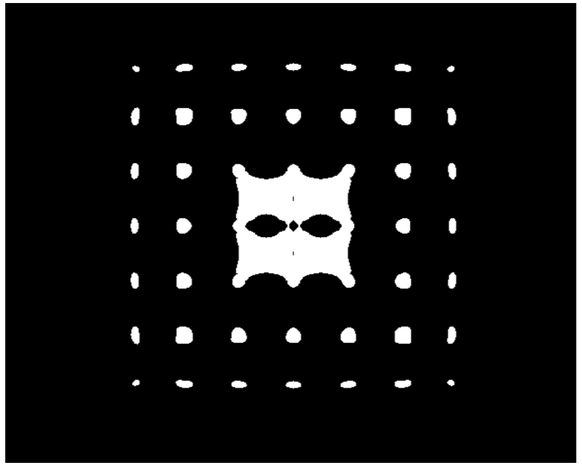
300 Model



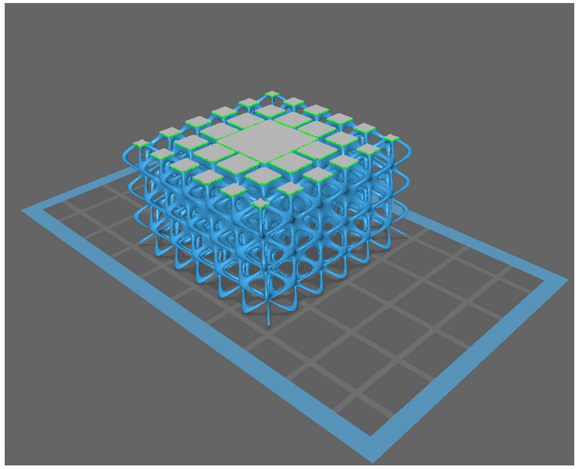
300 Sliced



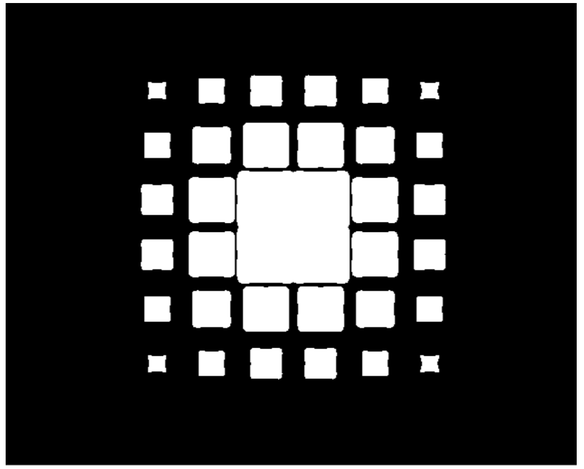
350 Model



350 Sliced

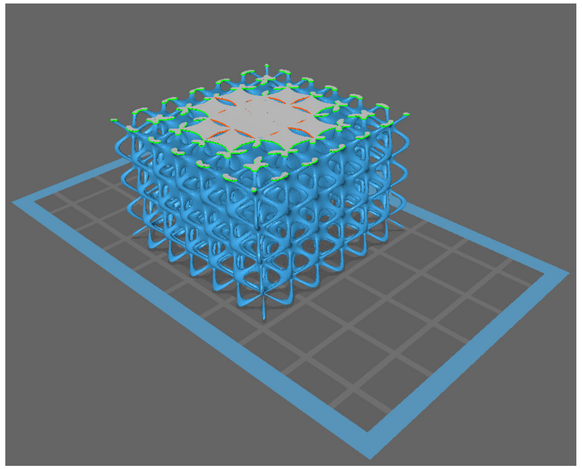


400 Model

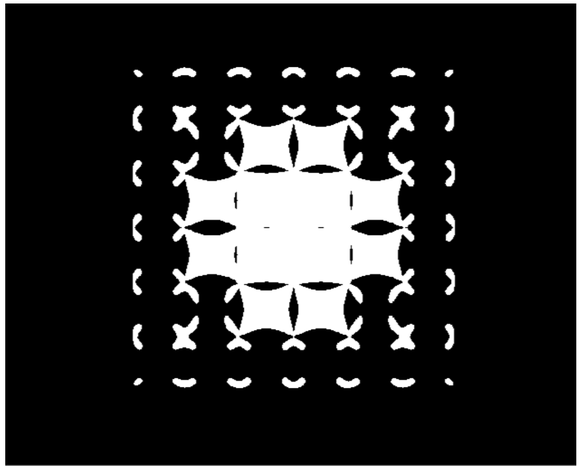


400 Sliced

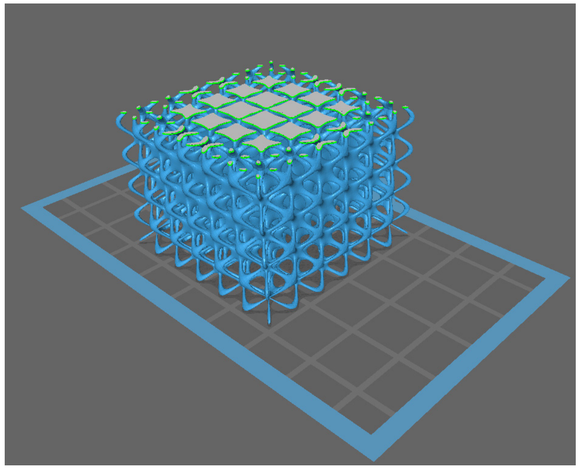
3D Printing Logistics



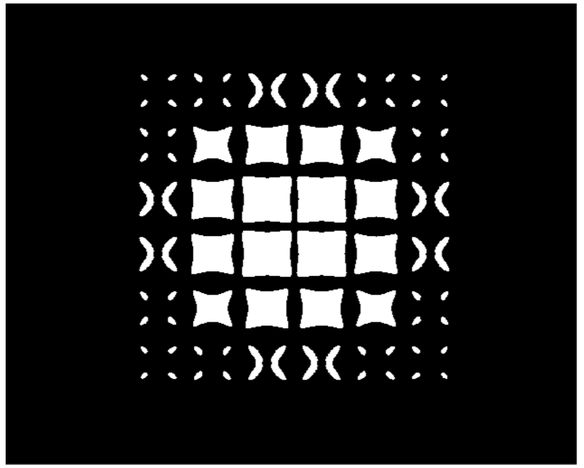
450 Model



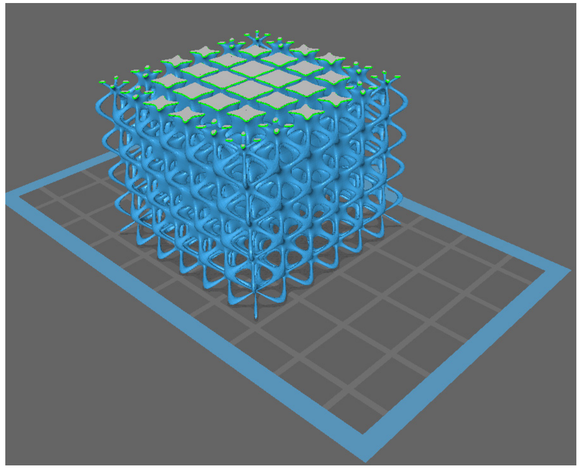
450 Sliced



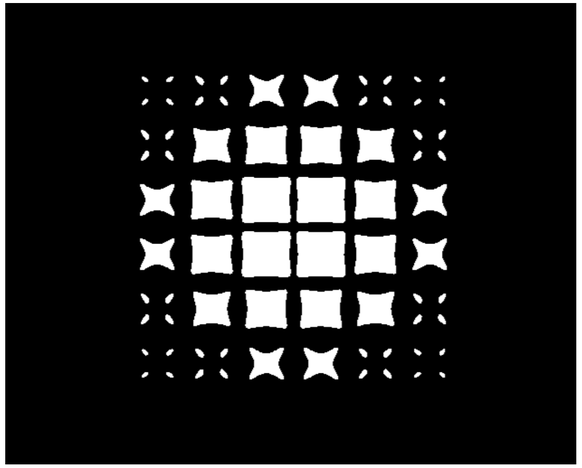
500 Model



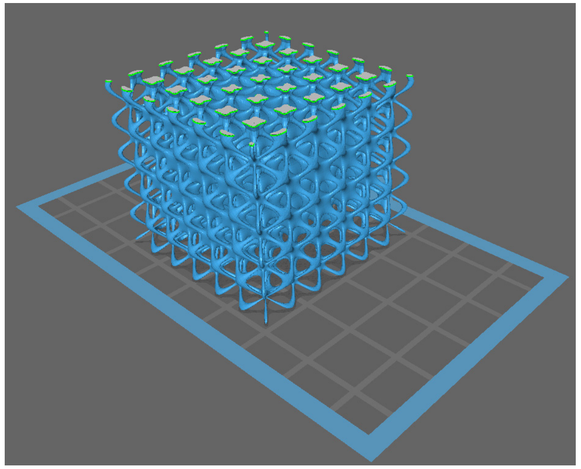
