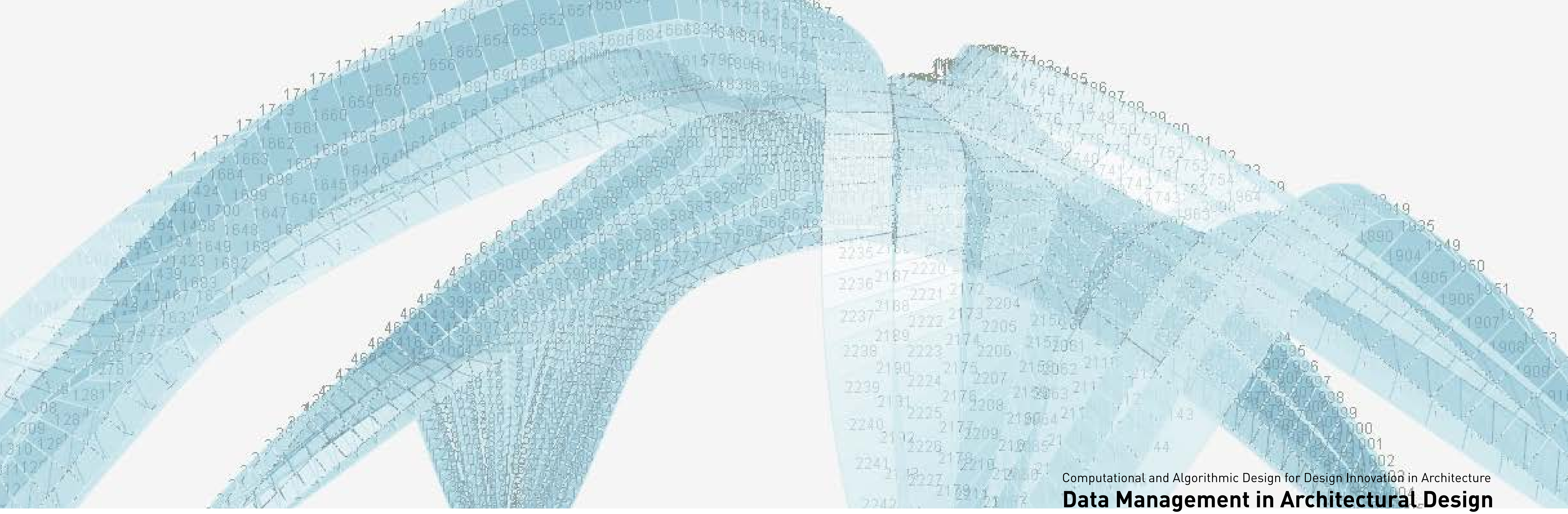


Computational and Algorithmic Design for Design Innovation in Architecture

Data Management in Architectural Design

H Architecture

Computational and Algorithmic Design for Design Innovation in Architecture

Data Management in Architectural Design

Part 1

H Architecture

This research paper was written and edited by Dongil Kim and Jake Jaekyung Han at H Architecture. 2017.

Computational and Algorithmic Design for Design Innovation in Architecture

Data Management in Architectural Design

Contents

Data, Information and Architecture

Introduction

Data is simply facts or figures — bits of information, but not information itself. When data is processed, interpreted, organized, structured or presented so as to make them meaningful or useful, they are called information. Information provides context for data. The careful sorting process can be a huge part of the design process in contemporary architectural practice.

This is an introduction to computational design and the prerequisite for more advanced topics in the field.

The "Data, Information and Architecture" introduces architects and designers to fundamental concepts and techniques in computational design. By the term "computational design" we mean a series of methods borrowed from computer science, computational geometry and other fields, and adapted to specific design problems such as form generation, design development, fabrication and analysis.

The fact that most design related fields [structural engineering, environmental engineering, fabrication etc..] rely increasingly on digital tools opens up the possibility for a cross disciplinary integration of techniques and data sets. This ability of numerical models to operate between disciplines will be addressed in this course both in terms of analysis of results coming from other fields as well as generation of appropriate outputs.

The goal is below:
First is to help architects and designers develop the skills necessary for creating or manipulating computational solutions for specific design problems. The understanding of the data structure and management is fundamental.

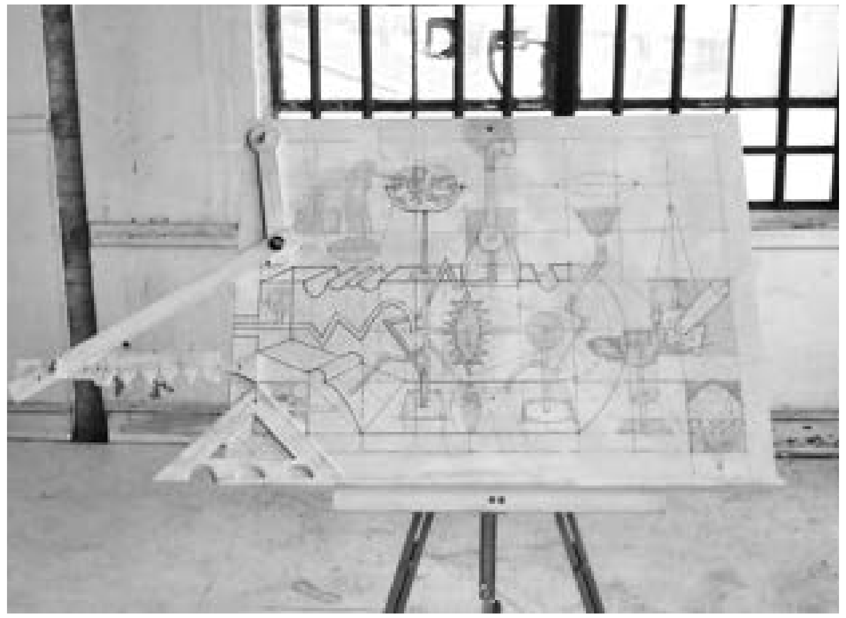
Second is to understand geometry generation and manipulation, the analysis of data from external sources, output of information and design evaluation. It covers a series of state of the art development of codes in visual programming language.

Architects and designers are expected to acquire some hands on experience in programming as this is the craft that underpins computational design through a set works. This will guide the advanced user of visual programming language, such as Grasshopper, (towards using a higher level of numerous data) in contemporary society.

Different sets of data management will be introduced with simplified principles and advanced examples. These will be given by using programming language, mainly Grasshopper, Python and C#. Architects and designers will be able to customize their process depending on their interests.

Mathematical concepts will be introduced in simple terms as needed. The first chapter is dedicated to building a solid understanding of data structures as well as introducing fundamental concepts from analytical and computational geometry.

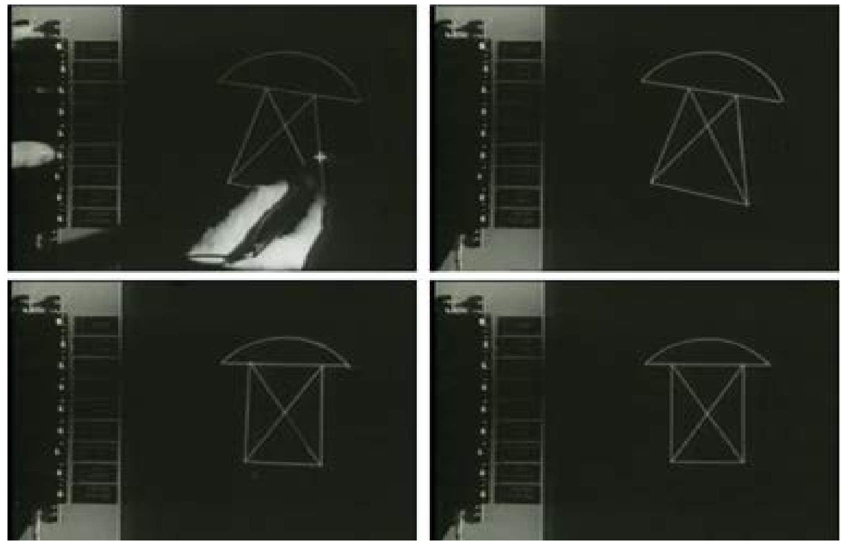
1901 – The "Drafting Machine" was patented for technical illustrations that became popular in the Industrial Revolution. It typically contained tools for accurate drawing like a protractor and compass.



Ivan Sutherland, Sketch pad, 1962

Computerization vs Computation

Informed Design Process for Design Innovation in Architecture



Architects and designers often confuse computerization and computation.
Computation is the procedure of calculating data that determines something by mathematical or logical methods, while computerization is the act of entering, processing, or storing information in a computer or a computer system. Computerization is about automation, mechanization, digitization, and conversion. Generally, it involves the digitization of entities or processes that are preconceived, predetermined, and well defined. In contrast, computation is about the exploration of indeterminate, vague, unclear, and often ill-defined processes; because of its exploratory nature, computation aims at emulating or extending the human intellect. It is about rationalization, reasoning, logic, algorithm, deduction, induction, extrapolation, exploration, and estimation. In its manifold implications, it involves problem solving, mental structures, cognition, simulation, and rule-based intelligence, to name a few.

Increased/ improved accessibility of data and information in all industries including architectural practice has changed contemporary society. Many social, environmental and especially structural and force data have now become crucial design resources in the algorithmic design and thinking process. Architects and designers should be equipped with the theoretical understanding and practical application of data through computational design.

Data as Design Driver

There is a series of information that has recently become increasingly available for designers and architects.
The first type is environmental data that has emerged with the increased popularity and importance of energy conservation in the 21st century. Honeybee and Ladybug are pioneering interfaces that allow for the simulation of various energy data such as daylighting and thermal aspects within the built environment.
Secondly structure and force data have become increasingly significant in the design process. And now designers use them as main factors and design drivers. Karamba a structural design tool and Kangaroo, a form finding tool, are the most prominent softwares used by architects and designers. The availability allows for the simplification of the design process that traditionally required the information to transition back and forth.

Lastly, the understanding of advanced geometry and digital fabrication methods is key in both design development and construction phase. Numerical fabrication systems and robotics within the architectural production industry allows architects and designers to adapt and accommodate it into the whole design and production process. This usually brings about optimization and efficiency in the construction phase.

The understanding of data and management becomes a crucial part of the design process. It can be described as below;

* Innovation of Initial Design by Information

* Efficient Process of Design in constant changes

* Analysis and Proof of Design for optimization

* Process of Production in Drawing and Construction phase

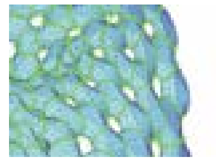
Top: Geco Grasshopper Simulation
ICD/ITKE Research Pavillion 2016 Structure Model
The Tower by CITA Interior
ICD/ITKE Research Pavilion 2016 Robotic Sewing Process

Environmental Data

Environmental Information Based Design

Rhinoceros + Grasshopper ——— Honeybee + Ladybug

Environmental Based Initial Massing in Pre-schematic
Facade system Analysis in Schematic and Design Development Phase


Physic Data

Structural Information Based Design

Rhinoceros + Grasshopper ——— KARAMBA 3D
Revit + Dynamo

Initial massing and Integrative Design System for parametric structure
in Pre-schematic & Schematic Phase

Material Data

Force Information Based Design

Rhinoceros + Grasshopper ——— Kangaroo

Column Free roof Structure, Stadium, Terminal, Airport,
Facade, Interior Feature in Pre-schematic & Schematic Phase

Urban and Geographical Information Based Design

Rhinoceros + Grasshopper ——— GIS, Local Code, Elk

Urban Design in Pre-schematic & Schematic Phase

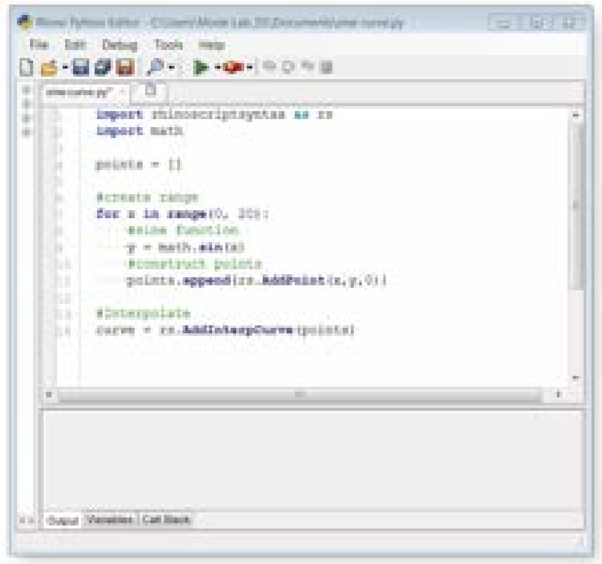
Process Data

Advanced Geometry Analysis in Design

Rhinoceros + Grasshopper ——— Lunch Box
Revit + Dynamo

Optimization and Realization for Paneling of Complex Geometry in Design Development & Construction Document Phase

Top: Python Interface
Bottom: Grasshopper Interface

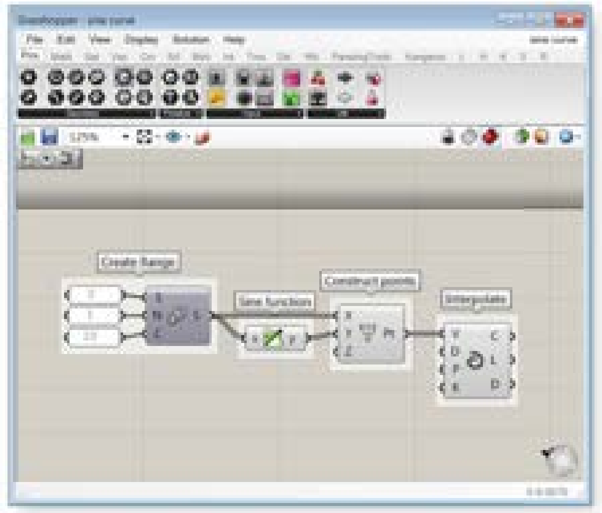


```
import rhinoscriptsyntax as rs
import math

points = []

#create range
for x in range(0, 20):
    #sine function
    y = math.sin(x)
    #construct points
    points.append(rs.AddPoint(x,y,0))

#Interpolate
curve = rs.AddInterpCurve(points)
```



Visual Programming Language

A visual programming language (VPL) is any programming language that allows users to create programs by manipulating program elements graphically rather than by specifying them textually. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols, used either as elements of syntax or secondary notations. For example, many VPLs (known as data-flow or diagrammatic programming) are based on the idea of "boxes and arrows", where boxes or other screen objects are treated as entities, connected by arrows, lines or arcs which represent relations. Grasshopper is the most popular visual programming language that can be used within the Rhinoceros environment. The combination of Grasshopper and Rhino allows us to have and define precise parametric control over models, creating exploratory design opportunities. Architects and designers can explore generative design work flows and higher levels of programming logic within a intuitive and graphical interface. The major features are listed below;

1. Graphic Algorithm Editor

2. Algorithm is a step by step procedure
   designed to perform an operation

3. Designing in Algorithms in Grasshopper
   that then automate tasks in Rhinoceros 3D

4. Manually and incrementally creating an algorithm
   using Rhinoceros 3D Environment and Re-writing the code.

Top Right: Munich Olympic Stadium Model at ILEK+Frei Otto
Bottom: Converted Gravity Model by Antonio Gaudi

British Museum Roof Structure, Poster & Partners

Energy & Form

Design Development, Analysis & Fabrication

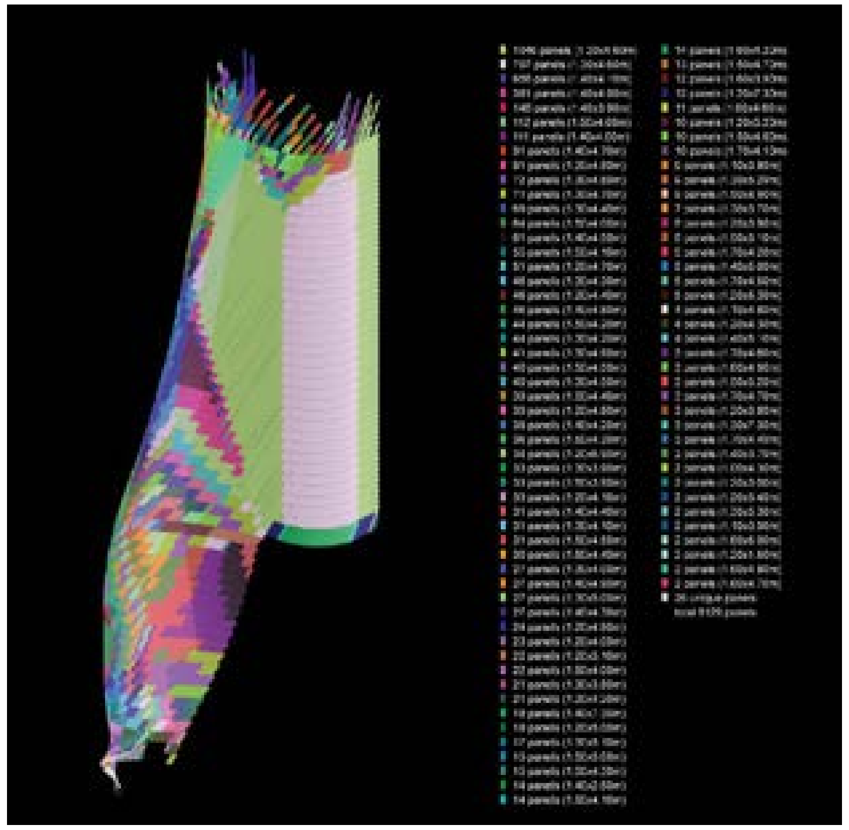
The purpose of this section is to introduce theoretical and technological investigations of synthetic morphologies from historical projects, which incorporate both external forces and intrinsic material logic, and to further develop them through contemporary computational design tools.

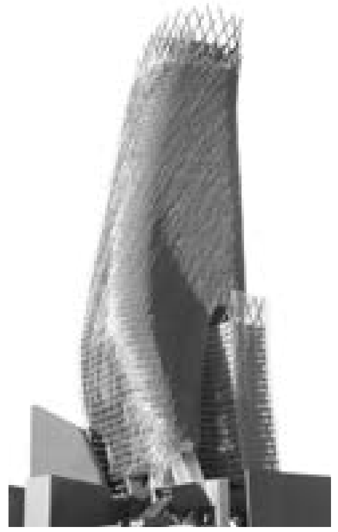
There have been rigorous investigations to design and discover novel shape formation strategies, involving synthetic design processes of physical force, material properties, and constraints in the making process in regards to optimizing material and structural redundancy. The key elements of Gothic architecture such as vaults, buttresses and arches are of the most seminal examples for the methodological integration of force and matter into formal and structural expression.

These are crucial examples of computation with physical forces. Under the spirit of synthetic formation of Gothic Architecture, Antonio Gaudi had explored advanced design methodologies for the physical visualization of force by using scale models made of chains and weighted strings for his major projects, such as the unfinished Church of Colonia Guell and Sagrada familia. The hanging chain models and it's real construction is a great way to show how he used the forces and material properties as drivers for design and methodologies of making.

The challenge of developing synthetic formation was carried over to the next generation of architects such as Frei Otto, Felix Candela, and Eladio Dieste. Even though

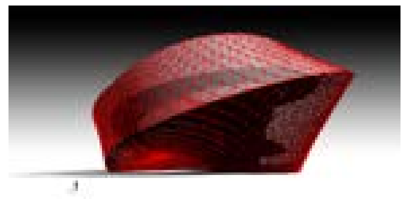
Top left: Morphosis Phare Tower Panelization Model,
Top right: Phare Tower Rendering

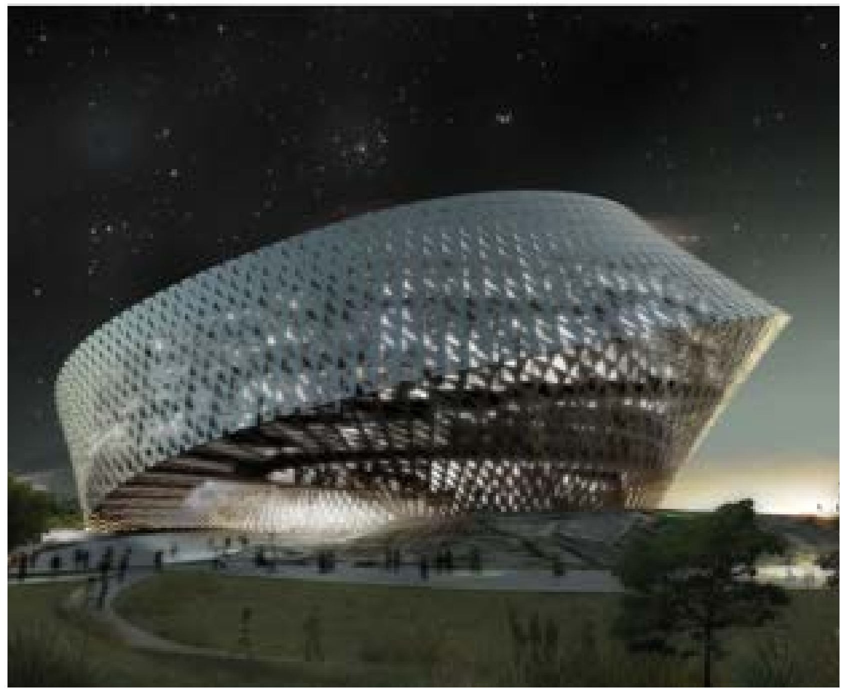




they each use distinctive material and geometries , their work has commonalities in that they implement a more advanced design method of force as the driver in the design process and within the material logic. Despite Gothic architecture's innovation in synthesizing material, structure, and geometry in design, the process required a significant amount of time and labor, and the construction process took several decades and even hundreds of years during that time. However, the development of computational tools and digital fabrication technology has been a game changer in this area. The application of modern computational design tools explore this topic, and simulate the force flow of more complex geometries. In addition, digital fabrication technology provides a more precise and fast method of creating systems. The roof structure of the British Museum by Norman Foster represents the very optimized solution of form that incorporates form, structure, and method of making. The grid shell is a fusion of the state-of-the-art engineering and economy of form. Its unique geometry is designed to span the irregular gap between

Left: National Library in Astana Structure Modeling by BIG
Right: National Library in Astana Rendering by BIG





the drum of the Reading Room and the courtyard facades, and forms both the primary structure and the framing for the glazing, which is designed to reduce solar gain. The "Phare Tower" by Morphosis embodies and reflects the information from thermal and sun vector data through the envelope system in order to minimize solar gain. BIG's National Library in Astana, Kazakhstan became a primary precedent for facade optimization using thermal data.

Unlike in engineering and science, in architecture, projects that utilized data and information as a design driver have not been widely considered nor recognised. However, the recent development of computational tools give designers an opportunity to explore deeper into force and material-based design methodologies. The technological efficiency of this type of design process requires lower energy costs, minimal waste in assemblage, less material, and reduced time. Through this work ,"Data, Information and Architecture", one can expect a wide-range of applications towards their designs that operate under similar systems of design procedures.

Fundamental

Fundamental

This section introduces the fundamental knowledge for advanced computational design. It includes understanding data structures such as 'list' and 'tree management'. And covers ways to generate mesh. It starts with generating meshes from points, from 2 curves and then introduces Lunchbox which is a highly developed NURBS surface based panelization plug-in by Nathan Miller at Proving Ground.

Secondly, the relationship between the sun and the object demonstrates the vector and the way the vector is used in both the geometry itself and the environmental data system. Ladybug and Honeybee are key background plug-ins for environmental analysis and representation.

Lastly, the spring particle system demonstrates the process of integrated and interdependent modeling system. Physic analysis and simulation achieved by Kangaroo Physics, developed by Daniel Piker, and SoFistik is perfect for point and vector based interdependent data systems.

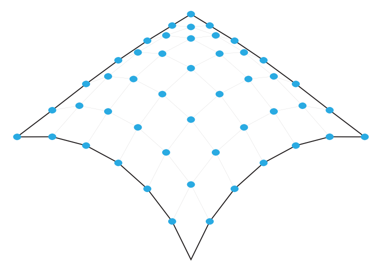
These three chapters lead designers and architects to explore informed design processes within complex data flow systems.

Introduction of Mesh

In the field of computational modeling, meshes are one of the most pervasive forms of representing 3D geometry. Mesh geometry can be a light-weight and flexible alternative to working with NURBS, and are used in everything from rendering and visualizations to digital fabrication and 3D printing. This chapter will provide an introduction to how mesh geometry is handled in Grasshopper. A mesh is a collection of vertices, edges, and faces that describe the shape of a 3D object.

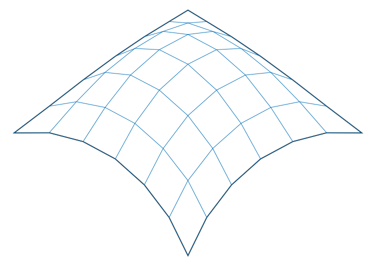
Vertice

A vertex is a single point. (The plural of vertex is "vertices").



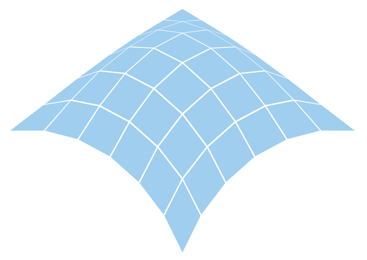
Edge

An edge is a straight line segment connecting two vertices.



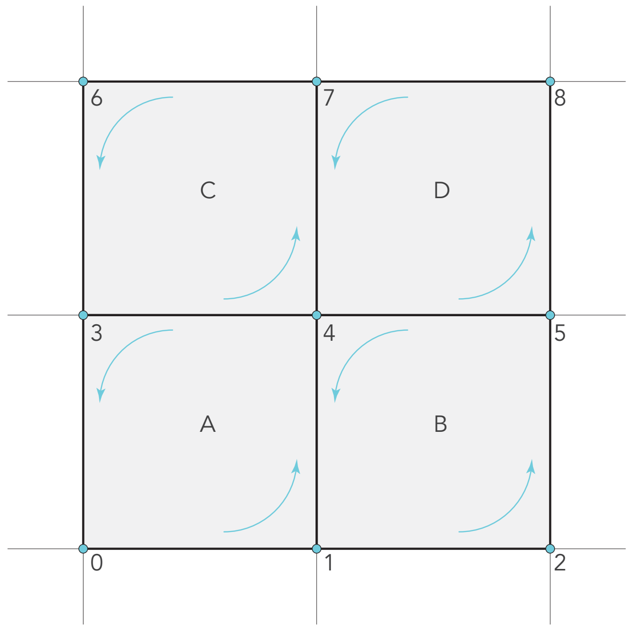
Face

A face is a flat surface enclosed by edges. (Some other applications call these "polygons")



Basic Anatomy of a Mesh

Grasshopper defines meshes using a Face-Vertex data structure. At its most basic, this structure is simply a collection of points which are grouped into polygons. The points of a mesh are called vertices, while the polygons are called faces. To create a mesh we need a list of vertices and a system of grouping those vertices into faces.
The vertices of a mesh are simply a list of points. Recall that a list in Grasshopper is a collection of objects. Each object in the list has an index which describes that objects position in a list. The index of the vertices is very important when constructing a mesh, or getting information about the structure of a mesh.
A face is an ordered list of three or four vertices. The "surface" representation of a mesh face is therefore implied according to the position of the vertices being indexed. We already have the list of vertices that make up the mesh, so instead of providing individual points to define a face, we instead simply use the index of the vertices. This also allows us to use the same vertex in more than one face.



VERTEX LIST	FACE LIST	MESH
0 (0,1,4,3)	A (0,1,4,3)	
1 (1,2,5,4)	B (1,2,5,4)	*MESH*
2 (3,4,7,6)	C (3,4,7,6)	
3 (4,5,8,7)	D (4,5,8,7)	

Mesh vs Nurbs

Parameterization
In the previous chapters, we saw that Nurbs surfaces are defined by a series of Nurbs curves going in two directions. These directions are labeled U and V, and allow a Nurbs surface to be parameterized according to a two-dimensional surface domain. The curves themselves are stored as equations in the computer, allowing the resulting surfaces to be calculated to an arbitrarily small degree of precision. It can be difficult, however, to combine multiple Nurbs surfaces together. Joining two Nurbs surfaces will result in a polysurface, where different sections of the geometry will have different UV parameters and curve definitions.

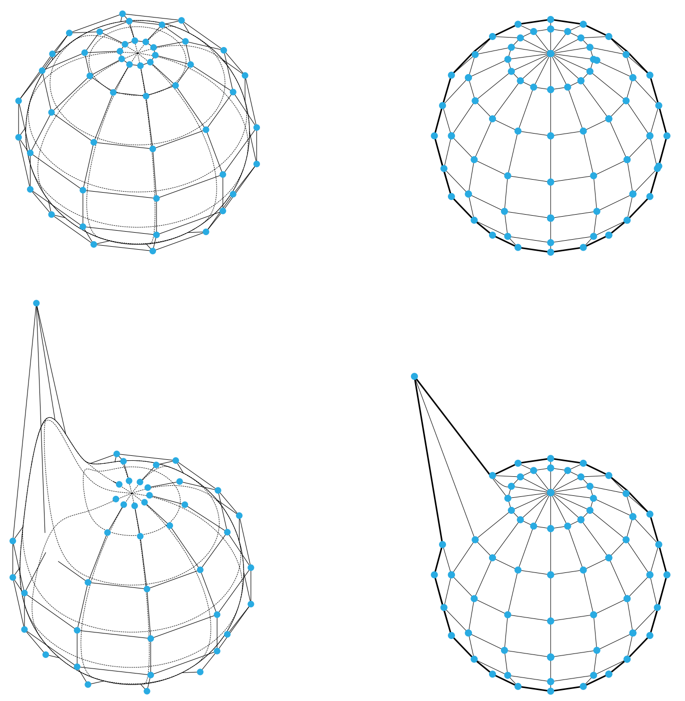
Meshes, on the other hand, are comprised of a discrete number of exactly defined vertices and faces. The network of vertices generally cannot be defined by simple UV coordinates, and because the faces are discrete the amount of precision is built into the mesh and can only be changed by refining the mesh and adding more faces. The lack of UV coordinates, however, allows meshes the flexibility to handle more complicated geometry with a single mesh, instead of resorting to a poly-surface in the case of Nurbs.

