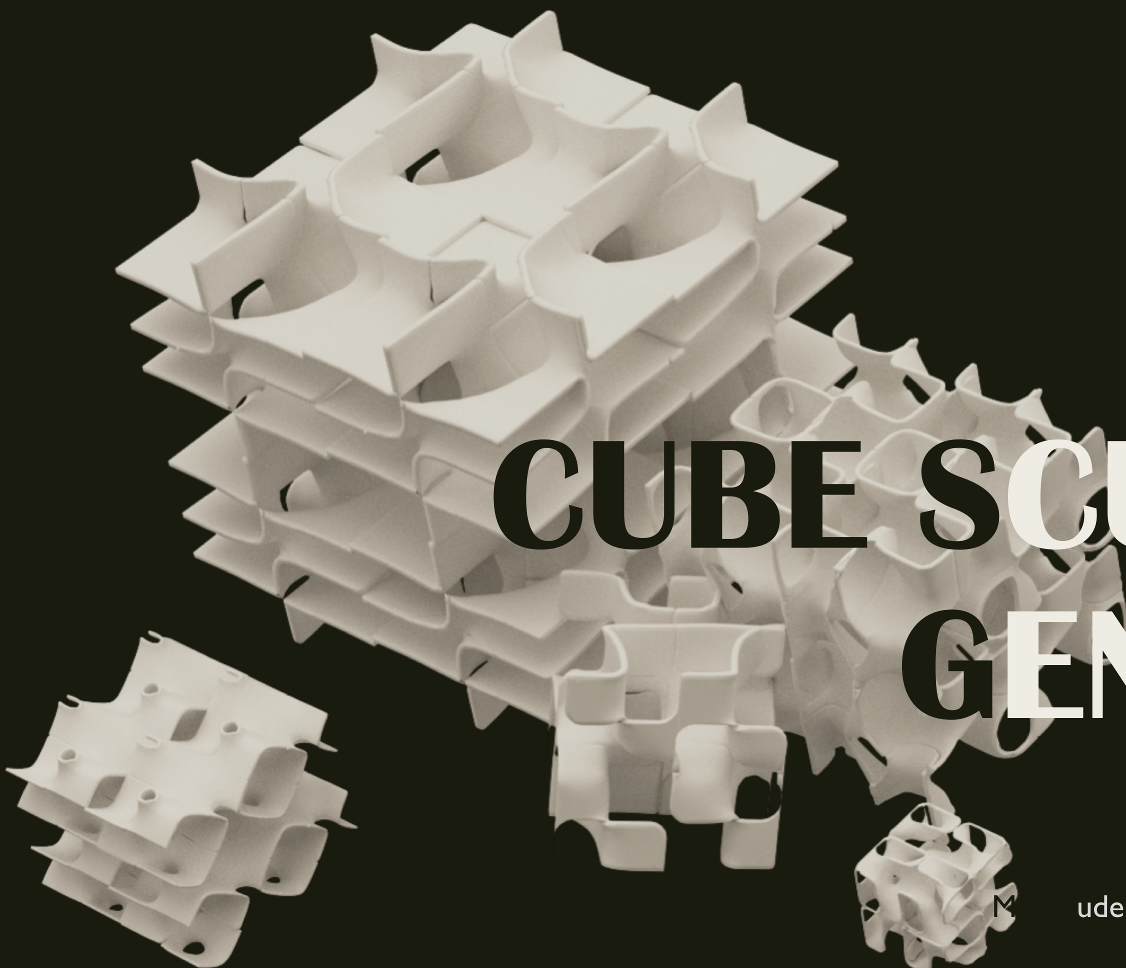

CUBE SCULPTURE GENERATOR

Yifei Wang

M udent in Civil and Environmental Engineering, CMU

Video link: https://youtu.be/NSK6CBjN2EA

Geometry Basis

Iso-Surface

An iso-surface is a three-dimensional analog of an isoline. It is a surface that represents points of a constant value (e.g. pressure, temperature, velocity, density) within a volume of space; in other words, it is a level set of a continuous function whose domain is 3D-space. It normally use the 3 dimensional coordinates to evaluate value for the point. Fig. 1 shows a grasshopper component from Millipede plugin, used in this project to achieve the iso-surface generation. It takes a box as geometry boundary, using resolution of each side to decide the number of points. Select points that have same value as the isoValue parameter, and create a mesh surface.