500 Sliced



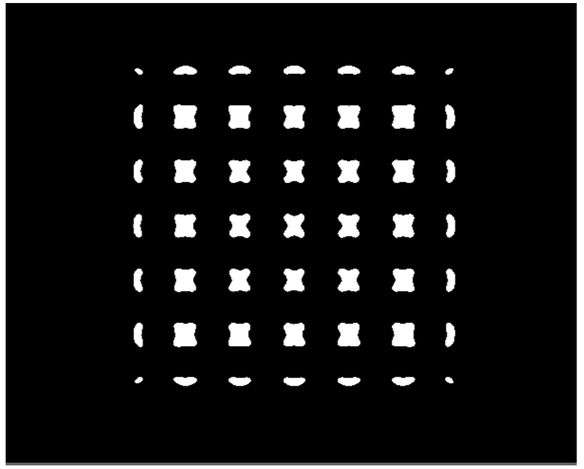
550 Model



550 Sliced

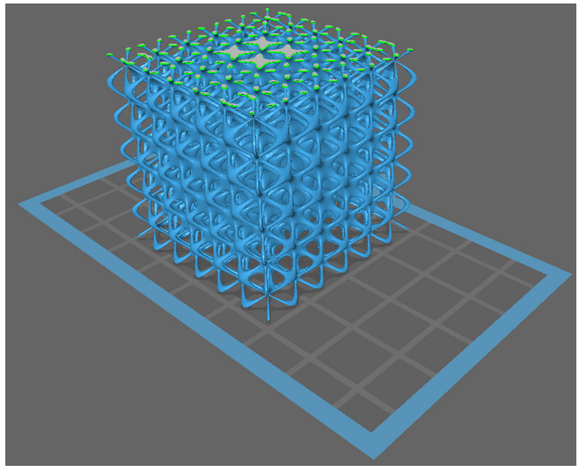


600 Model

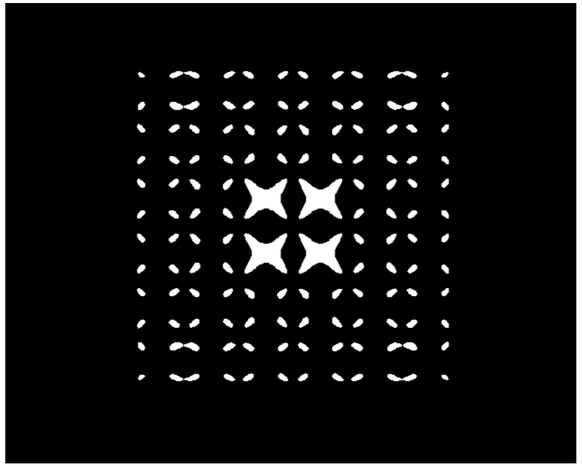


600 Sliced

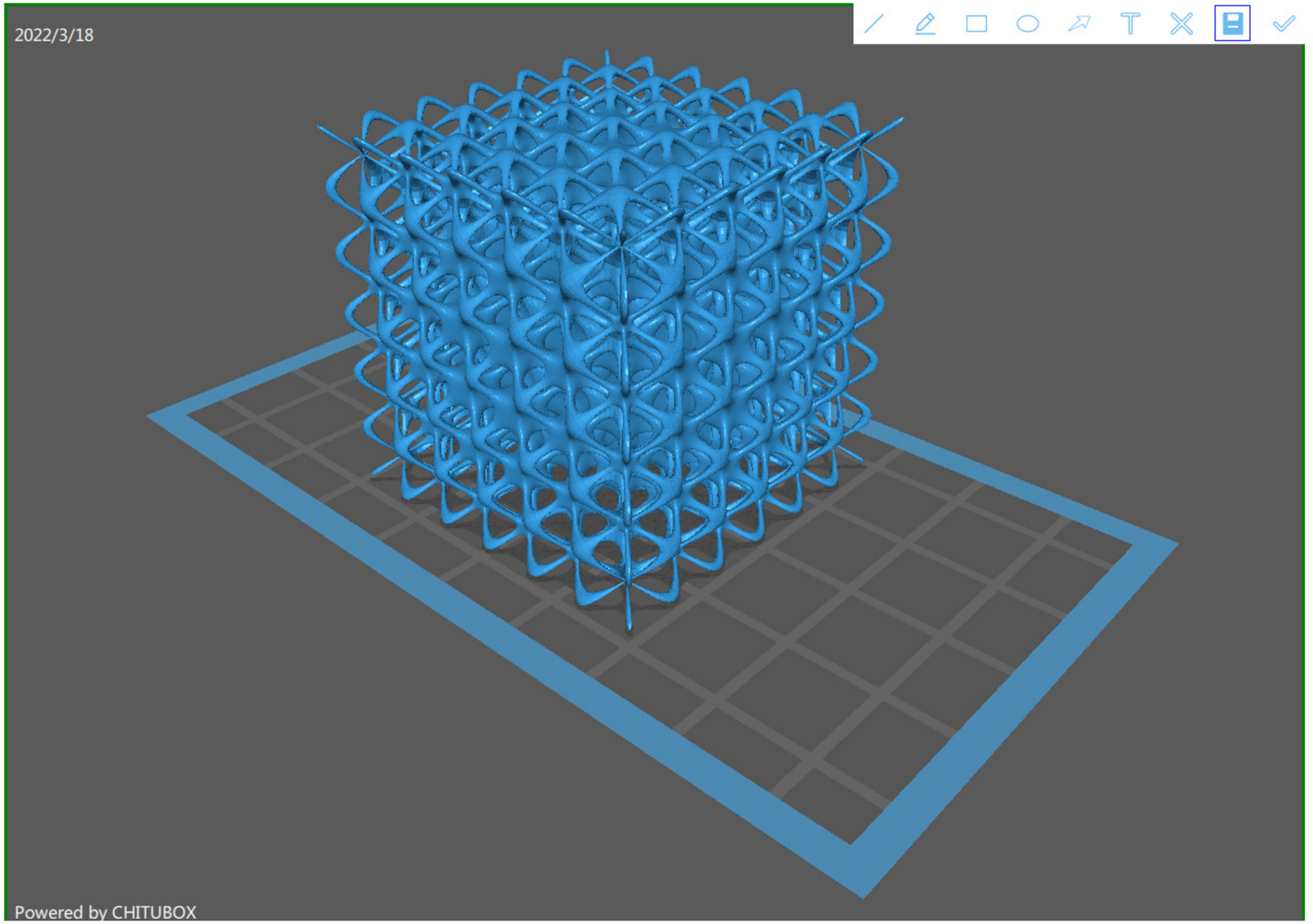
3D Printing Logistics



675 Model



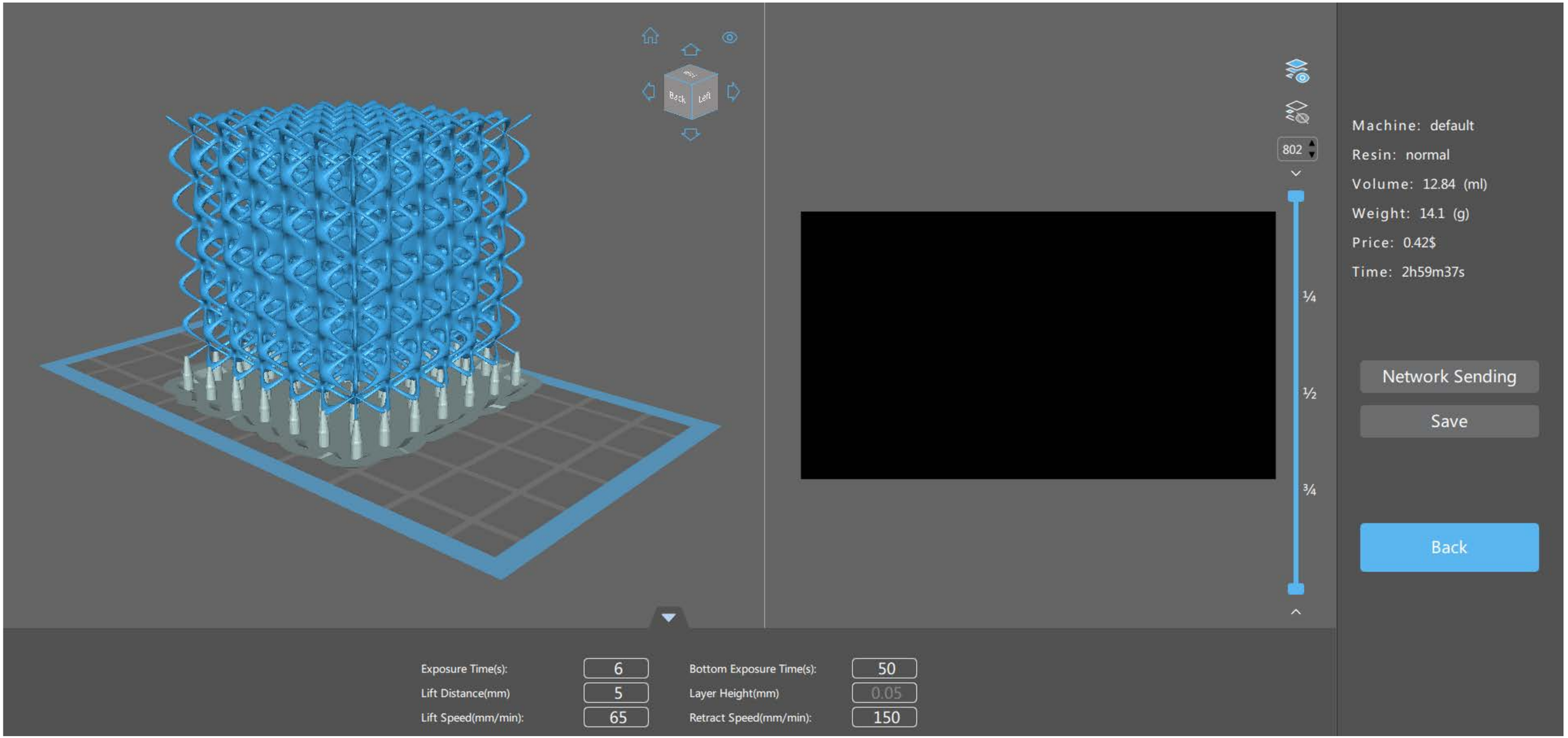
675 Sliced



Final Print Form

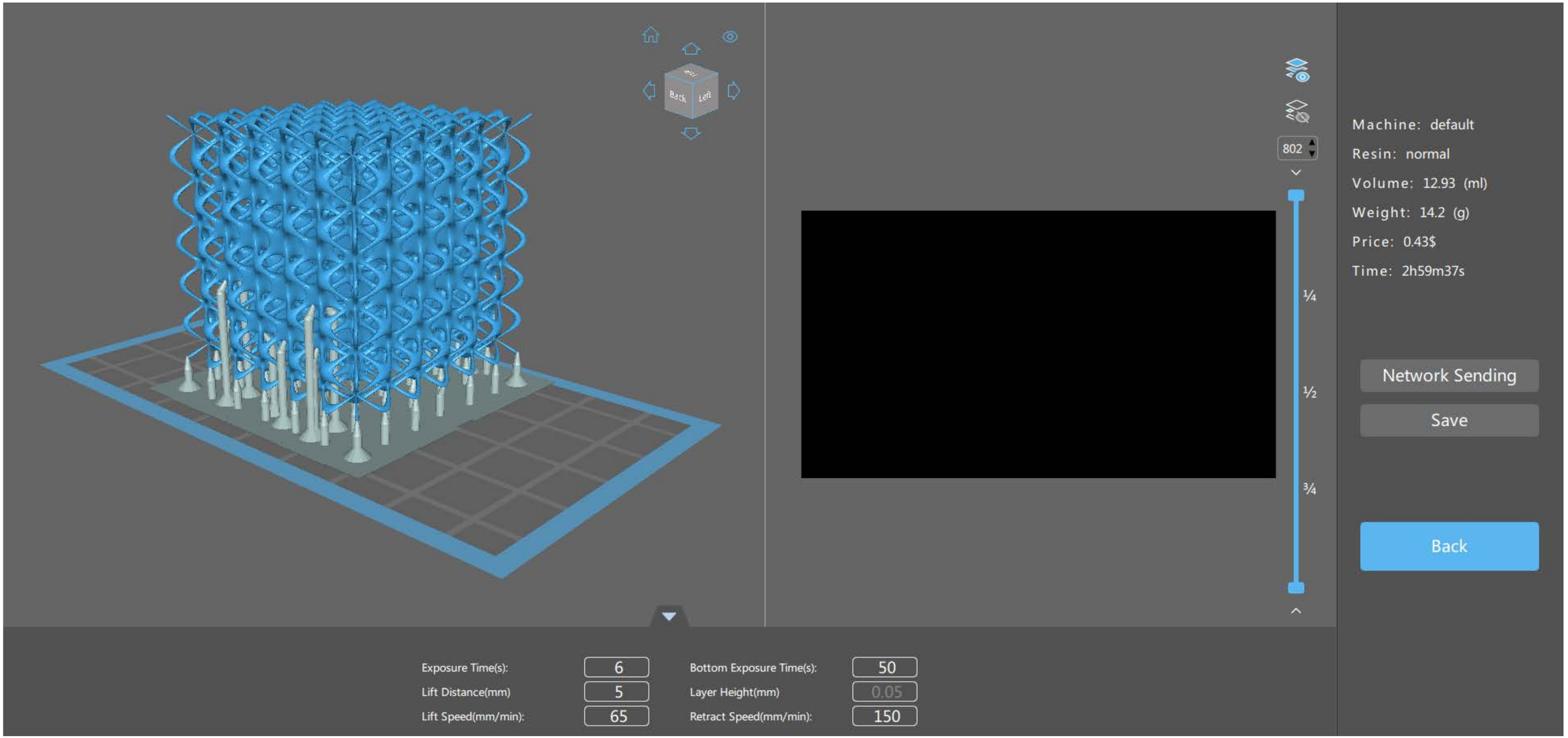
3D Printing Support Material Logistics

Heavy Support



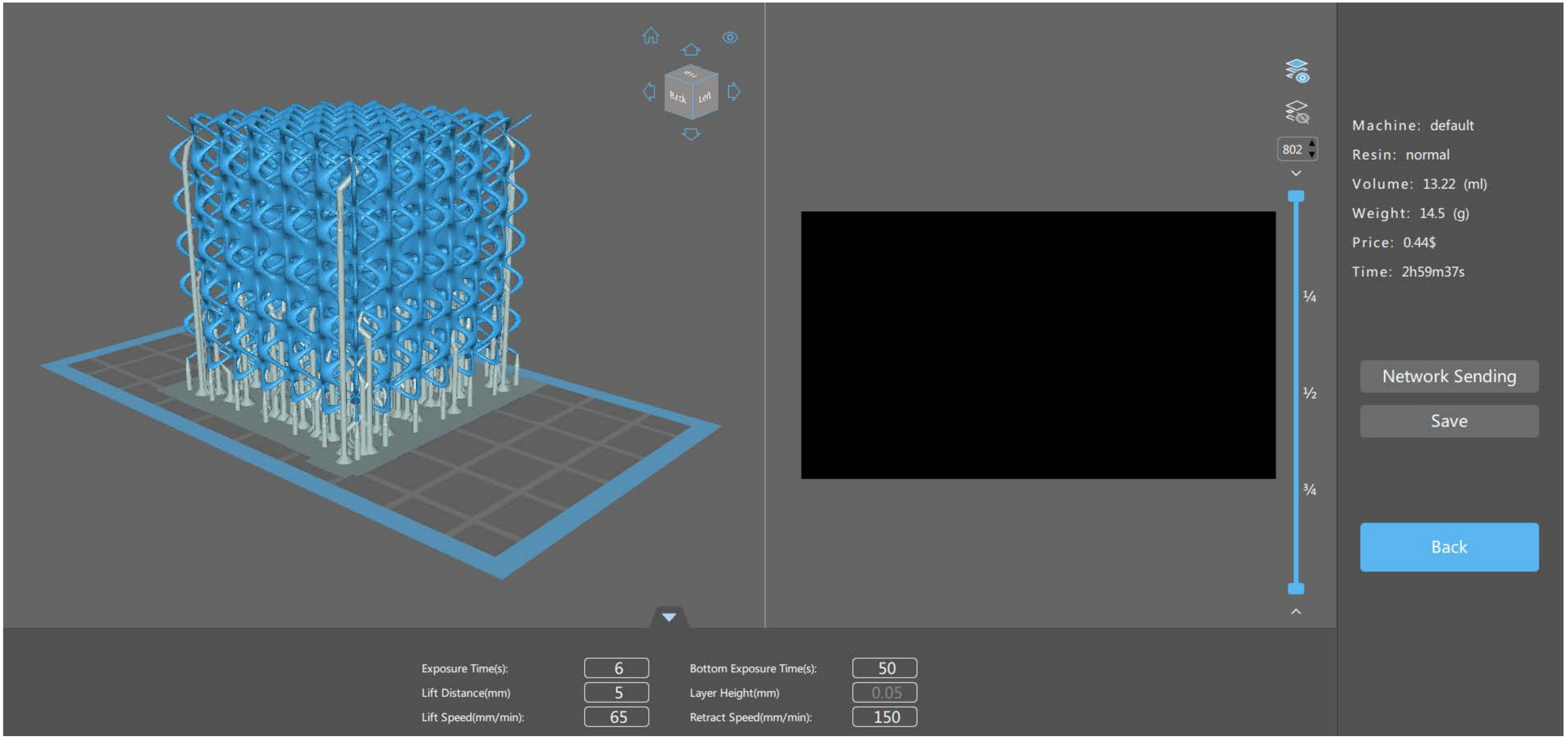
3D Printing Support Material Logistics

Medium Support

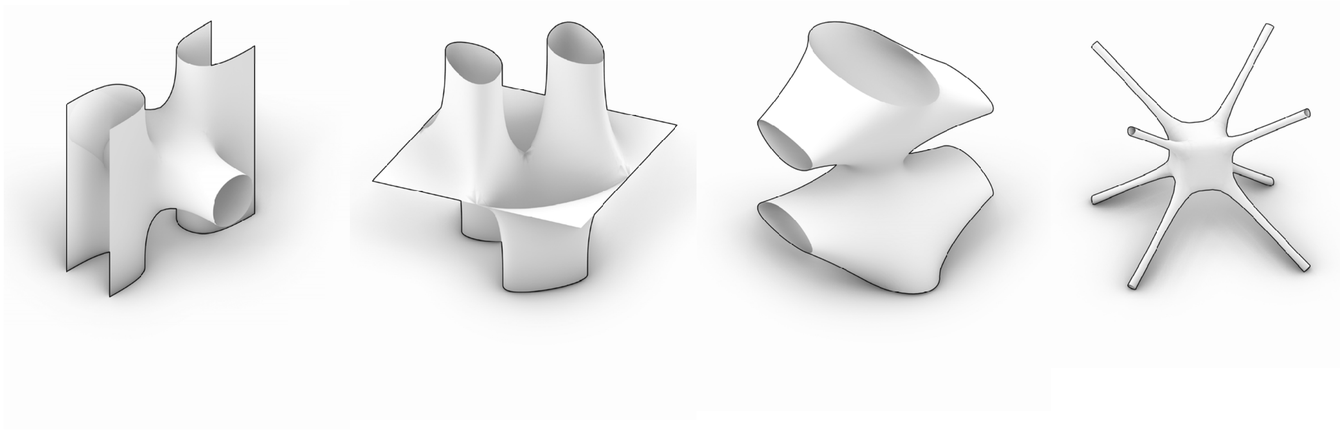


