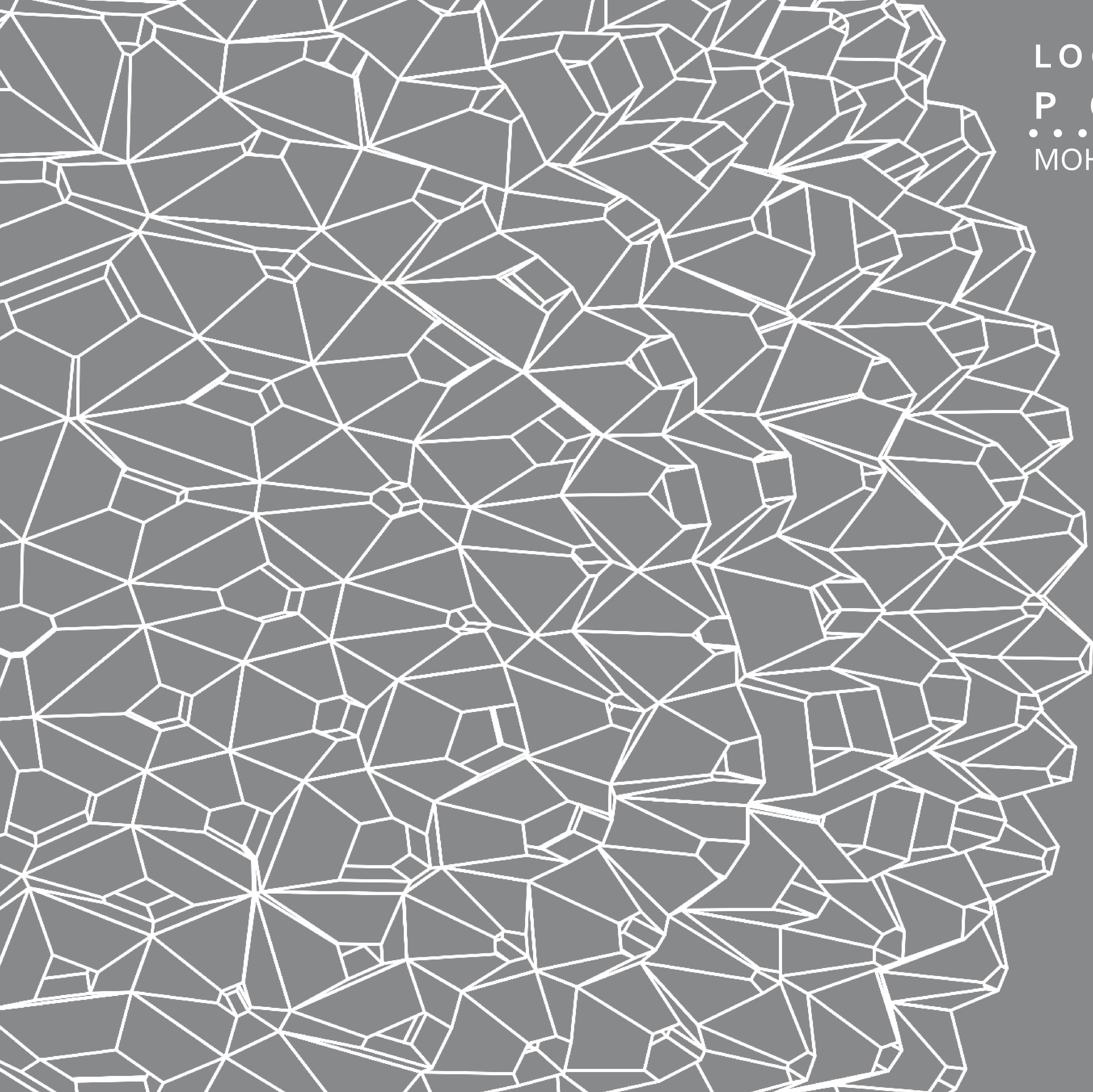


LOGIC TO ARTEFACT

PORTFOLIO

MOHAMMAD ZAKARIYA SHAIKH

CONTENTS
Page

INTRODUCTION
The Brief

The aim within this project is to investigate the methods of computational architecture and how computational systems can assist design and function.

The main outcome of the project is to develop a logic for a small structure to hold 1-3 people (a pavilion). This structure would need to be influenced by computational methods that will link to each other to form the structure. The computational methods have been researched within this portfolio and will look into the logic behind each method with examples of precedence and research development.

In a result of this module, we should have built new fundamental skills to assist us with future works in our university and professional careers.

The pavilion chosen to construct was a Bird Hide and looking into computational methods to assist in constructing a pavilion allowing users to bird watch with also having a sense of privacy from nature. With investigations looking into the form, structure, and façade this project helps us find computational methods to complete these functions.

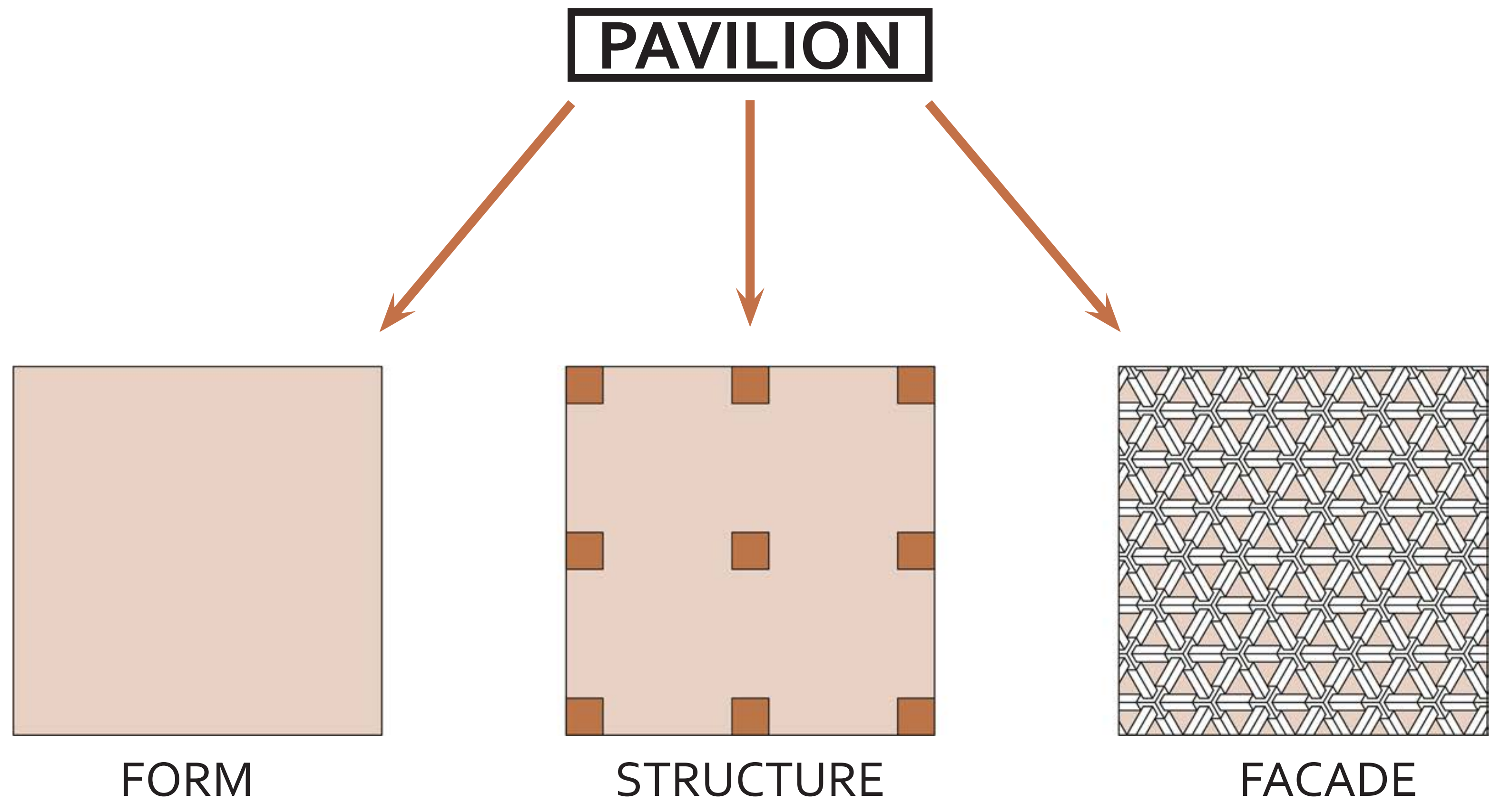


Fig.01: Bird Murmuration.

L-SYSTEMS

What are they?

Created in 1968 by Aristid Lindenmayer, Lindenmayer systems or more commonly known as L-systems use a set of specific rules to define a pattern algorithm over many iterations.

A common example of an L-system would be the use of fractals as shown in Fig.02 with the Seirpinski Arrowhead L-System.

The precedence of fractals comes from village planning within native African tribes, for example, the Ba-Ila village shown in Fig.03 where the oval cell pattern is replicated within each hut.

There are multiple variations of the type of L-systems available. The fractals made before can be seen as a Deterministic L-System, but fractals can be determined from either how the rules are perceived and/or what the end term of the sequence is.

AXIOM: A

RULES: A --> B-A-B
B --> A+B+A

KEY:
A,B: FIXED LENGTH LINE SEGMENT

+: POSITIVE 60 DEGREE TURN

-: NEGATIVE 60 DEGREE TURN

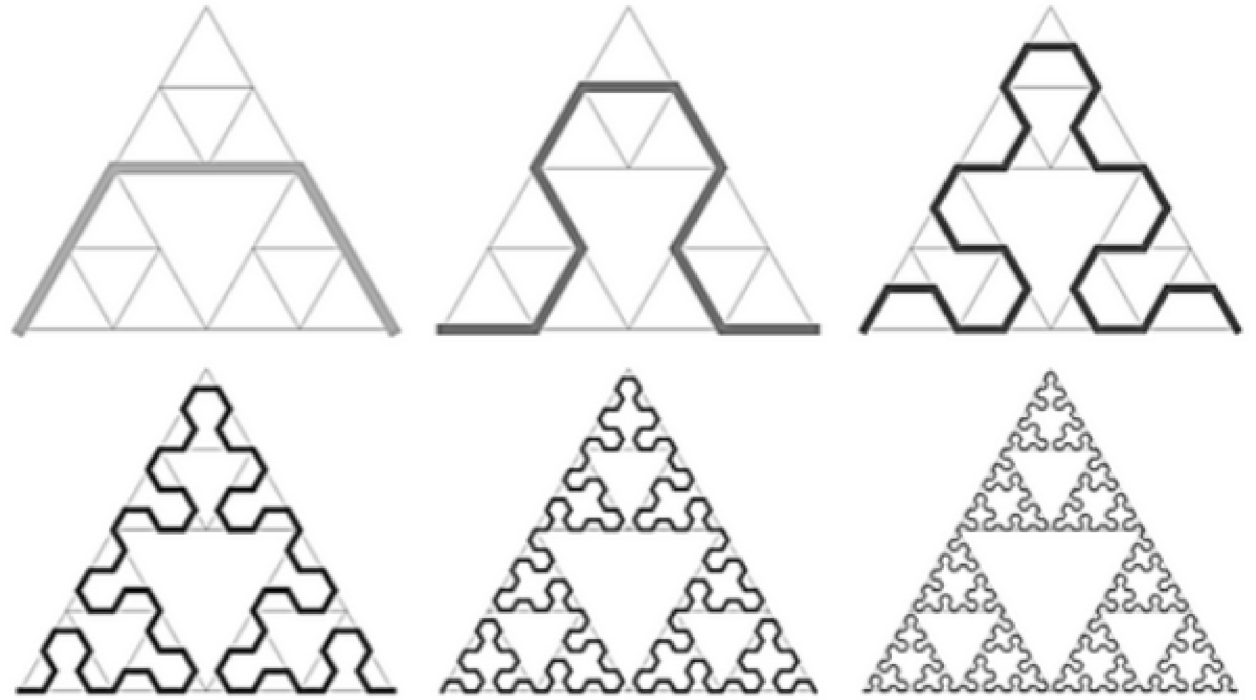


Fig.02: Seirpinski Arrowhead L-System (right), with rules followed (left).

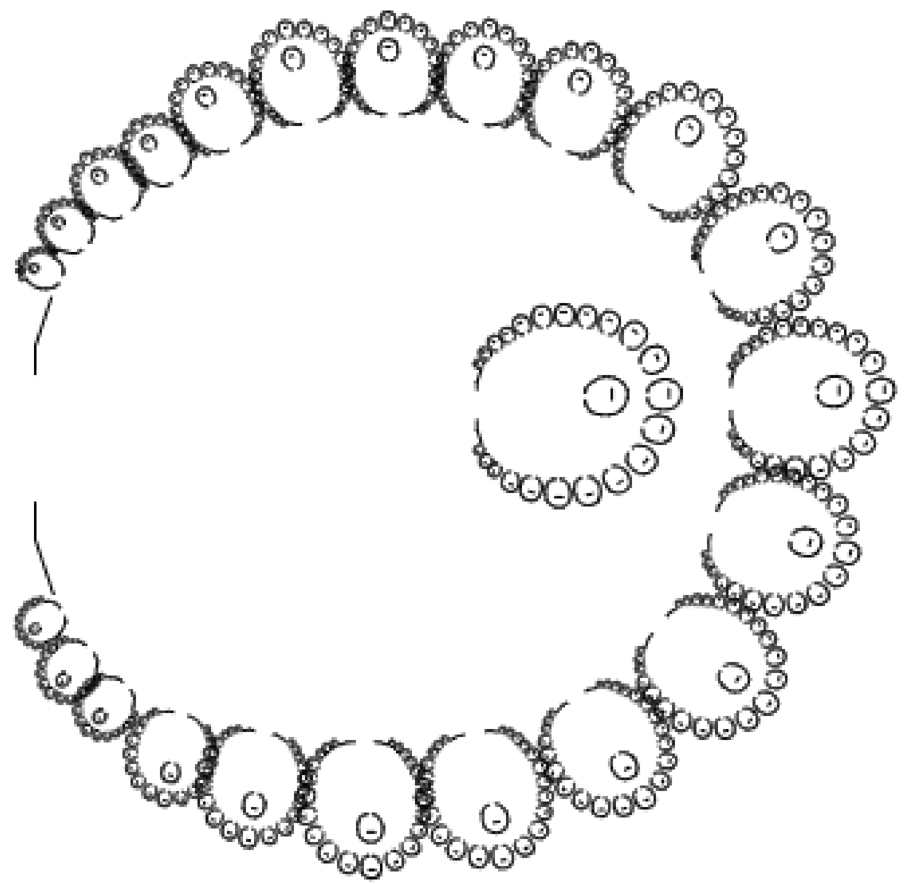


Fig.03: Fractal pattern in Ba-Ila village in Africa, Eglash, R. (1999).

L-SYSTEMS

Stochastic L-Systems

Looking into L-Systems the majority seemed very predictable as they all followed the same rules constantly. From this, the use of a stochastic L-System is put into place for a result to follow one of many rules. This allows for a more chaotic and unpredictable outcome. An example of this would be cellular automation where the result of the segment is dependent on its surroundings.

Within this project, I would want to focus on the façade and structure of the pavilion. Looking into the façade I would set my physical parameters as a flat square surface. Taking inspiration from cellular automation I would use a grid L-System which allows the square to divide into four equal segments radiating from the centre with each segment having an assigned characteristic being either void or shaded. This will then be repeated starting with 1 segment, then 4, 16, 64, 256, 1024, etc.