Note - While a mesh does not have implicit UV parameterization, it is sometimes useful to assign such a parameterization in order to map a texture or image file onto mesh geometry for rendering. Some modeling software therefore treats the UV coordinates of a mesh vertex as an attribute (like vertex color) which can be manipulated and changed. These are usually assigned and not completely defined by the mesh itself.

Local vs Global Influence

Another important difference is the extent to which a local change in mesh or NURBS geometry affects the entire form. Mesh geometry is completely local. Moving one vertex affects only the faces that are adjacent to that vertex. In NURBS surfaces, the extent of the influence is more complicated and depends on the degree of the surface as well as the weights and knots of the control points. In general, however, moving a single control point in a NURBS surface creates a more global change in geometry.

Zooming into a NURBS surface retains a smooth curve, while a mesh element has a fixed resolution
It is interesting to note that while NURBs surfaces are stored as mathematical equations, the actual visualization of these surfaces requires meshes. It is not possible for a computer to display a continuous equation. Instead, it must break that equation down into smaller parts, the result of which is that all rendering or display processing must convert NURBS to meshes. As an analogy, consider that even though we can store the equation of a line on a computer, in order to display that line, the computer must at some point convert the line into a series of discrete pixels on a screen to display.



Pros & Cons of Mesh

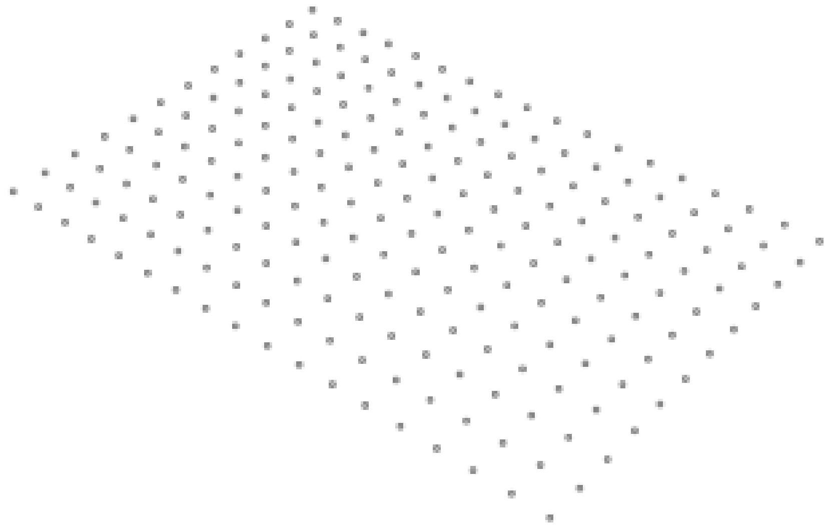
When we ask "What are the pros and cons of modeling with meshes?" we are really asking "What are the pros and cons of modeling with shapes that are solely defined by a set of vertices and a corresponding topological framework?" By framing the question this way, it is easier to see how the "simplistic" nature of a mesh is the critical aspect that would make a mesh either favorable or unfavorable to model with. Thus, whether to use a mesh will depend on the context of its application.

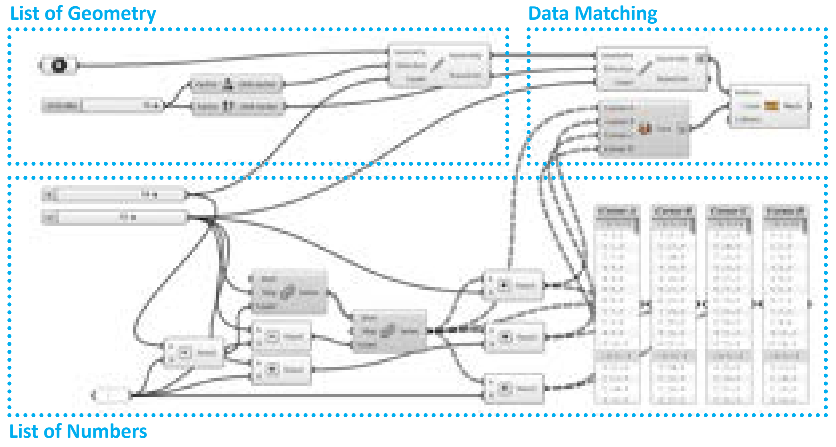
Meshes can be favorable in situations where:

* There must be an on going update of the rendering of a form that is changing in shape but not in face connectivity
* A discretized approximation of a curved geometry would suffice
* A low-resolution geometry must be systematically smoothed out (or articulated) using computational methods to arrive at a higher-resolution model.
* The low resolution model must be able to simultaneously support local and high resolution detail

Meshes can be unfavorable in situations where:

* Curvature and smoothness must be represented with a high level of precision
* True derivatives must be evaluated
* The geometry must be converted into a manufacturable solid
* The final form must be easily edited manually

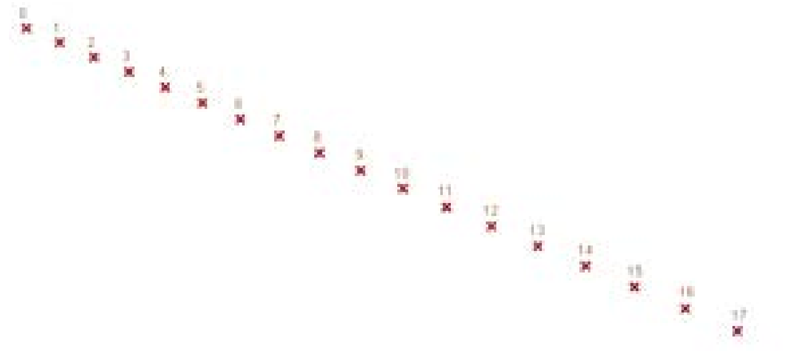


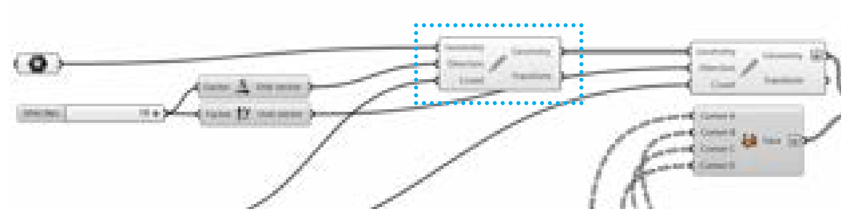


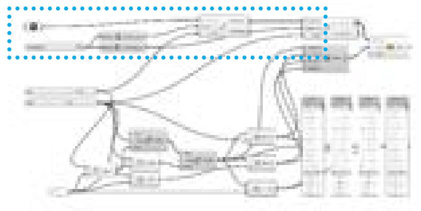
AP-01

Mesh from Point

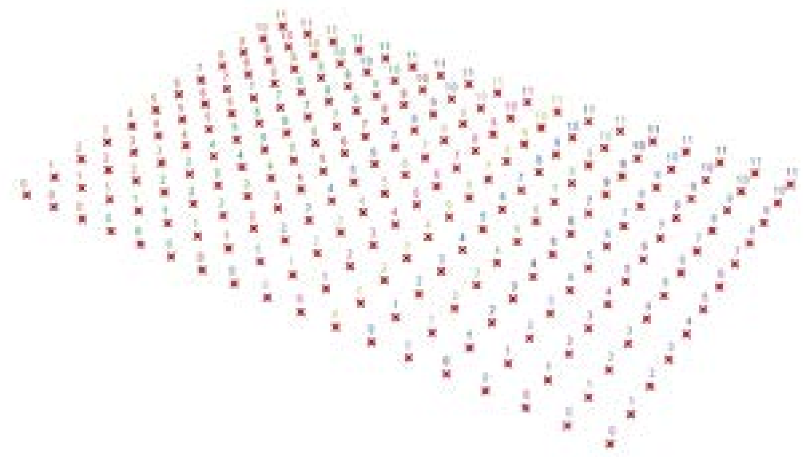
The physical & environmental simulation requires an understanding of mesh structure. This exercise introduces how to construct a mesh from single point with several parameters. The point is duplicated to another point with specific vectors given by an assigned parameter at the first of the code to create point matrix. Then, a rectangular mesh will be originated from the matrix by applying list management and data matching technique.

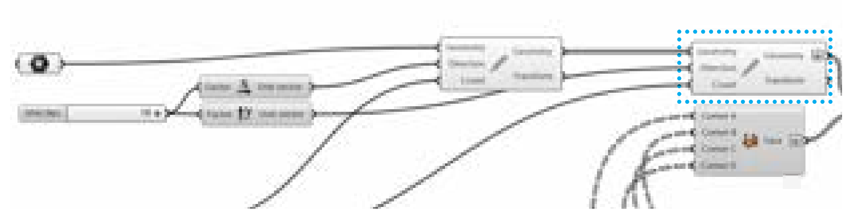


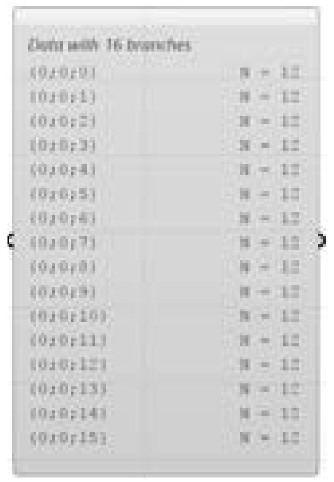




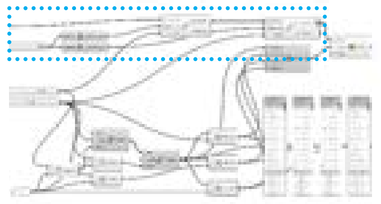
1. Connect Point component to the Geometry (G) input of the Linear Array.
2. Connect Spacing out of the Number Slide to the Factor(F) input of the Unit X. Connect Unit Vector (V) of the Unit X Output to the Direction (D) input of the Linear Array.
3. Connect V Output of Number slider to the Count (C) input of the Linear Array.



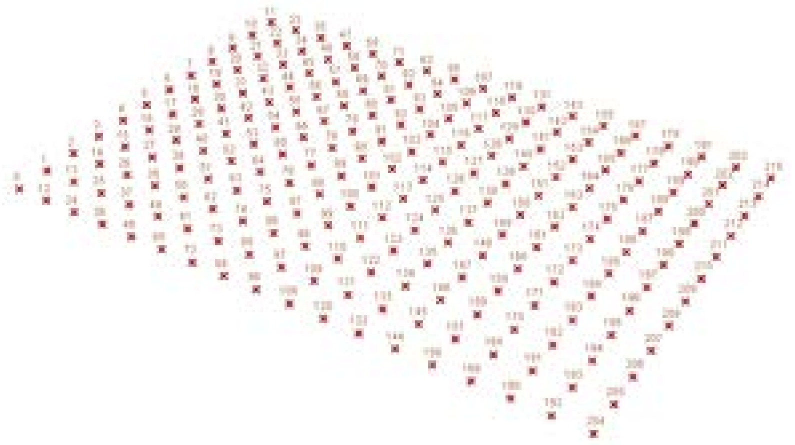


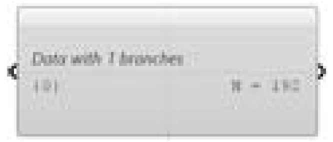


```
Data with 16 branches
{0;0;0}     N = 12
{0;0;1}     N = 12
{0;0;2}     N = 12
{0;0;3}     N = 12
{0;0;4}     N = 12
{0;0;5}     N = 12
{0;0;6}     N = 12
{0;0;7}     N = 12
{0;0;8}     N = 12
{0;0;9}     N = 12
{0;0;10}    N = 12
{0;0;11}    N = 12
{0;0;12}    N = 12
{0;0;13}    N = 12
{0;0;14}    N = 12
{0;0;15}    N = 12
```

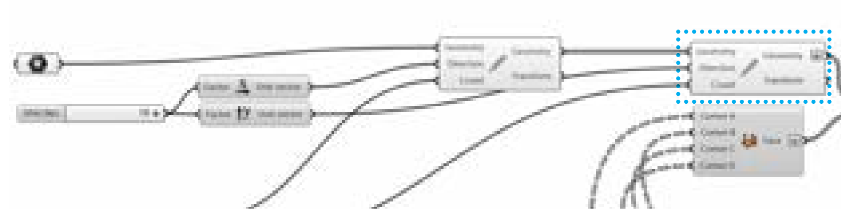


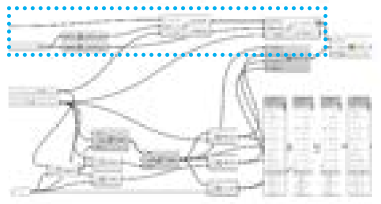
1. Connect Geometry of the Linear Array to the Geometry (G) input of the Linear Array.
2. Connect Spacing output of the Number Slider to the Factor (F) input of Unit Y.
   Connect Unit vector (V) Output to the Direction (D) input of the Linear Array.
3. Connect U output of Number slider to the Count (C) input of the Linear Array.





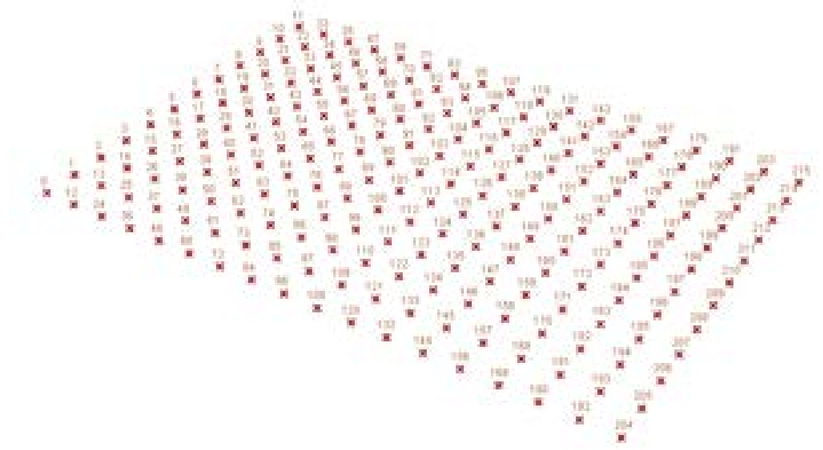
```
Data with 1 branches
{0}         N = 192
```

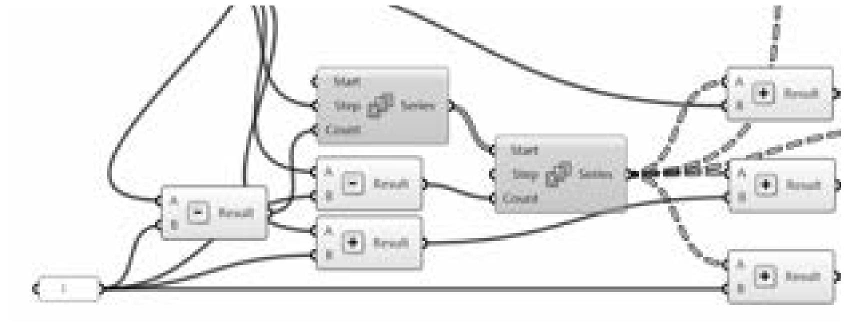


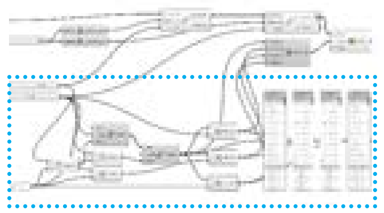


Flatten

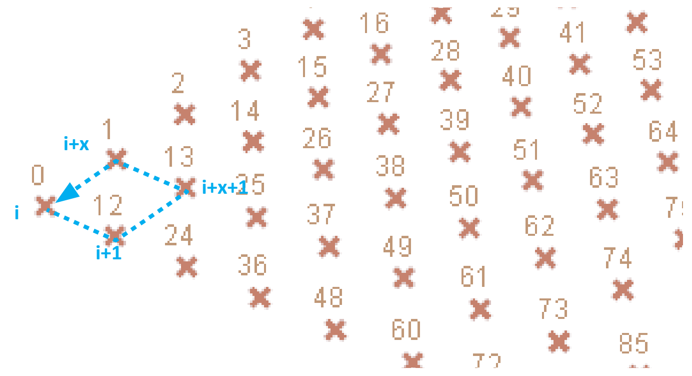
Flatten or graft items shown above are crucial techniques for list management. The list of points show different results depending on whether the item was flattened or grafted. Flattening lists every item in a single order with no hierarchy. Therefore, all items have individual numbers from 0 to {(18X12)-1} in this case.

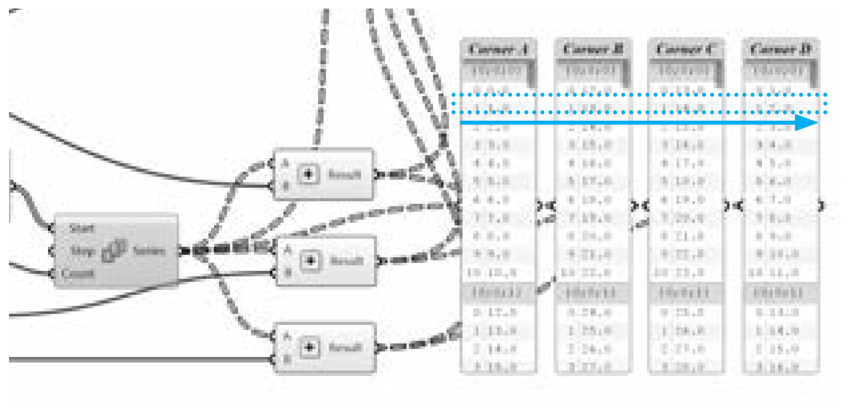


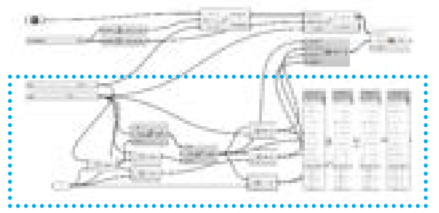




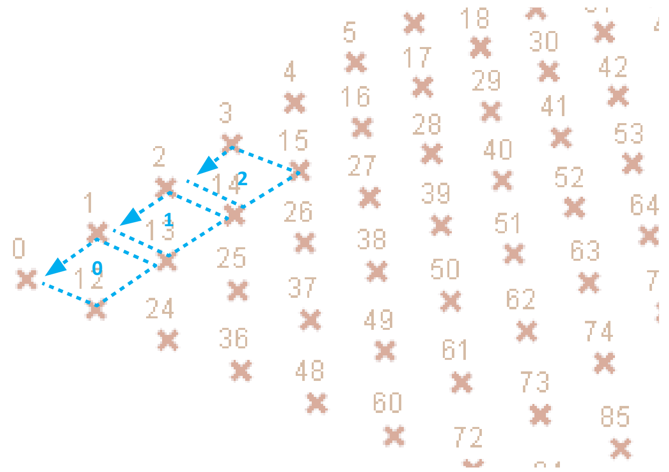
1. Connect U output of number slider to the Step (N) input of the Series component.
2. Connect Result (R) output of the Subtraction to the Count (C) input of the Series.
   Connect V output of number Slider to the A input of the Subtraction.
   Connect panel output of number 1 to the B input of the Subtraction.
3. Connect Series (S) output of the Series to the Start (S) input of the Series.
   Connect Result (R) output of the Subtraction to the Count (N) input of the Series.
   Connect U output of the number slider to A input of the Subtraction.
   Connect panel output of number 1 to the B input of the Subtraction.

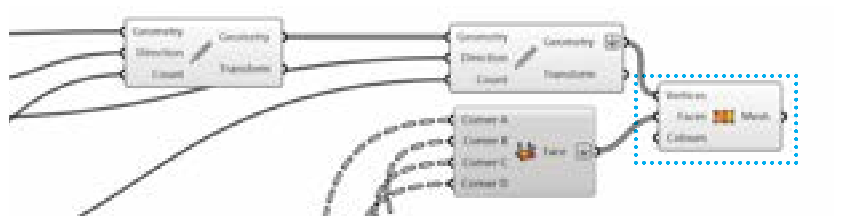


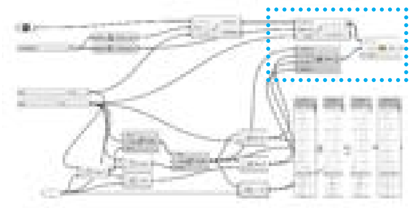




1. Connect Series (S) output of the Series to the Corner A input of the mesh quad.
2. Connect Result (R) output of the Addition to the Corner B input of the mesh quad.
   Connect Series (S) of the Series to the A input of the Addition.
   Connect U number slider output to the B input of the Addition.
3. Connect Result (R) output of the Addition to the Corner C input of the mesh quad.
   Connect Series (S) of the Series to the A input of the Addition.
   Connect Result (R) of Addition to the B input of the Addition.
4. Connect Result (R) output of the Addition to the Corner D input of the mesh quad.
   Connect Series (S) of the Series to the A input of the Addition.
   Connect panel output of number 1 to the B input of the Addition.



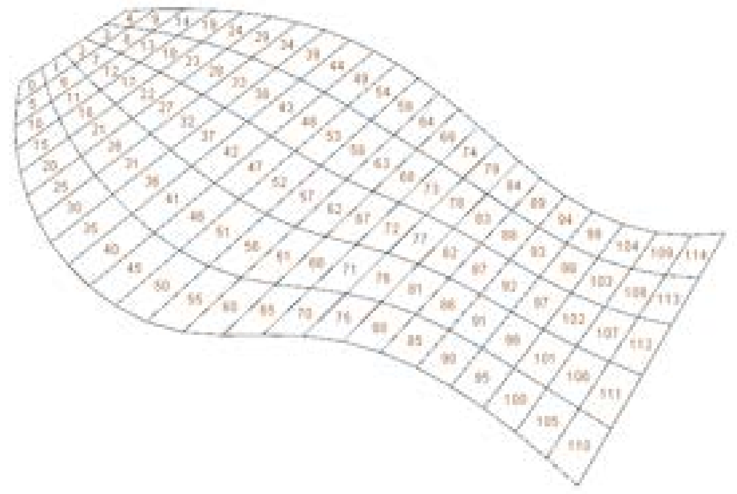


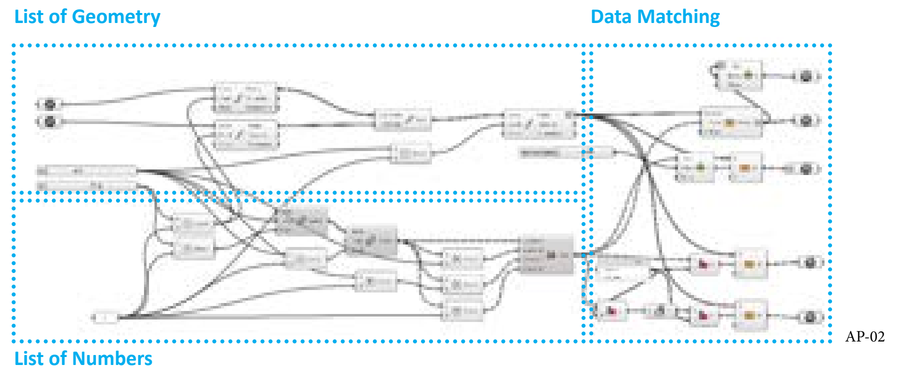


1. Connect Geometry (G) output of the linear array to the Vertice (V) input of the construct mesh.
2. Connect Face (F) of the mesh quad to the Faces (F) input of the construct mesh.

Result & Contribution

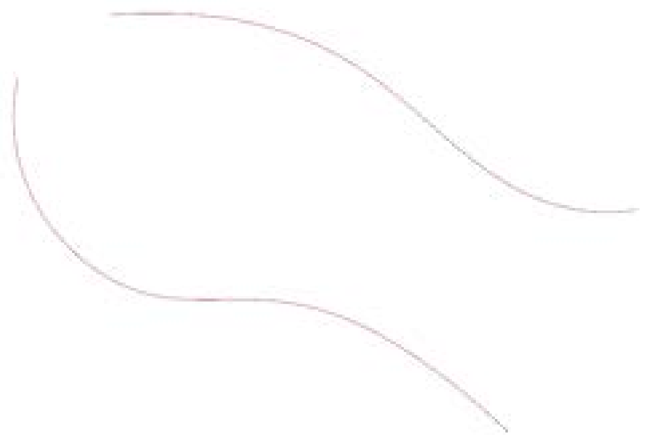
The first and foremost benefit that the designer can get from this mesh making process is that he/she can modify and regulate the mesh. A fundamental understanding of mesh generation allows for further morphological exploration. It is also important for environmental and physics simulation, because most of the simulation is computed based on mesh properties.

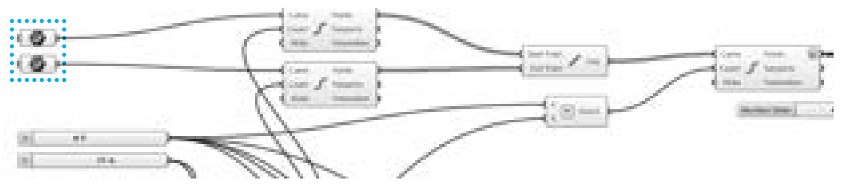




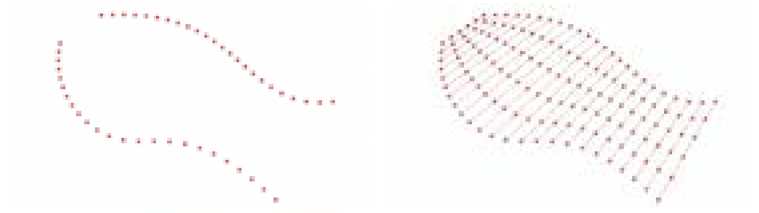
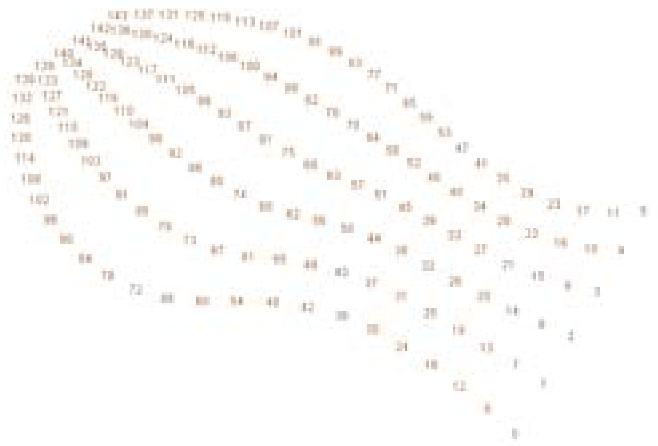
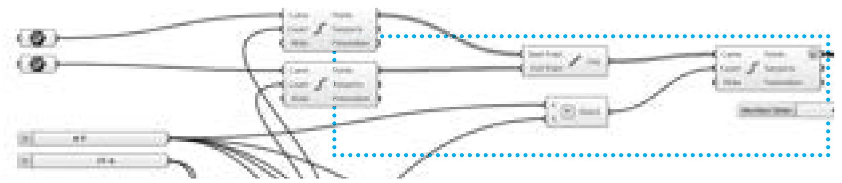
Mesh From two Open Curves

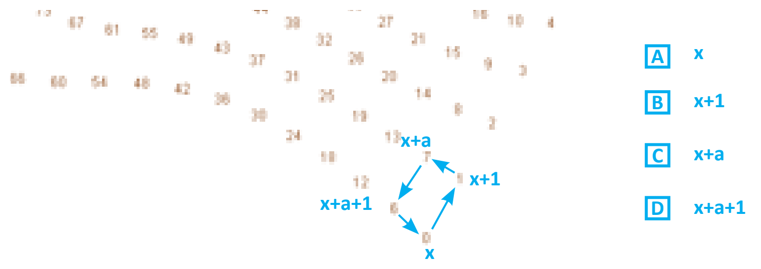
Breaking the UV direction is not favorable to create a mesh surface. For instance, a trimmed nurbs surface in Rhinoceros requires another layer of work for mesh generation. The principle of mesh generation is the same as the previous exercise.
(In that constructing a mesh from vertices.)

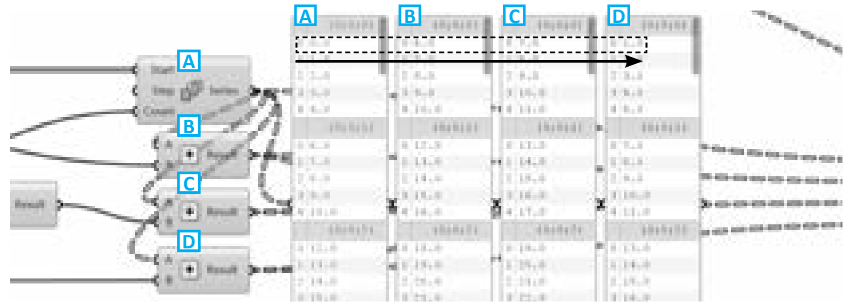


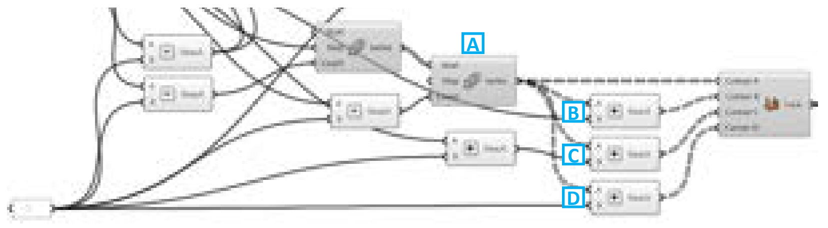


1. Connect two open curves to the Curve (C) input of the divide curve.
2. Connect V output of number slider to the A input of Subtraction, and connect panel number 1 to the B input of Subtraction.
3. Connect points from divide curve to the Start Point (A) input of the line.
   Connect points from divide curve to the End Point (B) input of the line.
4. Connect line(L) of the line to the Curve(C) input of the divide curve.
   Connect V output of number slider to the A input of Subtraction, and connect panel number 1 to the B input of Subtraction.