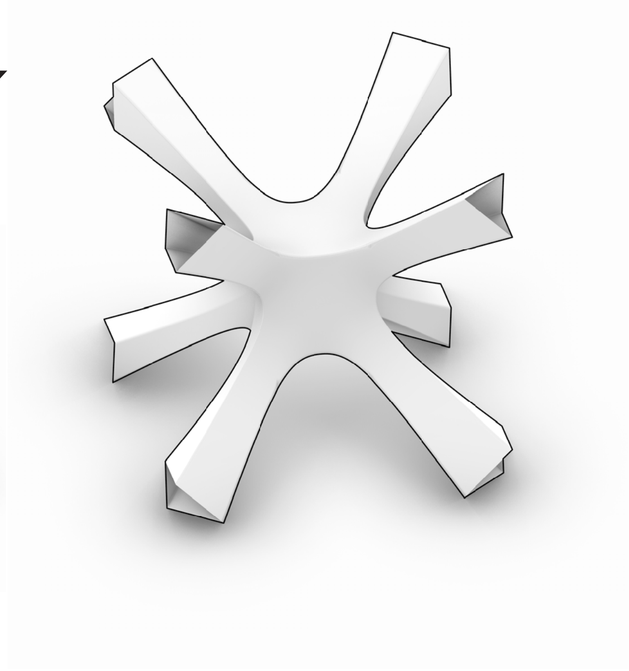
3D Printing Support Material Logistics

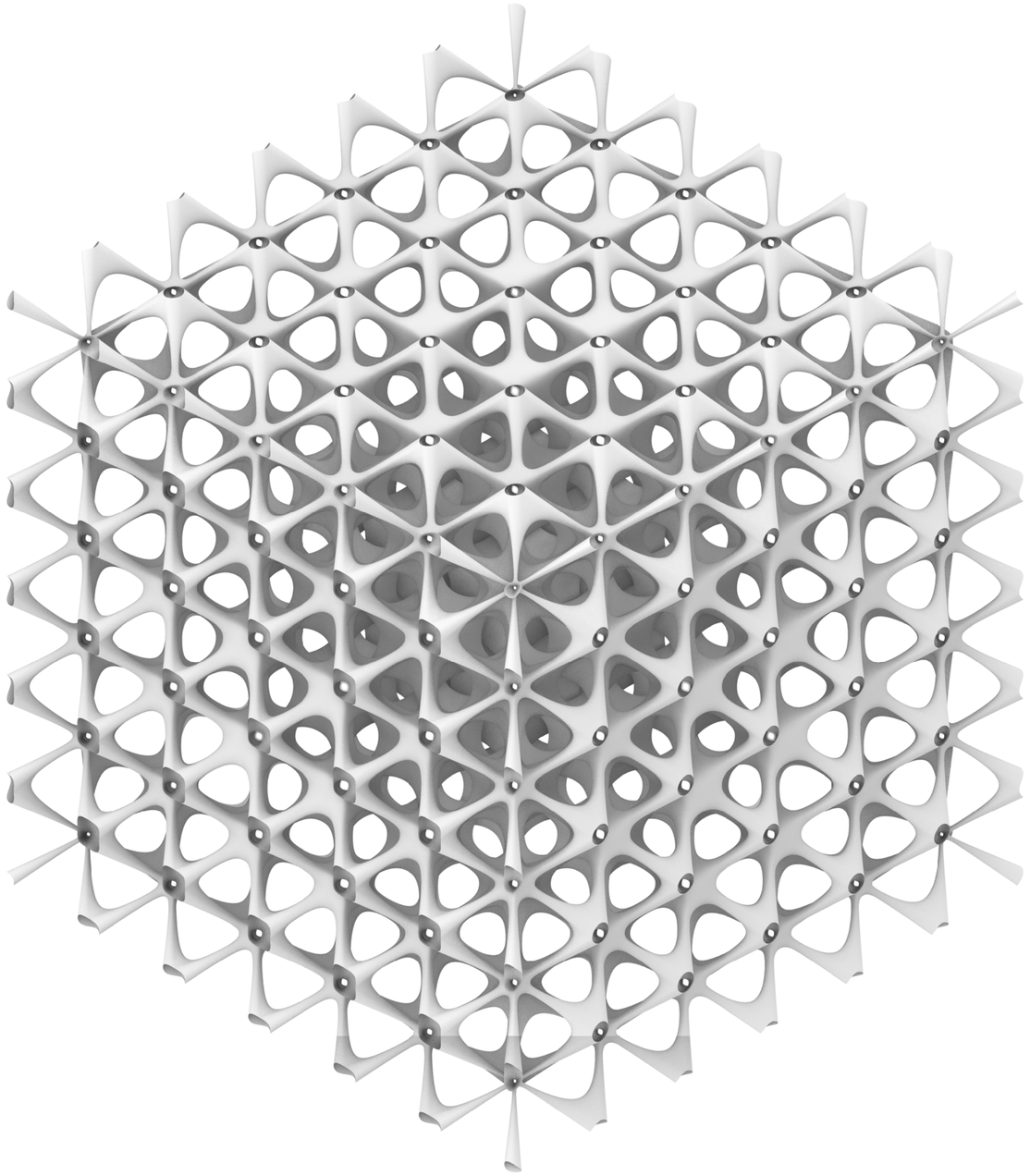
Light Support



Appendix



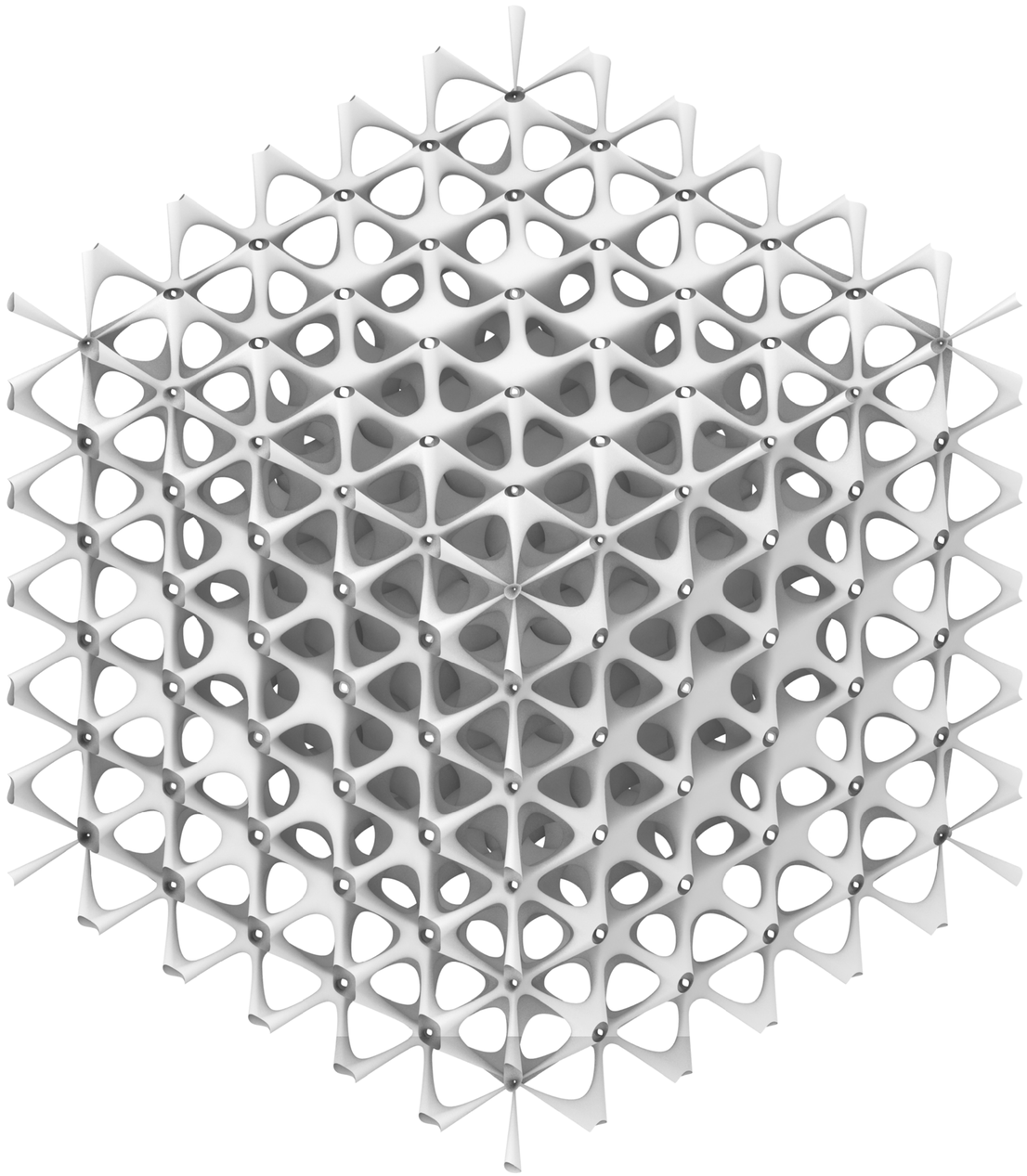




Iteration 1_Script Definition

Aggregating the form in a grid while creating a centre to outside gradient affecting the scaling of the central mesh vertices, slowly expanding the shape. Creating an aggregated form that is densest at the centre and becomes less dense as it expands.

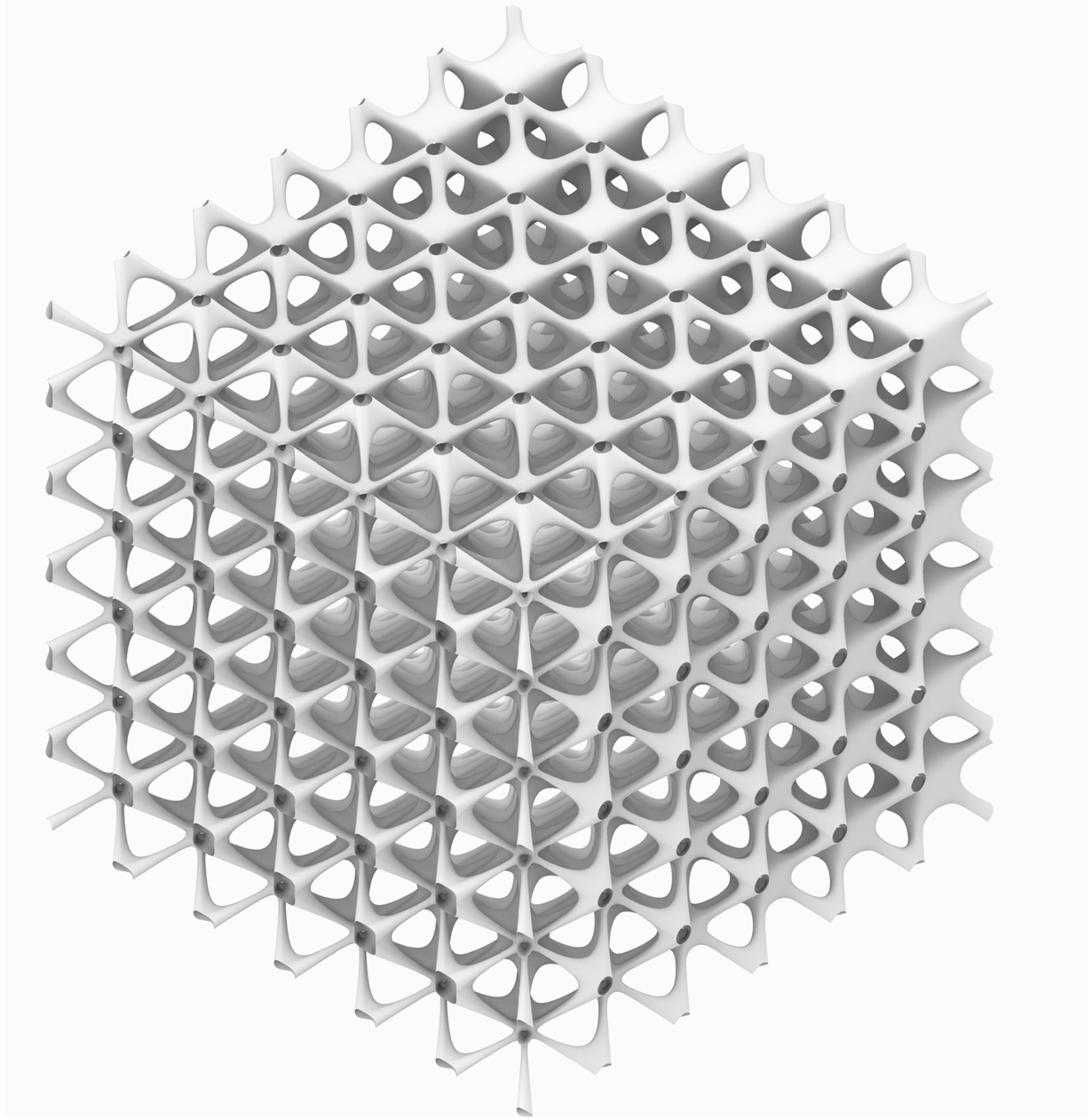
```