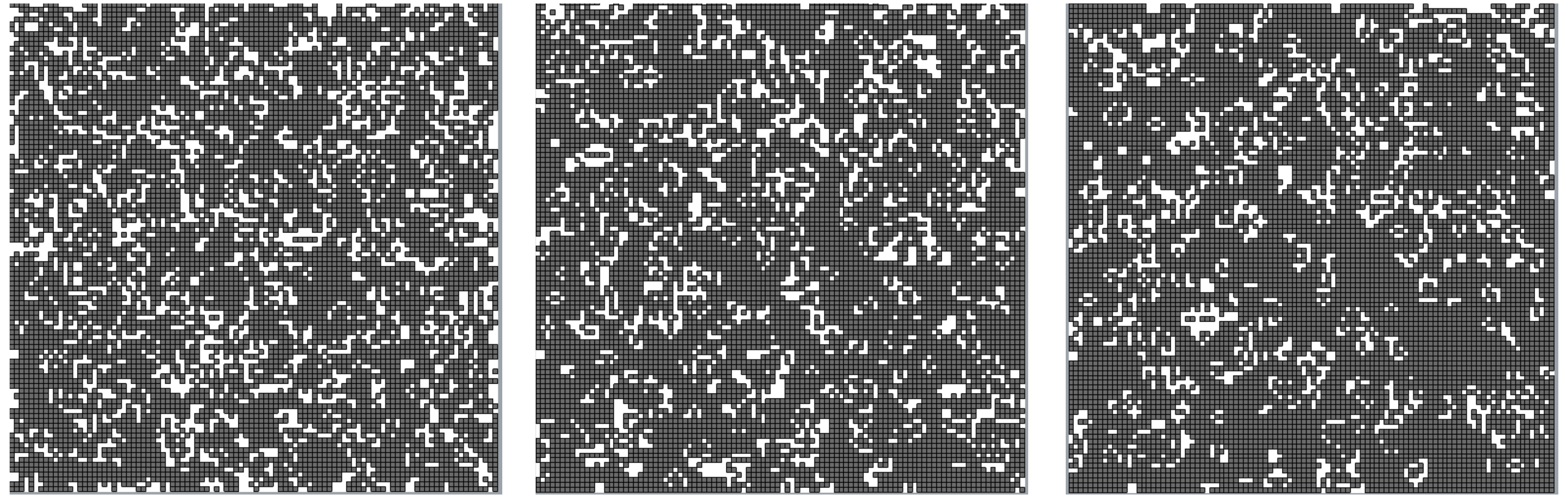


Fig.04: Example of cellular automata, from 1st iteration, 5th iteration and 10th iteration. (Left to right)

$$p_1 : F \xrightarrow{.33} F[+F]F[-F]F$$
$$p_2 : F \xrightarrow{.33} F[+F]F$$
$$p_3 : F \xrightarrow{.33} F[-F]F$$

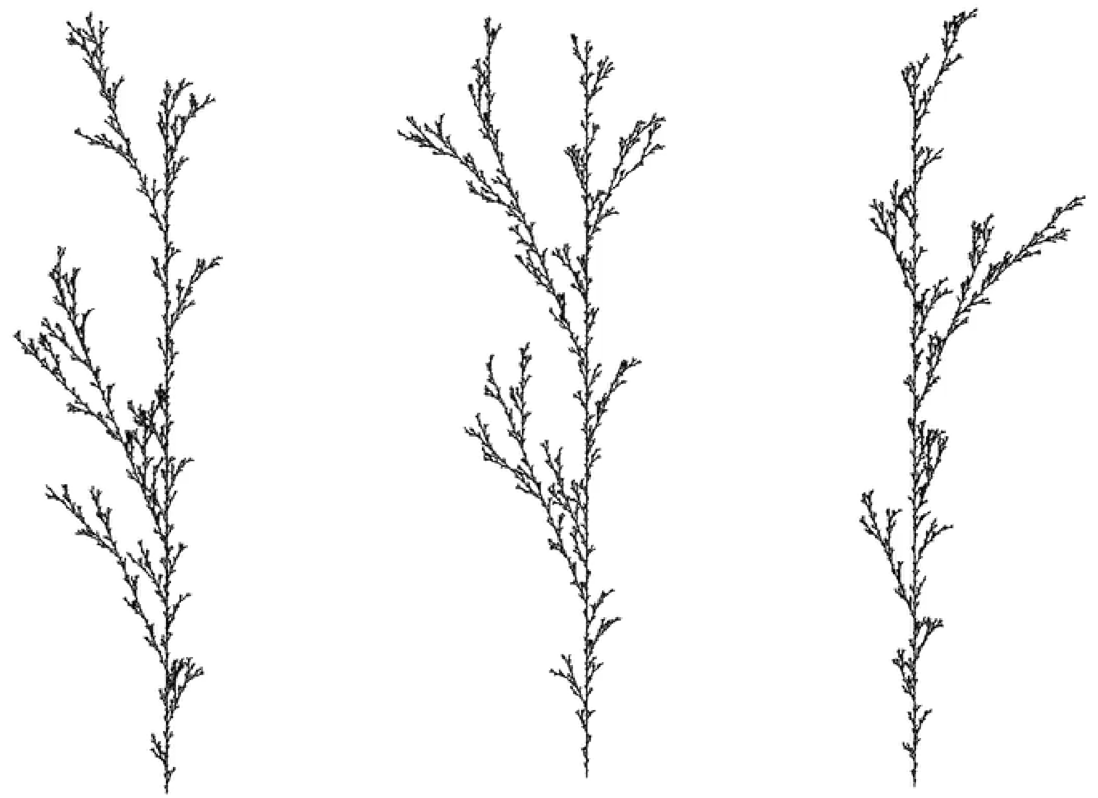


Fig.05: Stochastic L-system used within branching to create a different outcome on every iteration.

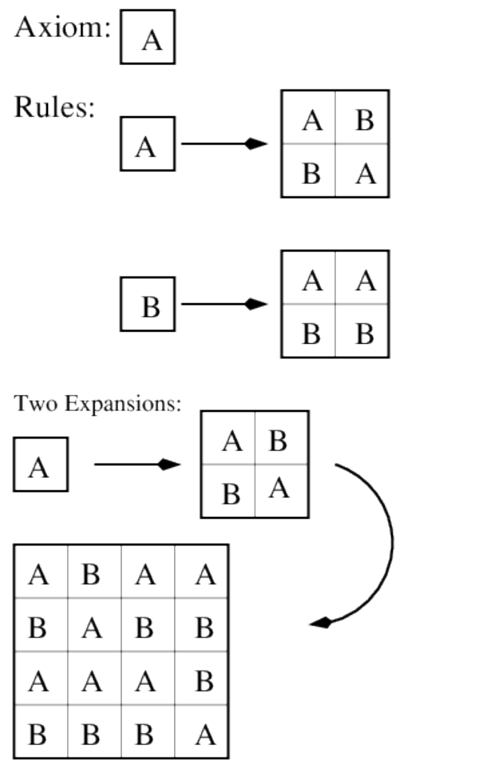


Fig.06: Example of a grid L-system. Ashlock, D., Gent, S. and Bryden, K. (2005)

SURFACE ALGORITHM
Predator-Prey

To find a solution for the task of finding an L-System to assist with allowing viewports while providing a sense of privacy. This was achieved using a stochastic grid L-System, with the rules inspired by the rules of the Predator-Prey relationship.

Based on the research of the auto-catalytic relationship between foxes and rabbits by Alfred J. Lotka shown in the book Shapes by Philip Ball how this relationship is put into the L-System format shown in Fig.06.

After the last step, the number of Rabbits increases due to the scarcity of Foxes.

When this relationship is mapped on a graph it is shown the majority of the time there are more rabbits at any point in time and when there are more Foxes than rabbits the population of Foxes quickly dies down.

For the Algorithm that I am researching, I would want to implement this predator-prey model onto a grid surface where Rabbits are shown as a shaded panel and Foxes are shown as an open panel to allow visibility.

When this relationship is mapped on a graph it is shown the majority of the time there are more rabbits at any point in time and when there are more Foxes than rabbits the population of Foxes quickly dies down. This allows for the rules to be applied to the Stochastic Grid L-System as mentioned before.

1. G + R = more R (with G representing Grass and R representing rabbits

2. R + F = more F with F representing Foxes)

3. F = D (with D representing Dead)

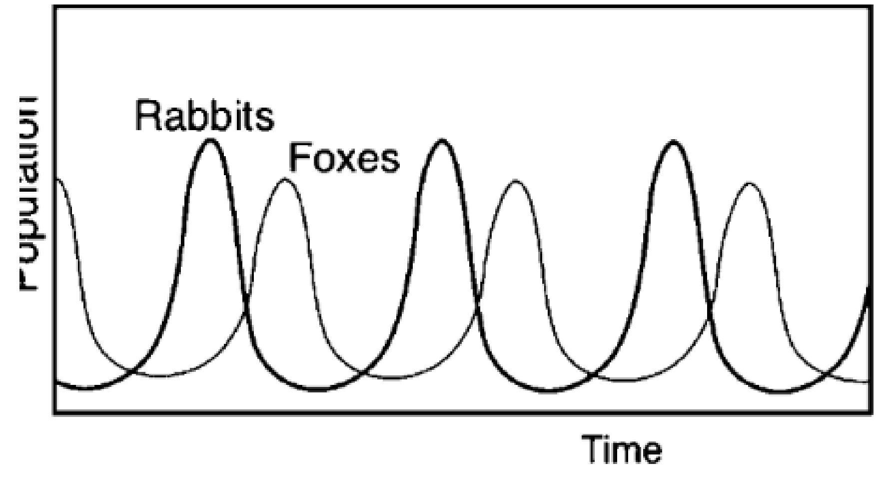


Fig.07: Auto-Catalytic relationship between Rabbits and Foxes with L-System rules (top). Ball, P (2009).

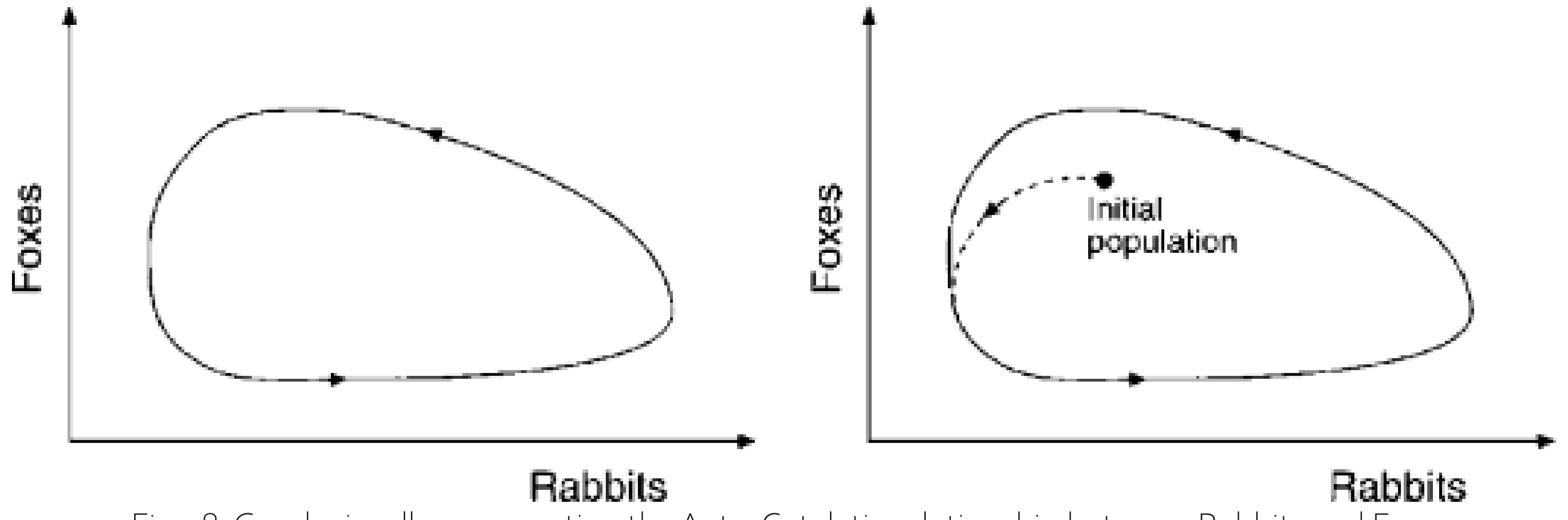


Fig.08: Graph visually representing the Auto-Catalytic relationship between Rabbits and Foxes. Ball, P (2009).

SURFACE ALGORITHM
Hypothesis

When researching the relationship between predator and prey, Lotka was attempting to find a relationship to the Belousov–Zhabotinsky reaction. This reaction shows an oscillation between two chemicals creating a wave-like pattern that is auto-catalytic.

There has been research attempting to replicate this reaction from mathematical models. An example is from the Institute for Molecular Physiology in Dortmund from the use of cellular automation which represents the results as a grid of cells. The results show an organised set of spiral geometry. However, the results have been varying from different institutions such as from Michael Hassell from, Imperial College, London. Where he focuses on host–parasitoid interactions and concludes a chaotic pattern.

From my testing of this algorithm, my initial thinking is that the results will replicate either one of these results.

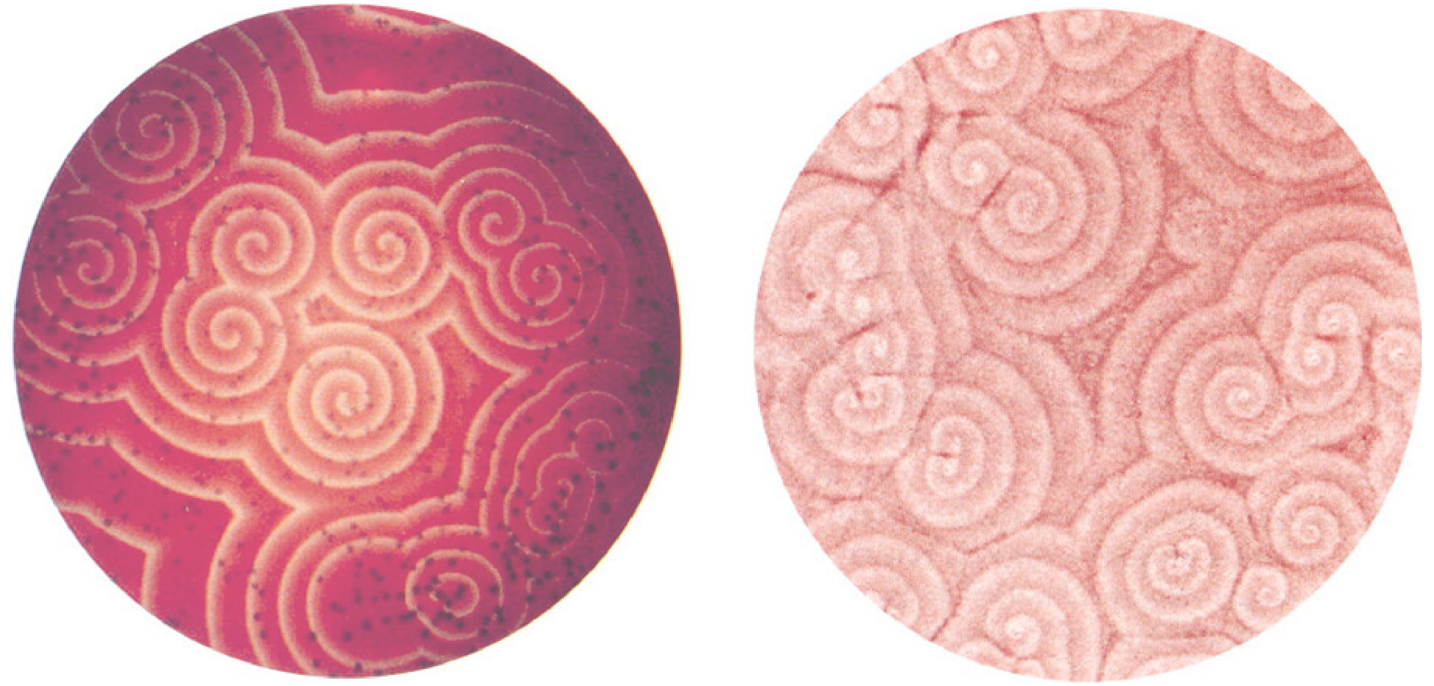


Fig.09: Belousov-Zhabotinsky reaction. Ball, P (2009).