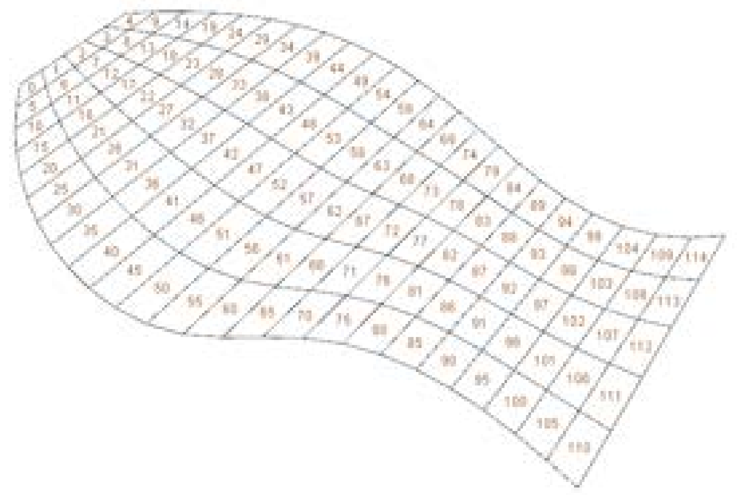


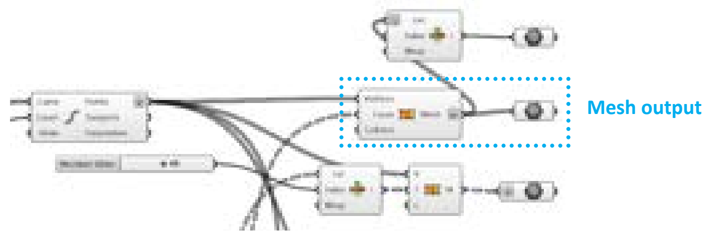


Point Matrix

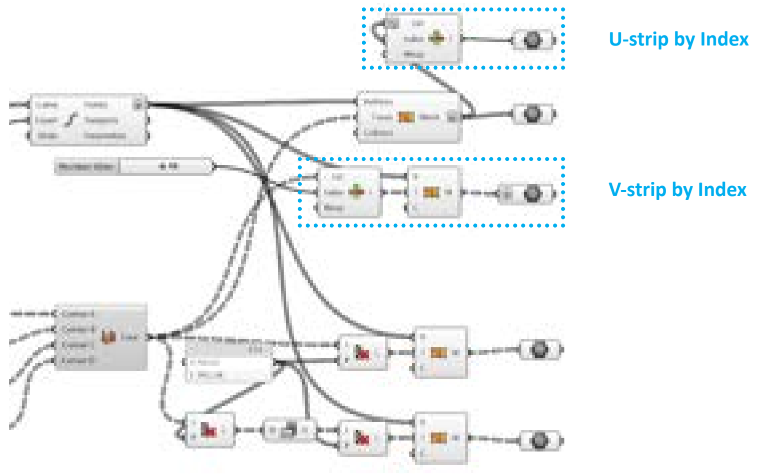
1. Connect the Points output from the first divide curve component to Start input of the line 2pt component.
2. Connect the Points output from the second divide curve component to End input of the line 2pt component.
3. Connect the Line ouput to the Curve input of divide curve component and connect the u-1 number to the Count input to create lists of points as a point matrix.

1. Connect Series (S) output of the Series to the Corner A input of the mesh quad.
2. Connect Result (R) output of the Addition to the Corner B input of the mesh quad.
   Connect Series (S) of the Series to the A input of the Addition.
   Connect U number slider output to the B input of the Addition.
3. Connect Result (R) output of the Addition to the Corner C input of the mesh quad.
   Connect Series (S) of the Series to the A input of the Addition.
   Connect Result (R) of Addition to the B input of the Addition.
4. Connect Result (R) output of the Addition to the Corner D input of the mesh quad.
   Connect Series (S) of the Series to the A input of the Addition.
   Connect panel output of number 1 to the B input of the Addition.



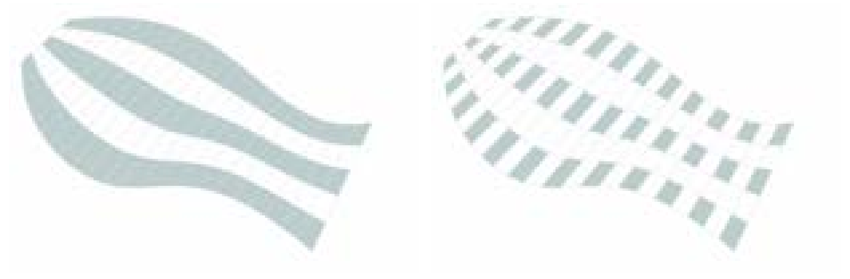


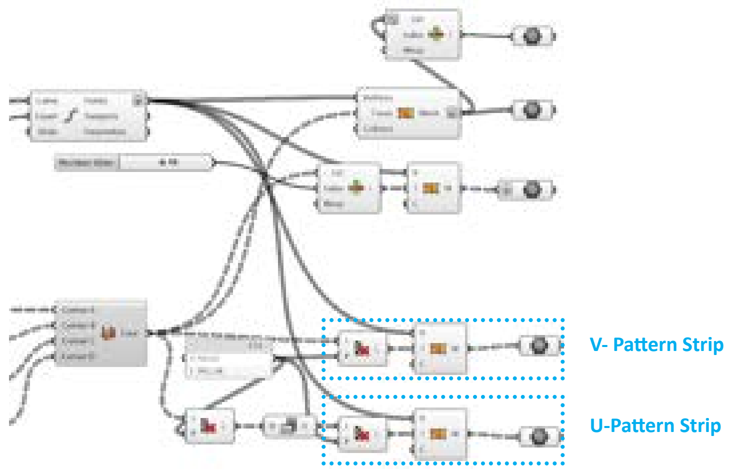
1. Connect Face (F) of the mesh quad to the Faces (F) input of the construct mesh.
2. Connect Mesh (M) of the construct mesh to the Mesh component.

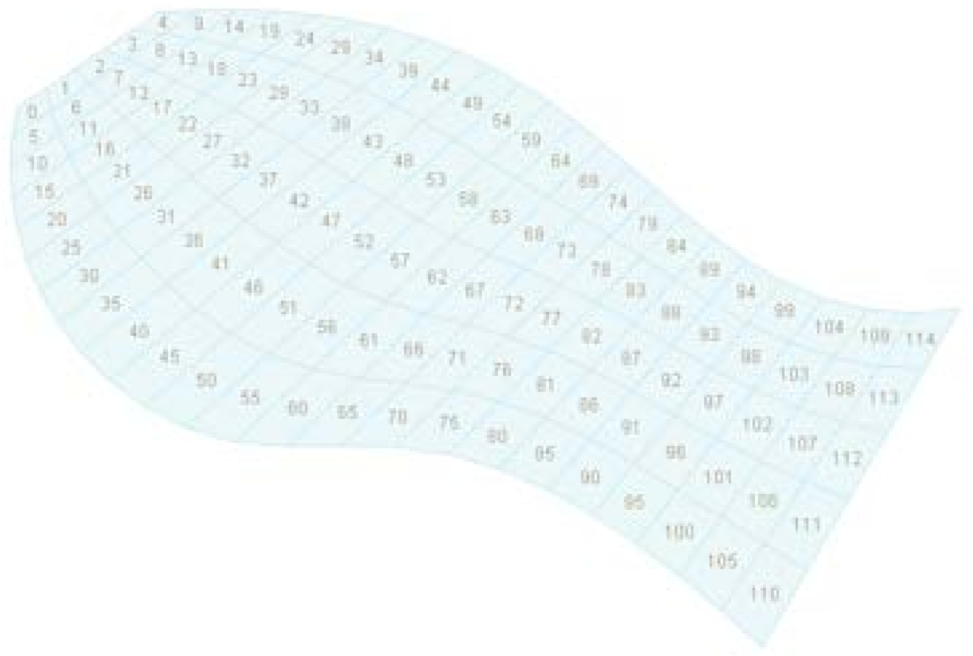


Data Management

Once the data is matched to generate mesh, by treating the list components the user can create various types of output. The code above is an example of how to create different types of geometry.





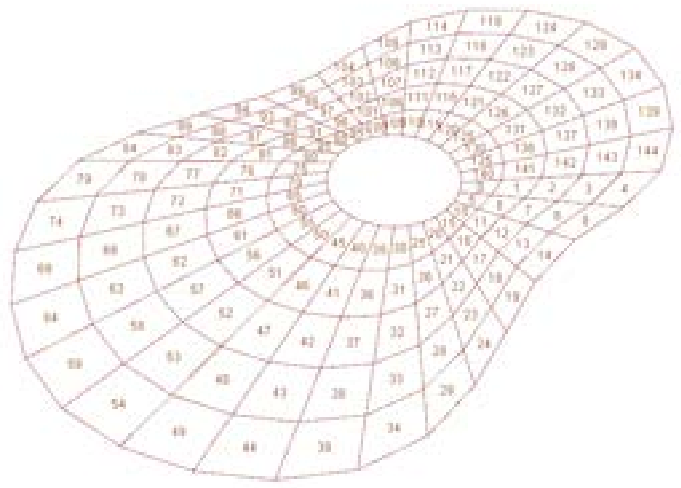


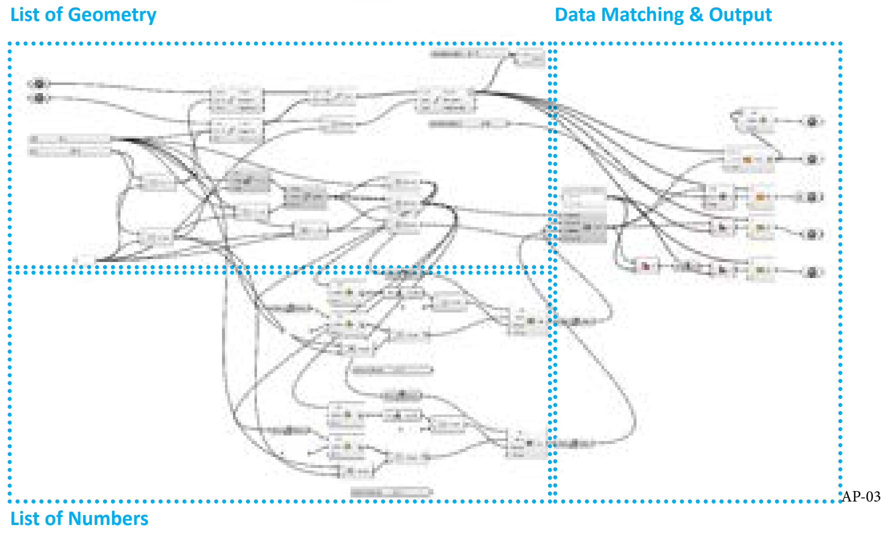
Data Access

The major benefit that architects obtain from this mesh generation process in a grasshopper environment is that every mesh face is aligned and listed so that it can be selected easily. With a simple code above, a designer can access an individual or specific group of mesh with certain parameters.

Result & Contribution

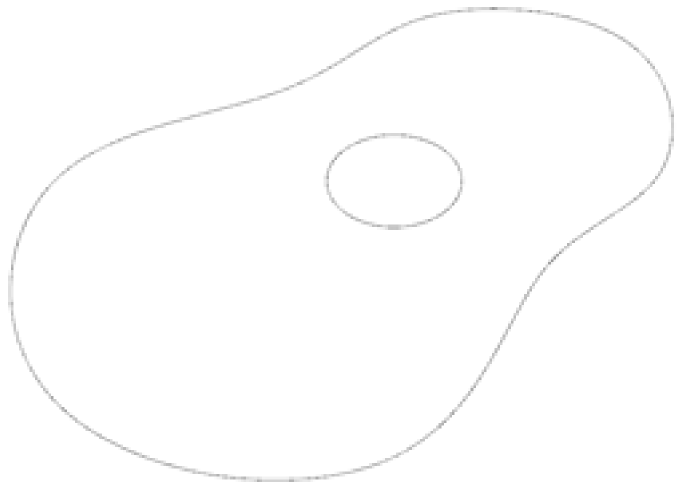
The process can be used in specific cases that require regular pattern on irregular surfaces. The limitation of this method is that the result always contains two other edges of mesh. Following session demonstrates how to deal with this problem.

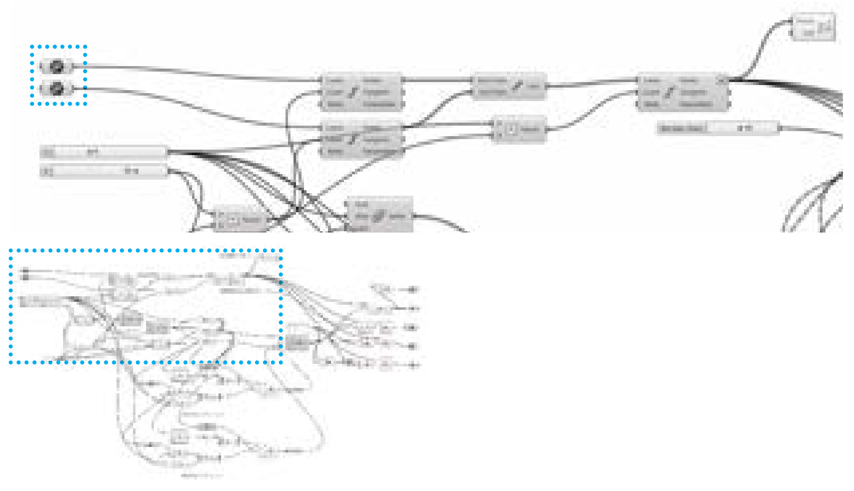




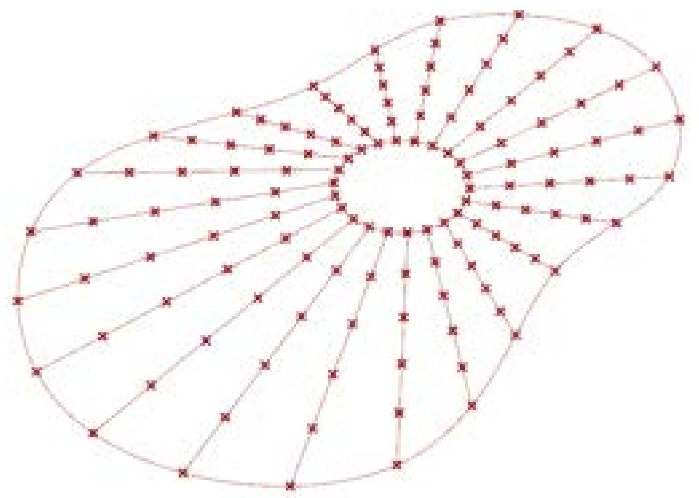
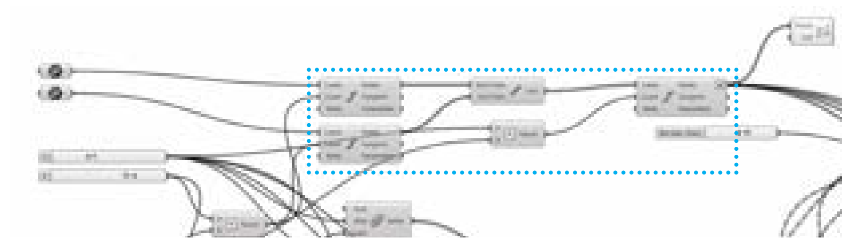
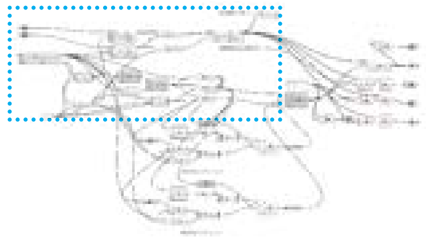
Mesh From two Closed Curves

This session covers the mesh generation in a different topology, specifically the donut type. This could be useful in case of surfacing the two closed curves with a regular mesh pattern such as a roof structure with two closed walls. Due to its shape, a critical part of this exercise is how to enclose a loop system using list management.



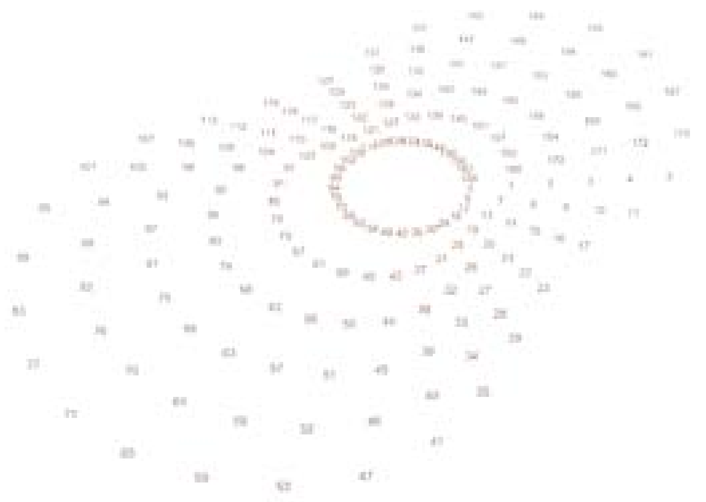
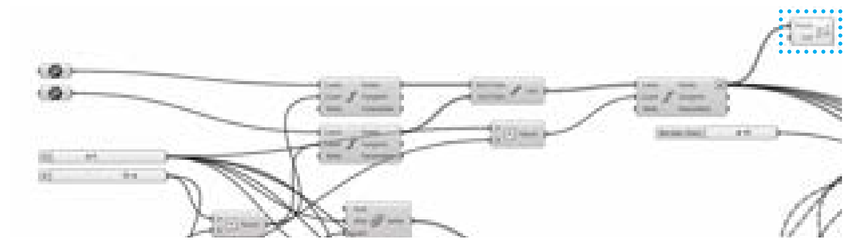
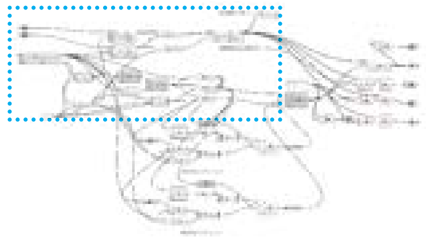


1. Connect two open curves to the Curve (C) input of the divide curve.
2. Connect V output of number slider to the A input of the Subtraction, and connect panel number 1 to the B input of the Subtraction.
3. Connect Points from divide curve to the Start Point (A) input of the Line.
   Connect Points from divide curve to the End Point (B) input of the Line.
4. Connect Line(L) of the Line to the Curve(C) input of the divide curve.
   Connect V output of number slider to the A input of the Subtraction, and connect panel number 1 to the B input of the Subtraction.

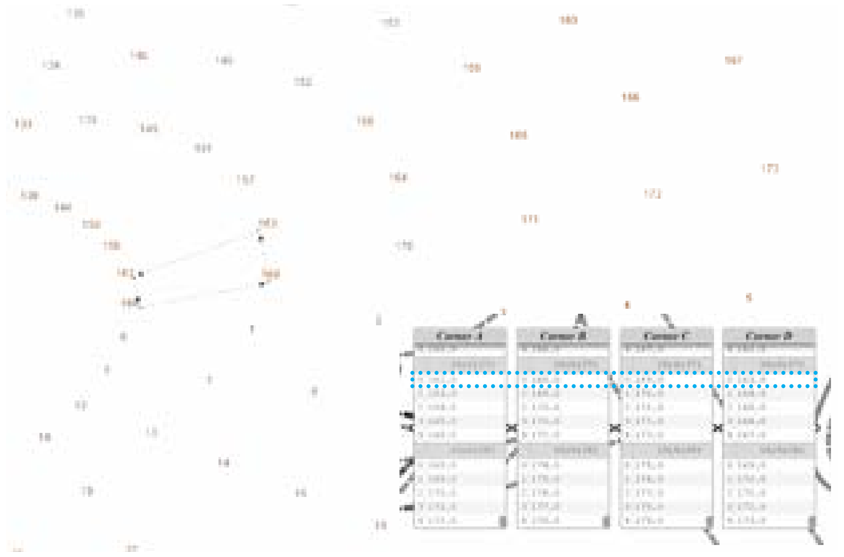
Point Matrix

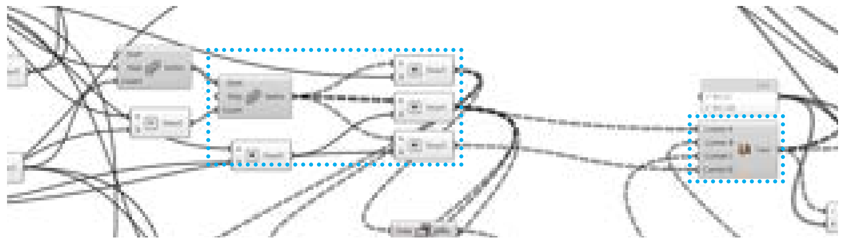
1. Input the Point with Linear array in the X direction with assigned spacing.
2. Linear array in the Y direction with assigned spacing.
3. Parameters for the Count inputs are equal to UV value of the mesh surface.
3. Flatten the Geometry output to sort all items into one branch.

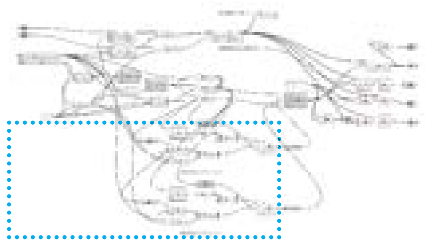




List Management - Loop

Once the Point matrix is created, the list of the point can be checked with the 'Point list' component. At this stage, the loop of the list is not created. So, the first and last data should be matched to create the ring shaped UV mesh.

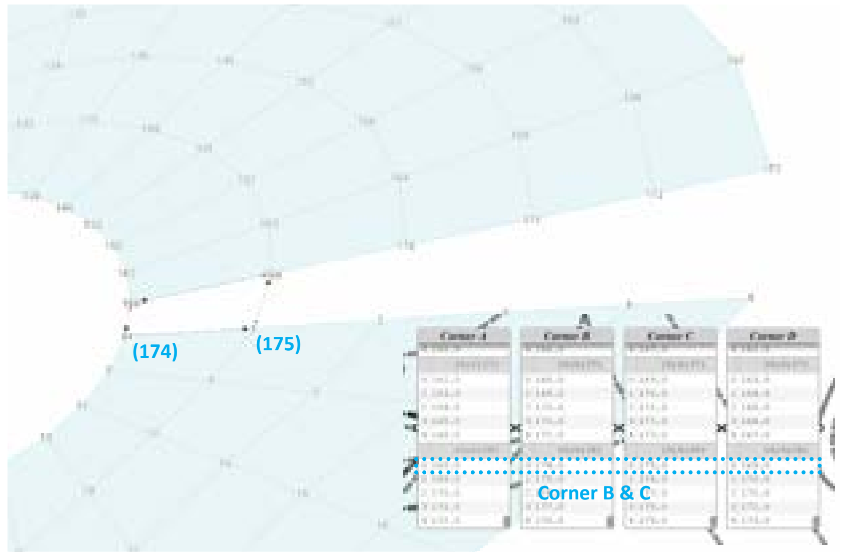


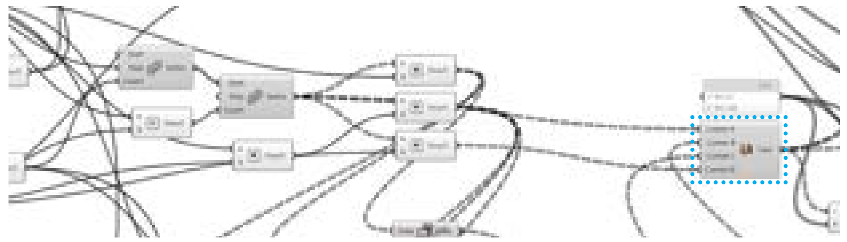


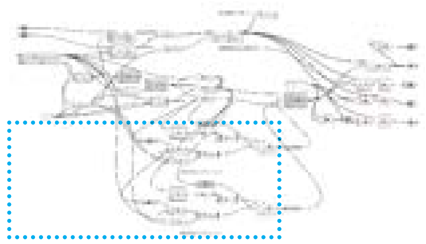


Data Matching

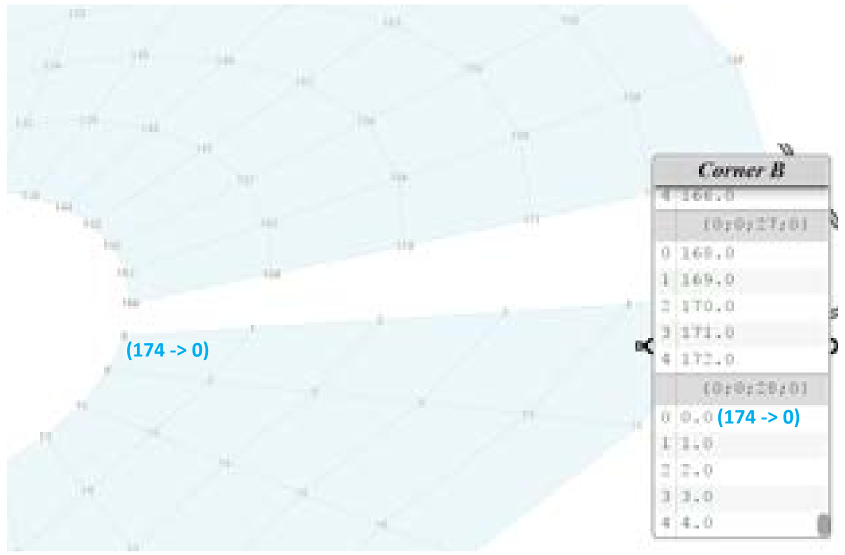
Once the Point matrix and locational numbers are set up, the data can be matched to make a visualization of the geometry. In this exercise, 'Construct mesh' component is used to match this data.





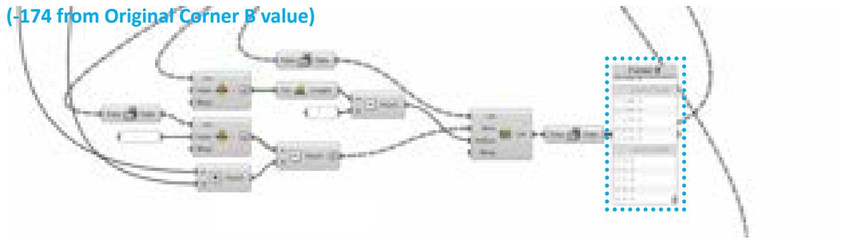


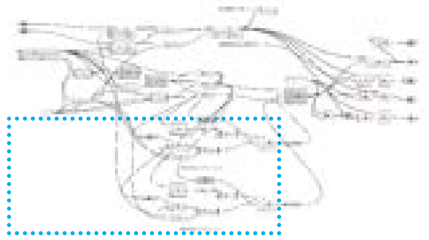
The last mesh strip is not created in this code at this step. Thus, this type of mesh requires special treatment on this issue. For example, numbers 0 and 1 of the Point list should be altered to 174 and 175 by inputing additional coding.



```
Corner B
4 166.0
   {0;0;27;0}
0 168.0
1 169.0
2 170.0
3 171.0
4 172.0
   {0;0;28;0}
0 0.0
1 1.0
2 2.0
3 3.0
4 4.0
```

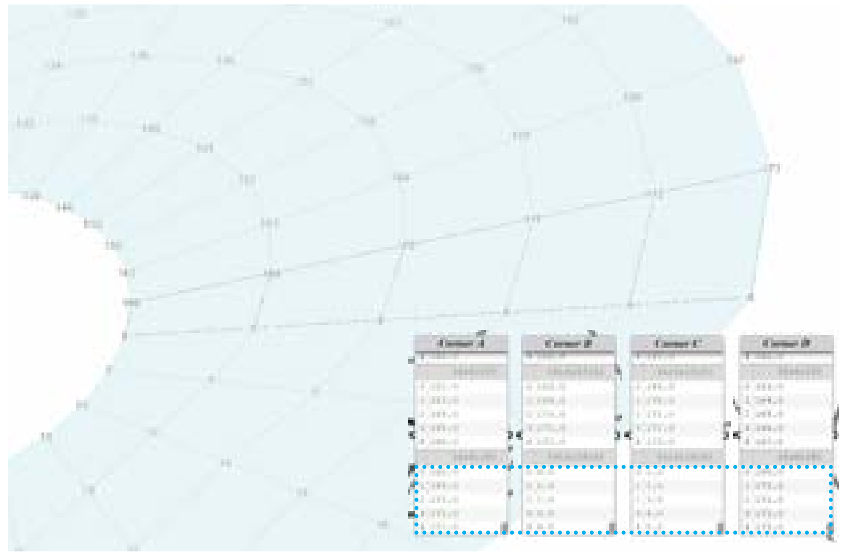
(-174 from Original Corner B value)



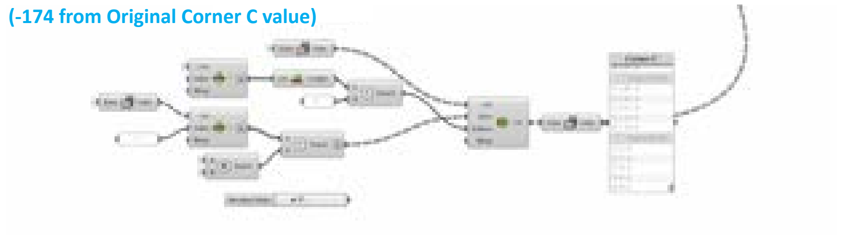


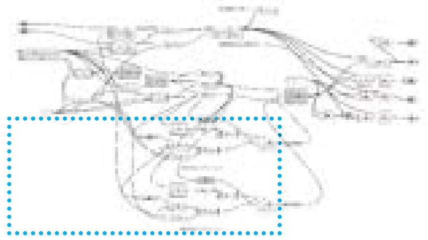
Data Treatment

In order to change the number '174' to '0', instead of just subtracting 174, there should be a proper method of delivering '174' as meaningful data. For example, if the code is just subtracting 174, the code will not work in any other scenario that would have a different UV number.