createMeshFromFile("data/Momie_Topology.obj");
            transformMesh(mat);

            //---------------------- trasnfromPArtofMesh

            //contruct array with vertex ids of the vertices that need to be
trasnfromed
            int src[] = {
4,5,6,7,8,9,10,11,12,13,18,19,20,21,22,23,24,25,26,27,32,33,34,35,36,37,42,43,44,45,
46,51,52,53,54,55,60,61,62,63,64,69,70,71,72,73,78,79,80,81,82,83,84,85,86,87,88,89
};// MOMIE_form
            int n = sizeof(src) / sizeof(src[0]);
            zIntArray ids(src, src + n);

            // construct a gradient parameter
            zVector u, v, w, c, attractor;
            attractor = zVector(0, 0, 0);
            c = cen;
            float maxL = diagLength;

            float distAttractor = cen.distanceTo(attractor);

            float mult = ofMap(distAttractor, 0, 4, 3, 0.5); ;// 4 -
ofMap(cen.length(), 0, maxL, 0.1, 2);

            //cout << mult << "--mult" << endl;

            //construct the transformation matrix
            zTransform TM;
            TM.setIdentity();
            u = zVector(1, 0, 0);
            v = zVector(0, 1, 0);
            w = zVector(0, 0, 1);
            u.normalize(); v.normalize(); w.normalize();
            u *= mult; v *= mult; w *= mult;

            //assign the values to the matrix
            TM.col(0) << u.x, u.y, u.z, 1;
            TM.col(1) << v.x, v.y, v.z, 1;
            TM.col(2) << w.x, w.y, w.z, 1;
            TM.col(3) << 0, 0, 0, 1;
```



Iteration 2_Script Definition

Expanding on option 1 and adding a higher scale factor to the effected vertices. Creating an aggregated form that is densest at the centre and becomes less dense as it expands.