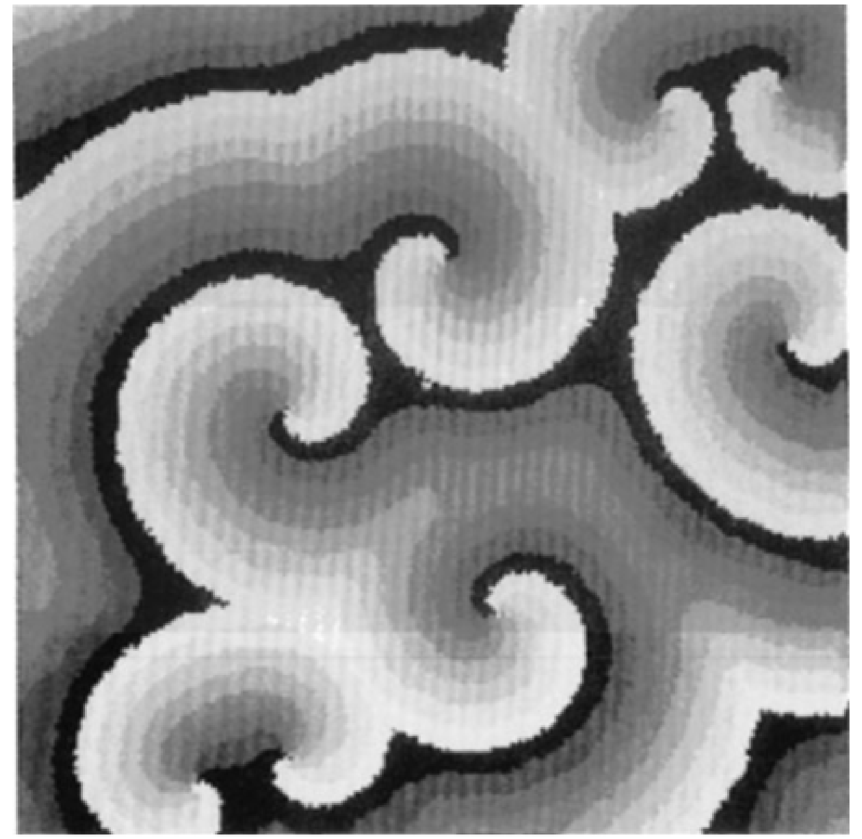


Fig.10:Chaotic mathematical model representing the BZ reaction, Imperial College London. Ball, P (2009).

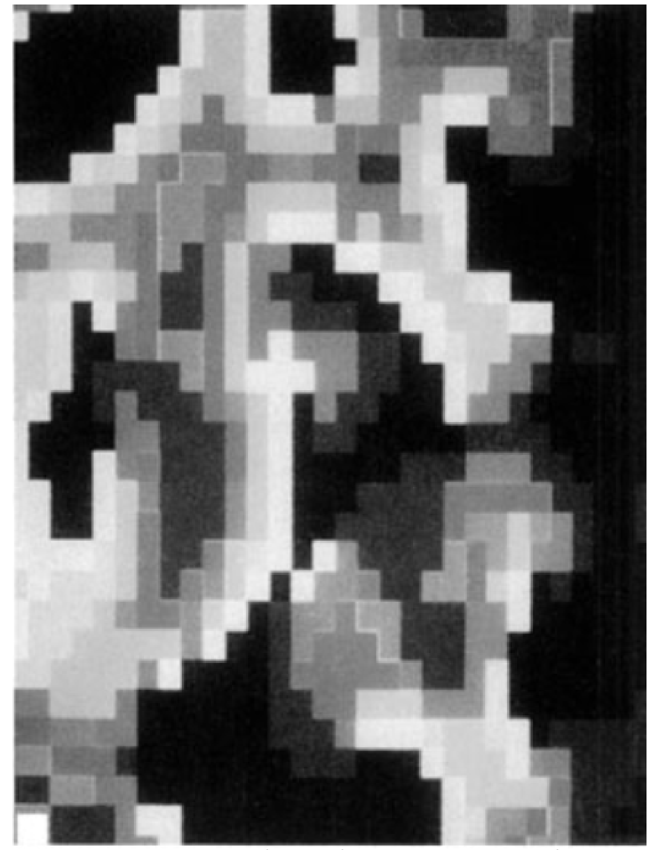


Fig.10: Organised mathematical model representing the BZ reaction, Institute for Molecular Physiology, Dortmund. Ball, P (2009).

SURFACE ALGORITHM

Rules Followed

Within this project, I would want to focus on the façade and structure of the pavilion. Looking into the façade I would set my physical parameters as a flat square surface. Taking inspiration from cellular automation I would use a grid L-System which allows the square to divide into four equal segments radiating from the centre with each segment having an assigned characteristic being either void or shaded. This will then be repeated starting with 1 segment, then 4, 16, 64, 256, 1024, etc.

The initial theory of this algorithm was to represent prey as H: Hide and predator as V: View allowing the predator (V) to perforate through the prey (H) creating radial attractor points within the surface.

The rules assigned resemble there being more prey than predator.

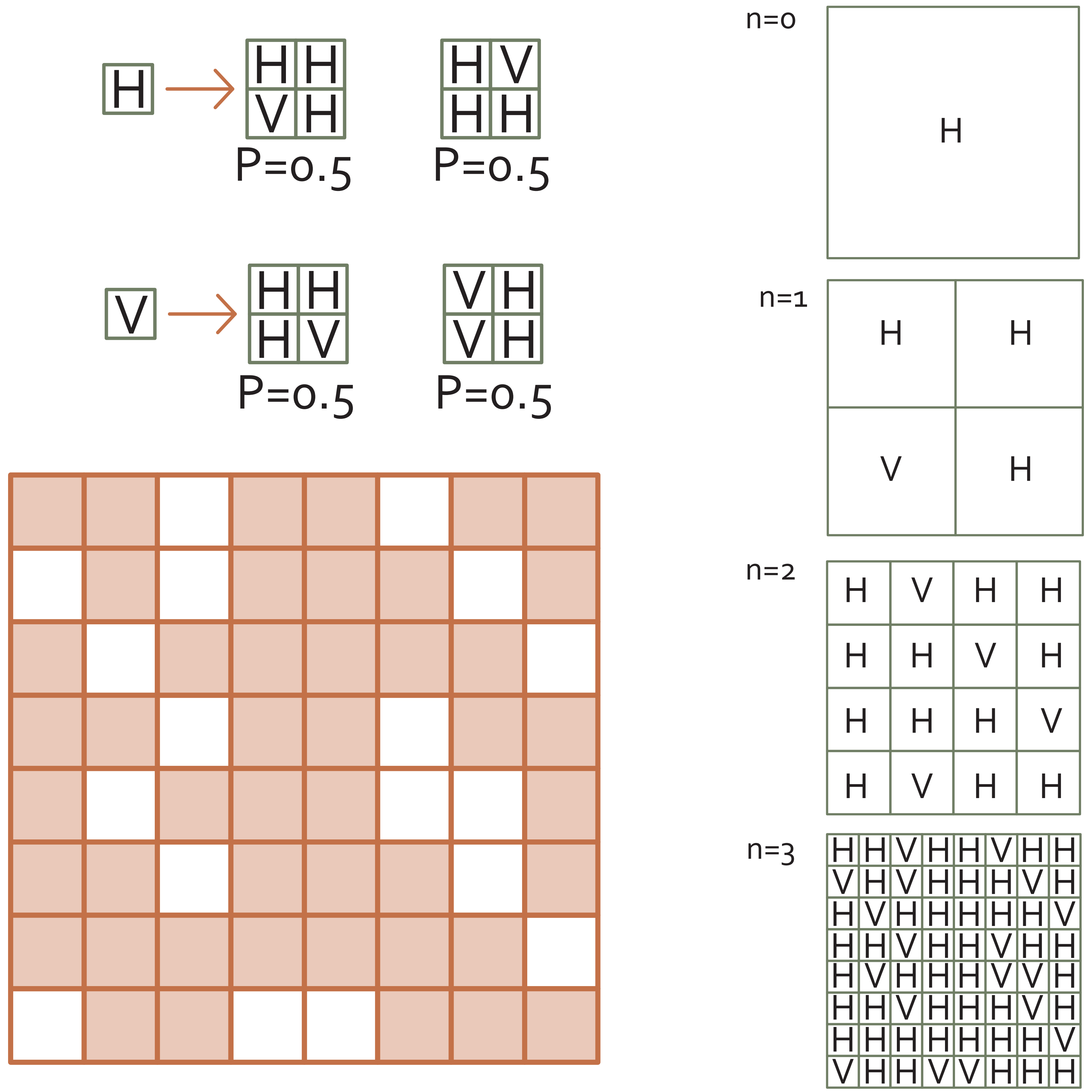


Fig.11: Diagram explaining the surface algorithm with rules (top left) and first four iterations (right).

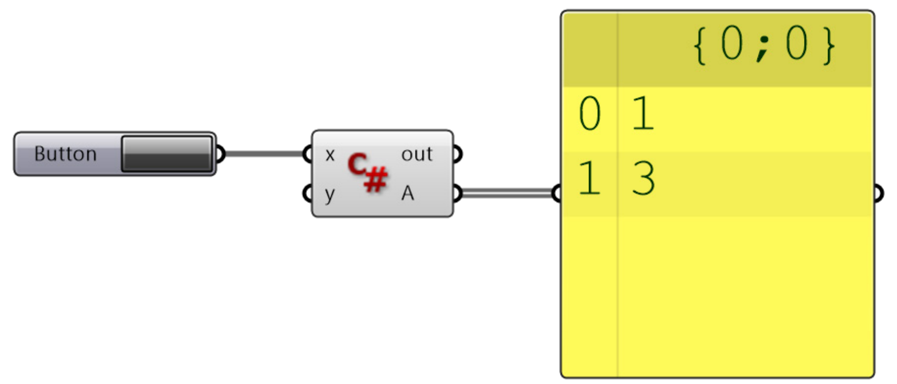
SURFACE ALGORITHM

Logic

The rules used for the algorithm predominate "H" which will be represented as a shaded square and will relate to the prey within the Predator-Prey model. This will allow for some viewpoints but also allow for a sense of privacy from nature.

As this is a stochastic search there is a probability element needed within the rules of the algorithm, this would be a similar random probability from when a coin is flipped.

To replicate this aspect, I created a C# component in Grasshopper recreating this randomness between two outcomes between 1-2 and 3-4.



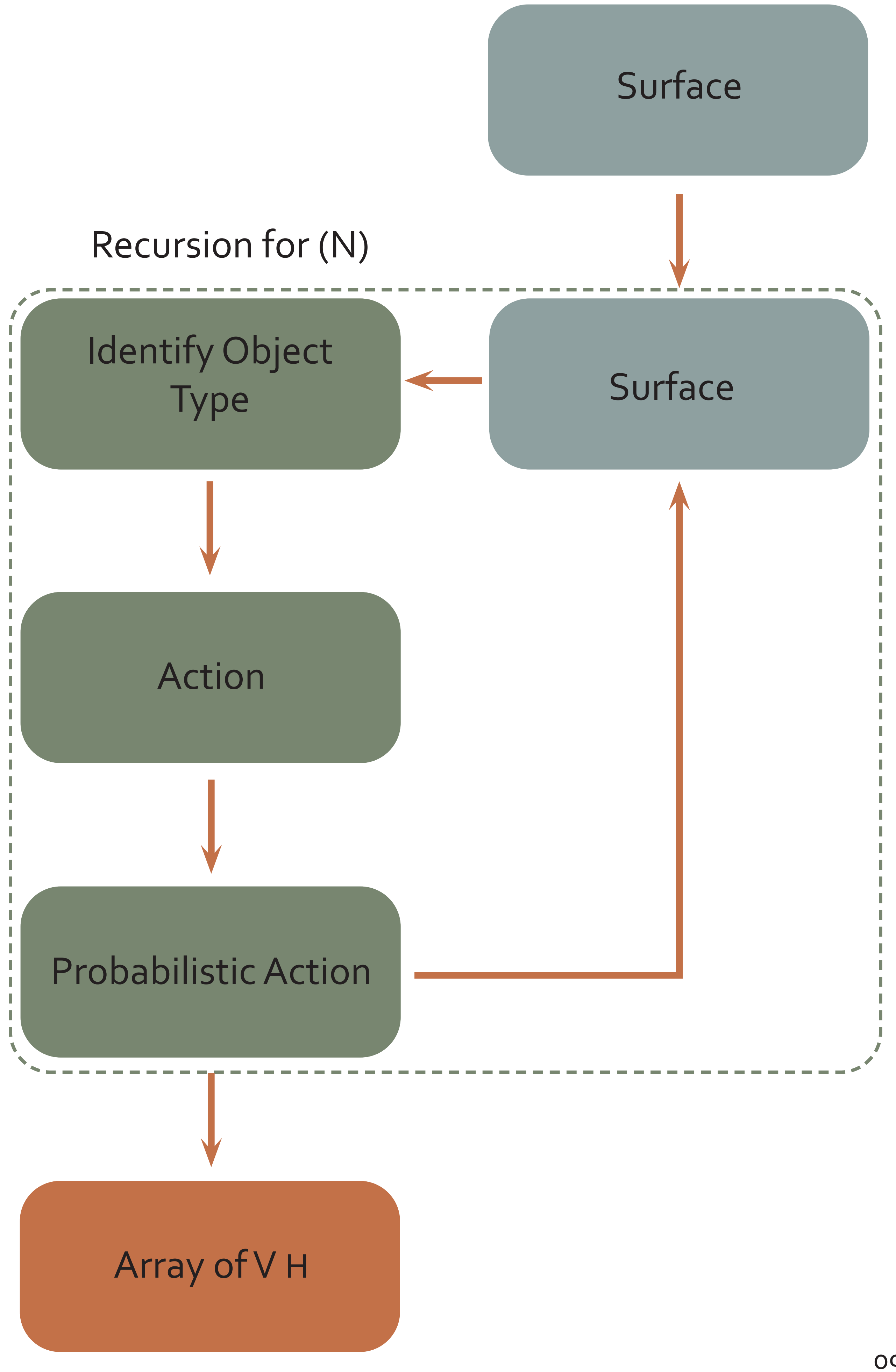

```
private void RunScript(bool x, object y, ref object A)
{
  //Random Number Generator

  var randoma = new Random();
  var returnlista = new List<double>();

  for (int i = 0; i < 1;i++)
  {
    returnlista.Add(randoma.Next(1, 3));
    returnlista.Add(randoma.Next(3, 5));
  }

  A = returnlista;
```

Fig.12: Grasshopper mapping with expanded C# component.