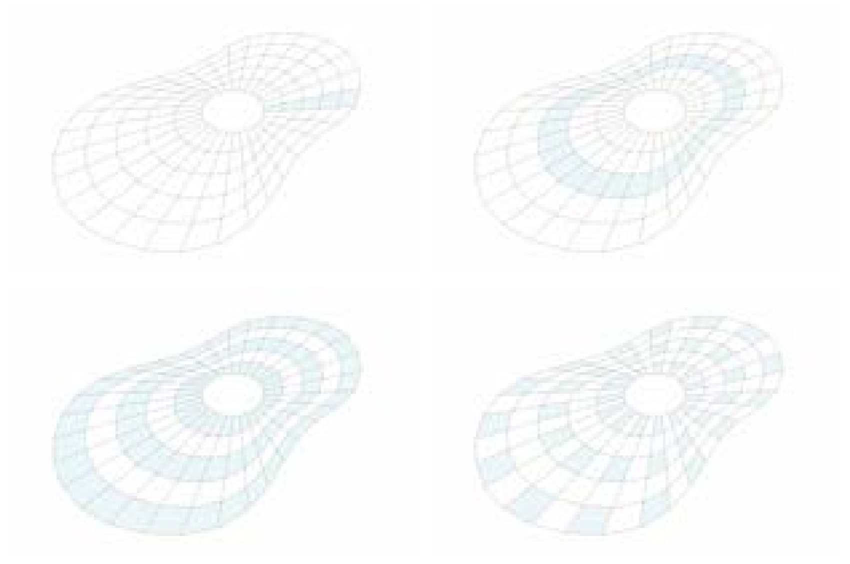


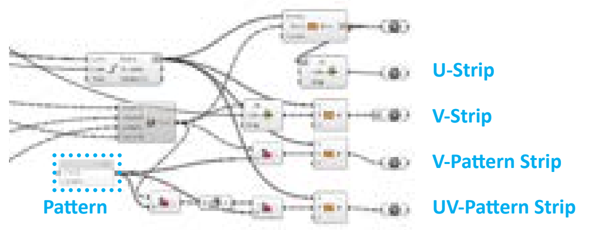
(-174 from Original Corner C value)

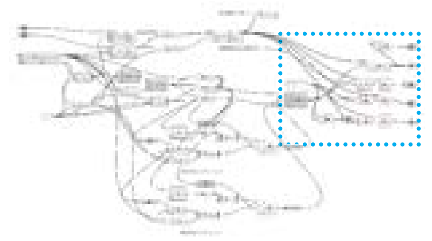


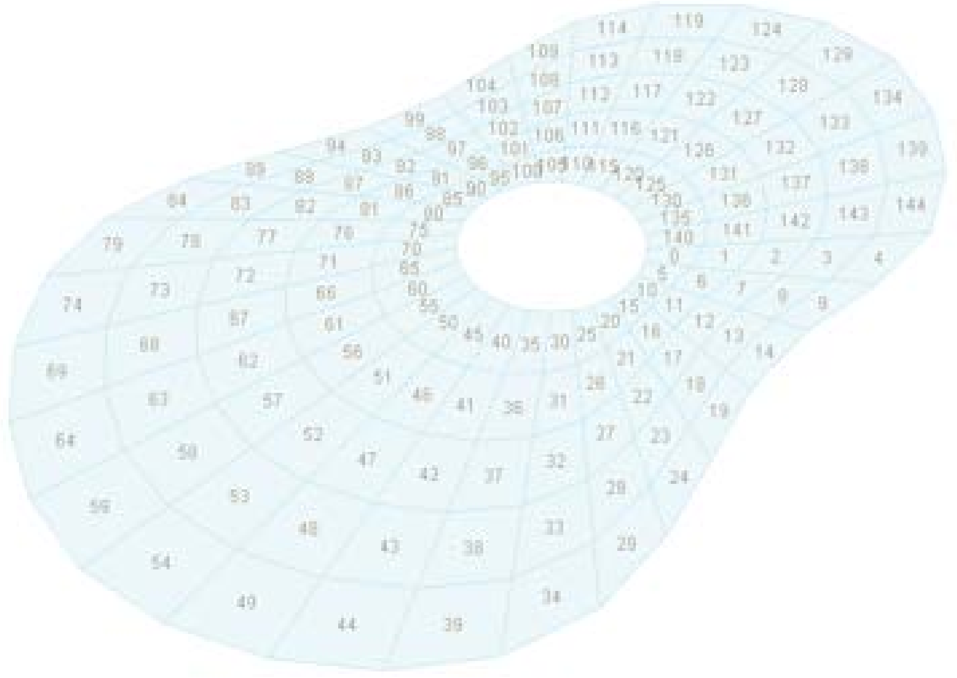


The code shown above is the method to achieve 174 with a special combination of the components. The components are making a result based on an assigned U V value, so the code has become robust enough to be applied in different computational circumstances.







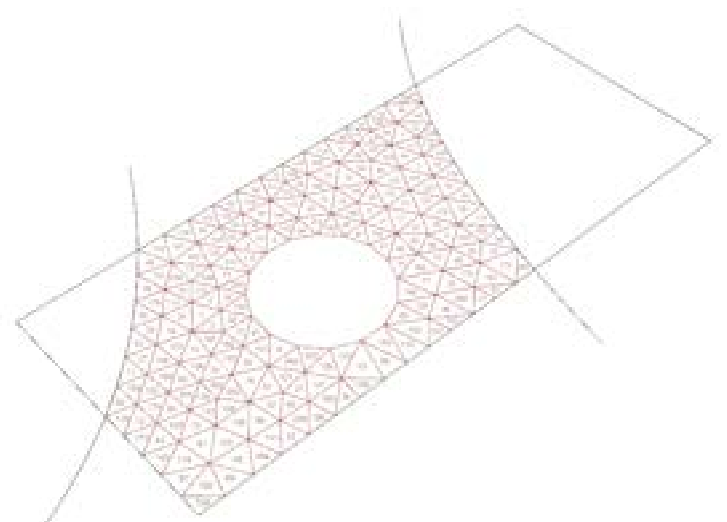


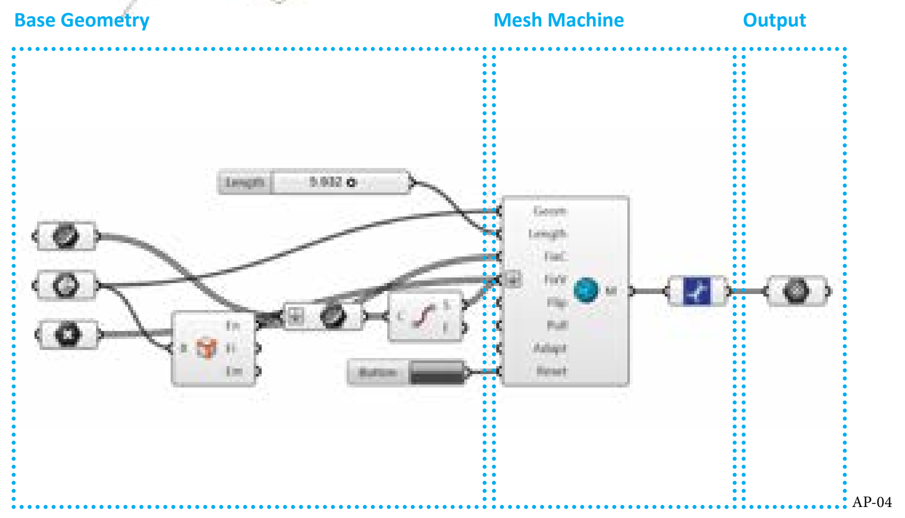
Data Access

The major benefit that architects can obtain from this mesh generation process in a grasshopper environment is that every mesh face is aligned and listed to be selected easily. With a simple code above, the designer has access to an individual or specific group of mesh with certain parameters.

Result & Contribution

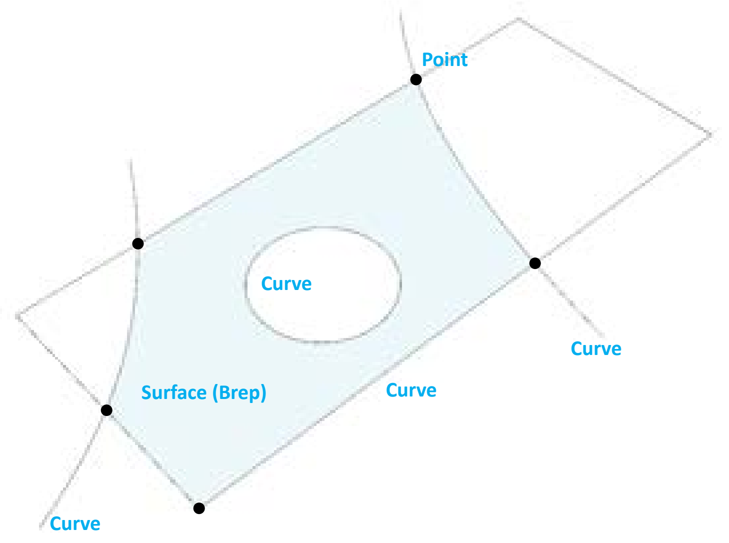
This process can be used in specific cases requiring regular pattern on irregular surfaces. For example, this code can create diverse curtain wall shapes on certain surfaces only by adjusting simple number parameters and can experiment diverse options efficiently in less time. However, the shortcomings of this method is that the result always contains 2 straight edges at the end. Next 2 sessions will show how to deal with this problem.

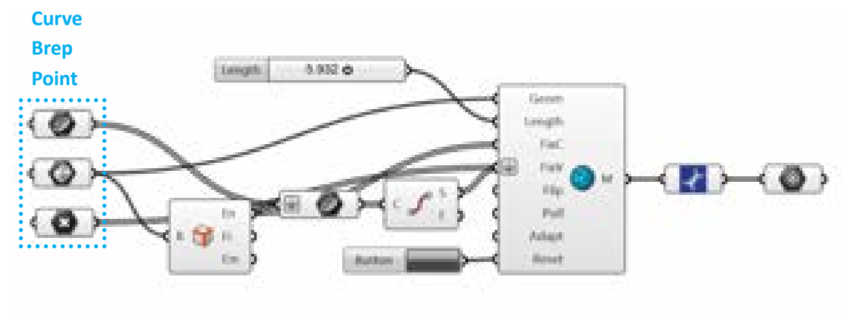




Mesh Machine by Daniel Piker

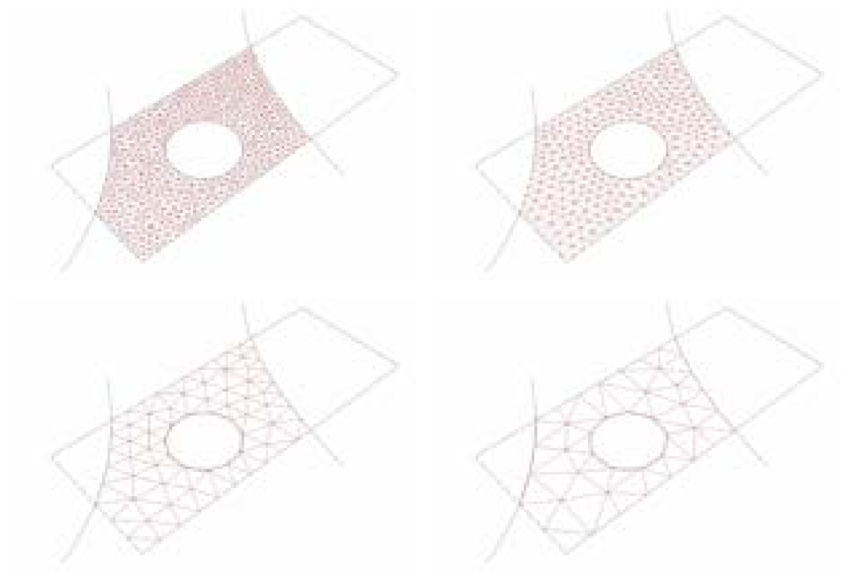
The Mesh Machine component is developed by Daniel Piker, as an addition to grasshopper. It is a unique tool that generates mesh. It creates a more uniform mesh topology, and also features mesh adaptability tools, mesh point distribution based on curvature, guides geometries to manipulate the meshes and mesh relaxation. As a quick and versatile way to generate mesh, Mesh Machine is a perfect component for force simulation in the initial stages of design.

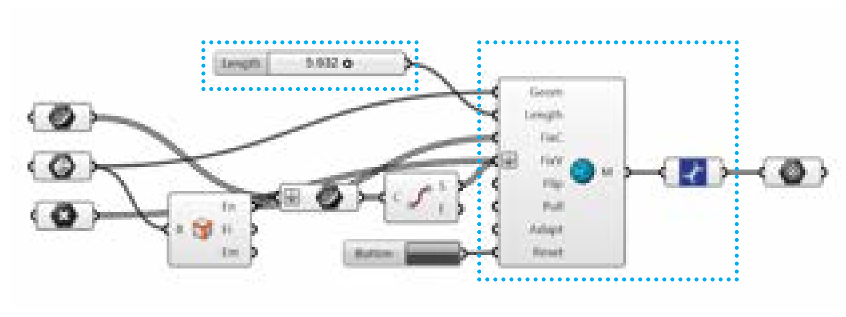




Initial geometry input setting

1. Set all the curves into the Curve component.
2. Set the surface of the closed boundary into the Brep component.
3. Set all the points into the Point component.
4. Connect Brep component to the Brep input of the Brep edge.
   Connect Naked edge from the Brep edge to Curve componenet.
   Connect Curve component to the Curve (C) input of the End points.
   Connect Curve Start points output of the End points to the Fix Vertices list (Fix V) of the Mesh Machine.
5. Connect Point component to the Fix Vertices List(FixV) input of the Mesh Machine.



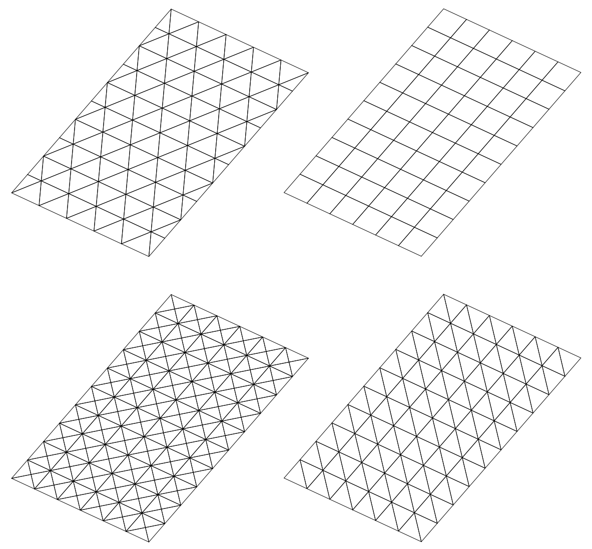


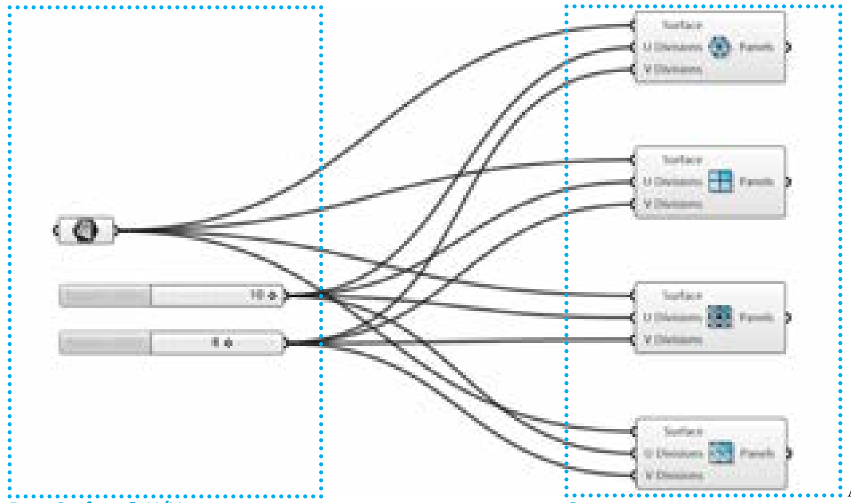
Mesh length setting

1. Click the on and off button when you first reset the geometry.
2. Move the length number slider to change the size and density of the mesh pattern.

Result & Contribution

The goal of the Mesh Machine is to generate a mesh with well distributed properties. This is essential for physic simulation because the neutral properties of a mesh represents the even distribution of its properties in a natural state as well as the artificial composites that have bending qualities.

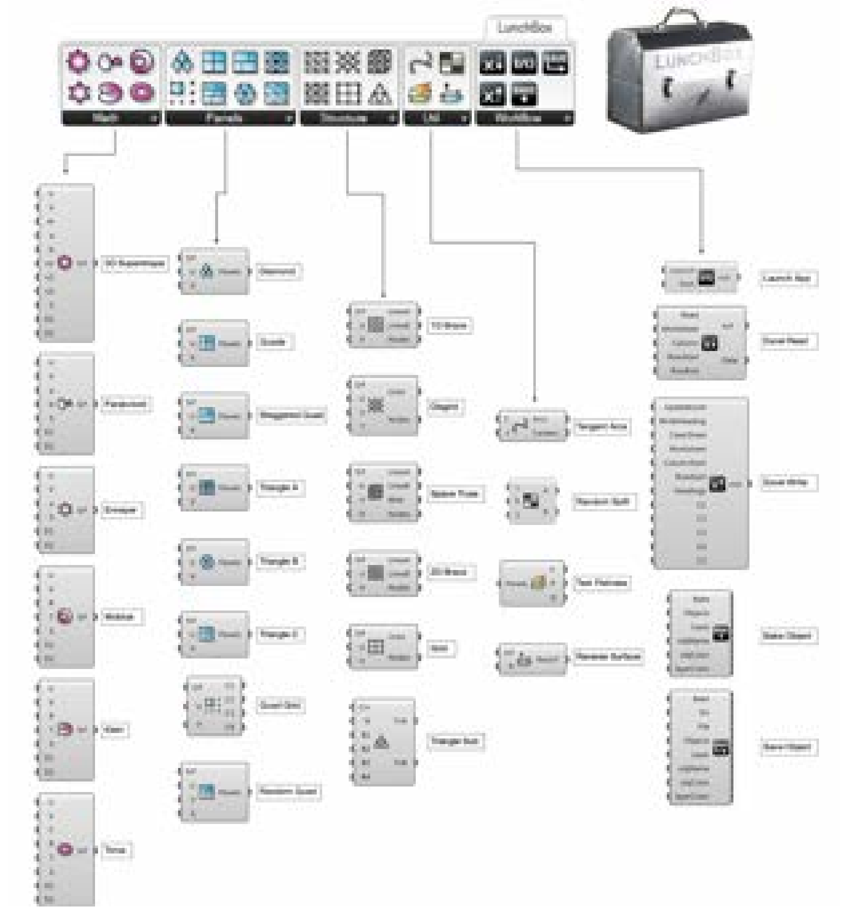




AP-05

Lunch Box by Proving Ground

LunchBox is for exploring mathematical shapes, paneling, structures and workflow. The grasshopper plug-in introduces new components for general machine learning implementations such as regression analysis, clustering, and networks as well. A key aspect of Lunch box for mesh generation is to simplify mesh organization,(but the panelization only works for untrimed surface.)



Data: Components for dataset management, XML, and JSON formats

Machine Learning: LunchBoxML components for regression, clustering, and neural networks.

Generate: Components for cool generative geometry.

Math: Create parametric surfaces and forms such as the Mobius, Klein, or 3D Supershape

Panels: Create paneling systems such as quad grids, diamonds, or triangles.

Structure: Create wire structures such as diagrids or space trusses.

Utility: Rationalize spline curves and reverse surfaces.

Workflow: Read and write Excel files, layer management, and automate baking and saving.

2. Vector _ Environmental Data

2.1. Sun Vector

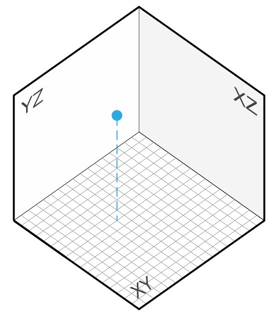
2.2. Thermal mass

Basic Anatomy of a Vector

A vector is a geometric quantity describing Direction and Magnitude. Vectors are abstract; ie. they represent a quantity, not a geometrical element.

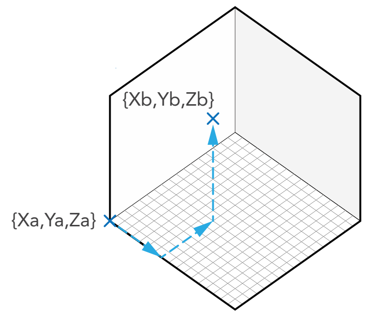
Vectors are indistinguishable from points. That is, they are both lists of three numbers so there is absolutely no way of telling whether a certain list represents a point or a vector. There is a practical difference though; points are absolute, vectors are relative. When we treat a list of three doubles as a point it represents a certain coordinate in space, when we treat it as a vector it represents a certain direction. A vector is an arrow in space which always starts at the world origin (0.0, 0.0, 0.0) and ends at the specified coordinate.

Planes are “Flat” and extend infinitely in two directions, defining a local coordinate system. Planes are not genuine objects in Rhino, they are used to define a coordinate system in 3D world space. In fact, it’s best to think of planes as vectors, they are merely mathematical constructs.



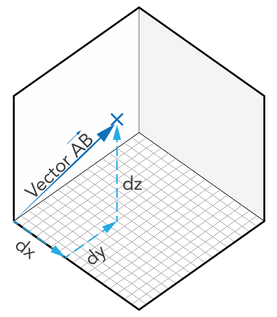
Plane

Planes are “Flat” and extend infinitely in two directions, defining a local coordinate system. Planes are not genuine objects in Rhino, they are used to define a coordinate system in 3D world space. In fact, it’s best to think of planes as vectors, they are merely mathematical constructs.



Point

Points in 3D space have three coordinates, usually referred to as [x,y,z]. Points in 2D space have only two coordinates which are either called [x,y] or [u,v] depending on what kind of two dimensional space we’re talking about. 2D parameter space is bound to a finite surface.

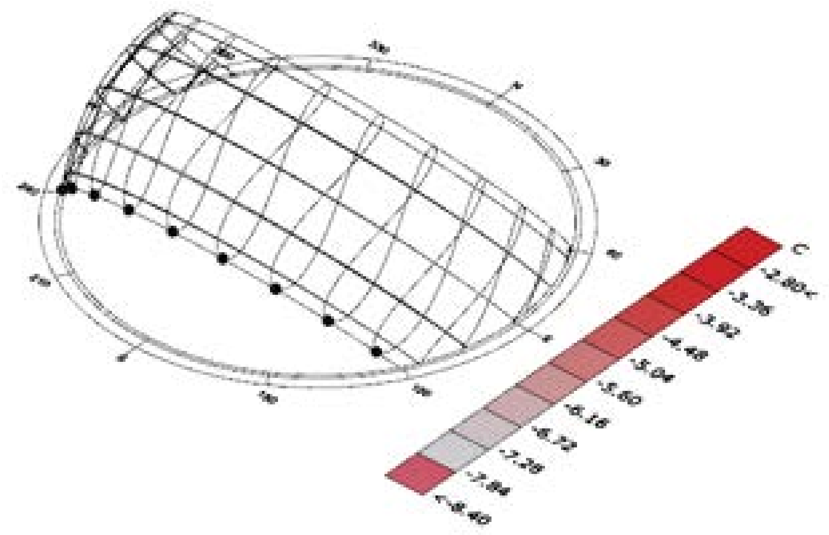


Vector

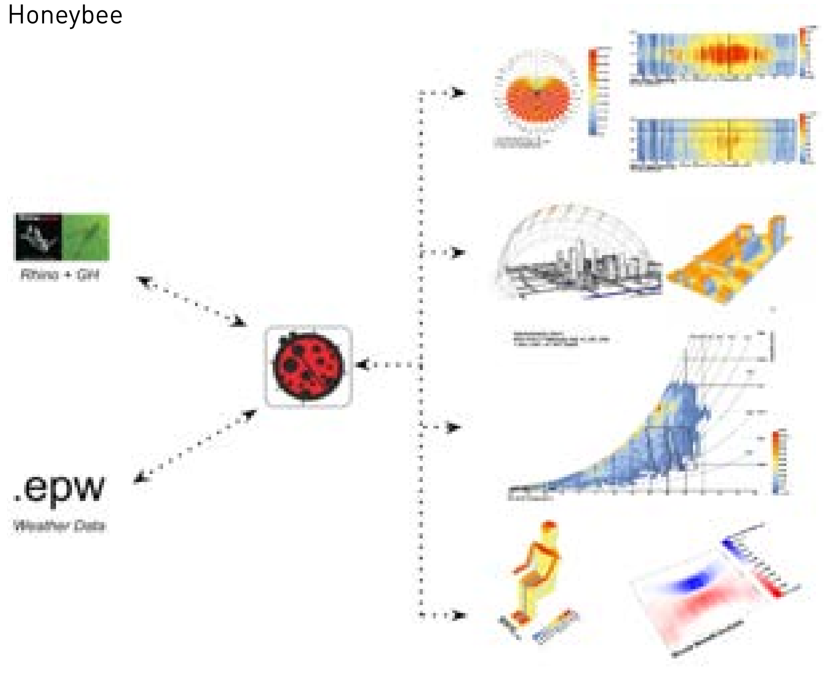
A vector is a geometric quantity describing Direction and Magnitude. Vectors are abstract; ie. they represent a quantity, not a geometrical element.

Sun Vector as Design Driver

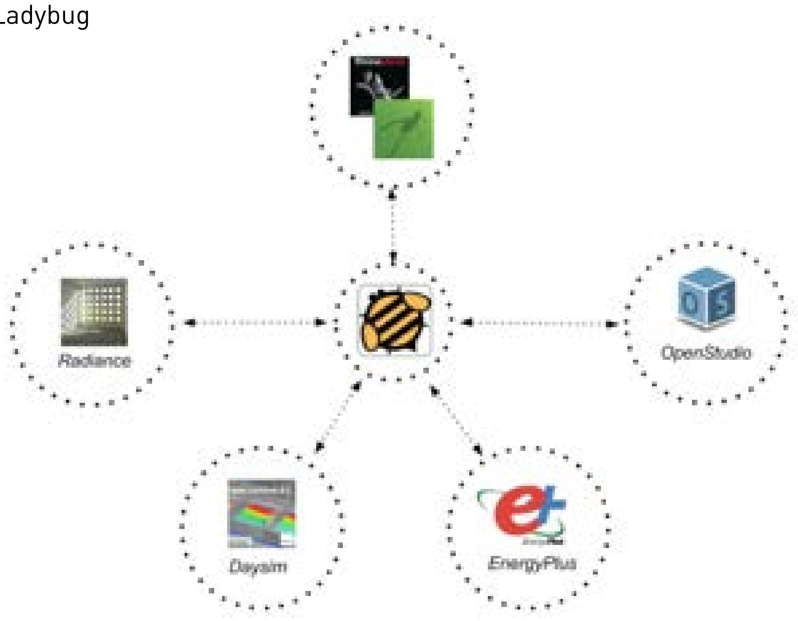
Vector contains main features: Point of origin, Direction and Length. It also can be represented as the relationship between origin of object to target objects. Parametric and computational design process have utilized the relationship heavily to identify multi-data into one single logic. A series of attractor definition is the most popular example for vector use in design process. Another usage of the vector in architectural practice is the sun vector with its thermal and daylight data. The environmental data-driven design process influences the field of architecture and urban design and the development of various environmental analysis and evaluation tool allows architects and designers to bring the data into design decision making process.



Honeybee



Ladybug

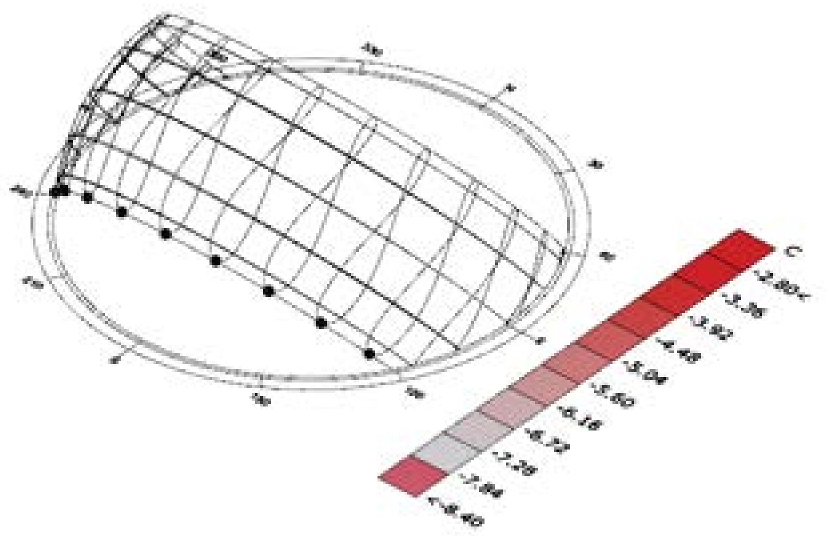


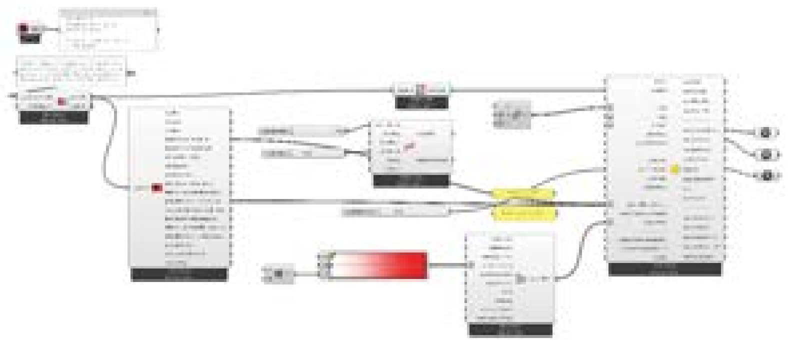
Honeybee & Ladybug as Environmental Analysis tool

Ladybug is a free and open source environmental plugin for Grasshopper to help designers create an environmentally-conscious architectural design. The initial step in the design process should be the weather data analysis; a thorough understanding of the weather data will, more likely, lead designers to high-performance design decisions.