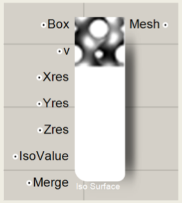

Fig. 1 Iso surface component

Minimal Surface

In parametric design field, the most famous previous works are observations of some minimal surfaces. Those surface has a mathematical concept that locally minimizes its area. These surfaces originally arose as surfaces that minimized total surface area subject to some constraint. Physical models of area-minimizing minimal surfaces can be made by dipping a wire frame into a soap solution, forming a soap film, which is a minimal surface whose boundary is the wire frame.

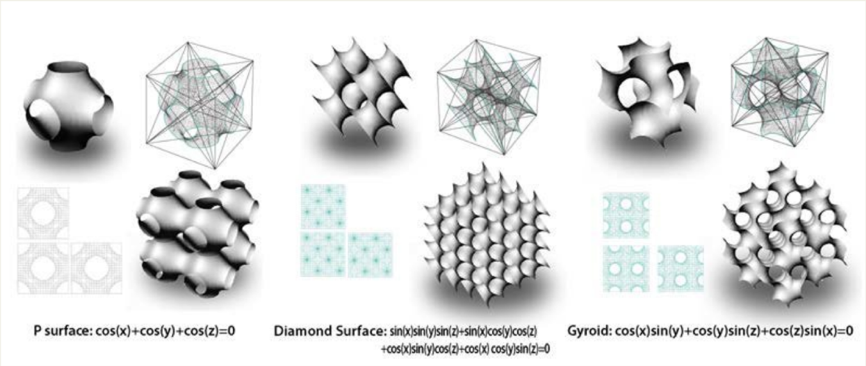

Fig. 2 Previous developed Triply Periodic Minimal Surface[1]

1. https://www.researchgate.net/figure/Triply-Periodic-Minimal-Surfaces_fig1_321756407

Mechanism

L-system

L-system is a dedicate application of rule-based system. It is introduced and developed by Lindenmayer, used to generate self-similar fractals such as iterated function by the idea of string replacement. Just like rule-based system contains initial state, rules, and controls, to be more specific, L-system consists of an initial string, a collection of product rules used to expand the symbols, and an alphabet of symbols to make strings acting as the control. Somehow, the relationship or generation between an initial state and the final state is detectable. With the increase of the iteration, the string is growing according to the ruleset.

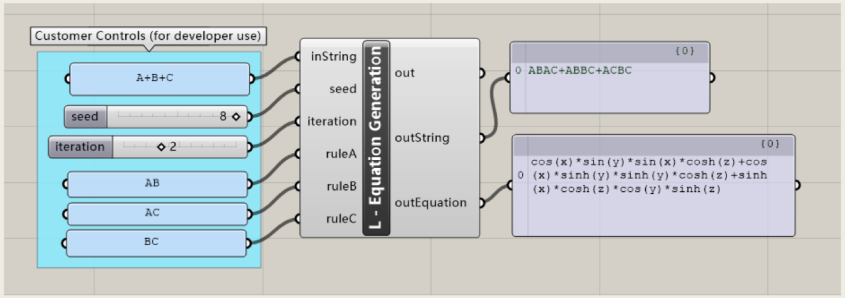

Fig. 3 Self-coded L-system Component

Table I

A	random.choice([sin(x), cos(x), sinh(x), cosh(x)])
B	random.choice([sin(y), cos(y), sinh(y), cosh(y)])
C	random.choice([sin(z), cos(z), sinh(z), cosh(z)])