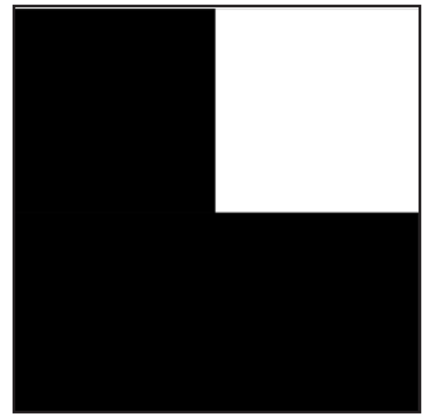


SURFACE ALGORITHM RESULTS

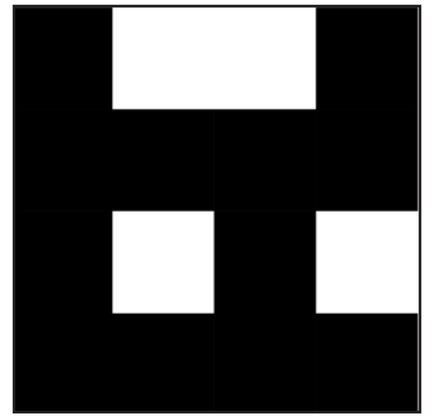
Variation 1

n=0

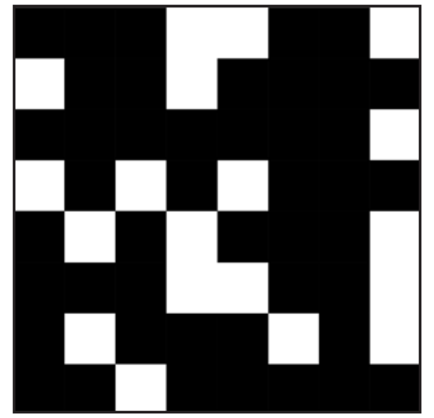
n=1



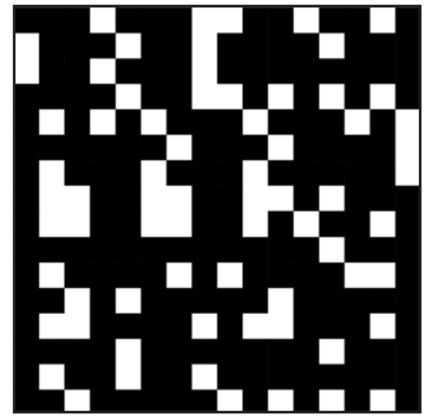
n=2



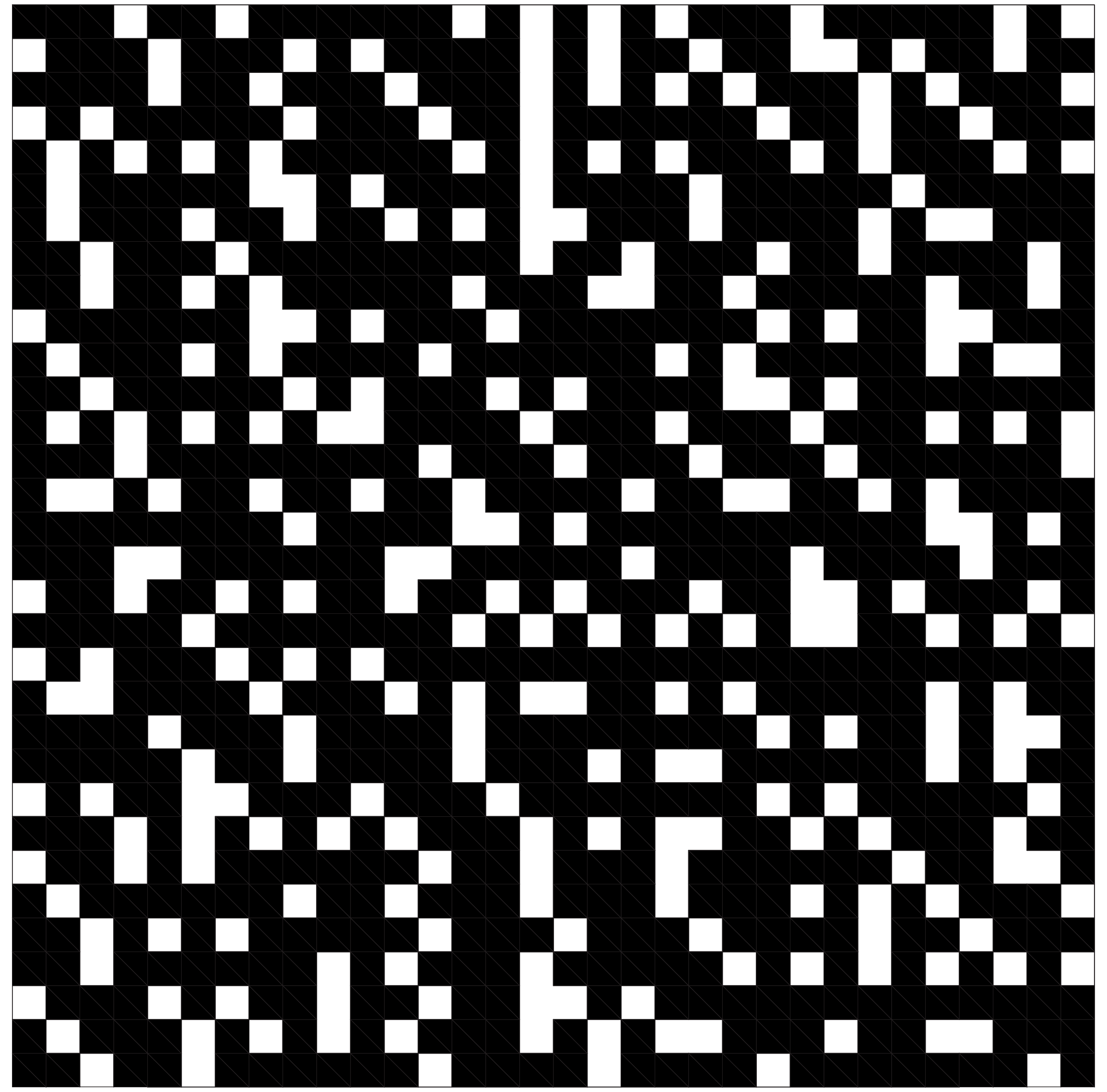
n=3



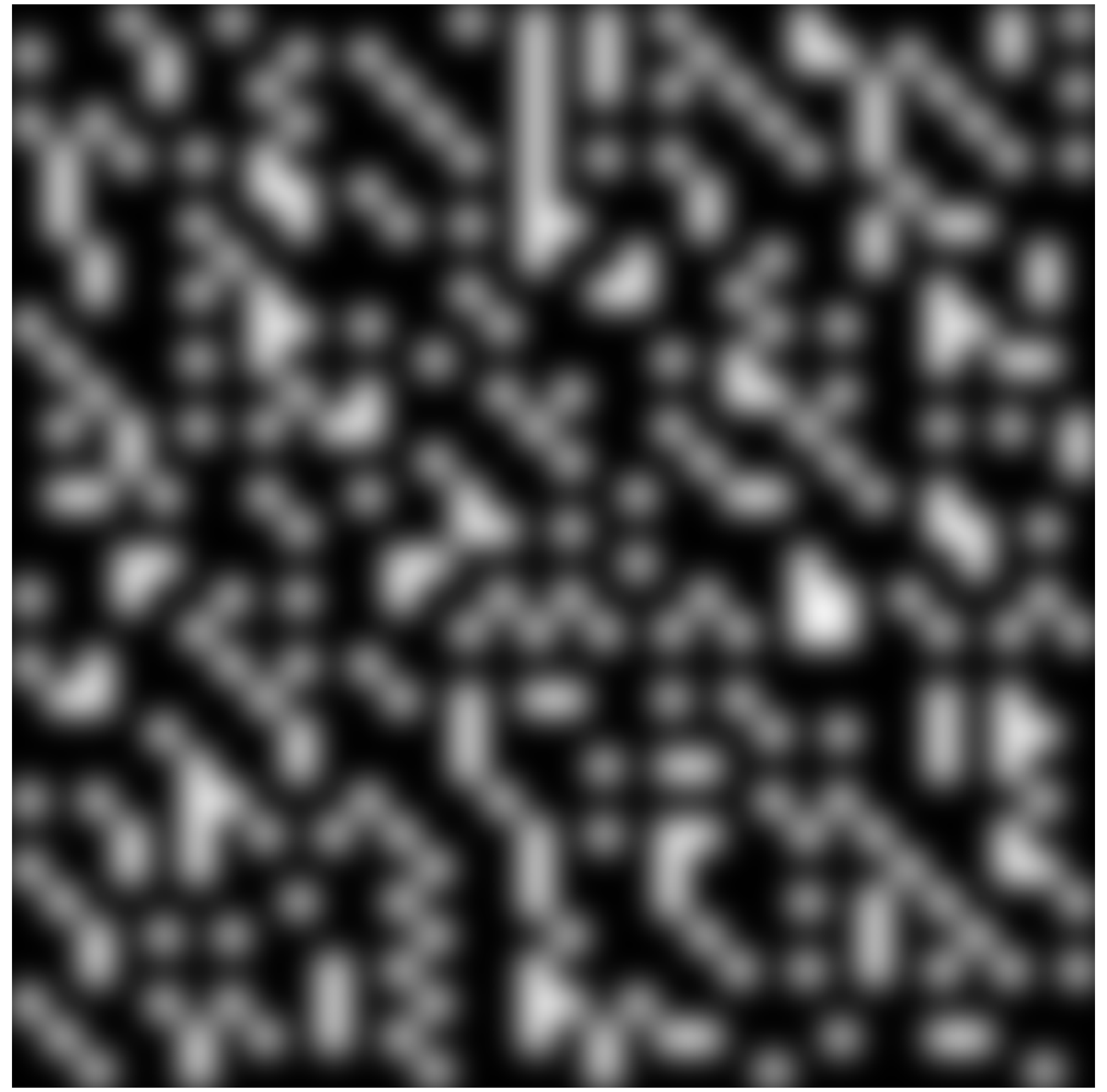
n=4



n=5



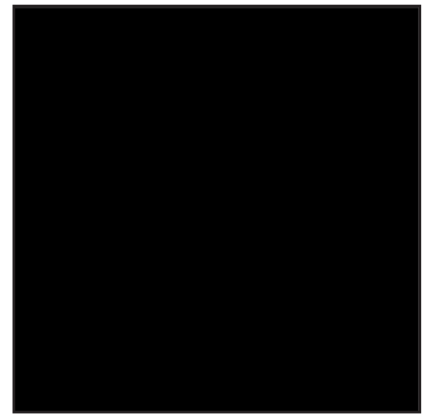
n=5 (Gaussian Blur: 70%)