Ladybug imports standard EnergyPlus Weather files (.EPW) in Grasshopper and provides a variety of 2D and 3D designer-friendly interactive graphics to support the decision-making process during the initial stages of design. The tool also provides further support for designers to test their initial design options for implications from radiation and sunlight-hours analyses results. Integration with Grasshopper allows for an almost instantaneous feedback on design modifications, and as it runs within the design environment, the information and analysis is interactive.

Honeybee connects Grasshopper3D to EnergyPlus, Radiance, Daysim and OpenStudio for building energy and daylighting simulation.

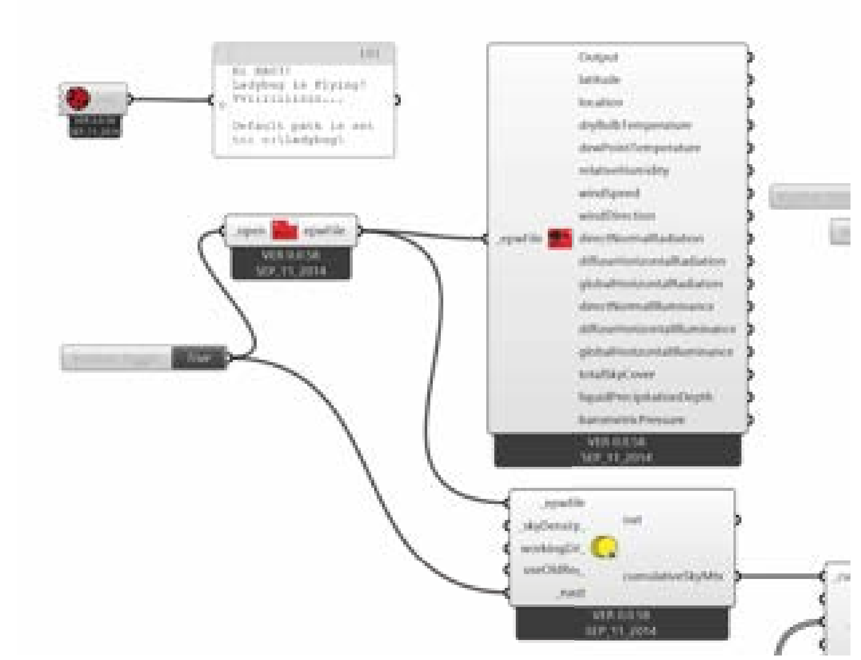


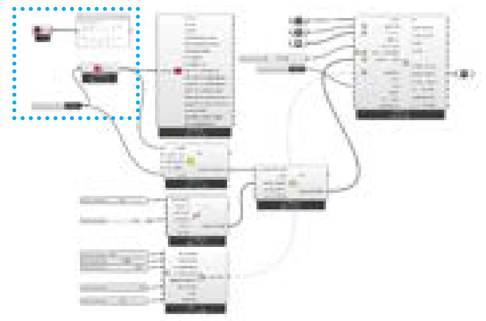


AP-06

Visualizing Sunpath and Vector

Sunpath is a fundamental aspect of environmental simulation. It can be visualized via ladybug's Sunpath component. For advanced environmental simulation, this function provides an intuitive understanding of the sun path as a vector. Honeybee and Ladybug are able to compute environmental data and reflect results onto the target surface with various visual parameters according to an assigned time, period and location from an epw file.

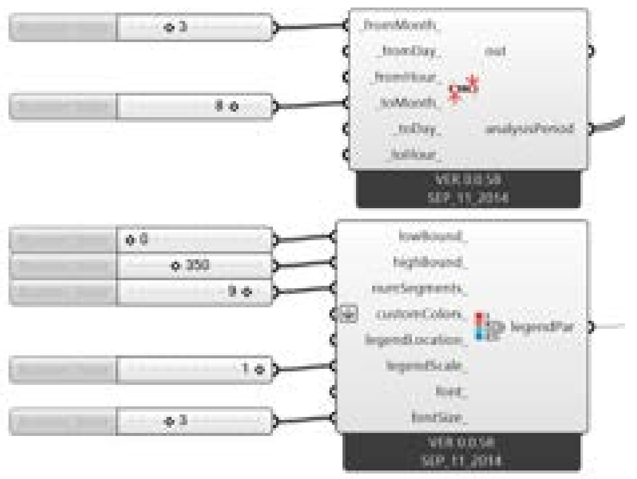


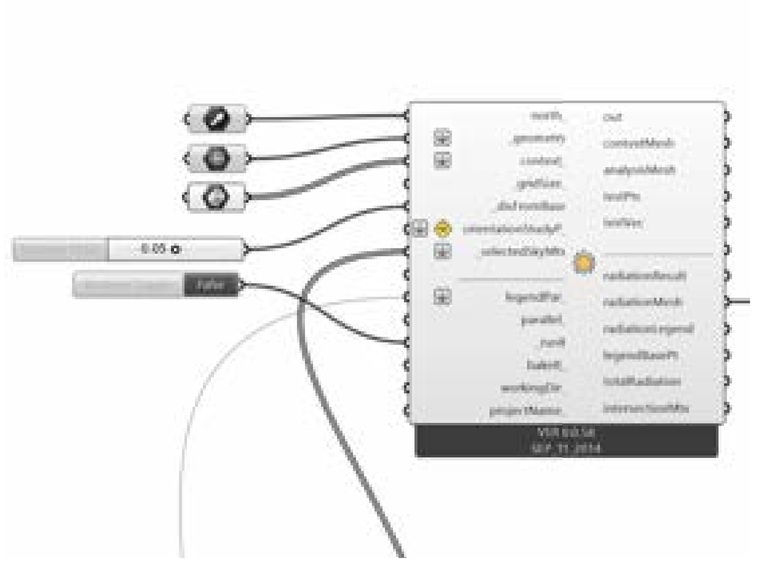


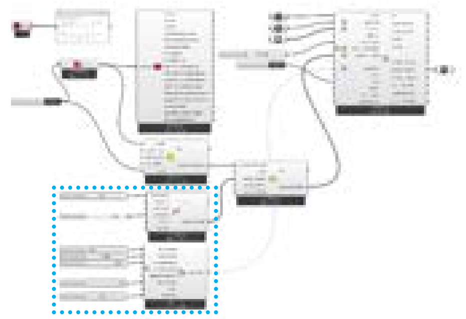
AP-07

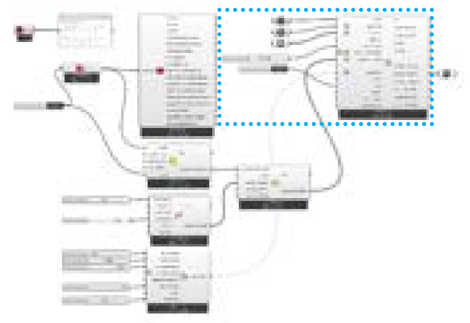
Weather data

Weather data is a crucial parameter for environmental simulation. Overall information for the simulation is based on this local weather data. The form of the file is EPW and the designer can easily access this data via 'energyplus' website or individually contact the country's weather forecast department. Once the file is loaded via 'open epw weather file' component, it has to be connected to the 'GenCumulativeSkyMtx' component.







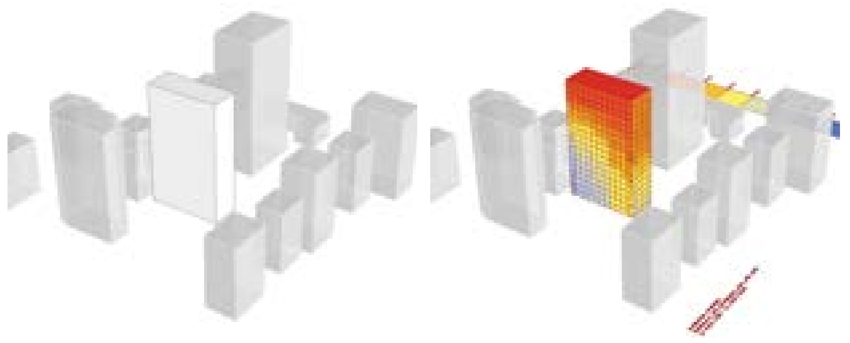
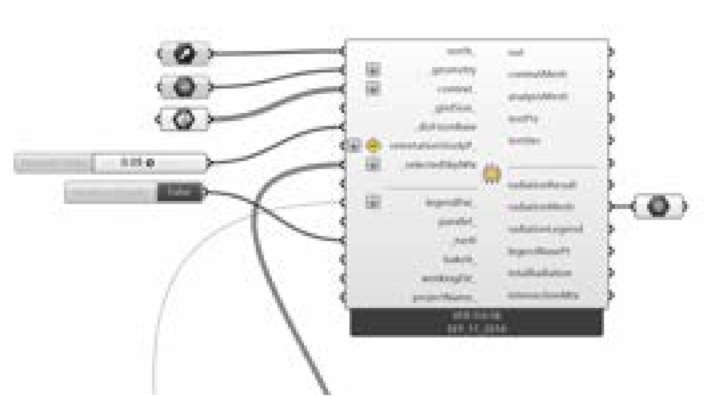
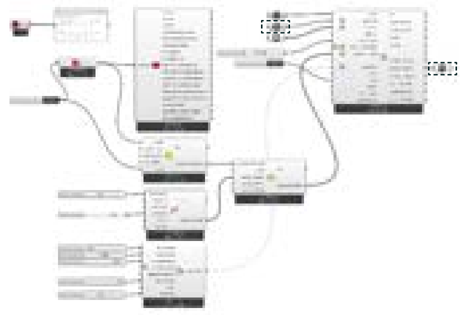


Analysis Period & Legend Parameter

Since the weather changes every minute, the designer has to set-up the analysis period with careful consideration. It can be inserted into the simulation via the 'Analysis Period' component with start and end time. Legend parameter is about visualization of the Graph. By adjusting the parameters, the legend will appear with a setting in the Rhinoceros view port.

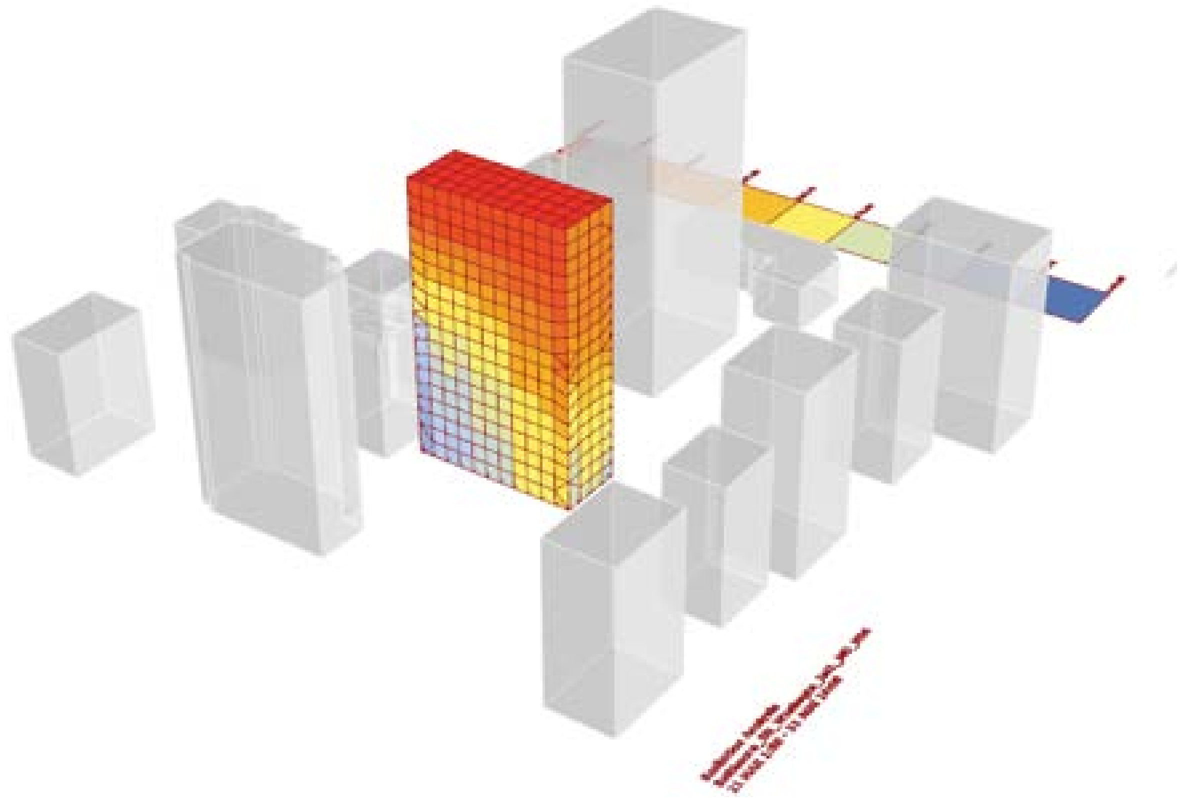
Thermal Simulation

Environmental simulation requires input of objective mesh as 'Geometry', surrounding circumstances such as 'Context', specific vector designated as North direction as 'North', and sky matrix which is set up according to the very first EPW weather file as 'v'. By accumulating this data, the simulation creates two important outputs - Radiation Mesh & Radiation Result.

Radiation Mesh

Radiation Mesh output is the result of the simulation. Each Individual mesh face has a different surface color and these colors can be modified in the legend parameter setting.



Result & Contribution

The major benefit of this simulation is that the designer can get precise data according to the epw file which is an accumulation of the local weather records. Not only can the designer define the result by setting the specific point of location, time and period, but can also use it to explore optimal results from the defined parameters.

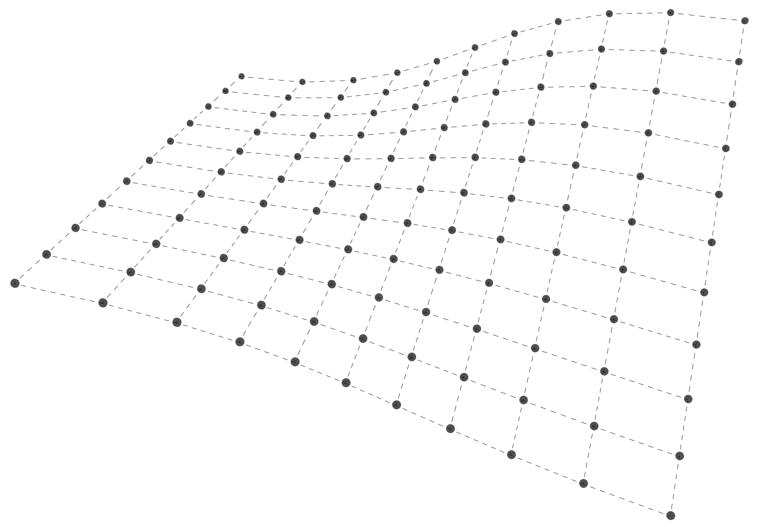
3. Physic simulation

Interdependency of Data _ Spring Particle System

Spring-Particle System

Spring-particle systems are based on lumped masses, called particles, which are connected by linear elastic springs. Each spring is assigned a constant axial stiffness, an initial length, and a damping coefficient. Springs generate a force when displaced from their rest length. External forces can be applied to the particles, as in the case of gravitational acceleration. To solve for the equilibrium geometry of spring-particle systems, there are many techniques with varying degrees of efficiency and stability.

Spring-particle systems are well-known in computer science for creating physical simulations by using the equilibrium position of each particle. The development of computational design in the field of architecture allows designers to access spring -particle systems for finding structural forms and geometrical exploration. Kangaroo, a grasshopper plug-in, and Karamba, a structural design grasshopper plug-in allows the user to interact with force simulation from two dimensional catenary curve to three-dimensional networks.



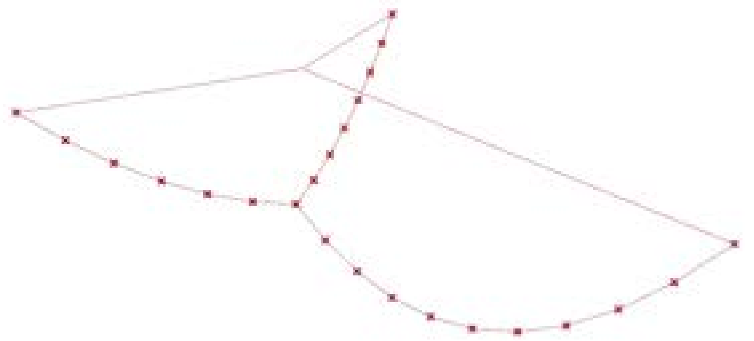
Kangaroo Physics & Karamba

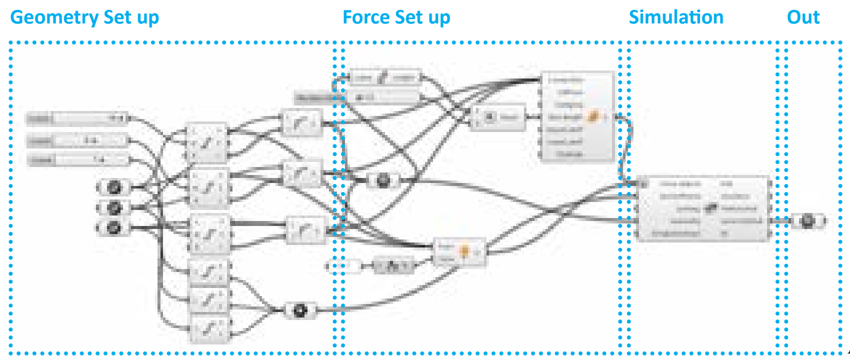
Kangaroo is a live physics engine for interactive simulation, form-finding, optimization and constraint solving. The main feature of Kangaroo physics is to simulate various types of physic behaviors through the spring-particle system.

In addition, Karamba 3D is a parametric structural engineering tool which provides accurate analysis of spatial trusses, frames and shells. It is an interactive, parametric finite element program. It lets you analyze 3-dimensional beam and shell structures under arbitrary loads.

Both physic and structural simulation tools serve as integrative form generators and structural optimization tools under the spring-particle system.

The section of physic simulation focuses on a series of form finding techniques with Kangaroo Physics. The understanding of spring-particle system and basic forces allow advancement into more combinatory and complex design challenges.

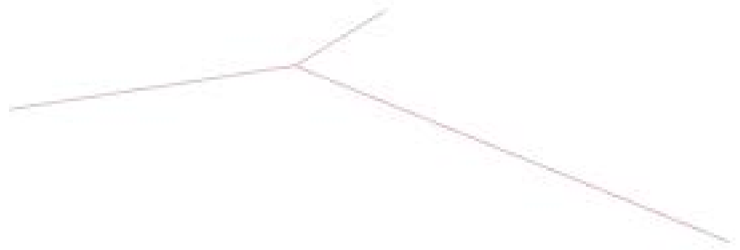


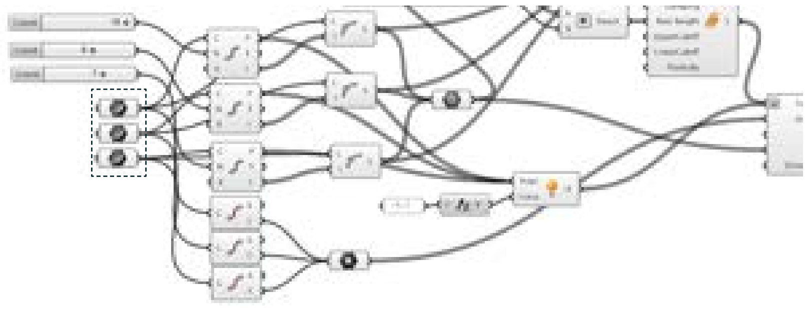


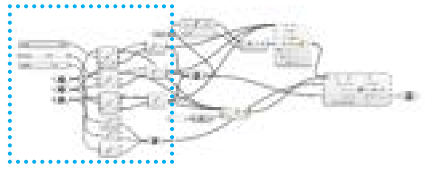
AP-08

Catenary Curve

Catenary is one of the most prevailing force-driven forms used in the field of architecture. This code shows the relationship between particles and springs and segmented polylines and vertices from the curve. The catenary form is made up of unary and spring force which represents the gravity and material properties.

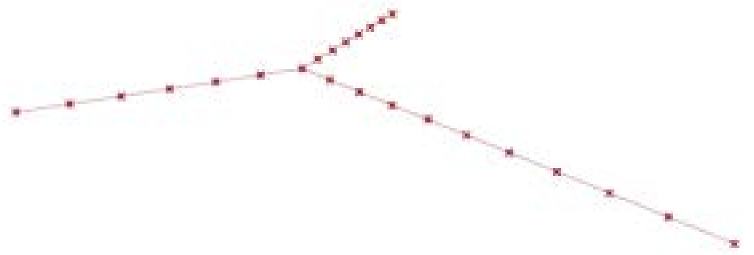


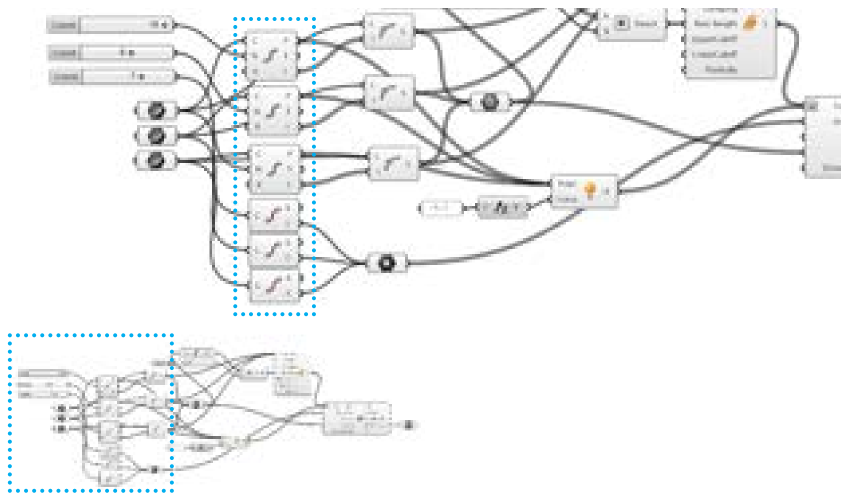




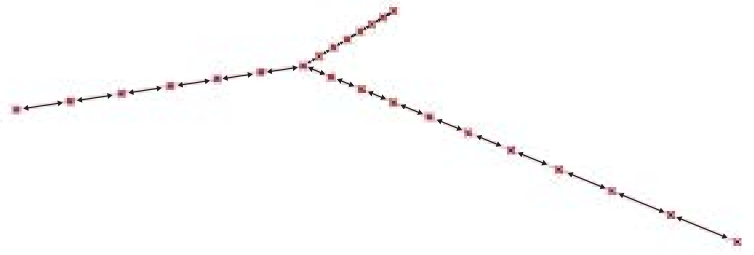
Point Matrix - List of Geometry

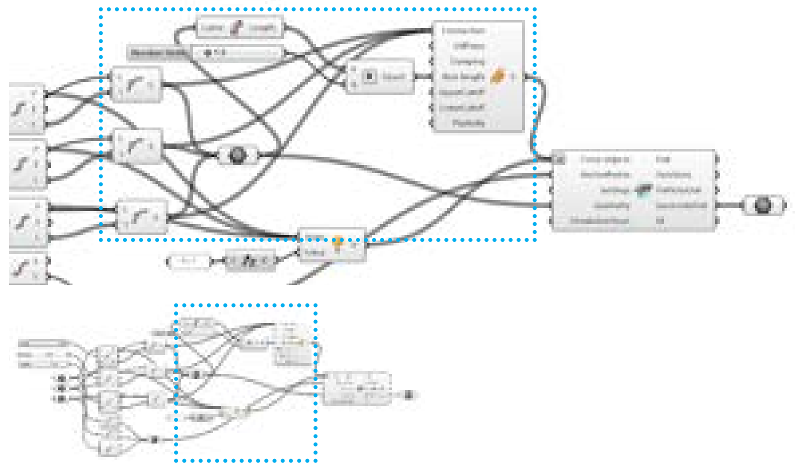
1. Draw 3 curves sharing one end point.
2. Set an open curve for each curve component.



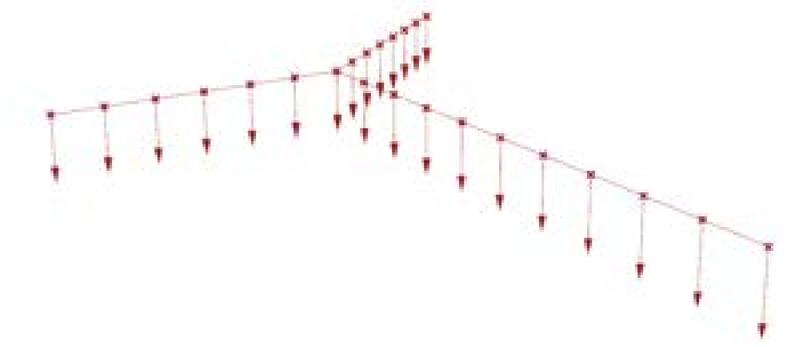


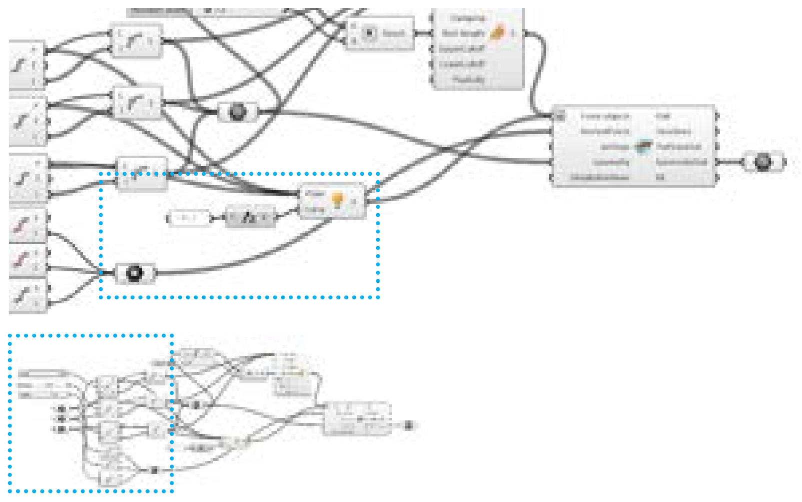
1. Connect curve output to curve (C) input of the divide curve.
2. Connect each number segment input of divide curve to the "Count" number slider.
3. Connect curve to the curve input of End point.





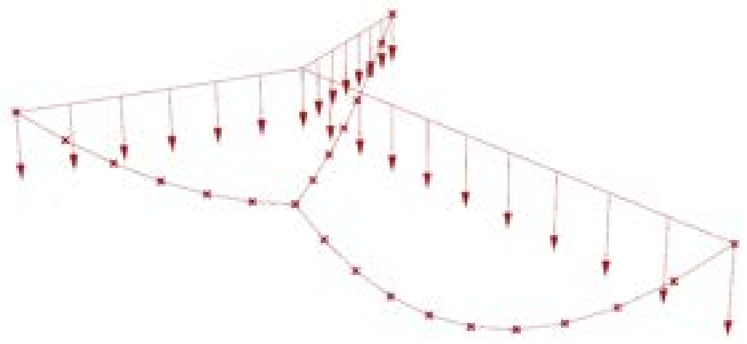
1. Connect Parameter as list (t) output to parameter as list (t) input of the shatter.
2. Connect segment as list (s) output to Connection (C) input of the Spring from Line.
3. Connect Segment as list to geometry, and Connect the geometry output to the curve input of the length.
4. Connect the length output of the length to the A input of the multiplication.
5. Set a number slider for B input of the Multiplication.
6. Connect the result output of the multiplication to the rest length input of the spring from l line.

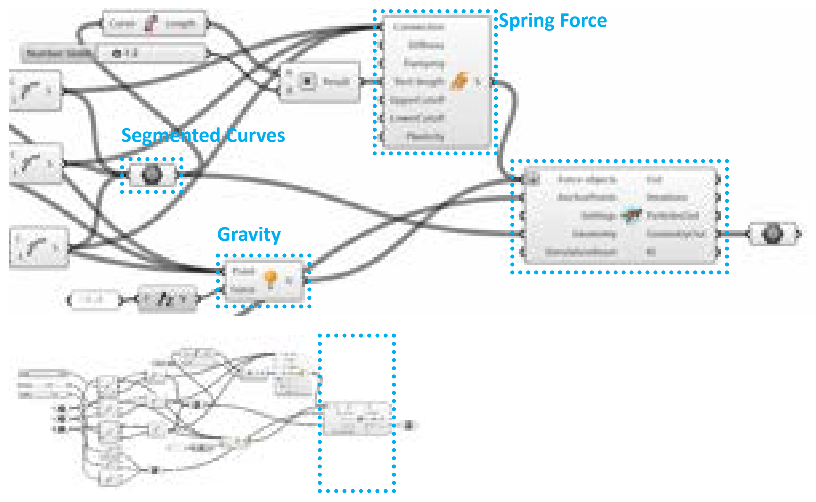




Force Set up : Gravity Force

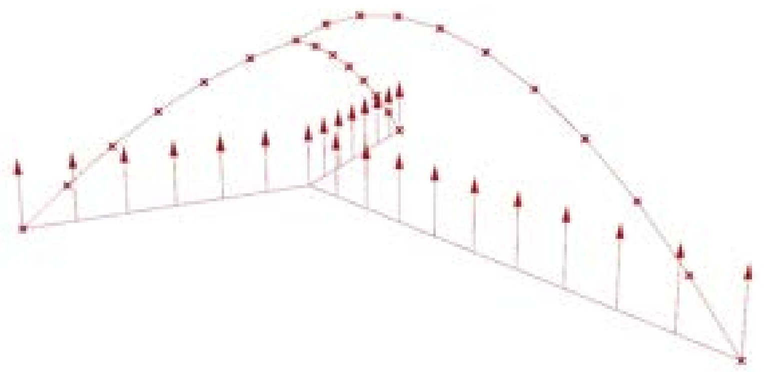
1. Connect the Point output of the divide curve to the Point input of the Unary force.
2. Connect Panel “-9.8” as a gravity force to Force input of Unary force.