For this project, the output string from L-system is then translated into a equation with a selection from a series trigonometric functions. The alphabet of the L-system contains “A”, “B”, “C” and operators “+”. To translate the string, “A”, “B”, “C” respectively represent a selected function of a list with independent variables x, y, and z , see Table. Fig. 4 shows the code programmed to replace the string to a equation.

```python
def element(x):
    i1 = str("sin(" + x + ")*")
    i2 = str("cos(" + x + ")*")
    i3 = str("sinh(" + x + ")*")
    i4 = str("cosh(" + x + ")*")
    return [i1, i2, i3, i4]
    
X = element("x")
Y = element("y")
Z = element("z")

def equationGen(Str):
    eq = ""

    random.seed(seed)
    for i in range(len(Str)):
        c = Str[i]
        if c == "A":
            eq += random.choice(X)
        elif c == "B":
            eq += random.choice(Y)
        elif c == "C":
            eq += random.choice(Z)
        elif c == "+":
            eq = eq[:-1]
            eq += c
        elif c == "-":
```

Fig. 3 Code for Translation

Surface Generation

The surface is generated in a workflow in grasshopper. The evaluate component takes a function and variables x, y, z as input, and get the output results. The Range and Cross Reference components are used to create an array of points. Since the trigonometric function has periodic pattern from –pi to pi. The domain of Range is set as –pi To pi, N is the divided steps. The result from Evaluate Component is a list of 9261 values.

$$(20+1)^3 = 9261$$

These values are then assigned to points created in the iso-surface component to be selected and create mesh. Ideally, we are looking for points with value of zero, however, zero value will result in an invalid mesh, so a minimal values is normally used to solve this issue.

After the creation of the mesh, I smooth the geometry with approach from online tutorial. In addition, I used a box array of 8 unit cells to visualize the surface continuity.

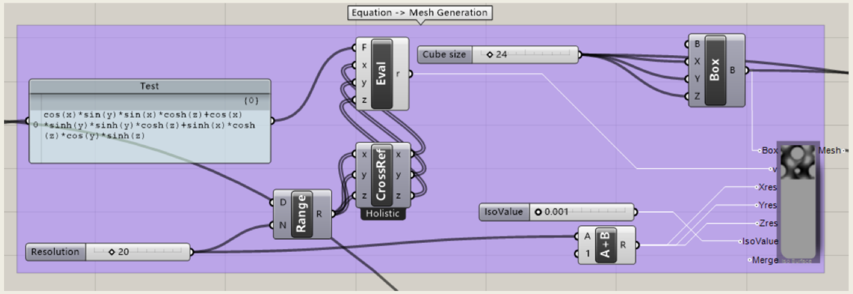

Fig. 4 Workflow for Surface Generation

For this project, the output string from L-system is then translated into a equation with a selection from a series trigonometric functions. The alphabet of the L-system contains "A", "B", "C" and operators "+". To translate the string, "A", "B", "C" respectively represent a selected function of a list with independent variables x, y, and z , see Table. Fig. 4 shows the code programmed to replace the string to a equation.

YouTube Tutorial: https://www.youtube.com/watch?v=mhYrVlvbN4k

Products

3D Sculpture

I selected some of the interesting mesh and assigned a reasonable thickness to them that meet the printing machine's settings. I baked the meshes with thickness to Rhinoceros in different layers. Before export the file to a printable stl file, I used the check mesh commend, and it shows out as a good mesh for all of them.