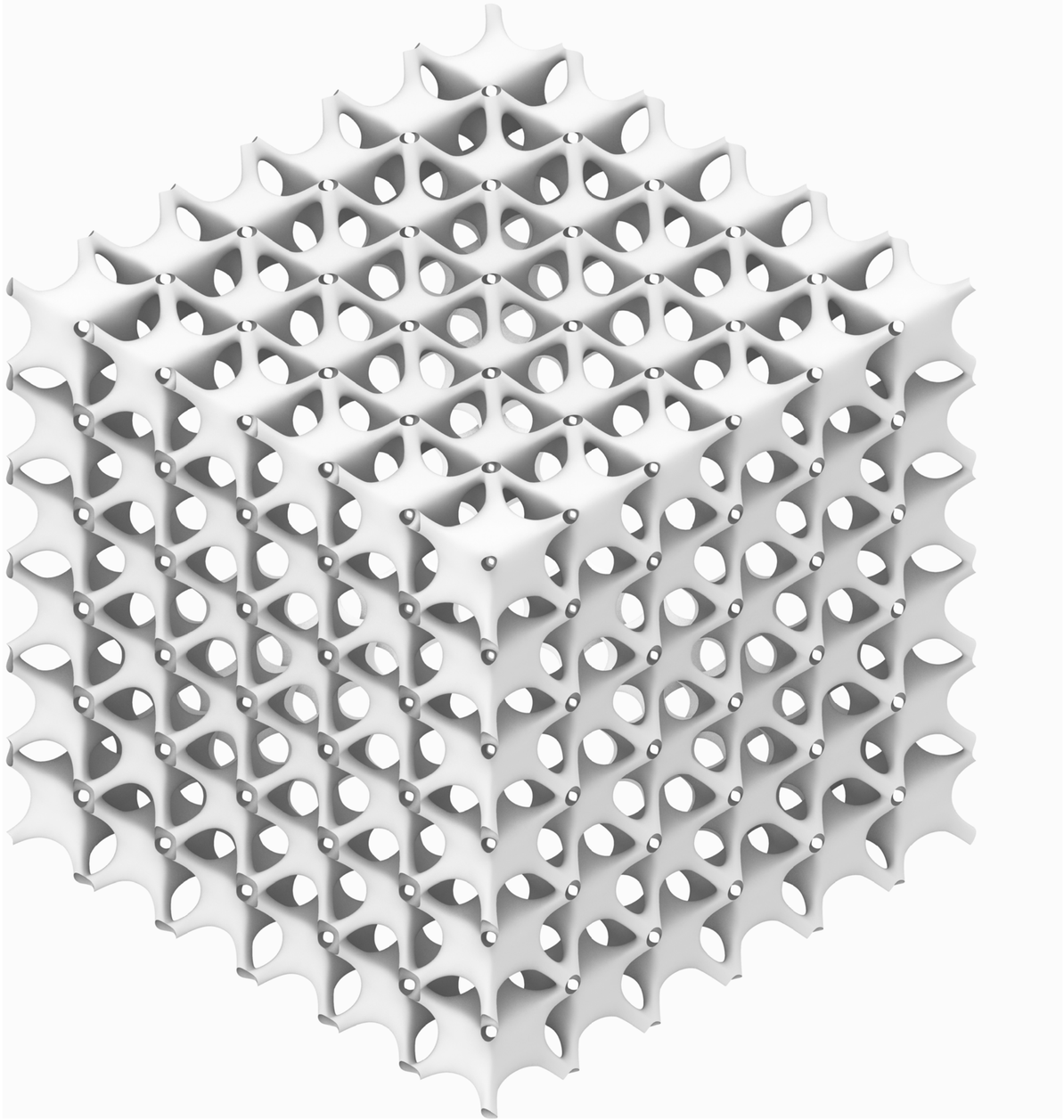
```
createMeshFromFile("data/Momie_Topology.obj");
            transformMesh(mat);

            //---------------------- trasnfromPArtofMesh

            //contruct array with vertex ids of the vertices that need to be
trasnfromed
            int src[] = {
4,5,6,7,8,9,10,11,12,13,18,19,20,21,22,23,24,25,26,27,32,33,34,35,36,37,42,43,44,45,
46,51,52,53,54,55,60,61,62,63,64,69,70,71,72,73,78,79,80,81,82,83,84,85,86,87,88,89
};// MOMIE_form
            int n = sizeof(src) / sizeof(src[0]);
            zIntArray ids(src, src + n);

            // construct a gradient parameter
            zVector u, v, w, c, attractor;
            attractor = zVector(3, 3, 3);
            c = cen;
            float maxL = diagLength;

            float distAttractor = cen.distanceTo(attractor);

            float mult = ofMap(distAttractor, 0, 4, 4, 0.5); ;// 4 -
ofMap(cen.length(), 0, maxL, 0.1, 2);

            //cout << mult << "--mult" << endl;

            //construct the transformation matrix
            zTransform TM;
            TM.setIdentity();
            u = zVector(1, 0, 0);
            v = zVector(0, 1, 0);
            w = zVector(0, 0, 1);
            u.normalize(); v.normalize(); w.normalize();
            u *= mult; v *= mult; w *= mult;

            //assign the values to the matrix
            TM.col(0) << u.x, u.y, u.z, 1;
            TM.col(1) << v.x, v.y, v.z, 1;
            TM.col(2) << w.x, w.y, w.z, 1;
            TM.col(3) << 0, 0, 0, 1;
```



Iteration 3_Script Definition

Aggregating the form in a grid while creating a left to right gradient in central mesh vertices, slowly expanding the shape.