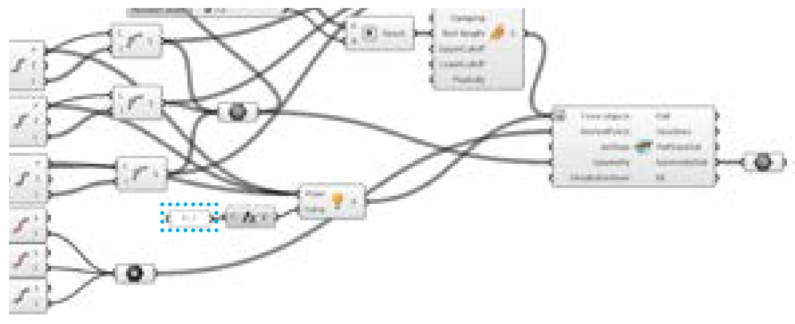


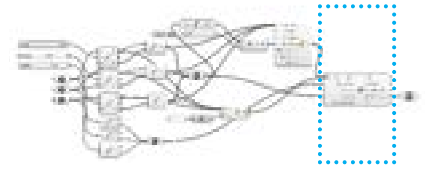


Simulation : Kangaroo Physics

1. Connect End point of the End curve to the Point component, and connect the output of the Point to the Anchor points input of the Kangaroo Physics.
2. Connect Spring (S) output of the Spring from line to the Force objects input of the KangarooPhysics
3. Connect Unary force output to the Force objects input of the Kangaroo Physics.
4. Flatten the Force objects input of Kangaroo Physics.

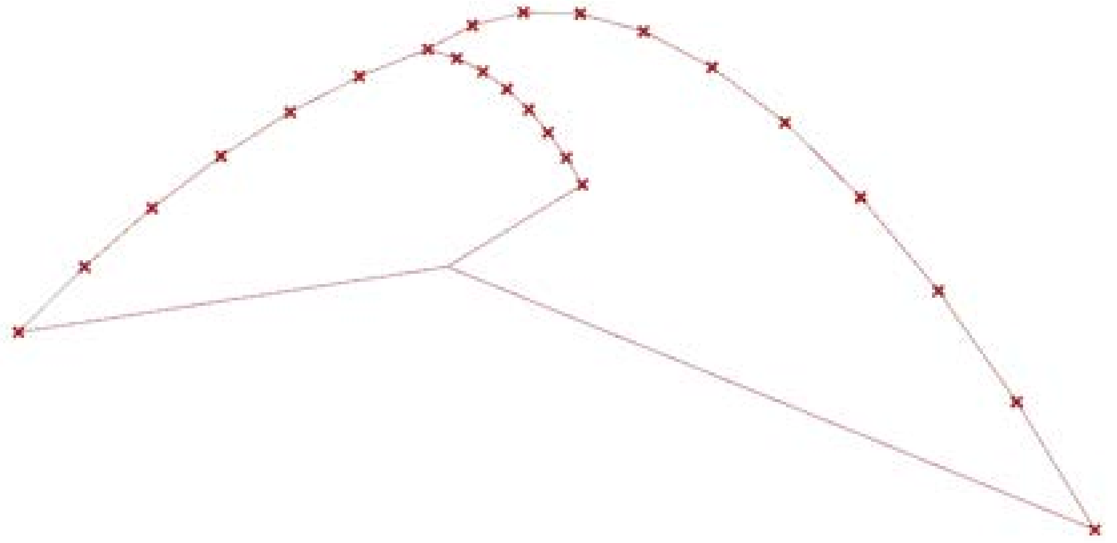






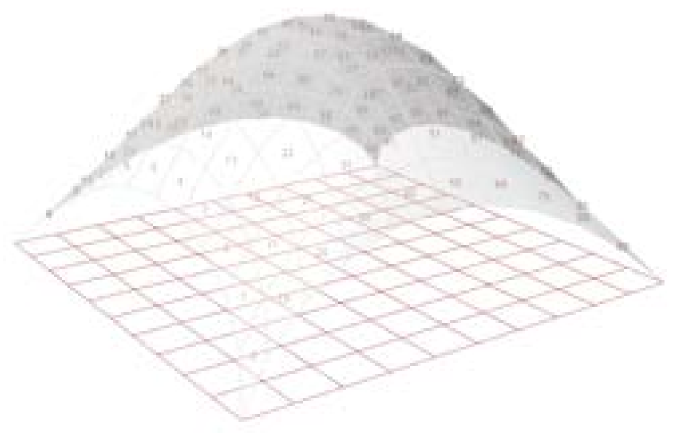
Reverse Gravity Force : Tension to Compression

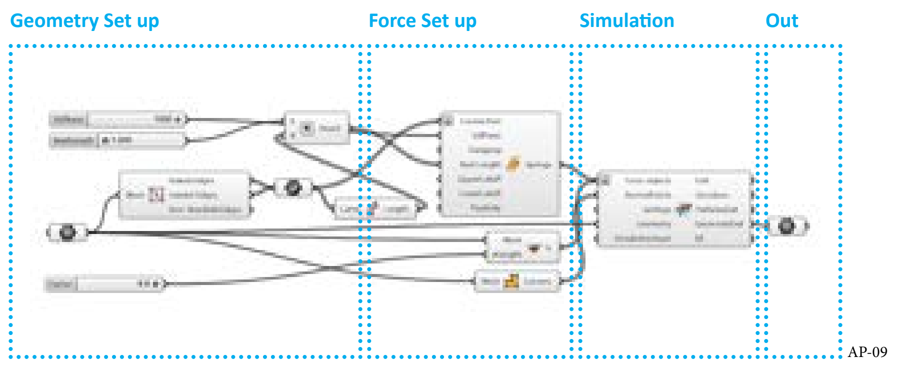
1. Make the value of Force positive (-9.8 --> 9.8) to reverse gravity direction.



Result & Contribution

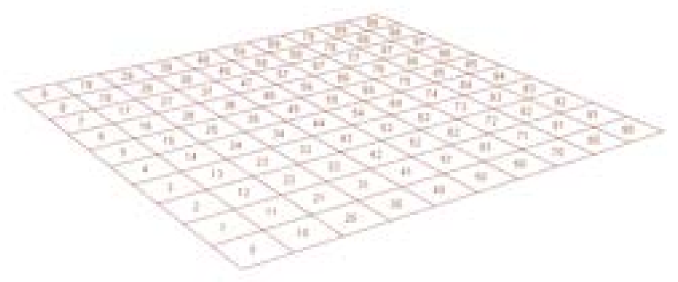
The principle of catenary simulation is from Gaudi's series of experimental models. However, the simulation is only able to represent morphological results. So, the process should incorporate research on specific materials and forces such as wind, live load, dead load and seismic lateral loads for further development.

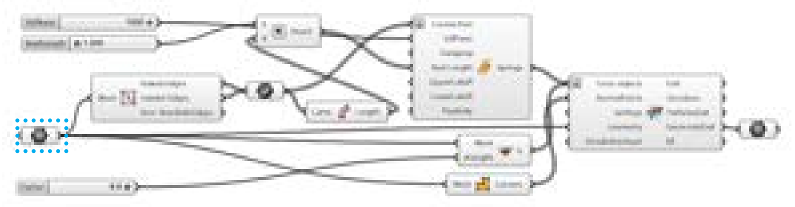


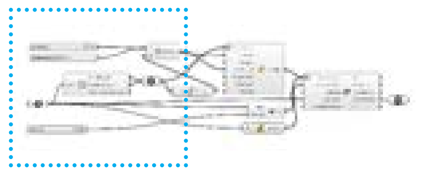


Catenary Surface

As an extension of the previous exercise, this session is about connecting the curves like a net and creating a collaborative flow of force on it by only applying gravitational force, which is one directional. Under the principle of catenary, the gravity(one-directional) enforces different vectors during simulation. The synthetic result represents a specific shape which is more natural than any other shape artificially derived.





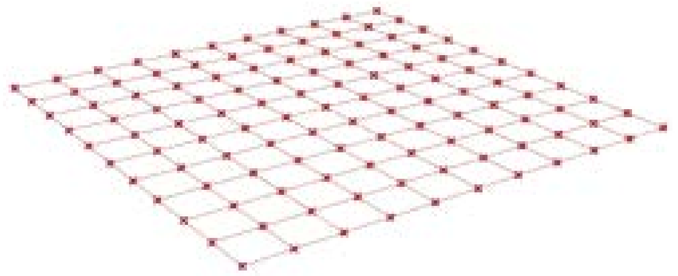


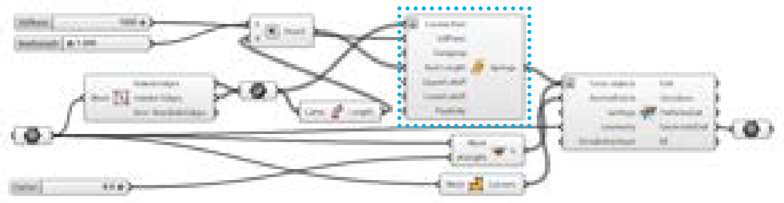
Base Geometry - Mesh

1 Set a mesh into the Mesh Component.

2. Connect the mesh to the Mesh(M) input of the Mesh edge.

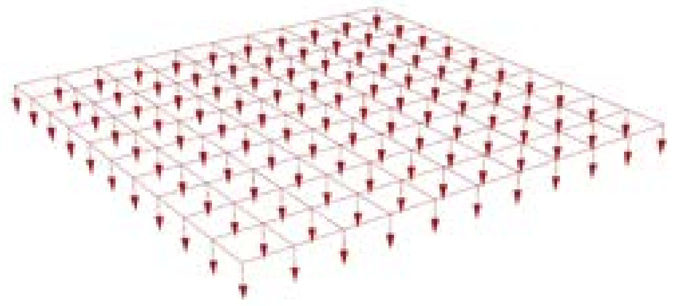
3. Connect Naked edges and Interior edges output to the curve component.

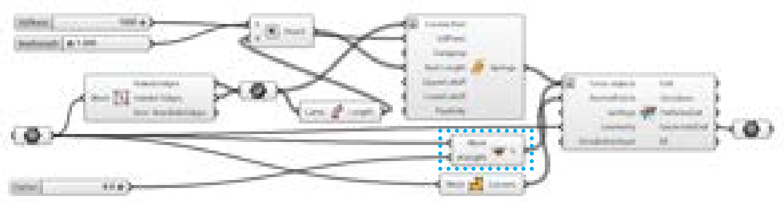




Force Object Set up: Spring Force

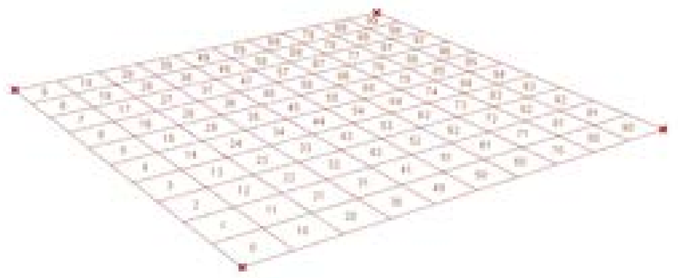
1. Connect curve to the Curve input of the Length, and connect Length output to the Rest length input of the Springs from line.
2. Connect Rest length number slider to the A input of the Multiplication.
3. Connect a number slider to the Stiffness input of the Spring from line. (Initial value for stiffness is 1.000)
4. Connect Curve to the connections input of the Spring from line.

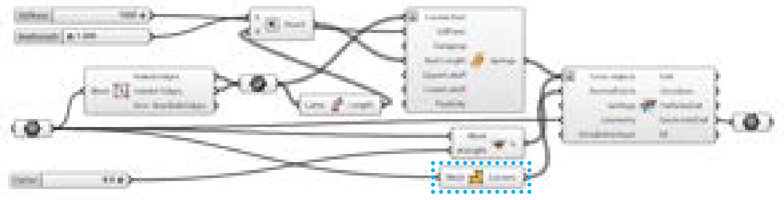




Force Set up : Gravity Force

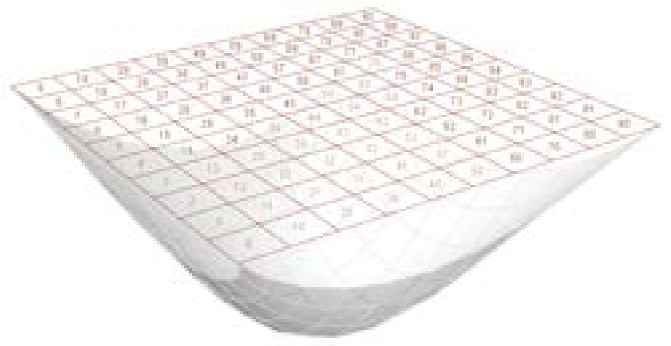
1. Previous Session's Unary force and its neighboring components can be simplified by using 'Gravity' Component.
2. Connect the mesh output to the mesh input of the Gravity.
3. Set a number slider to the strength input of the gravity.

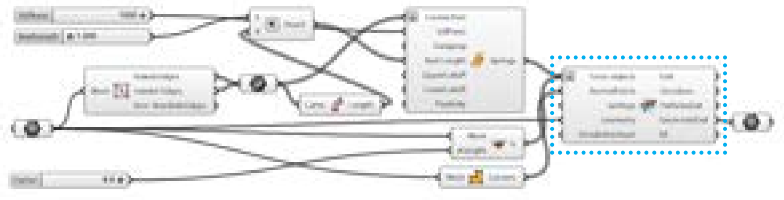




Anchor Point

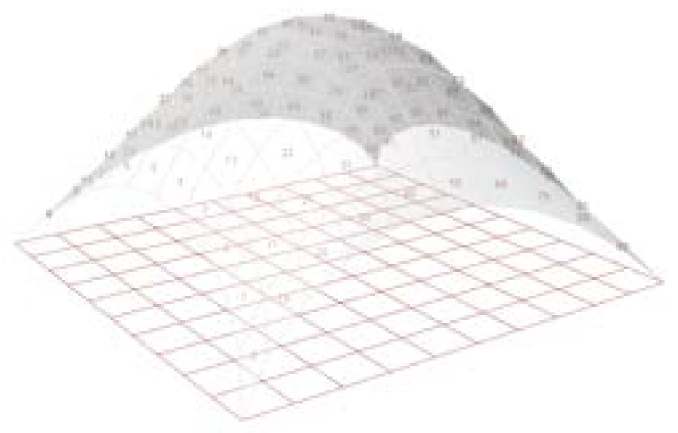
1. Connect the mesh output to the Mesh input of the Corner.
2. Connect the Corner output to the Anchor points of the Kangaroo Physics.

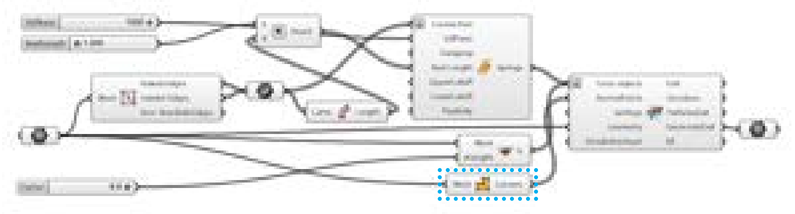


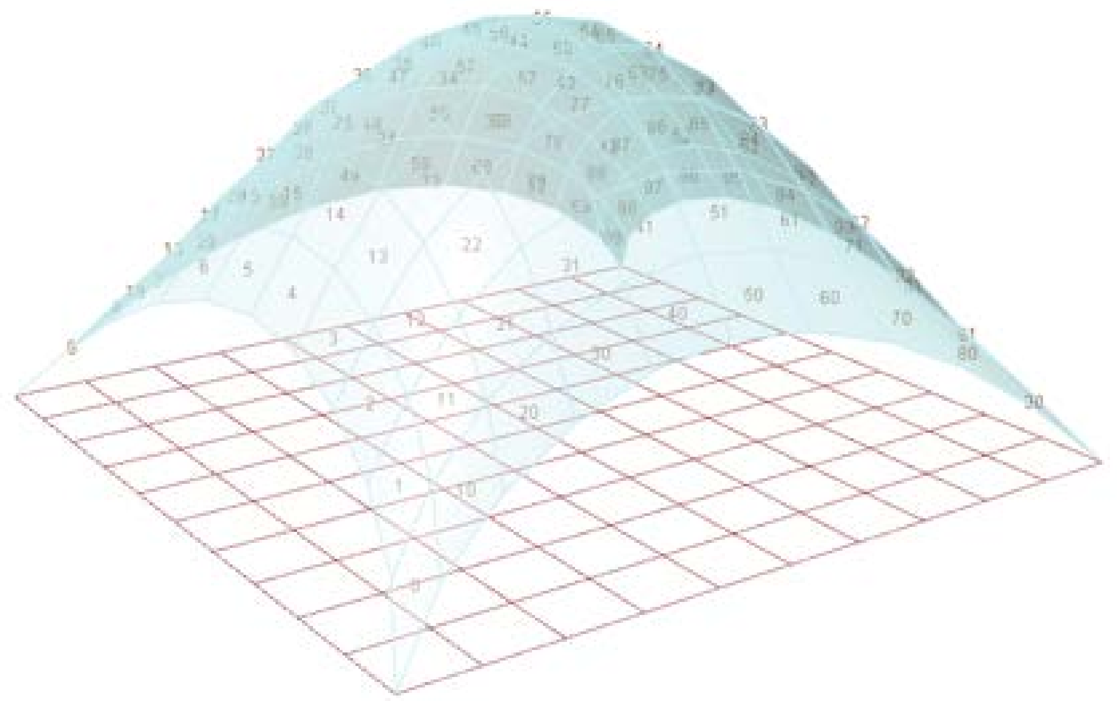


Simulation : Kangaroo Physics

1. Connect Spring (S) output of the Springs from Line to the Force objects input of the Kangaroo Physics.
2. Connect Gravity (G) output of the Gravity to the Force objects of the Kangaroo Physics.
3. Double click the Kangaroo Physics icon and press the play button for simulation.





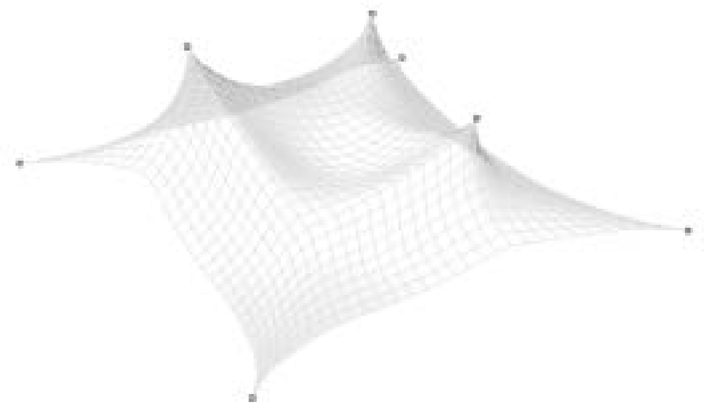


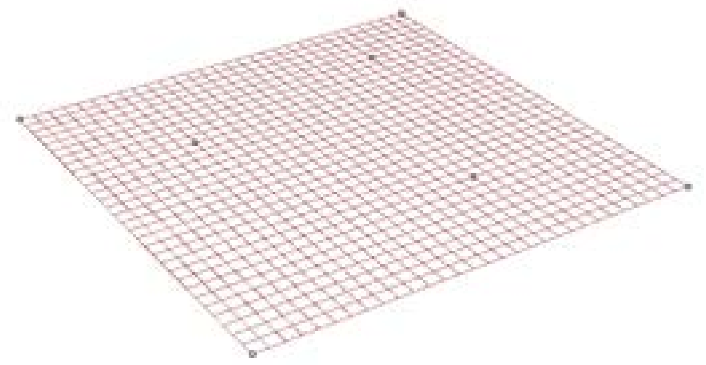
Reverse Gravity Force : Tension to Compression

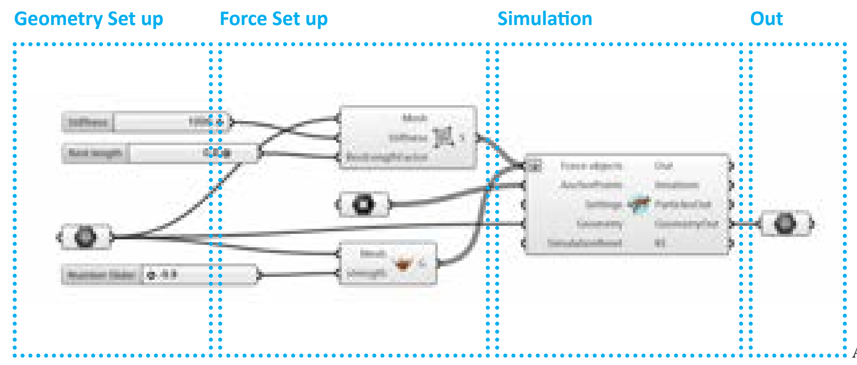
1. Connect Factor number slider range from negative to positive floating number to the strength input of Gravity.

Result & Contribution

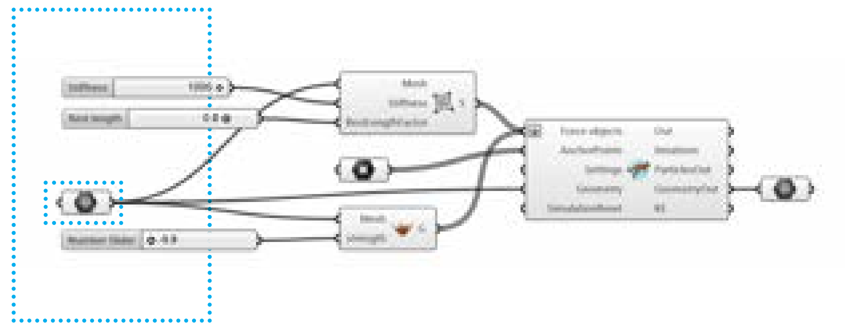
Final process of mirroring the shape to make it float upwards is necessary if the members or plates are designed to resist compression forces. The major benefit for architects to use this simulation is that they can gain the skill to create funicular shapes, which are optimized from an engineering perpective.







AP-10

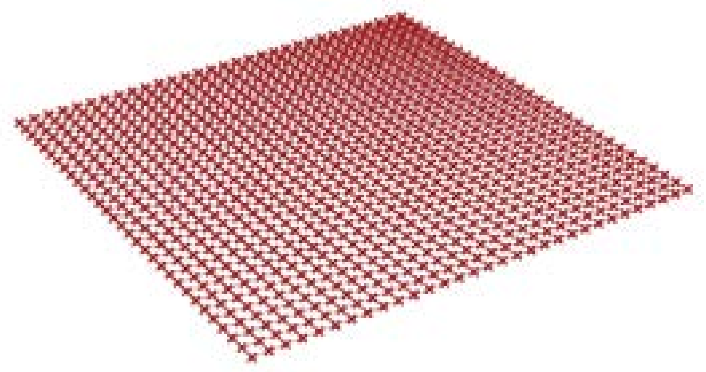


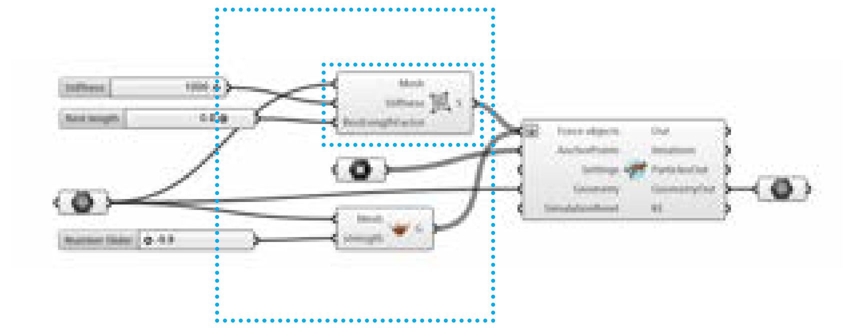
Tensile Form

Tensile form shows similar behaviors to the catenary form finding model. However, the critical difference between these two forms is the existence of external force. In tensile structures, this external force could be shown as a mast or another tensile cable that holds a specific point of the surface. So, the designer is able to identify the tension and compression points with design intention under the external forces.

Base Geometry - Mesh

1. Set a mesh plane to the Mesh component.
2. Set points from the corner of the plane and the points from the plane.

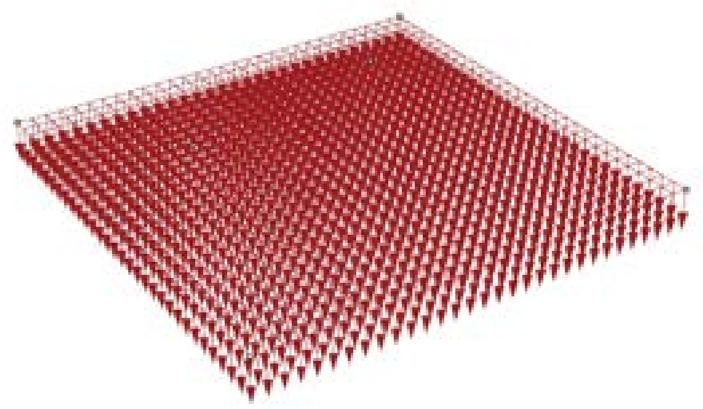


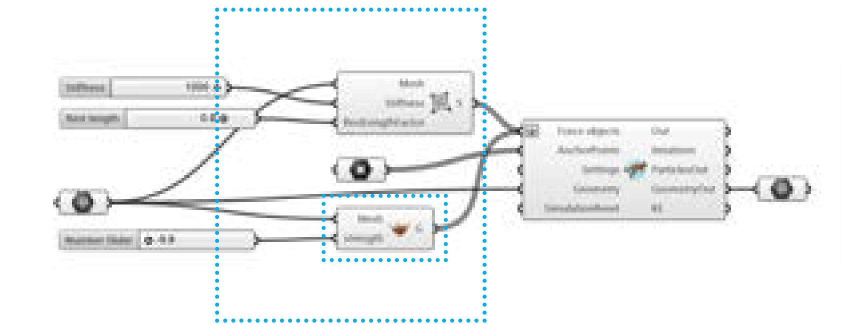


Force Set-up: Spring

1. Set a Stiffness number slider to the Stiffness input of the Spring from mesh.
2. Connect mesh output to the mesh input of the Spring form mesh.
3. Set the Rest length number slider to the Rest length factor input of the spring from mesh.

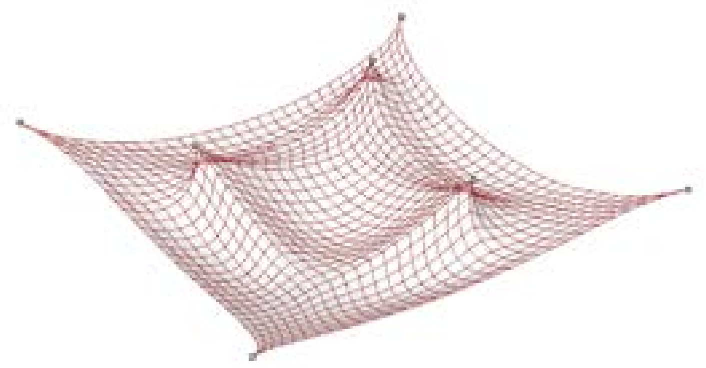
Previous session's 'spring from line' component and its neighboring components can be simplified by using mesh from spring component.

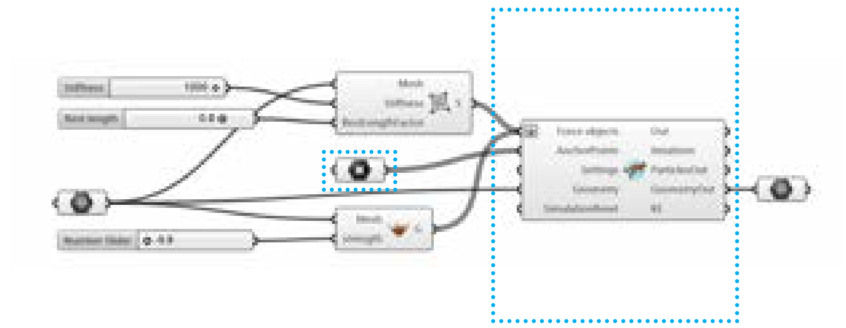




Force Set-up: Gravity

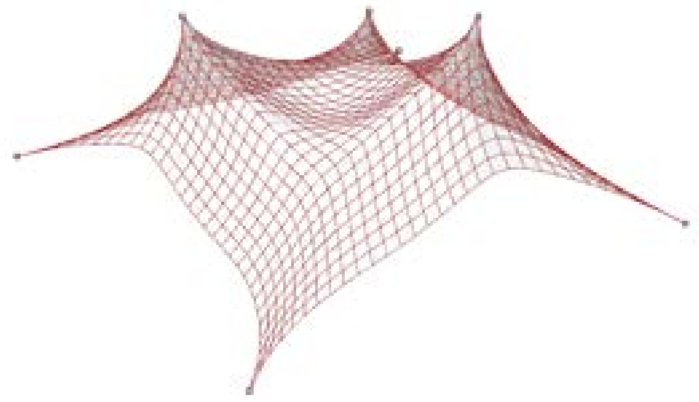
1. Drag and drop Gravity.
2. Connect mesh to the Mesh input of the Gravity.