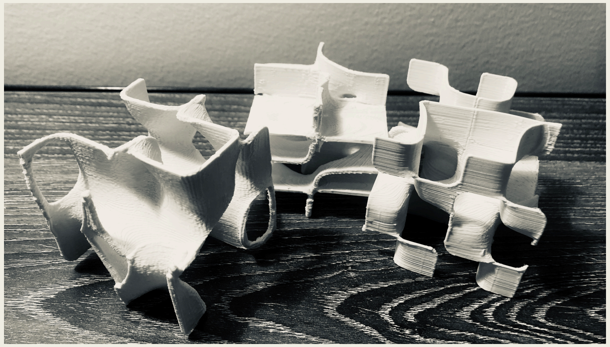

Fig. 5 3D Printed Design Solution Representatives

User Interface

I create a cleaner user interface for the developed generative system using Human UI, a grasshopper plugin. In the popup window, there will be textboxes and sliders as parametric controls, a 3D view to display the result geometry, and other relevant result element, such as the out string and the area of the surface as a indicator for the complexity of the sculpture.

Therefore, instead of directly connect panel and slider to the L-system component, I use value listeners to get the value from interface and connect to the generative system.

Fig. 6 User Interface

There are some other effort to further adjust the position of elements and format the windows. After the above, when user trigger the window with a Boolean toggle, he or she can customize the slider controls as well as the initial string and the ruleset. I would think the designer using this system might not need to know the underlying algorithm, so I decide only display the final string in the window.

YouTube Tutorial: https://www.youtube.com/watch?v=mhYrVlvbN4k

T-SNE Analysis

I choose 60 geometries from two set of rules as dataset for analysis, see Table 2. T-SNE is a dimensional reduction method for unsupervised learning and can visualize cluster results. See Fig. 6 for the workflow and explanation of the required data structure.

To conduct the analysis, I reduce the resolution of the mesh, to has less data but not affect the geometry result. The result from Evaluate component is used as ***gene*** that represent the geometry. There are two attributes that make it appropriate to do so, (i) all the results have the same length, and (ii) the results are one to one correspond to a geometry. So I carefully internalize the data and mesh and make sure that they are respectively related. I build the data tree by connect the data to a component called Flatten Tree, which can assign the path for a data and become a branch of the data tree, see Fig. 9 for example.

The issue from calling each geometry is that grasshopper will recognize the input objects as a whole geometry. To solve this, I use Ungroup that take all 60 mesh objects to create a list of the mesh with the order of connection.

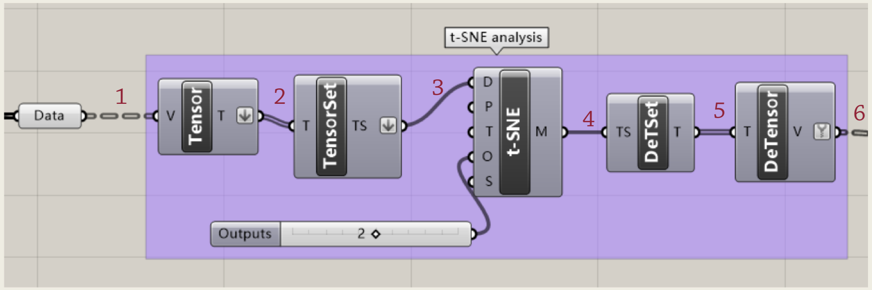

Fig. 8 Self-coded L-system Component

1. Data tree: 60 items each has 729 values
2. Tensor: 60 tensor each has 729 dimensions
3. Tensor set: a tensor set contains 60 tensors
4. Tensor set (after dimensional reduction): a tensor set contains 60 tensor
5. Tensor: 60 tensors each has 2 dimensions
6. Data tree: 60 items each has 2 value

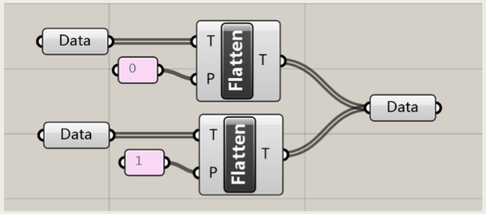

Fig. 8 Self-coded L-system Component

Then, the reduced dimension for each geometry is used as a motion vector, consequently, the results with higher similarity will have more similar vector. Therefore, the new positions of objects that analyzed as similar will be closer to each other after the movement.

Fig. 9 Cluster Analysis Result with T-SNE