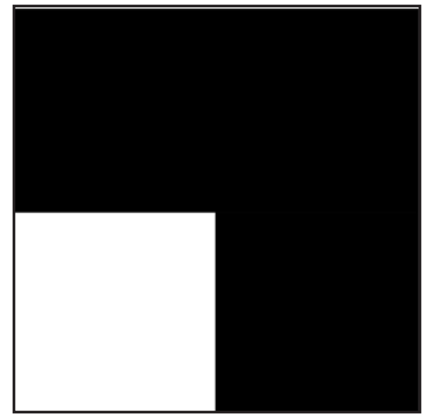
SURFACE ALGORITHM RESULTS

Variation 2

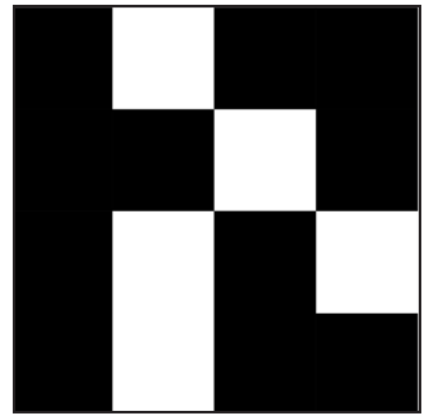
n=0



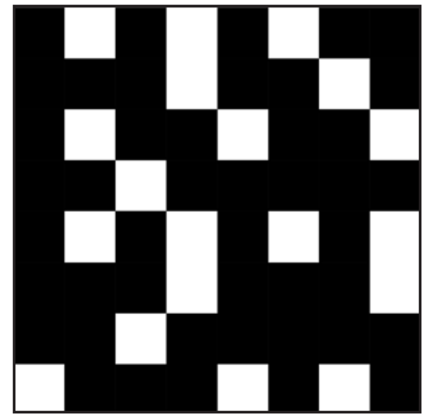
n=1



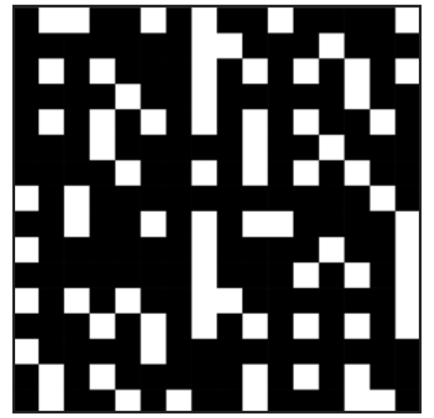
n=2



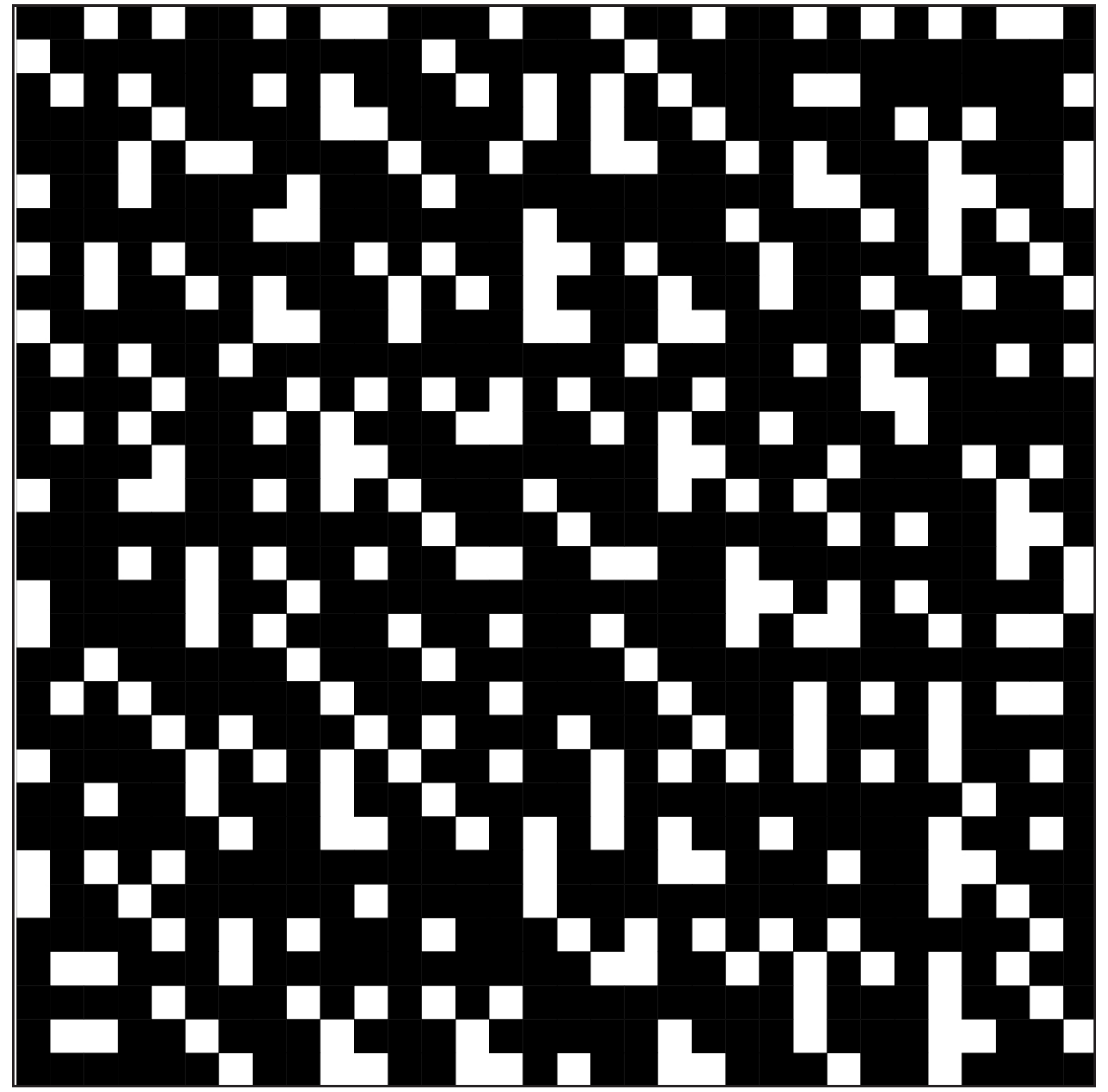
n=3



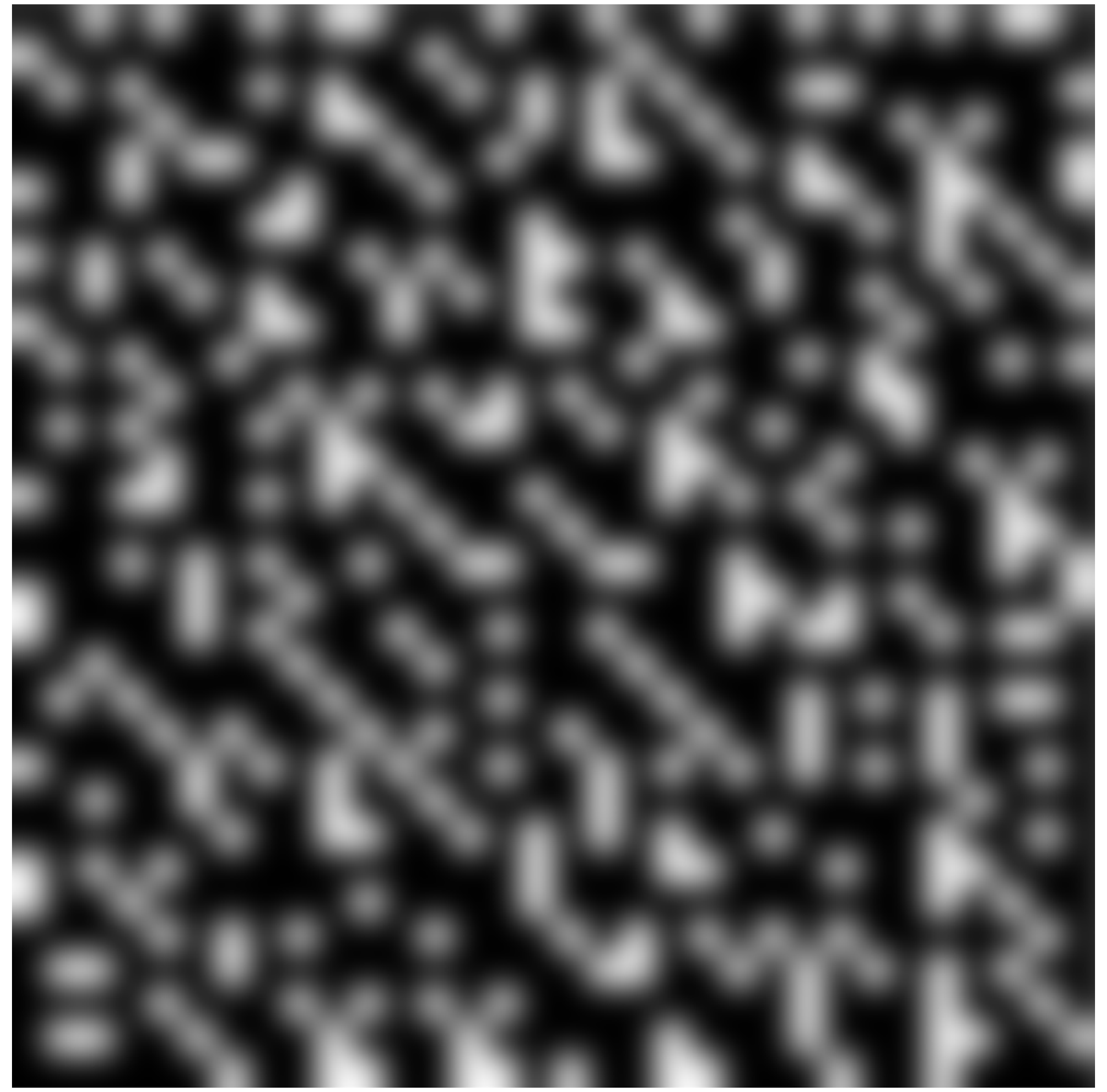
n=4



n=5

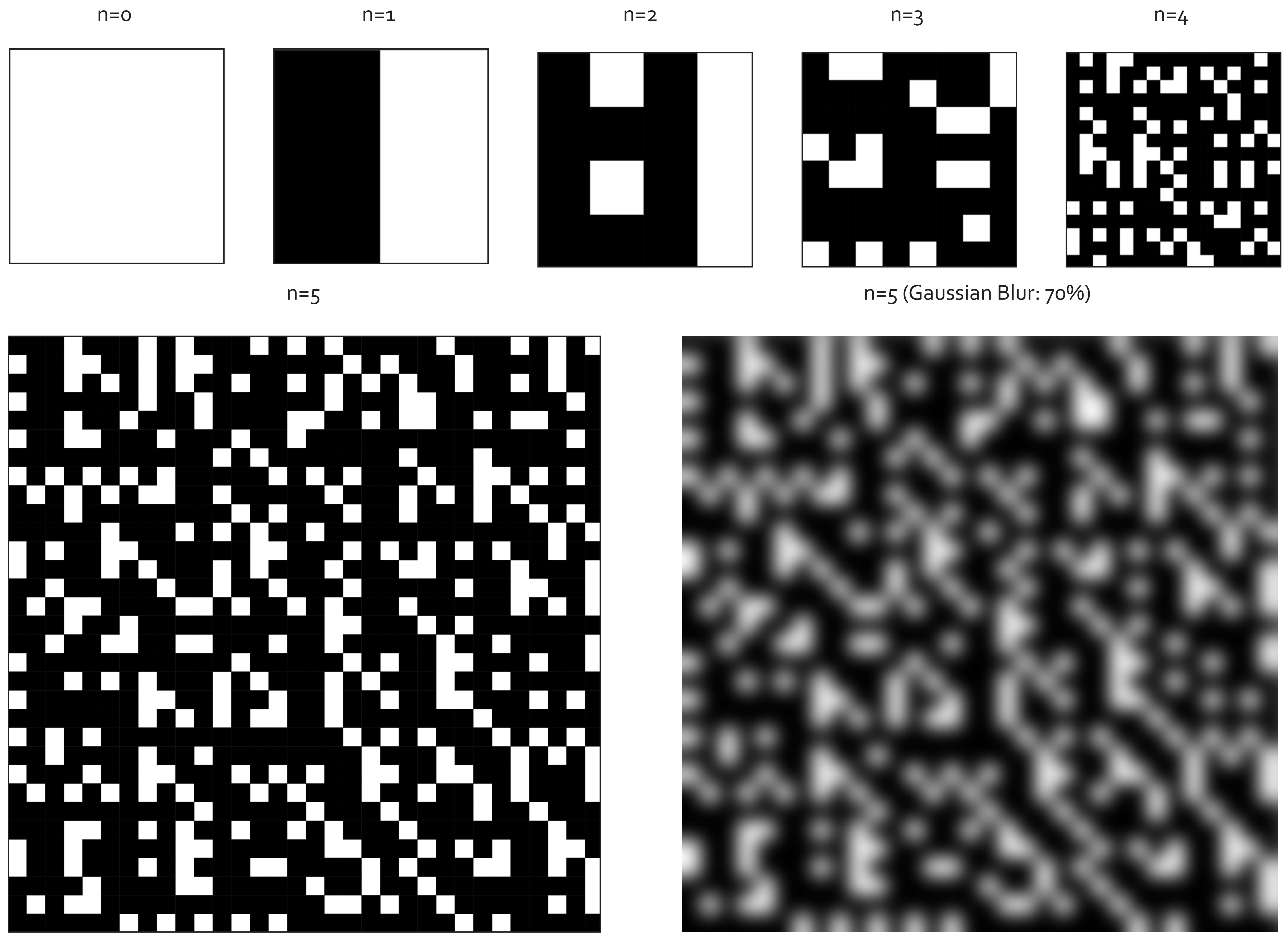


n=5 (Gaussian Blur: 70%)



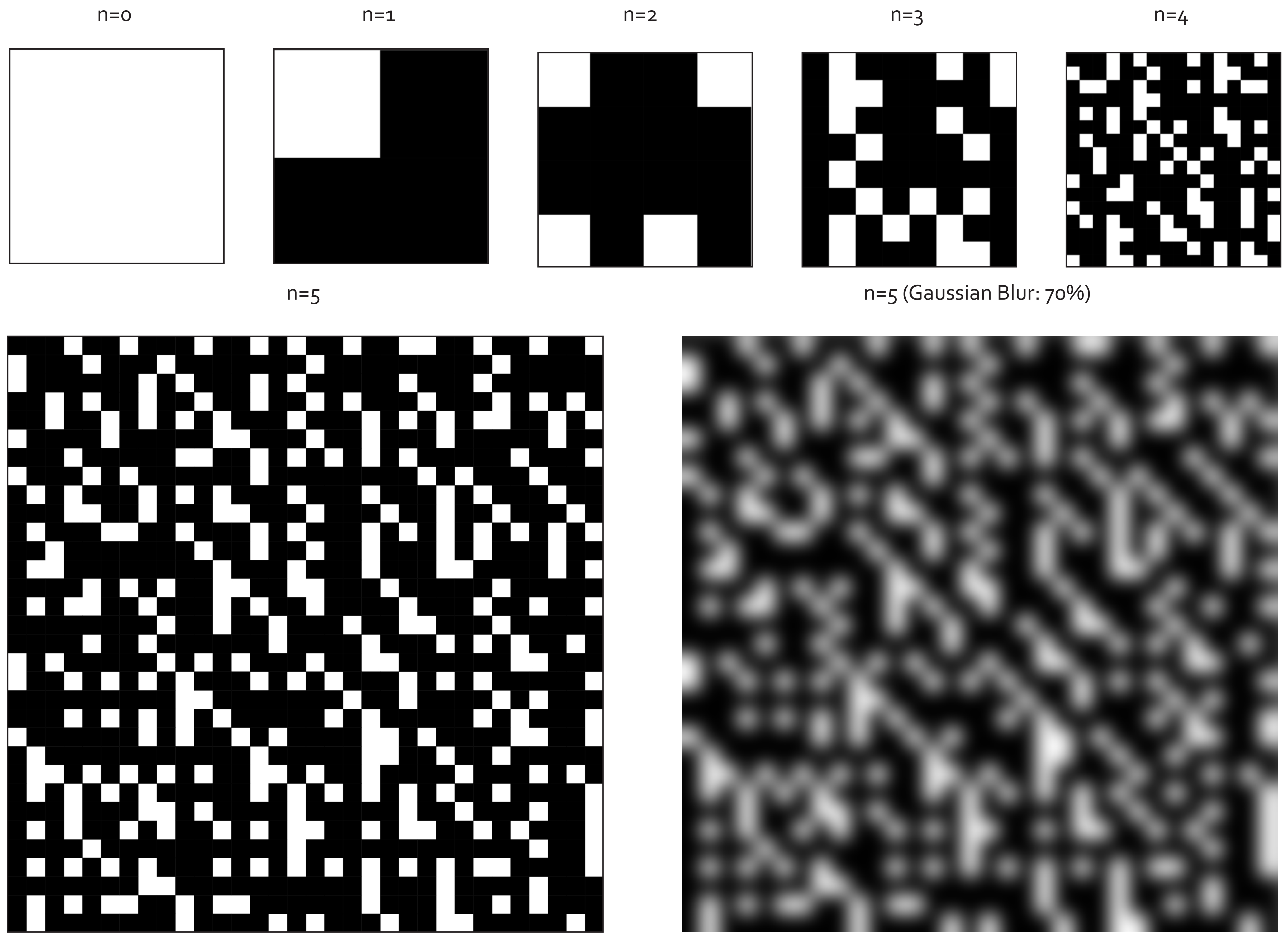
SURFACE ALGORITHM RESULTS

Variation 3



SURFACE ALGORITHM RESULTS

Variation 4



VORONOI
Rules Followed

The Voronoi structure is used as a packing system in the creation of cells from points. This can be seen in many elements within nature such as the wing of the dragonfly and the stacking of bubble formations.

As most natural patterns are structurally stable I wanted to see if a Voronoi-based grid will be more beneficial compared to a traditional grid method. Research from Giulia Angelucci and Fabrizio Mollaioli shows that using a Voronoi-like grid system for buildings not only is beneficial for structure but also reduces the material needed.

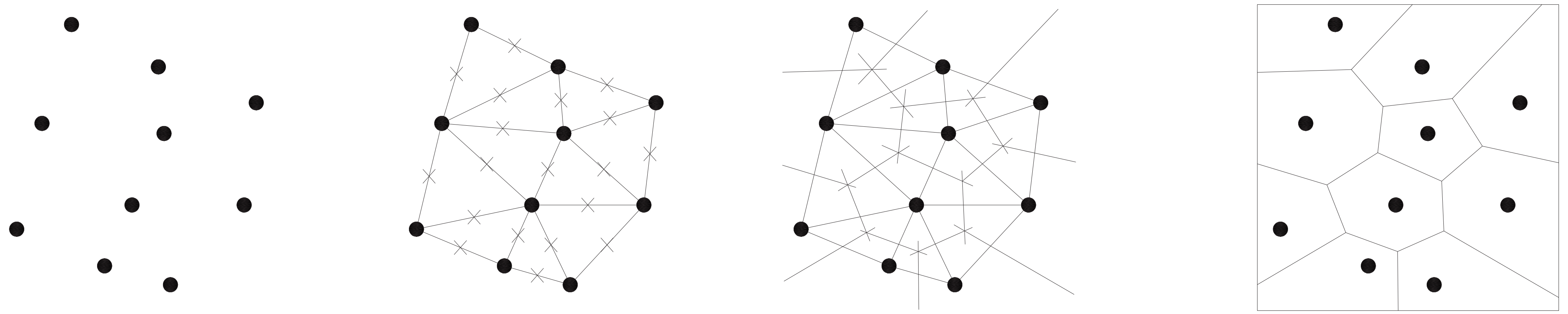


Fig.13: Development of a voronoi structure from a populated area of points.

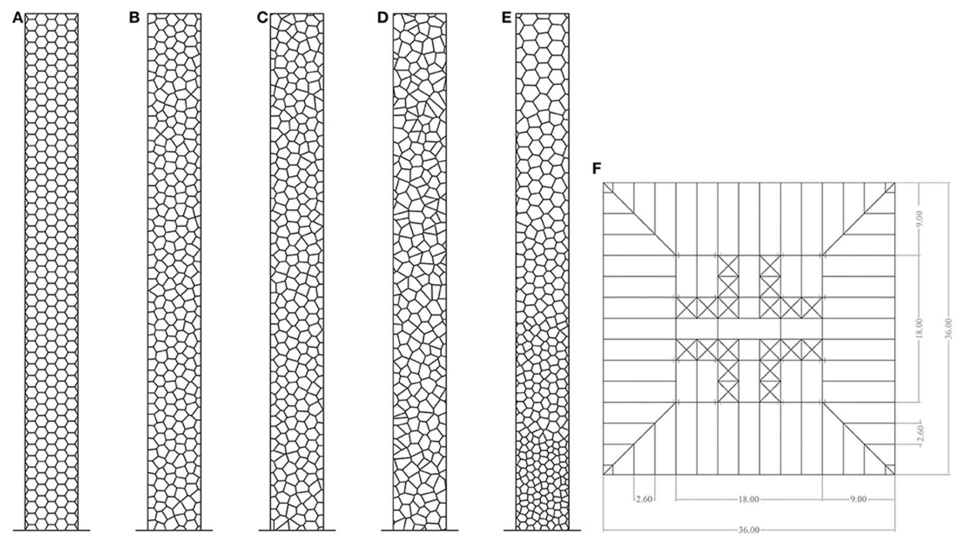


Fig.14:Research showing less material needed for a voronoi structure compared to a grid formation. Angelucci, G. and Mollaioli, F. (2018)