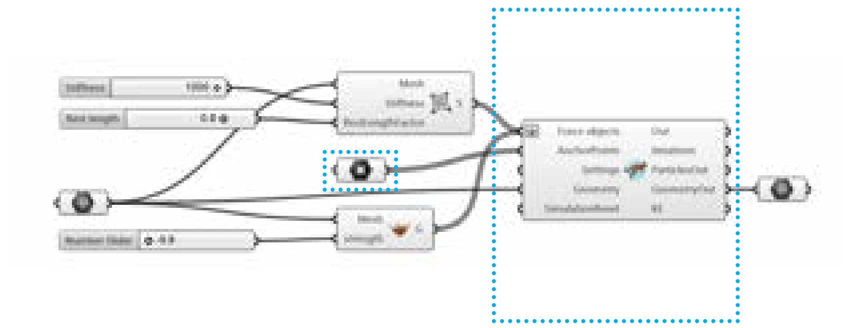




Anchor Point

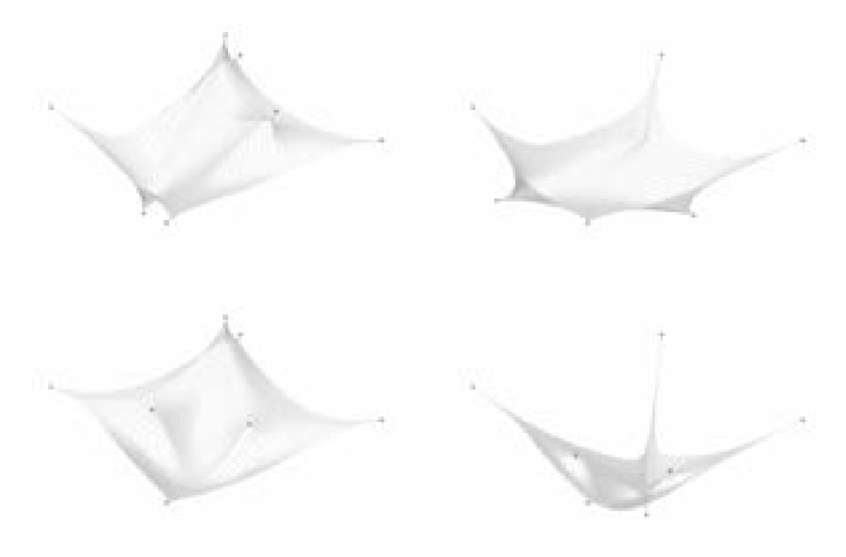
1. Set points from the corner of the mesh and points from the surface.

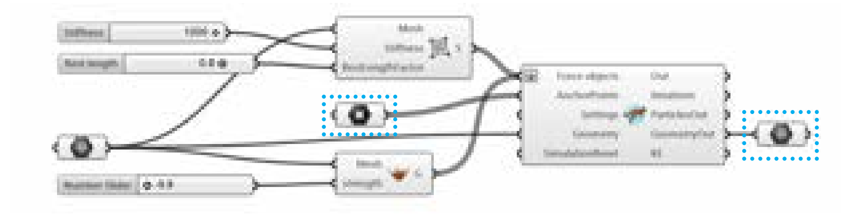




Simulation : Kangaroo Physics

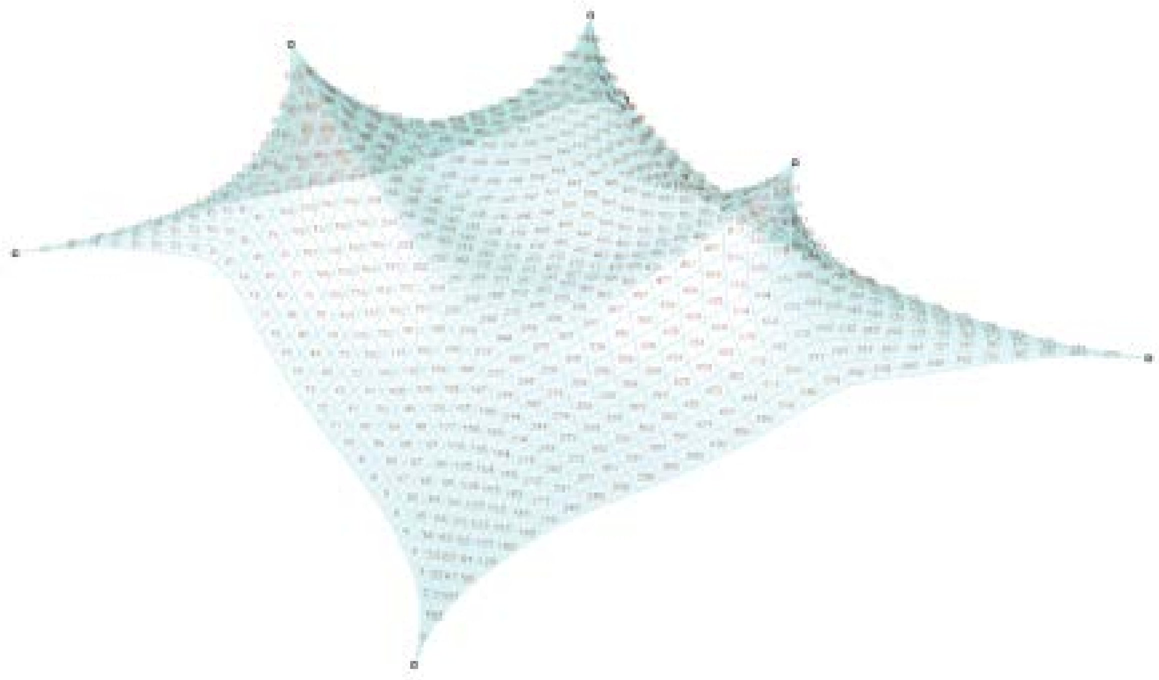
1. Flatten the Forces.
2. Connect points to the Anchor points input of the Kangaroo physics.
3. Connect original mesh to the Geometry input of the Kangaroo physics.





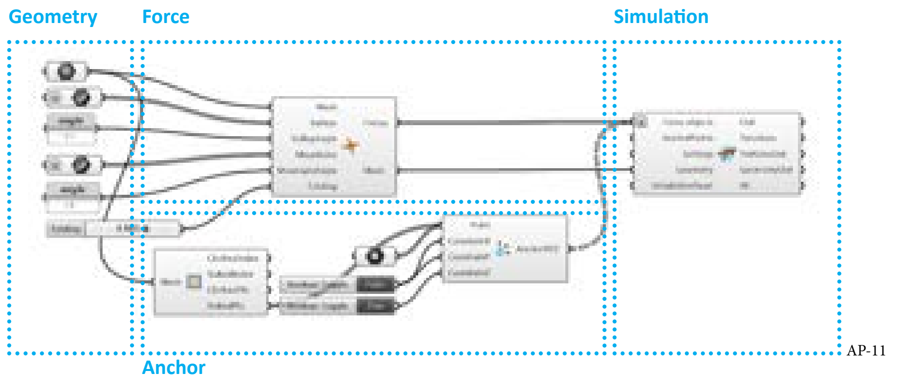
Anchor Adjustment

Moving the Anchor point geometries in rhino, it is now interacting with Kangaroo Physics. Once the point is fixed, simulation will slowly reach its optimized form in terms of given force.



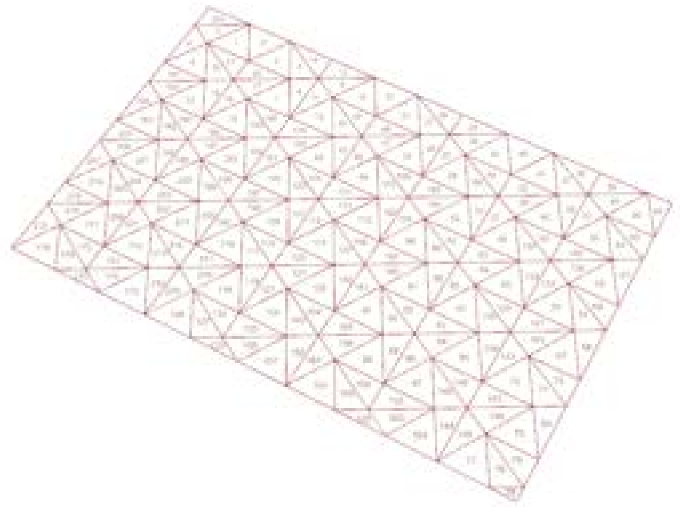
Result & Contribution

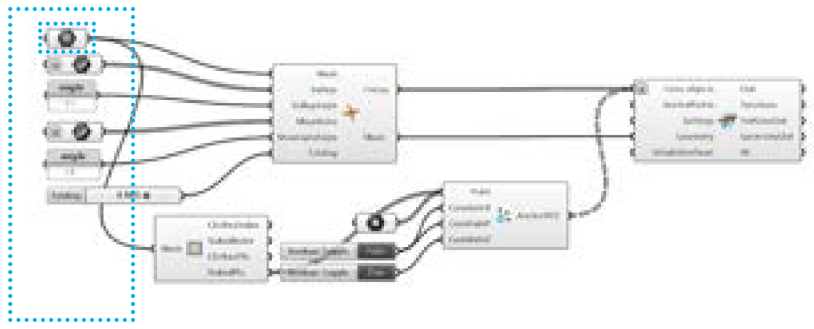
This is needed to simulate the shape of the membrane/tensile structures which has a mast or cables as mentioned earlier. This code is great for having an initial idea of soft material that is usually hard to visualize in a typical cad environment.



Folding Form

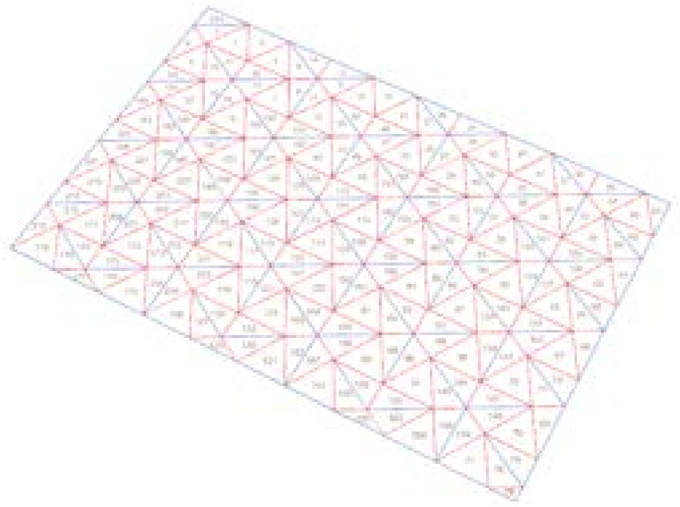
There is a wide range of applications for folding behavior. Folding is also one of the most often used technique in both architecture and engineering fields. Many of the designs are created based on this logic but are difficult to translate into a 3d model and do simulations with. This session introduces a pathway to simulate this type of folding technique through computation.

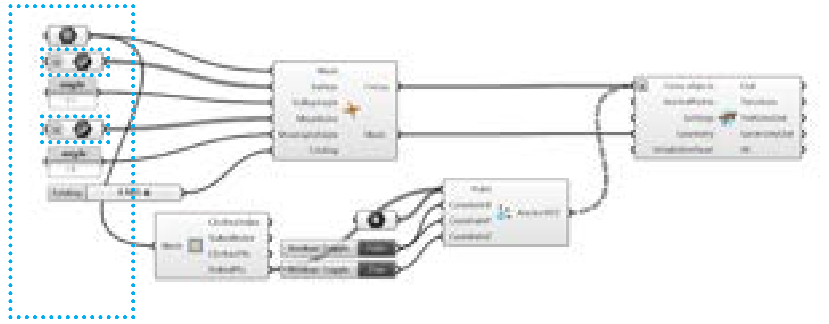


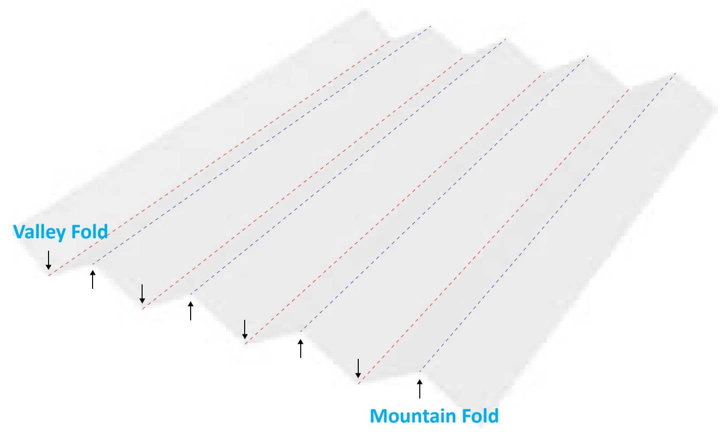


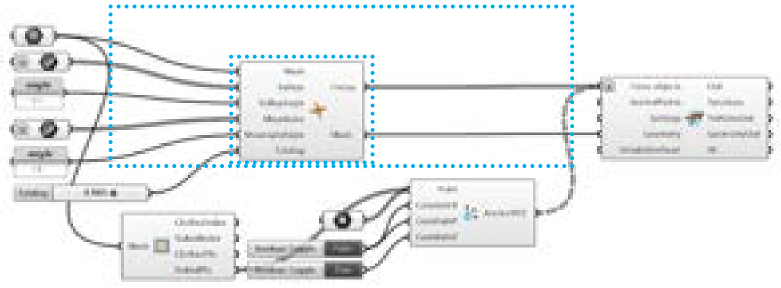
Base Geometry - Mesh

1. Set a mesh plane to the Mesh (M) input of the Origami.
2. Set curves to the Valleys input of the Origami.
3. Set curves to the Mountains input of the Origami.
4. Set angle as a pi and -pi, and connect to the Valley angle and Mountain angle input of the Origami.







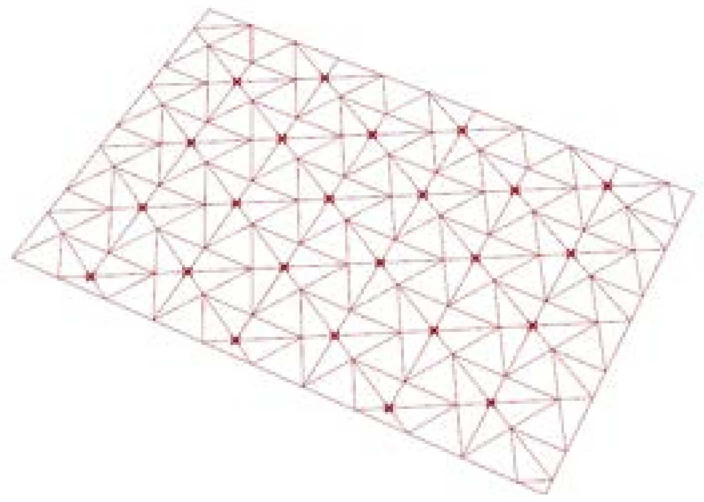


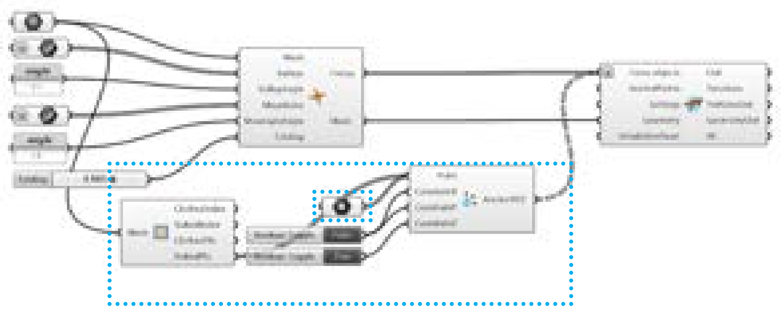
Ron Resch Pattern

This pattern is very typical in the world of Origami (or Paper folding) and is referred to in this session of folding simulation. The Ron Resch Pattern starts from a triangular grid which is aligned in zig-zag pattern(Red lines), and another triangular folding which is folded in a different direction crossing the triangular grid (Blue lines). These creasing patterns are folded into 3 dimensions with specific form and can be used as a design driver.

Valley & Mountain Fold

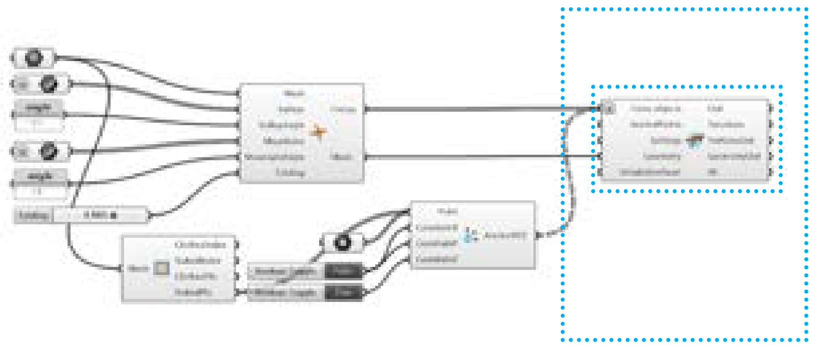
Valley & Mountain is the naming system for the creasing pattern, meaning that these two different groups have different directionality in terms of folding.





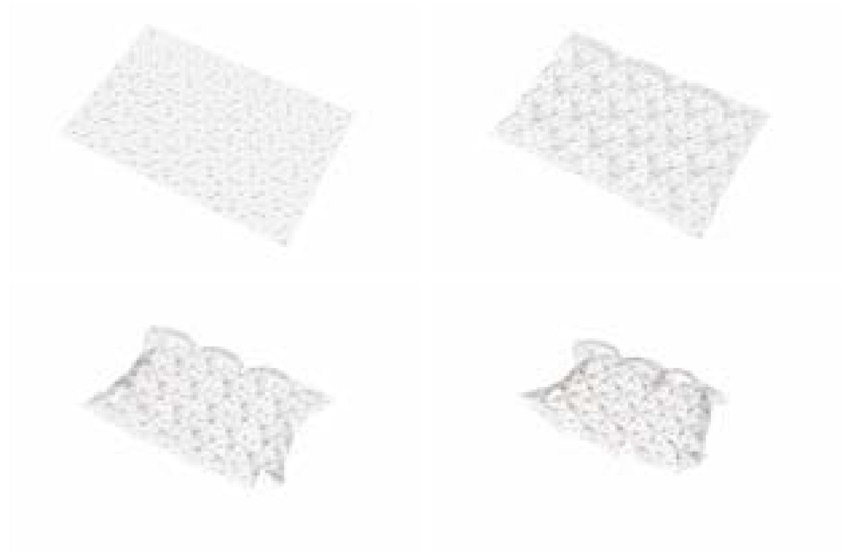
Anchor Point

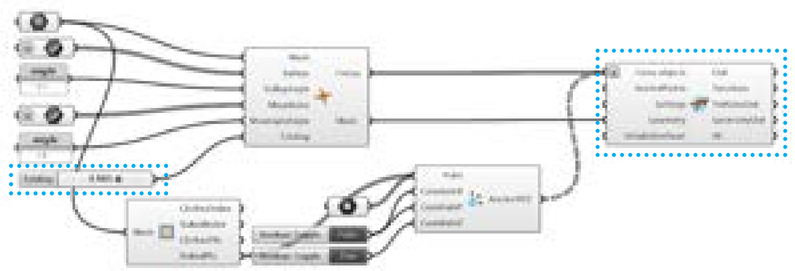
1. Set all the intersection points of the Valley fold and connect to the Point input of Anchor XYZ.
2. Set points from boundary of the mesh plan to the Point input of Anchor XYZ.
3. Set the Boolean Toggle false to contrain X, Y input of Anchor XYZ.
4. Set the Boolean Toggle true to contrain Z input of Achor XYZ.



Simulation : Kangaroo Physics

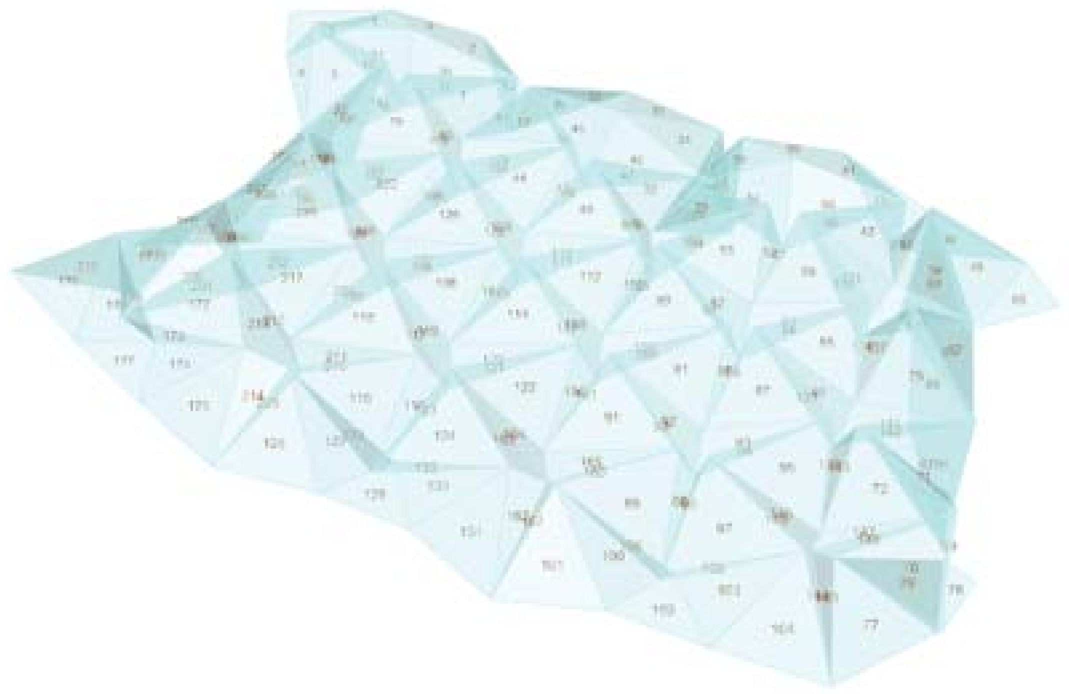
1. Connect Force of the origami to the Force objects input of the Kangaroo Physics.
2. Connect Anchor XYZ output to the Force objects input of the Kangaroo Physics.
3. Connect Mesh output of the origami to the Geometry input of Kangaroo physics.





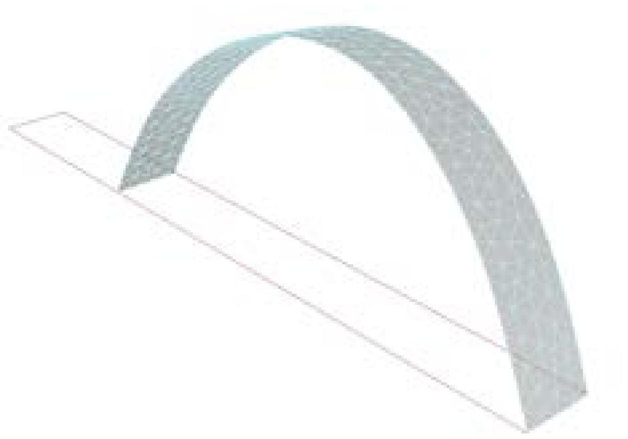
Force Set-up: Folding

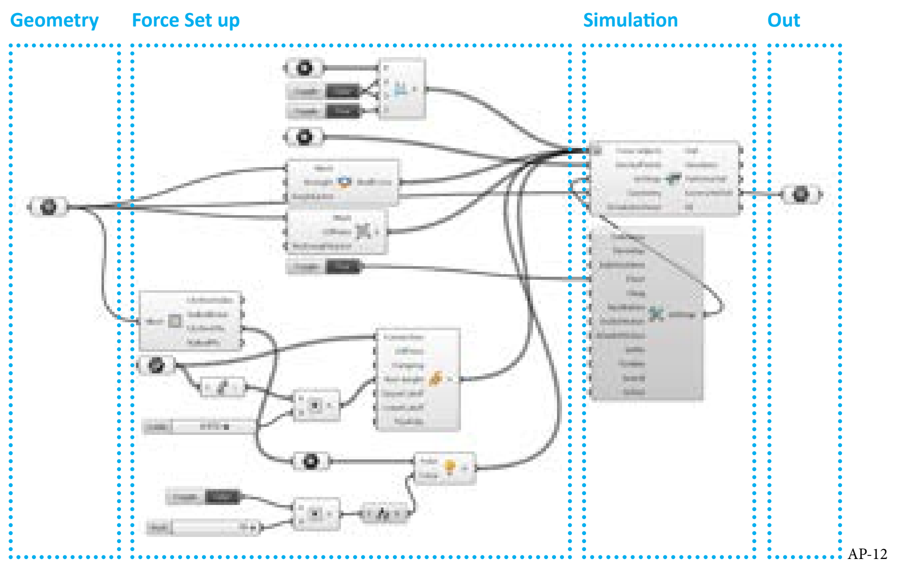
1. Move the Folding number slider to change the degree of folding strength from 0 to 1.
2. ‘0’ refers to a flat surface and ‘1’ represents a perfectly folded state.



Result & Contribution

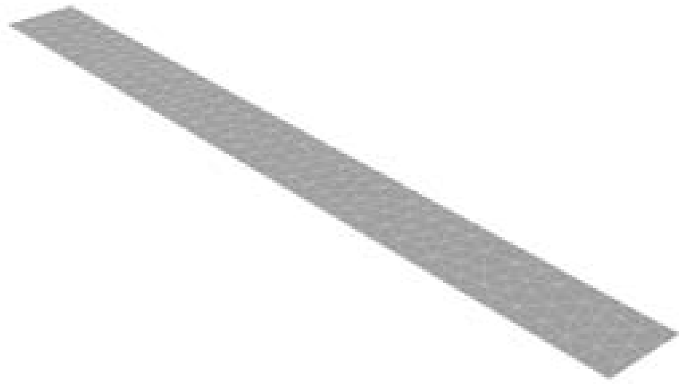
Folding has inspired designers within the field of design and architecture, but needs to be furthuer developed in terms of technique and application. This exercise allows architects and engineers to access the design process. Even though it can be applied in small prototype scales such as a paper size, it still requires further exploration in defining architectural details. For instance, the joinery between the surfaces. It is also important to develop this area further for folding in large scale applications.

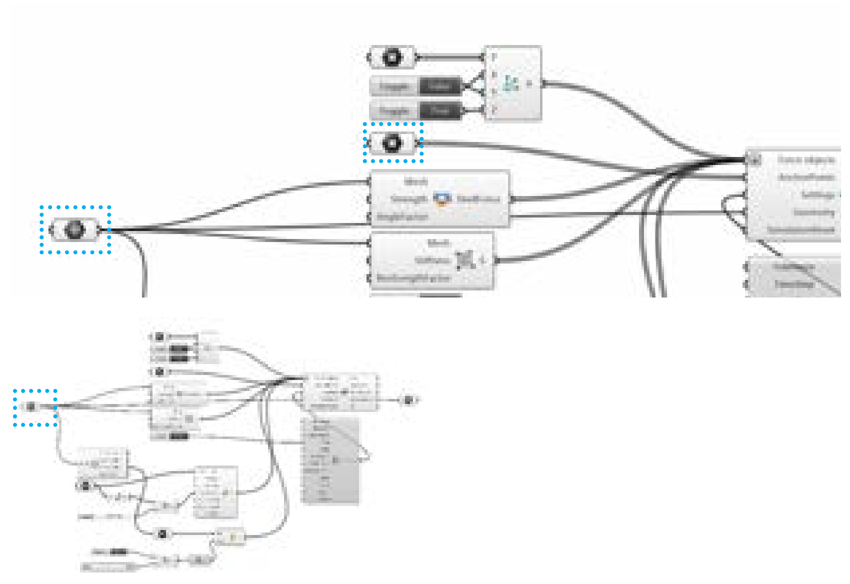




Bending Form

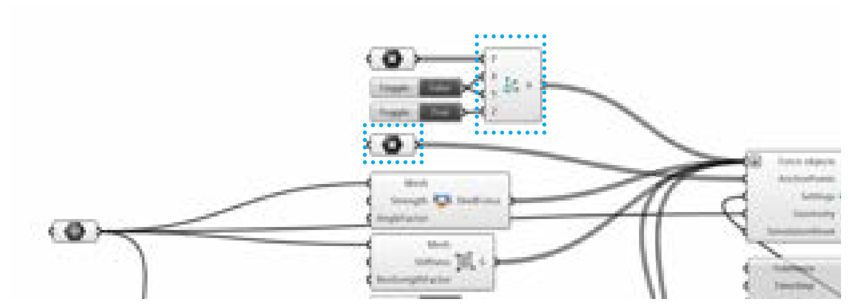
Bending form is one of the most delicate forms that a designer can create with this physic simulation tool. The form can be created by applying the unfolding force on every edge of the mesh over the Spring-particle system.

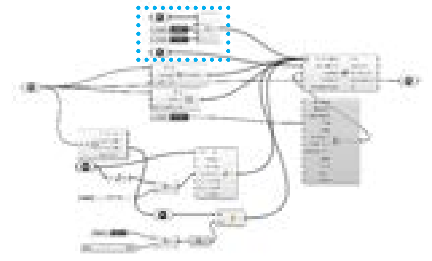




Mesh Preparation

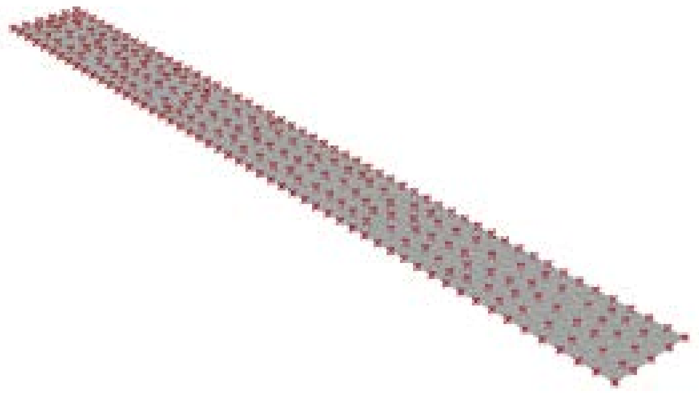
1. Use a Mesh Machine to remesh the component.
2. Use a mesh from the Mesh Machine as the base geometry for bending simulation.