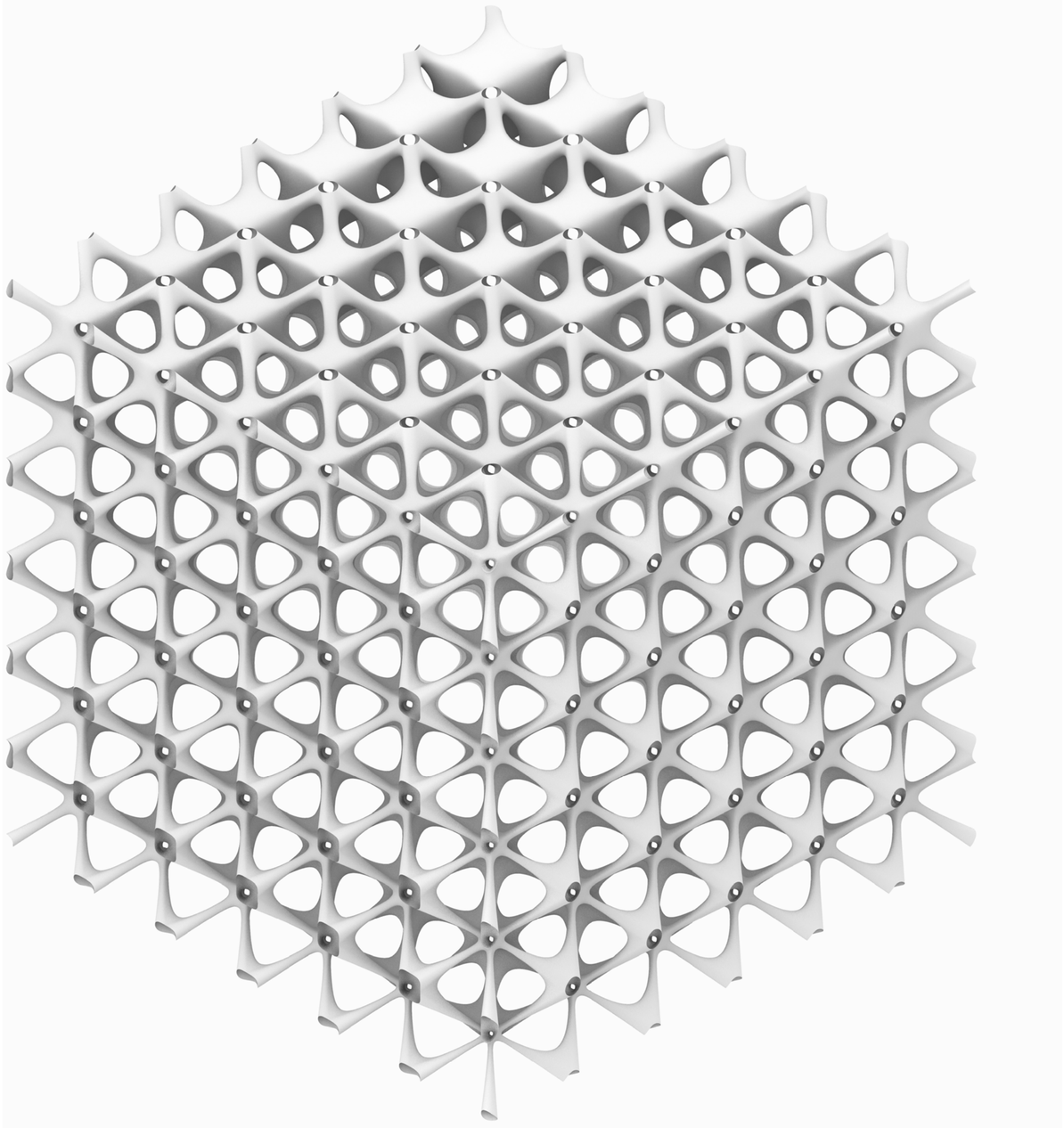
```
      // create crss3d mesh
            createMeshFromFile("data/Momie_Topology.obj");
            transformMesh(mat);

            //---------------------- trasnfromPArtofMesh

            //contruct array with vertex ids of the vertices that need to be
trasnfromed
            int src[] = {
4,5,6,7,8,9,10,11,12,13,18,19,20,21,22,23,24,25,26,27,32,33,34,35,36,37,42,43,44,45,
46,51,52,53,54,55,60,61,62,63,64,69,70,71,72,73,78,79,80,81,82,83,84,85,86,87,88,89
};// MOMIE_form
            int n = sizeof(src) / sizeof(src[0]);
            zIntArray ids(src, src + n);

            // construct a gradient parameter
            zVector u, v, w, c, attractor;
            attractor = zVector(0, 0, 0);
            c = cen;
            float maxL = diagLength;

            float distAttractor = cen.distanceTo(attractor);

            float mult = ofMap(id, 0, 215, .5, 2); ;// 4 - ofMap(cen.length(), 0,
maxL, 0.1, 2);

            //cout << mult << "--mult" << endl;

            //construct the transformation matrix
            zTransform TM;
            TM.setIdentity();
            u = zVector(1, 0, 0);
            v = zVector(0, 1, 0);
            w = zVector(0, 0, 1);
            u.normalize(); v.normalize(); w.normalize();
            u *= mult; v *= mult; w *= mult;

            //assign the values to the matrix
            TM.col(0) << u.x, u.y, u.z, 1;
            TM.col(1) << v.x, v.y, v.z, 1;
            TM.col(2) << w.x, w.y, w.z, 1;
            TM.col(3) << 0, 0, 0, 1;
```



Iteration 4_Script Definition

Aggregating the form in a grid while creating an outside to centre gradient affecting the scaling of the central mesh vertices, slowly expanding the shape. Creating an aggregated form that is densest at the outside corners.

```
// create crss3d mesh
            createMeshFromFile("data/Momie_Topology.obj");
            transformMesh(mat);

            //---------------------- trasnfromPArtofMesh

            //contruct array with vertex ids of the vertices that need to be
trasnfromed
            int src[] = {
4,5,6,7,8,9,10,11,12,13,18,19,20,21,22,23,24,25,26,27,32,33,34,35,36,37,42,43,44,45,
46,51,52,53,54,55,60,61,62,63,64,69,70,71,72,73,78,79,80,81,82,83,84,85,86,87,88,89
};// MOMIE_form
            int n = sizeof(src) / sizeof(src[0]);
            zIntArray ids(src, src + n);

            // construct a gradient parameter
            zVector u, v, w, c, attractor;
            attractor = zVector(0, 0, 0);
            c = cen;
            float maxL = diagLength;

            float distAttractor = cen.distanceTo(attractor);

            float mult = ofMap(distAttractor, 0, 4, .5, 2); ;// 4 -
ofMap(cen.length(), 0, maxL, 0.1, 2);

            //cout << mult << "--mult" << endl;

            //construct the transformation matrix
            zTransform TM;
            TM.setIdentity();
            u = zVector(1, 0, 0);
            v = zVector(0, 1, 0);
            w = zVector(0, 0, 1);
            u.normalize(); v.normalize(); w.normalize();
            u *= mult; v *= mult; w *= mult;

            //assign the values to the matrix
            TM.col(0) << u.x, u.y, u.z, 1;
            TM.col(1) << v.x, v.y, v.z, 1;
            TM.col(2) << w.x, w.y, w.z, 1;
            TM.col(3) << 0, 0, 0, 1;
```



Iteration 5_Script Definition

Aggregating the form in a grid while creating a diagonal gradient affecting the scaling of the central mesh vertices, slowly expanding the shape. Creating an aggregated form that is densest at the upper corner and becomes less dense as the form is aggregated throughout the voxels.

```
createMeshFromFile("data/Momie_Topology.obj");
	transformMesh(mat);

	//---------------------- trasnfromPArtofMesh

	//contruct array with vertex ids of the vertices that need to be
trasnfromed
	int src[] = {
4,5,6,7,8,9,10,11,12,13,18,19,20,21,22,23,24,25,26,27,32,33,34,35,36,37,42,43,44,45,
46,51,52,53,54,55,60,61,62,63,64,69,70,71,72,73,78,79,80,81,82,83,84,85,86,87,88,89
};// MOMIE_form
	int n = sizeof(src) / sizeof(src[0]);
	zIntArray ids(src, src + n);

	// construct a gradient parameter
	zVector u, v, w, c, attractor;
	attractor = zVector(3, 3, 3);
	c = cen;
	float maxL = diagLength;

	float distAttractor = cen.distanceTo(attractor);

	float mult = ofMap(distAttractor, 0, 8, 2.5, 0.75); ;// 4 -
ofMap(cen.length(), 0, maxL, 0.1, 2);

	//cout << mult << "--mult" << endl;

	//construct the transformation matrix
	zTransform TM;
	TM.setIdentity();
	u = zVector(1, 0, 0);
	v = zVector(0, 1, 0);
	w = zVector(0, 0, 1);
	u.normalize(); v.normalize(); w.normalize();
	u *= mult; v *= mult; w *= mult;

	//assign the values to the matrix
	TM.col(0) << u.x, u.y, u.z, 1;
	TM.col(1) << v.x, v.y, v.z, 1;
	TM.col(2) << w.x, w.y, w.z, 1;
	TM.col(3) << 0, 0, 0, 1
```