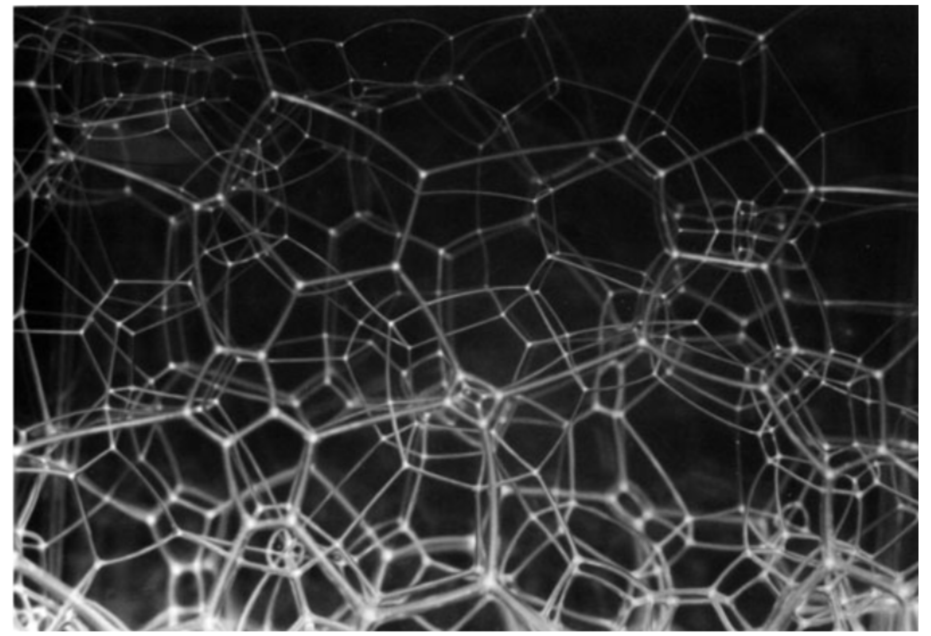
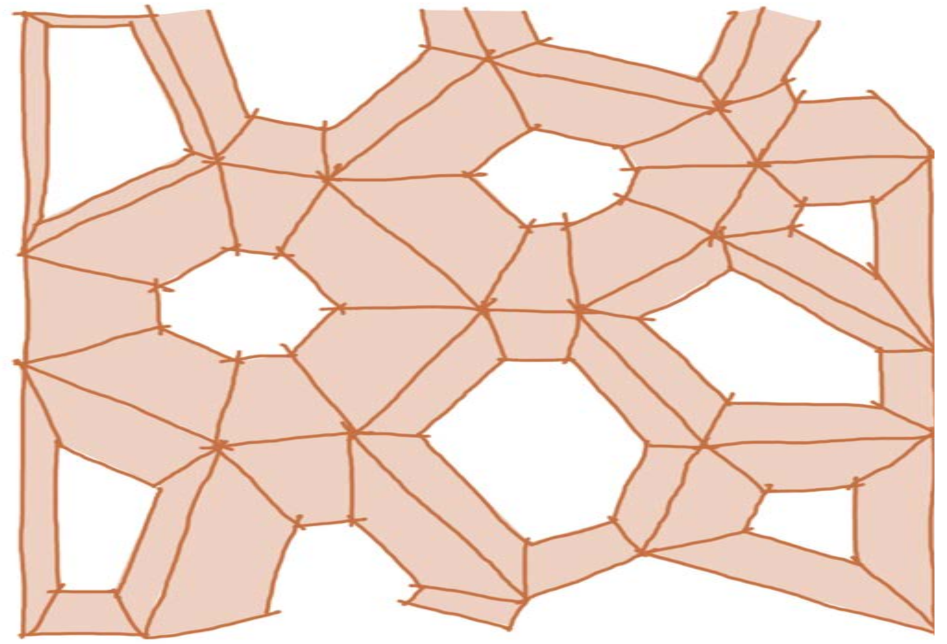
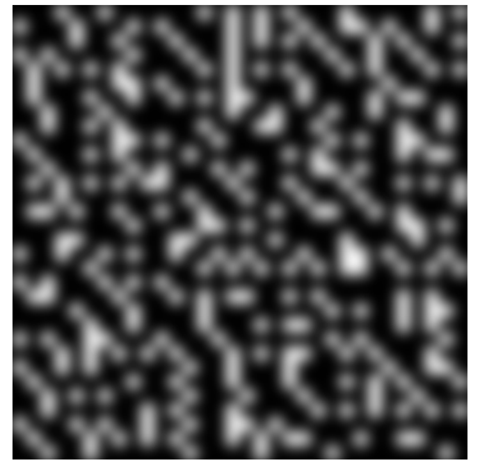


Fig.15: Structure of stacked bubbles demonstrating the strength of a voronoi structure. Ball, P (2009).

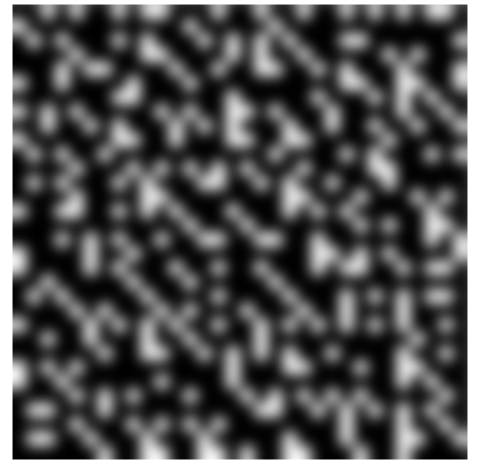
VORONOI
Logic

By using this method of packing I will create windows from the creation of the Voronoi cells the size will be determined by the previous Grid algorithm results. This will take effect as an attractor with the lighter areas having the most impact on the size of the openings. This can be seen within the logic provided.





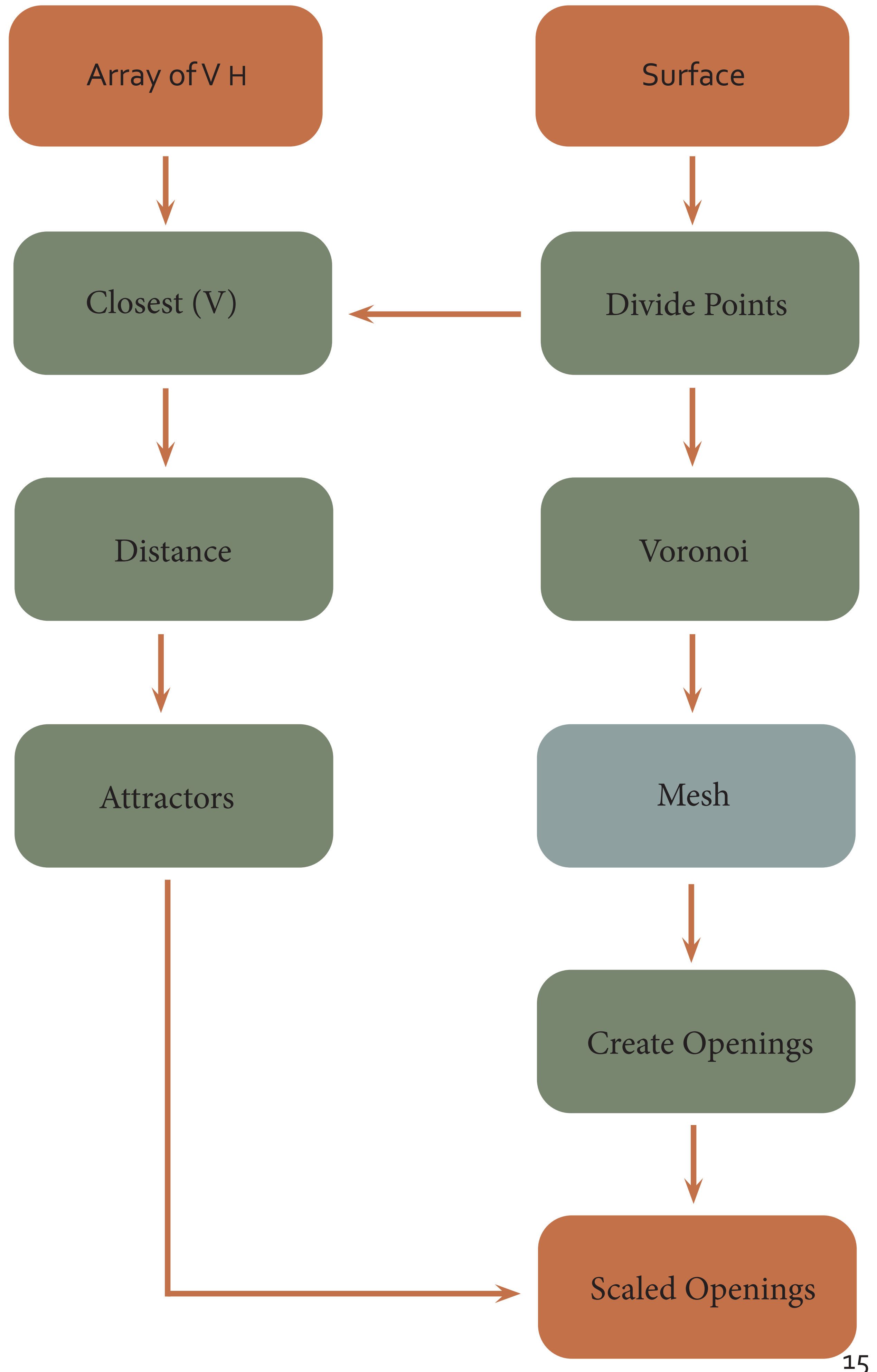
Surface Algorithm result 1



Surface Algorithm result 2

Surface Algorithm result 3

Surface Algorithm result 4



VORONOI RESULTS

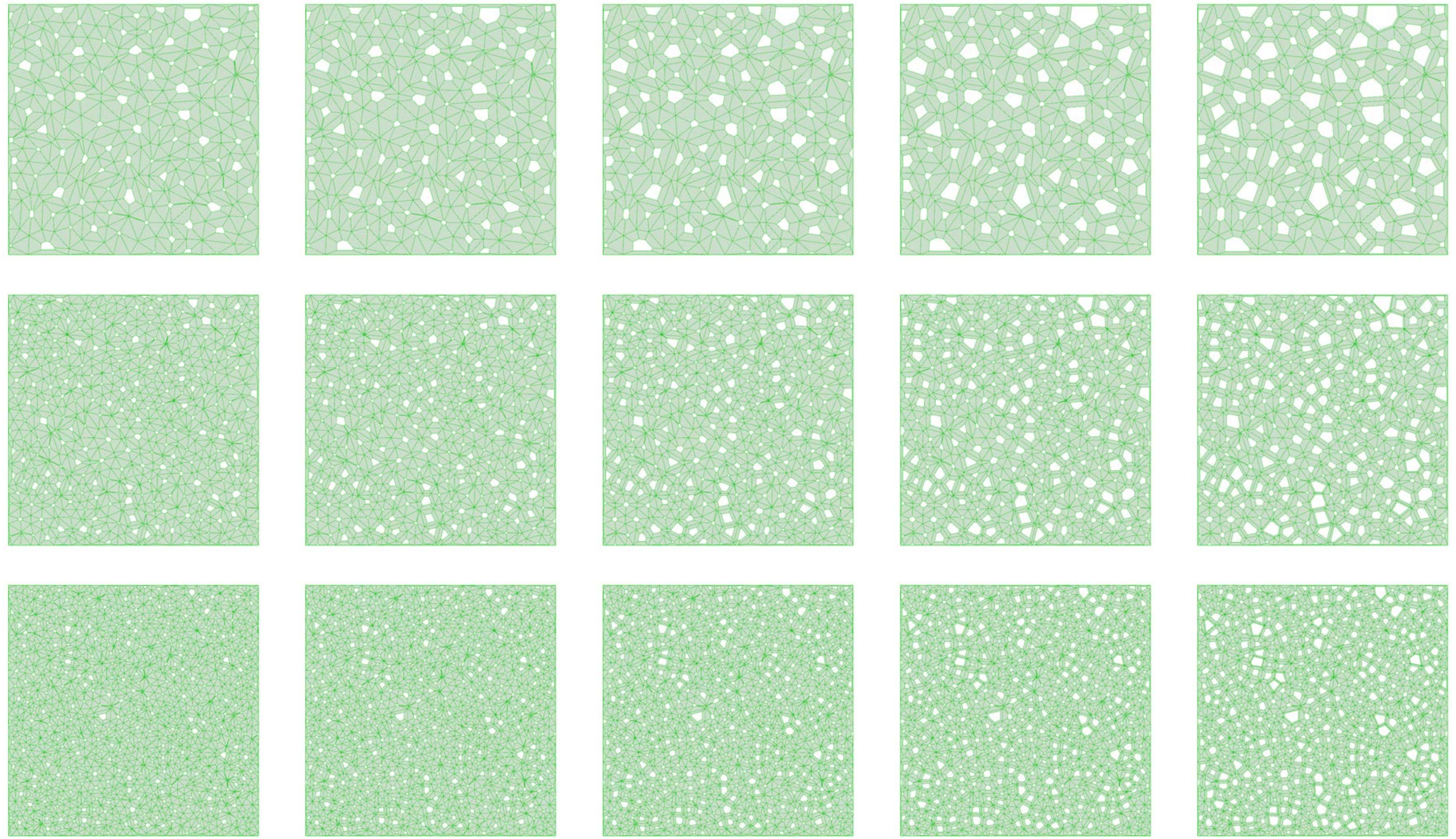
Grid 1

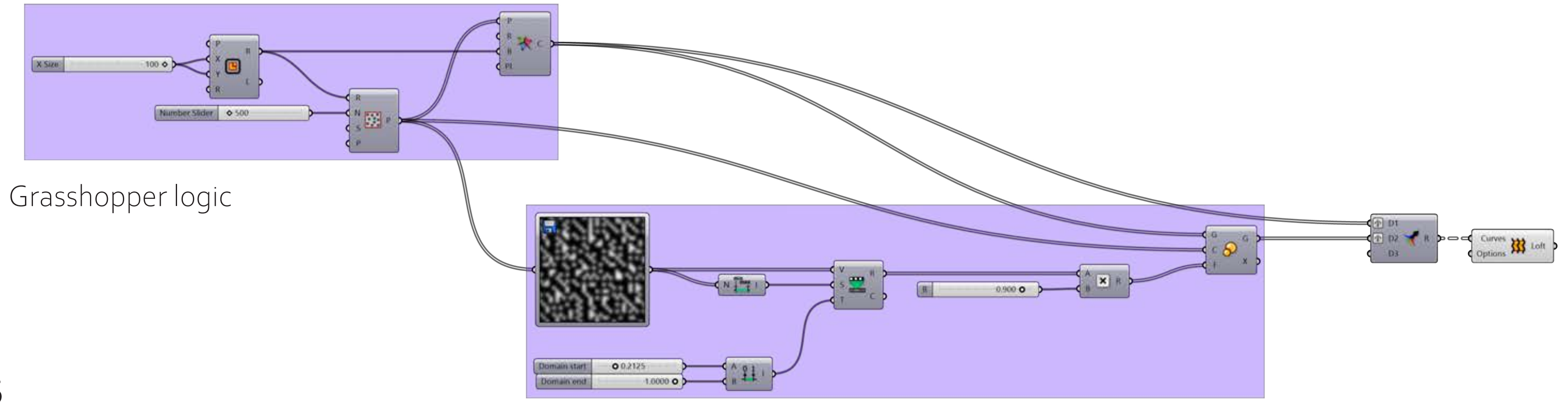
0.5 Scale | 0.6 Scale | 0.7 Scale | 0.8 Scale | 0.9 Scale

100 Cells

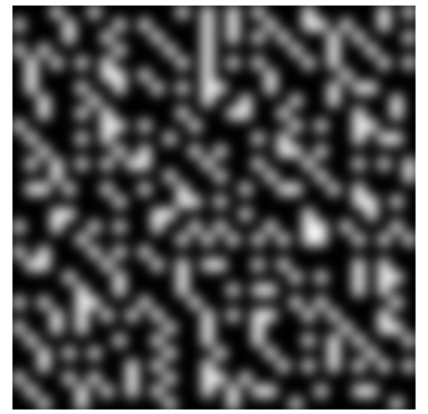
250 Cells

500 Cells





Grasshopper logic



Surface Algorithm result 1

VORONOI RESULTS

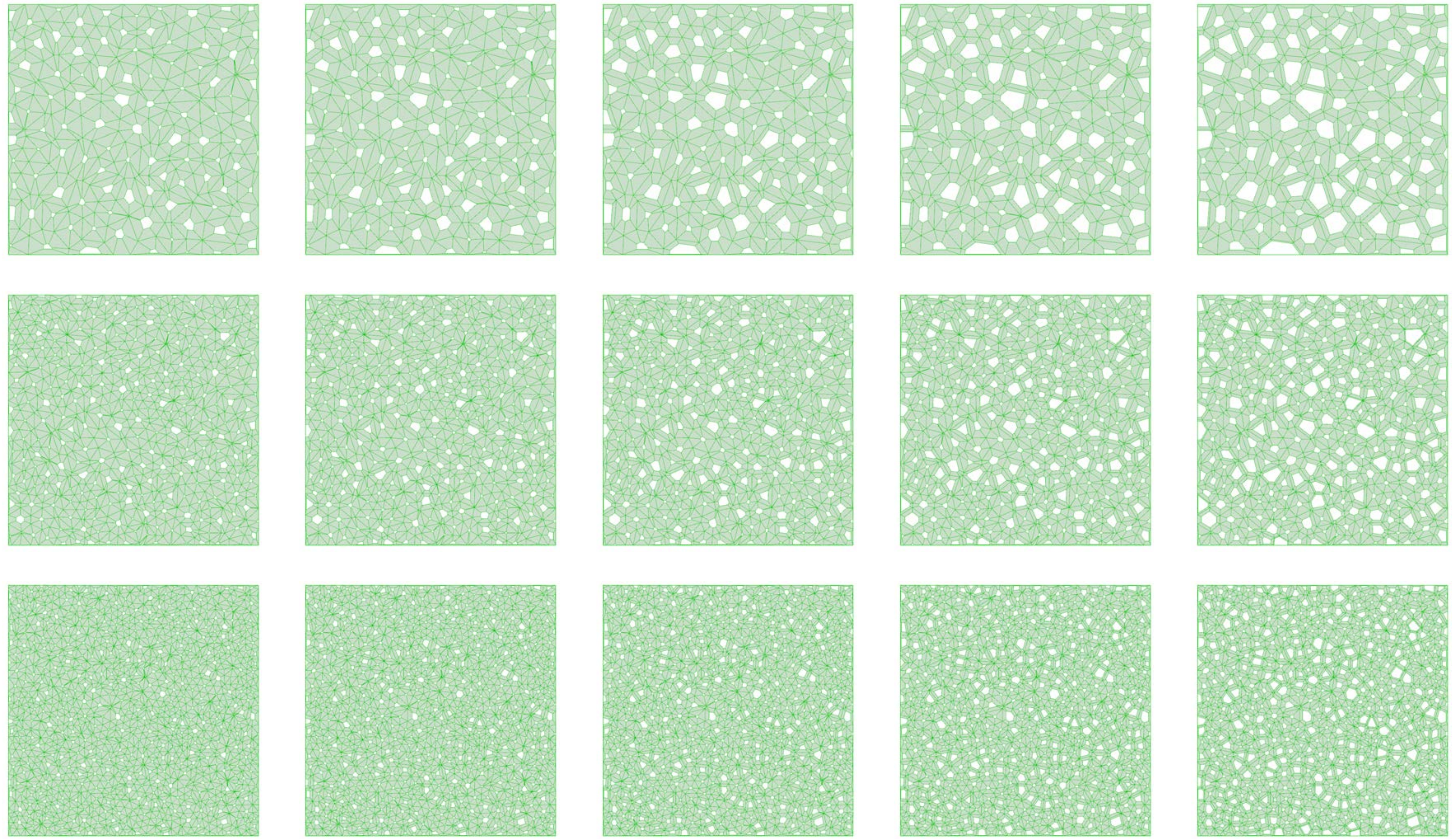
Grid 2

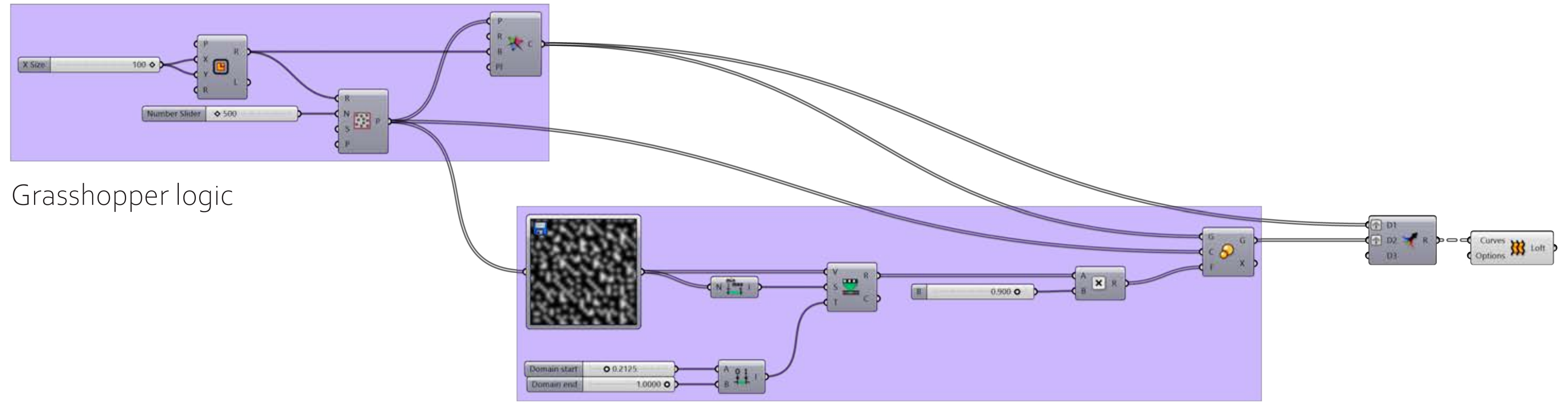
0.5 Scale | 0.6 Scale | 0.7 Scale | 0.8 Scale | 0.9 Scale

100 Cells

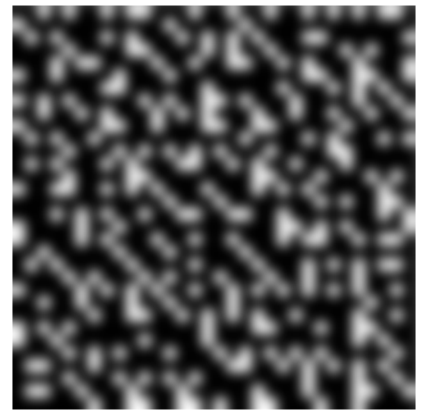
250 Cells

500 Cells





Grasshopper logic



Surface Algorithm result 2

VORONOI RESULTS

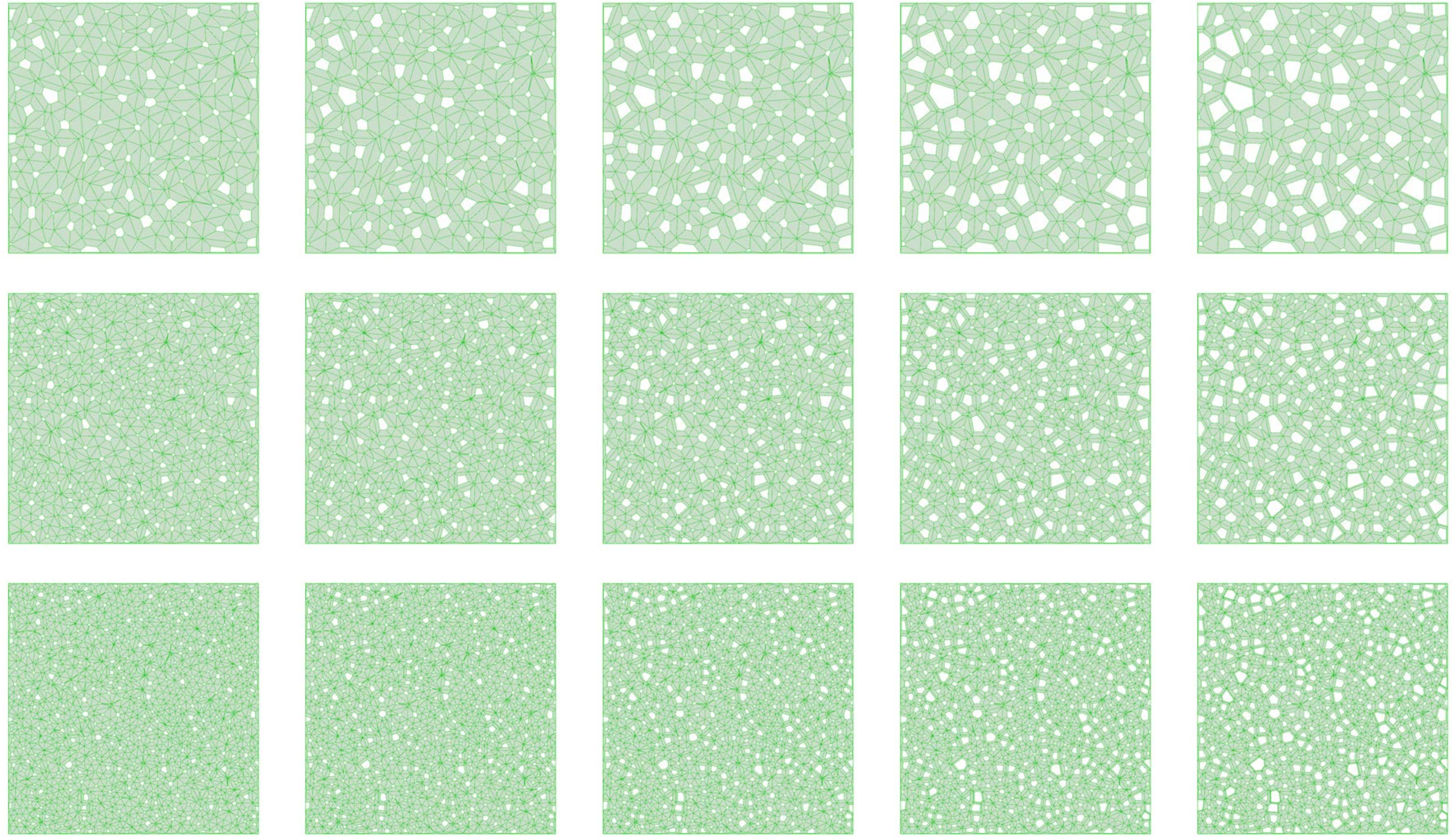
Grid 3

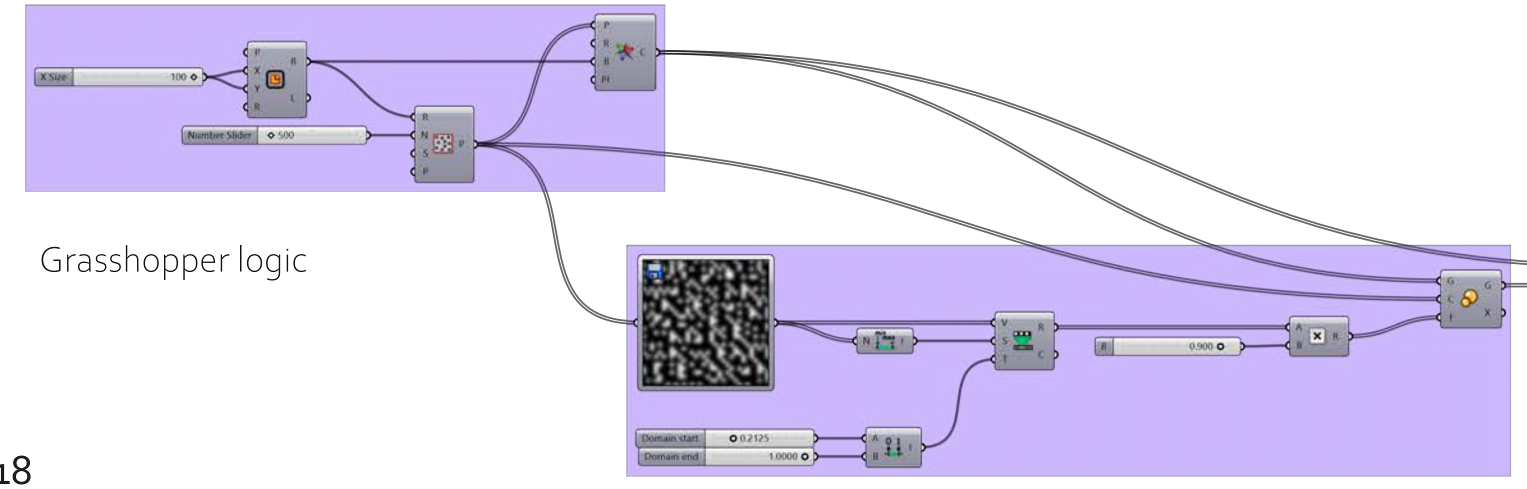
0.5 Scale | 0.6 Scale | 0.7 Scale | 0.8 Scale | 0.9 Scale

100 Cells

250 Cells

500 Cells





Grasshopper logic

Surface Algorithm result 3

VORONOI RESULTS

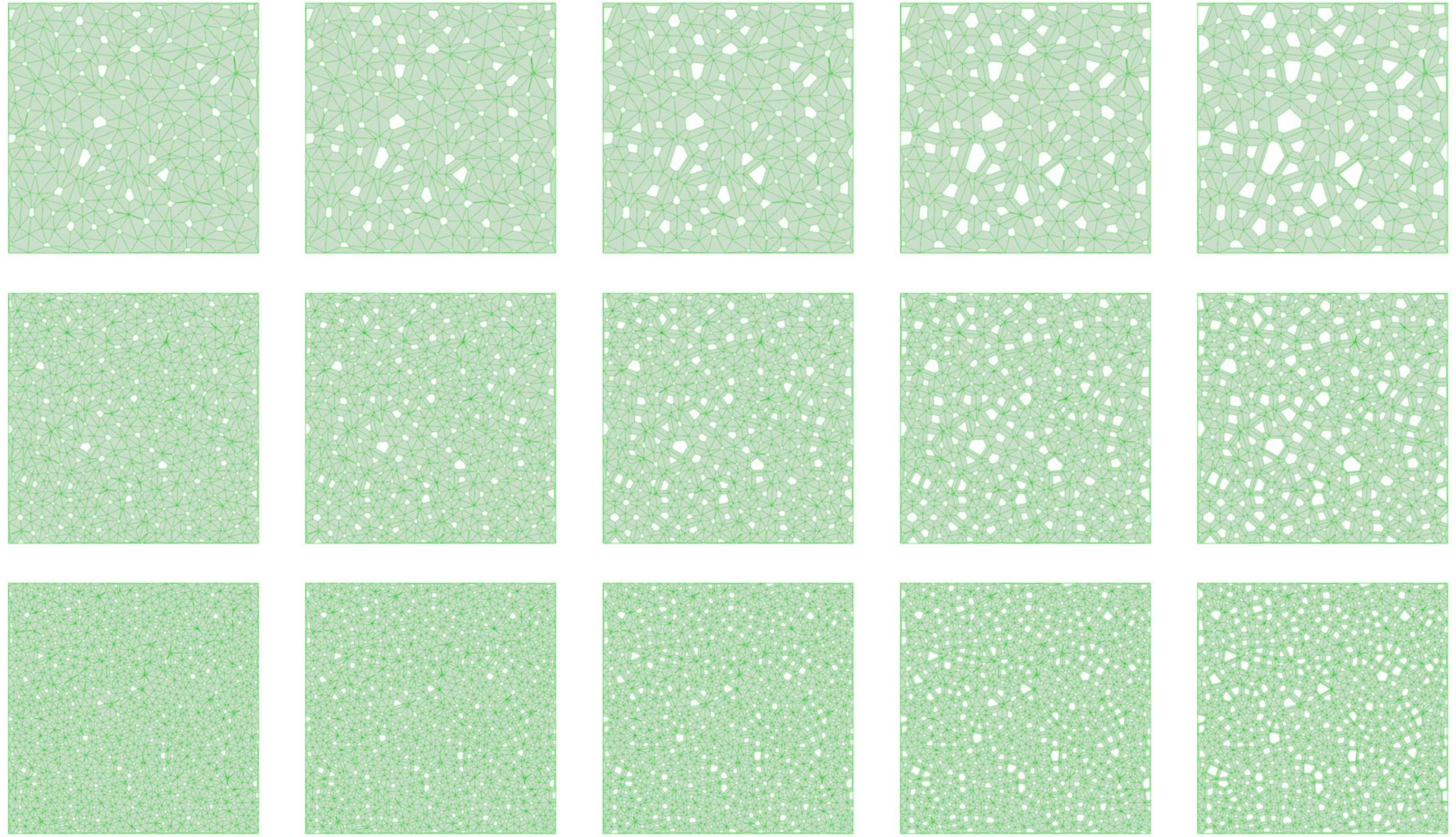
Grid 4

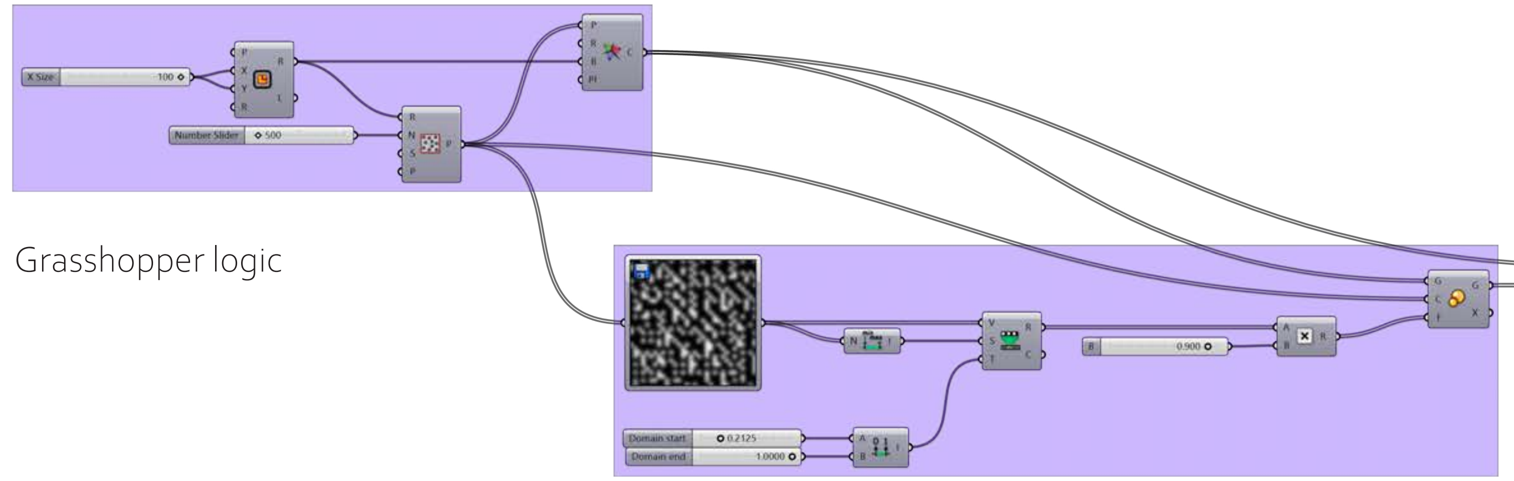
0.5 Scale | 0.6 Scale | 0.7 Scale | 0.8 Scale | 0.9 Scale

100 Cells

250 Cells

500 Cells





Grasshopper logic

Surface Algorithm result 4

FABRICATED MODEL
Images

The model has been created from the Grid 1 Voronoi result of 250cells/0.8 scale. This was then 3D printed as this was the most efficient method for model fabrication as there are many openings to account for. The model came out very sturdy due to the natural packing function of the Voronoi system, if this was to be fabricated to a further level, I would have used concrete to test the compressive strength of the structure.


Opening view 1

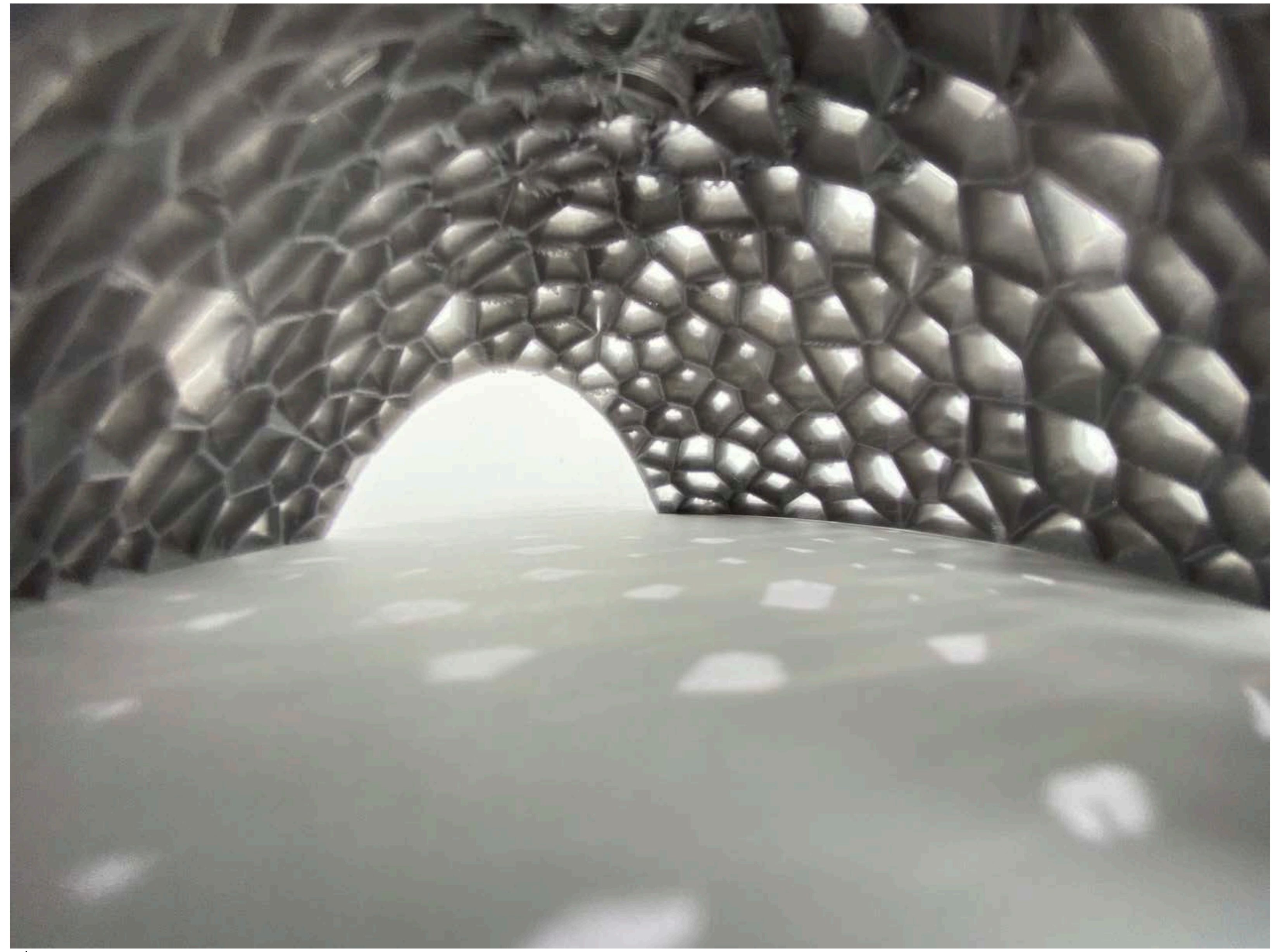

Side view 1


Aerial View


Opening view 2

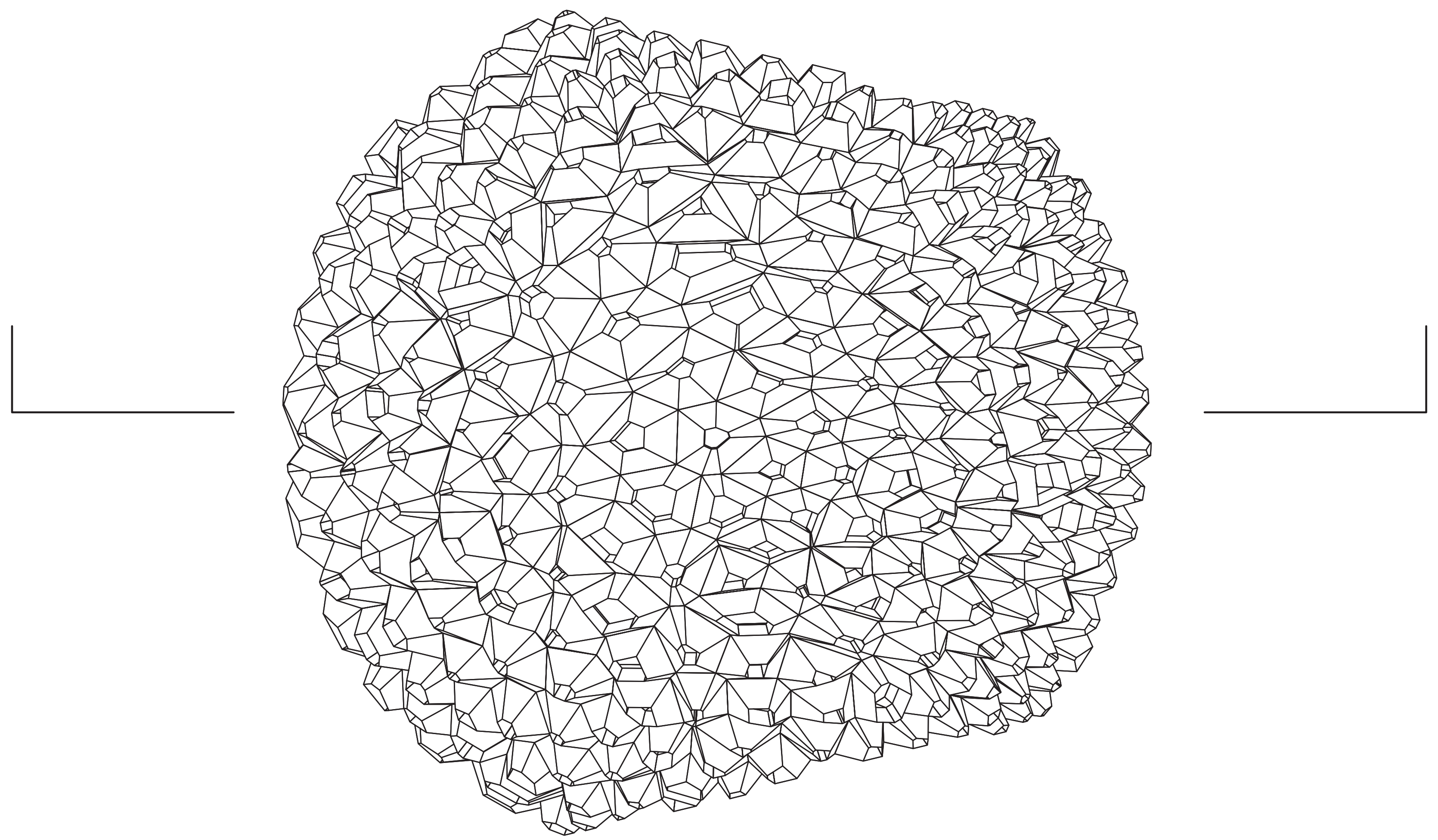

Side view 2

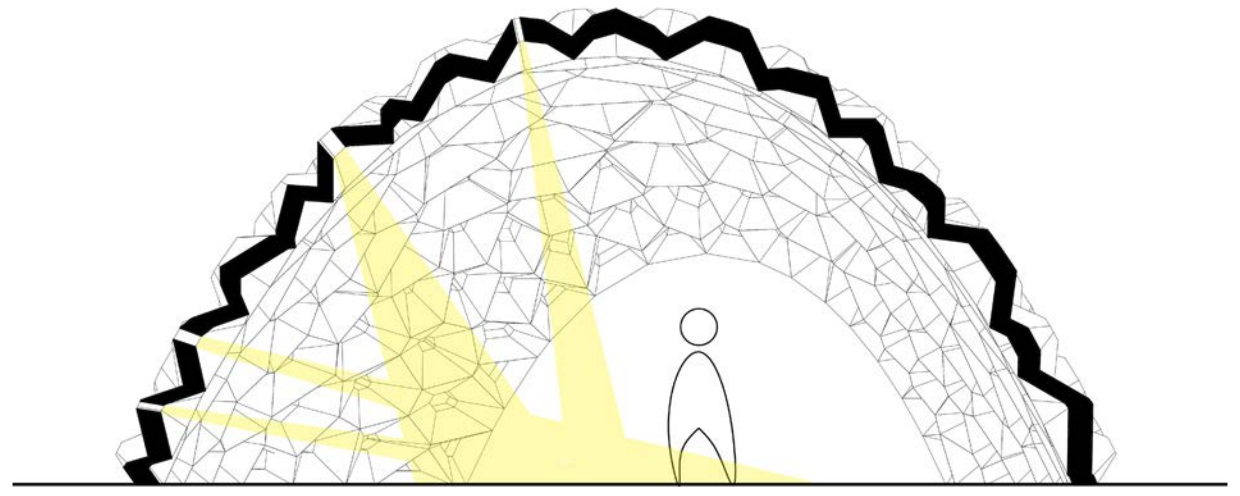

Opening view


Interior perspective view

FABRICATED MODEL
User Interaction

From the perspective of the viewer, the sense of viewing and privacy are synonymous with each other due to the scattered openings. The pavilion allows for a 360-degree view as birds will not only be seen in front of a building but flying around a surrounding habituating nearby nature.





CONCLUSION
A new future?

This module has given me a different outlook on computational design. Previously I thought it was just a shortcut to creating a parametric structure, but completing the module computational design helps develop a narrative within design allowing designers and architects to research different aspects of a design that would be impossible without these functions.

In the realisation of L-systems, they are one of the fundamentals of computational architecture and pattern algorithms. I look forward to using these methods in my future design developments.

REFERENCES

Angelucci, G. and Mollaioli, F. (2018) Voronoi-like Grid Systems For Tall Buildings. Frontiers in Built Environment [online]. 4 [Accessed 25 March 2023].

Ashlock, D., Gent, S. and Bryden, K. (2005) Evolution of L-systems For Compact Virtual Landscape Generation. Proceedings of the Iee Congress on Evolutionary Computation, Cec [online]., pp. 2760-2767. [Accessed 11 February 2023].

Ball, P. (2009) Nature's patterns : a tapestry in three parts. Shapes. Oxford: Oxford University Press.

Ball, P. (2011) Branches: Nature's Patterns: a Tapestry in Three Parts. Oxford: Oxford University Press.

Eglash, R. (1999) African Fractals: Modern Computing and Indigenous Design [online]. New Brunswick, NJ: Rutgers University Press. [Accessed 08 February 2023].

Prusinkiewicz, P. (1990) The Algorithmic Beauty of Plants. Springer-verlag. New-York.

Santell, J (2019) L-Systems. Available from: https://jsantell.com/l-systems/ [Accessed 11 February 2023].

Terzidis, K. (2006) Algorithmic architecture. Princeton: Architectural Press.

IMAGE REFERENCES

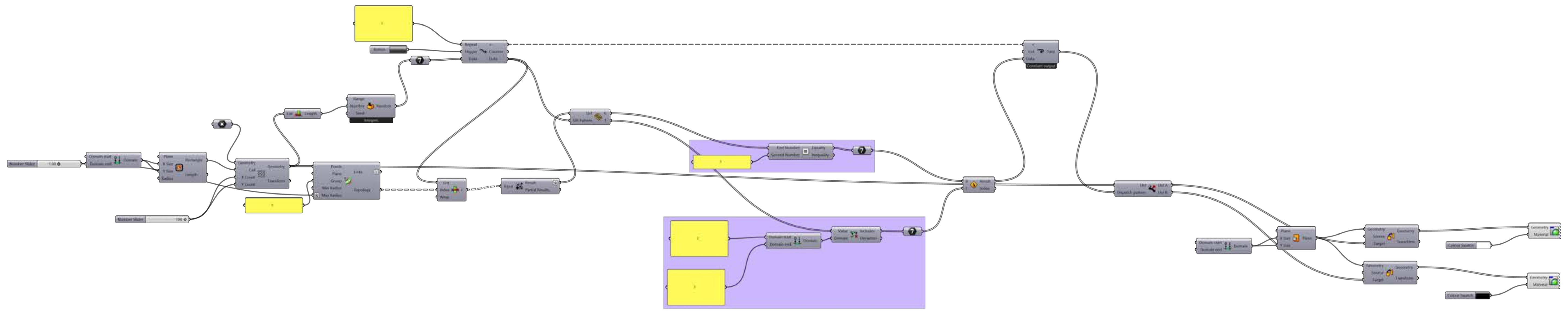
Fig. 01: Solkær,S (2020). Søren Solkær starlings. [Photo]

Fig.02: Bartoshesky, H (2021). Sierpinkski Arrowhead. Available from: https://www.hbartoshesky.com/l-system-page#results-L-system [Accessed 29 March 2023].

Fig. 05: Santell, J (2019). Stochastic L-systems. Available from: https://jsantell.com/l-systems/ [Accessed 11 February 2023].

APPENDIX

Cellular Automata



APPENDIX

Final Pavilion Structure