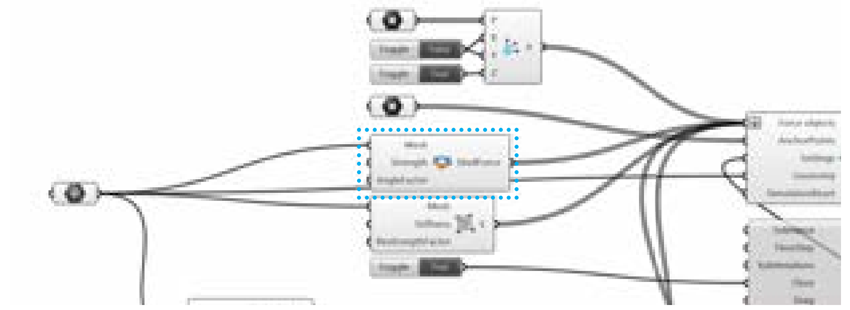


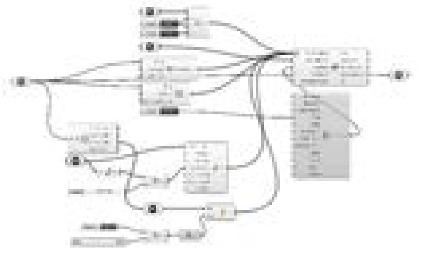


Anchor Point

1. Set Anchor points on the corner of the short edge with two points.
2. Set Anchor XYZ from the other corner of two points.
3. Set False on Constraint X and Y.
4. Set True on Contraint Z.



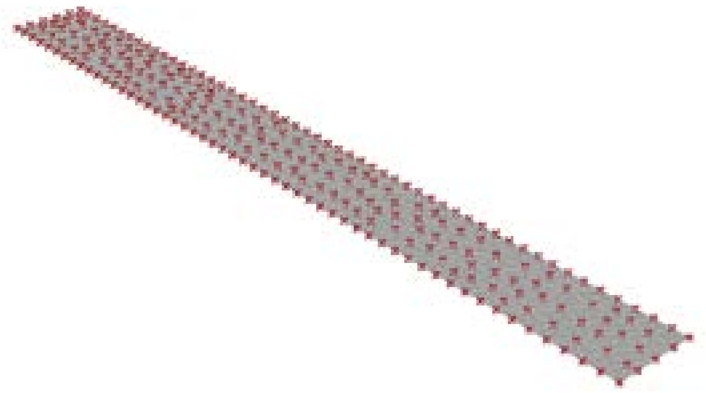
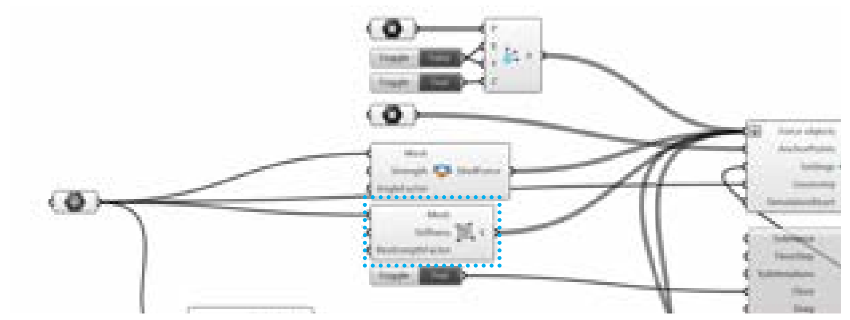
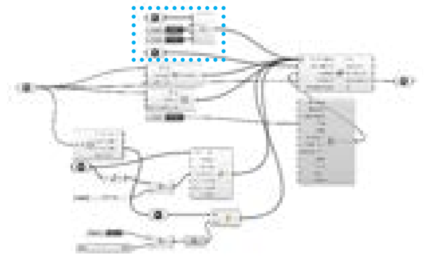




Force Set-up: Shell Force

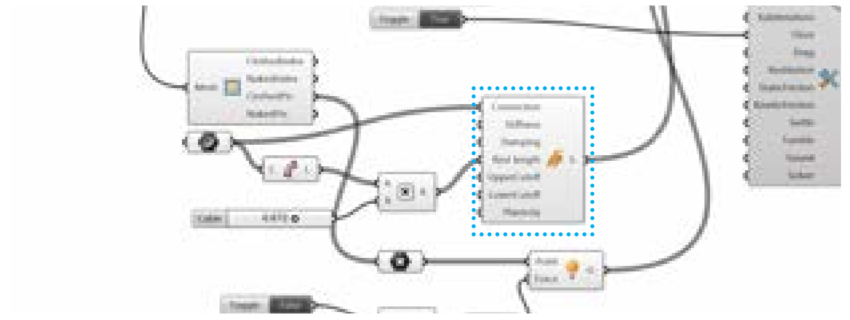
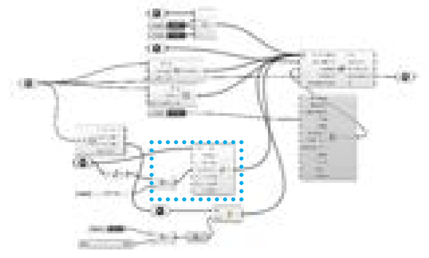
Shell force component representing the force works on every edge of the mesh to unfolding the neighboring surfaces to reach to the flat status with assigned force.

1. Connect Mesh to Shell Force component.
2. Connect 'ShellForce' output to Force objects of the Kangaroo physics.

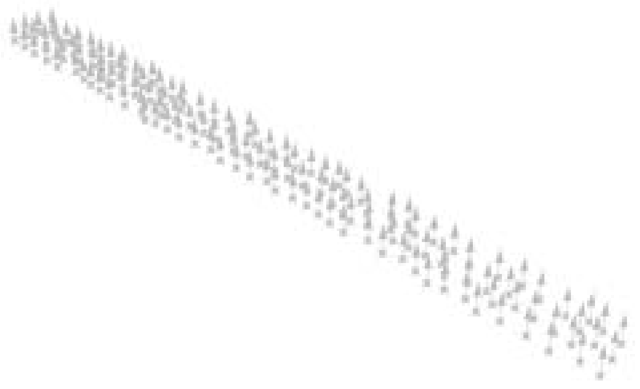
Force Set-up: Spring

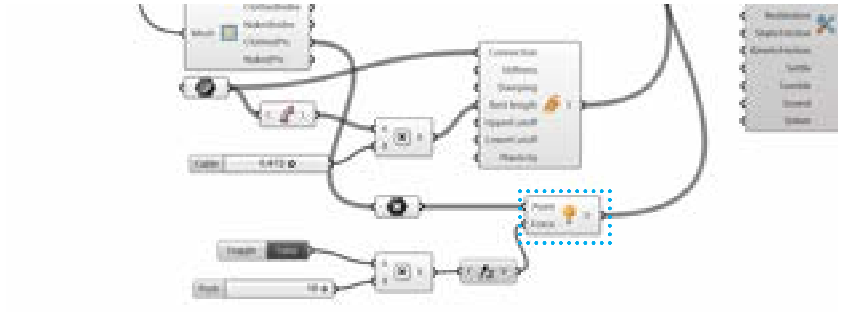
1. Set Anchor points from the corner of two points from the short edge.
2. Set Anchor XYZ from the other corner of the two points.
3. Set False on Constraint X and Y.
4. Set True on Constraint Z.

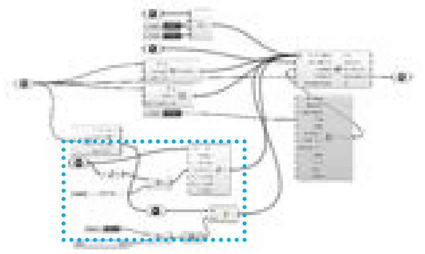



Force Set-up: Spring for Cable contractor

1. Set two curves from the edge of the long side.
2. Connect curve to the Connection input of the Springs.
3. Connect curve to the curve input of the length and connect A input of the Multiplication.
4. Set number slider to the B input of the multiplication.
5. Connect Result input of the mulplication to the Rest length input of the springs.





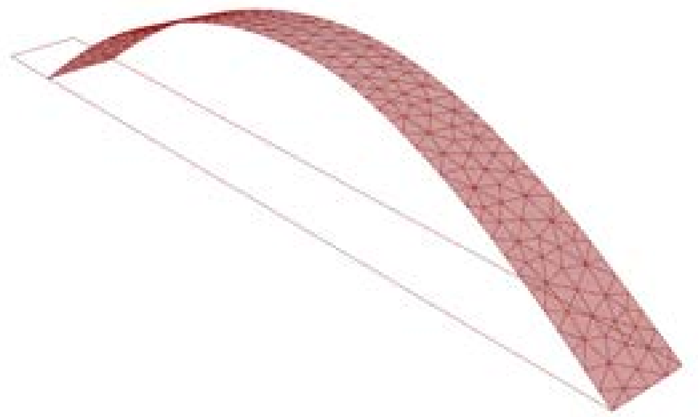


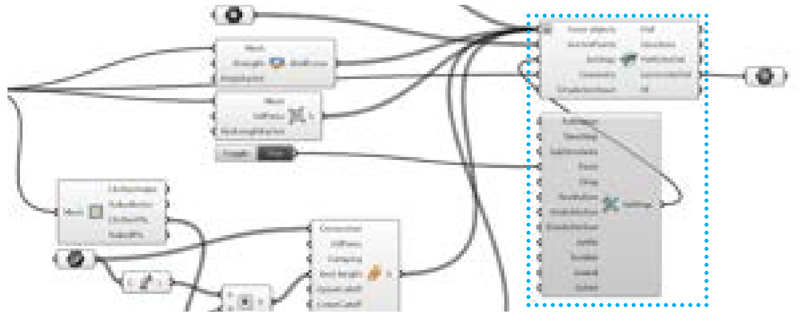
Force Set-up: Temporal Gravity

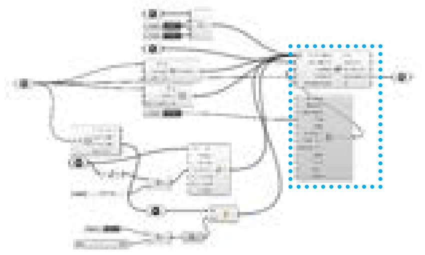
1. Connect Mesh to the Mesh input of the Naked vertice.
2. Connect ClothedPts to the point and connect the Point to the Point input of the Unary force.
3. Connect Unit Z to the Force input of the Unary Force.
4. Connect Result(R) output of the Multiplication to Factor input of the Unit Z.
5. Connect Boolean toggle to A input of the Multiplication.
6. Connect number Factor to the B input of the Multiplicaion.

Temporal Gravity :

This is Unary force object for the Gravity force application. Since the bending is requiring the initial direction of the bending, This needs to be applied at the very first of the simulation. However, to make the simulation to have a valid result, this component needs to be turned off just after the bending has the initial direction.

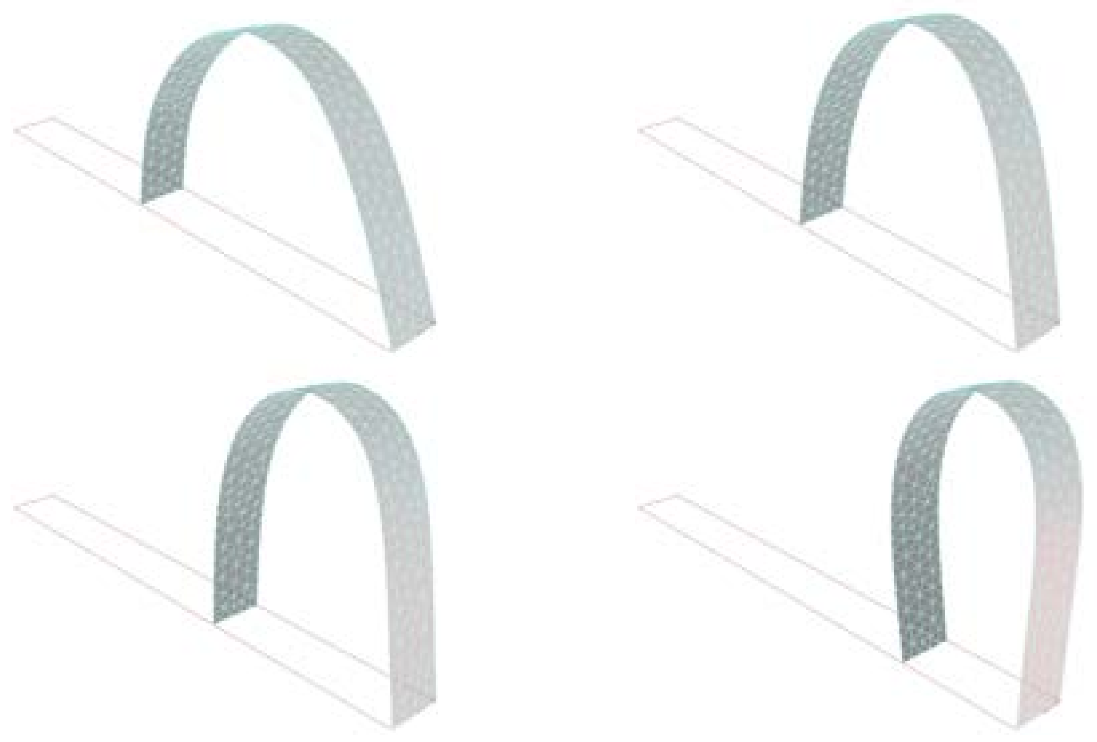


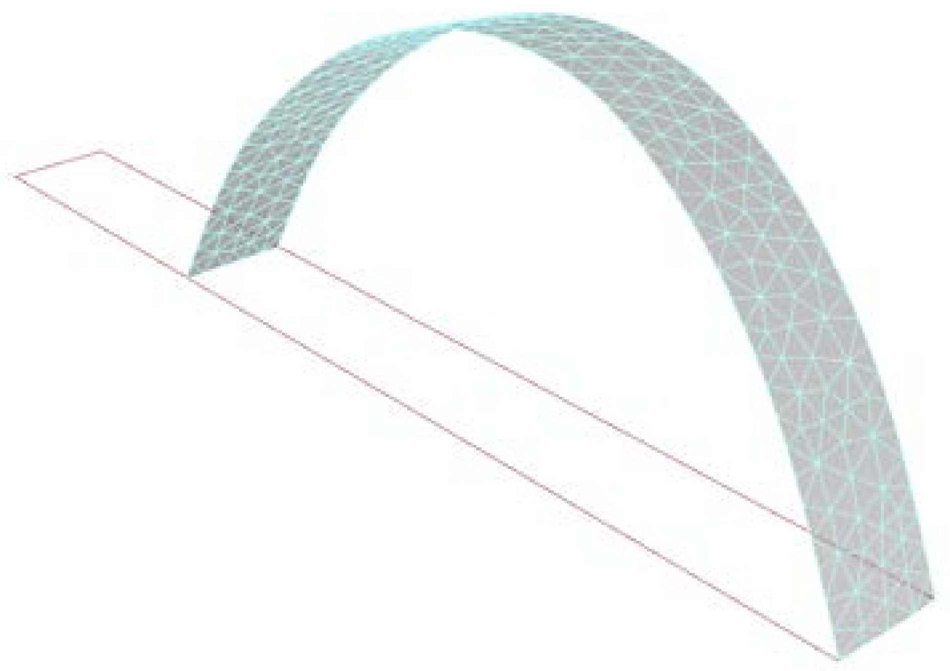




Kangaroo Setting

1. Connect Anchor XYZ output, Hinges(Shell Force) output of the Shell, Springs output of spring from mesh, and Unary force (U) output to the Force objects input of the Kangaroo Physics.
2. Set a Kangaroo Setting, and Connect Boolean toggle True to the floor input of the Kangaroo Setting.
3. Connect the mesh to the geometry input of the Kangaroo Physics.



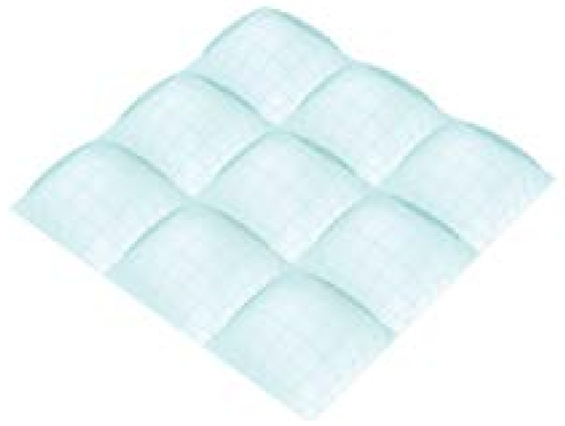


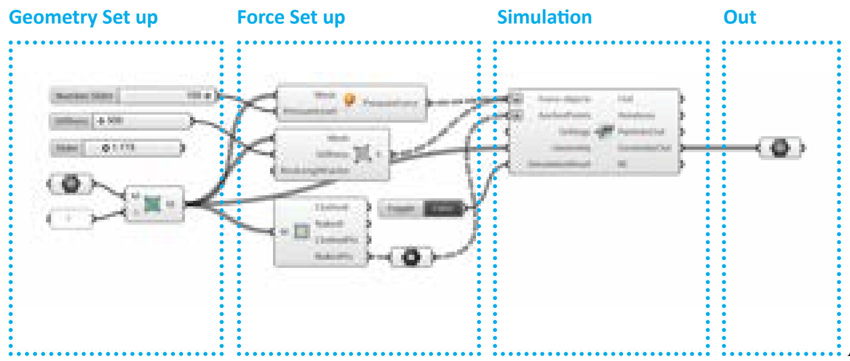
Simulation : Kangaroo Physics

1. Double click the Kangaroo Physics icon and press play button for simulation.
2. Regulate the Cable parameter to adjust the degree of contraction between two points.
3. Set 'False' for the boolean toggle that connected to the Unary force component to turn- off the temporal gravity force.
4. Wait for the optimized bending shape until the geometry stops moving.

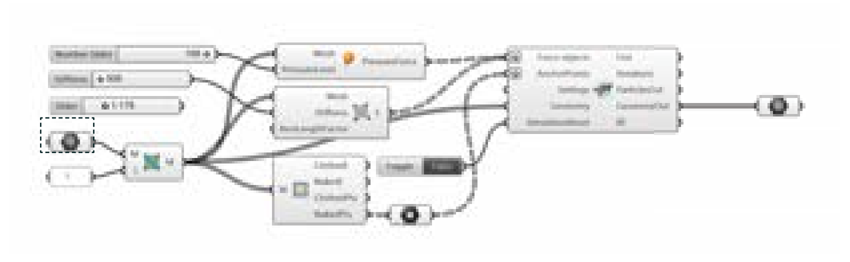
Result & Contribution

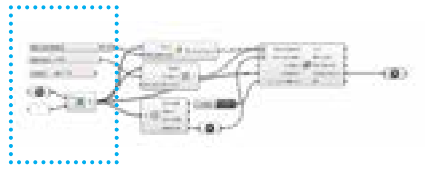
Bending form creates the most natural curvature in a given setting due to the fact that every edges of the mesh tries to unfold with the same amount of force. The bended geometry itself can resist much more weight or stress only with a single surface as the engineered form.





AP-13



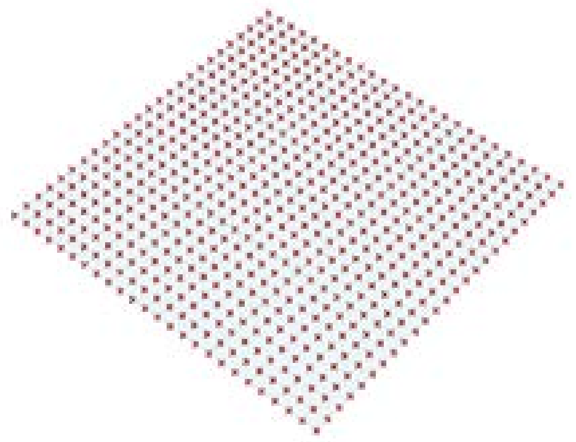


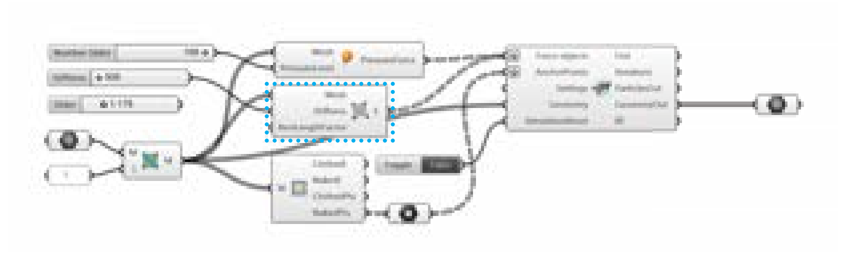
Inflation Form

The Inflation session is about how to create a balloon-like shape. It is sharing the same logic with the previous Kangaroo sessions which originates from the spring-particle system. This simulation uses pressure force as a main driver for the form making process.

Base Geometry_mesh

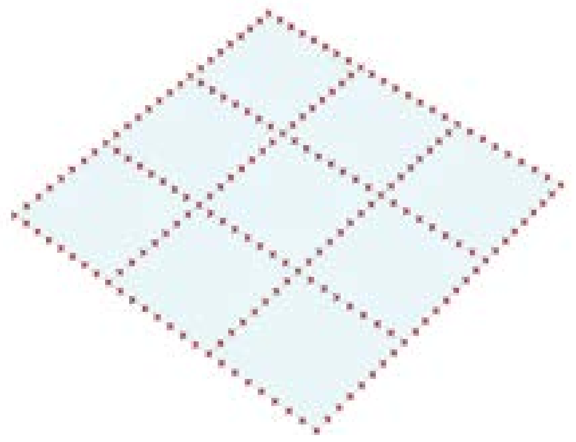
1. Set a Mesh Plane and connect mesh to the Mesh input of the Byparent.
2. Set a degree number for the sub division and connect to the Level input of the Byparent.

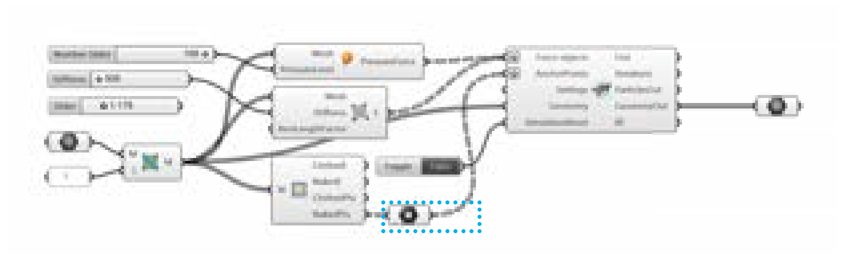




Force Set-up: Spring

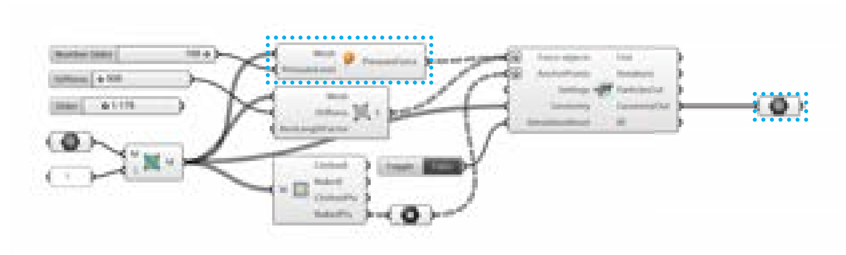
1. Connect the Mesh of List output of the Byparent to the Mesh input of the Spring from mesh.
2. Connect the Number slider to the Stiffness input of the Spring from mesh.





Anchor Point

1. Connect the Mesh as the List output of the Byparent of the Mesh input of the Naked vertice.
2. Connect nakedpts to the point component.
3. Connect Point to the Anchor points input of Kangaroo Physics.



Force Set-up: Pressure

1. Connect mesh output of the byparent to the mesh input of the mesh pressure.
2. Connect number slider to the presure level input of the mesh pressure.
3. Connect the pressure force from mesh pressure to force object input of Kangaroo physics.

Result & Contribution

This simulation is valid for the representation of the pneumatic structures or delicate materials like fabric or membrane, for example, design simulation of the ETFE facade can be done with this.

4. Optimization

Optimization through Evolutionary theory in Computation

There is a great article of evolutionary generic algorithms by David Rutten at the AAG10 conference in Vienna on September 21st 2010. It demonstrates Evolutionary Theory using Rhinoceros, Grasshopper, and Galapagos for the topic.

"There is nothing particularly new about Evolutionary Solvers or Genetic Algorithms. The first references to this field of computation stem from the early 60's when Lawrence J. Fogel published the landmark paper "On the Organization of Intellect" which sparked the first endeavours into evolutionary computing. The early 70's witnessed further forays with seminal work produced by -among others- Ingo Rechenberg and John Henry Holland. Evolutionary Computation didn't gain popularity beyond the programmer world until Richard Dawkins' book "The Blind Watchmaker" in 1986, which came with a small program that generated a seemingly endless stream of body-plans called "Bio-morphs" based on human selection. Since the 80's the advent of the personal computer has made it possible for individuals without government funding to apply evolutionary principles to personal projects and they have since made it into the common parlance.

The term "Evolutionary Computing" may very well be widely known at this point in time, but they are still very much a programmers tool. 'By programmers for programmers' if you will. The applications out there that apply evolutionary logic are either aimed at solving specific problems, or they are generic libraries that allow other programmers to piggyback along. It is my hope that Galapagos will provide a generic platform for the application of Evolutionary Algorithms to be used on a wide variety of problems by non-programmers."

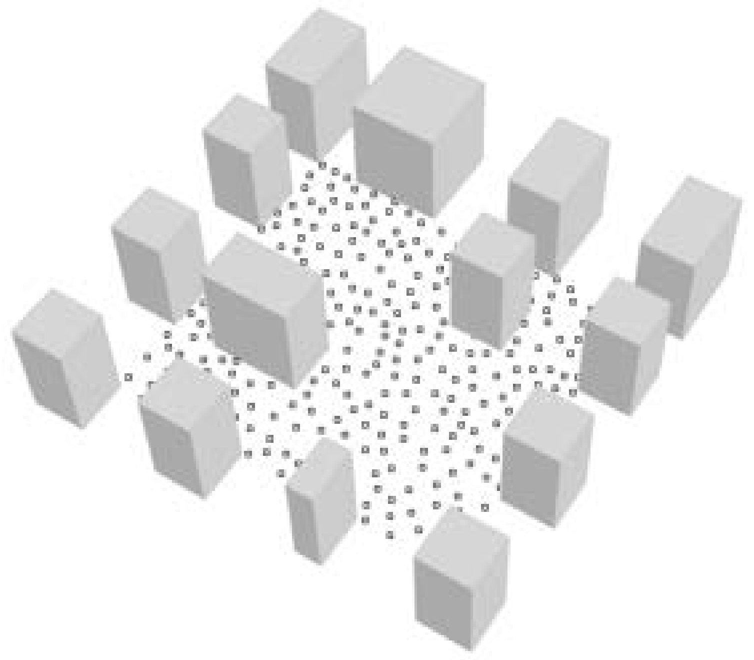
Even though Evolutionary Algorithms are slow and do not guarantee a solution (predefined as "good enough"), It is more flexible to carry the problem and easy to deal with. The high degree of interaction of the tool "Galapagos" is quite a unique feature of the application.

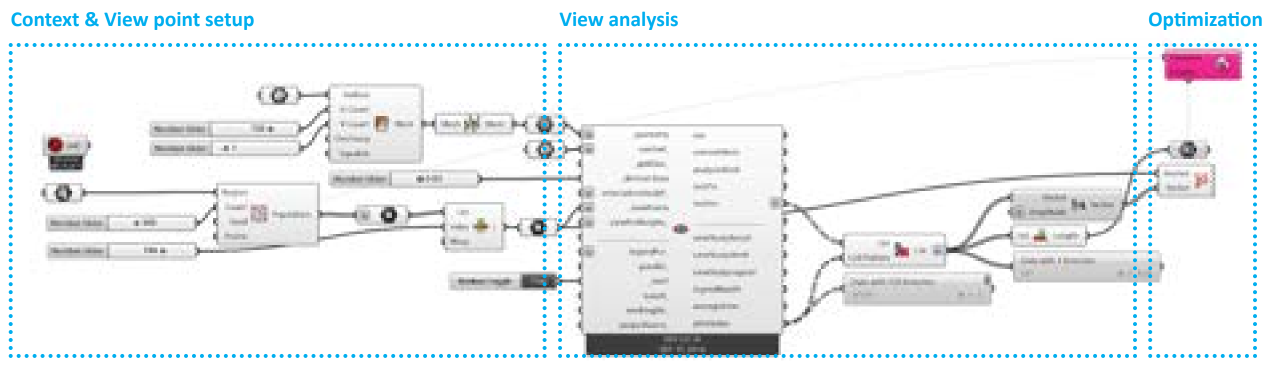
Galapagos-Evolutionary Theory

Data driven design is on the rise and with the advent of sophisticated and powerful software this process becomes easier and more manageable over time. Knowing how to use these tools effectively will allow you to make smarter and more informed design decisions early on in the design process.

Galapagos uses an evolutionary-based algorithmic solver. In evolutionary models, a population of candidate solutions is maintained, and new candidate solutions are generated randomly by mutating or recombining variants in the existing population. Periodically, the population is pruned by applying a selection criterion (a fitness function) that allows only the better candidates to survive onto the next generation. Iterated over many generations, the average quality of the solutions in the candidate pool gradually increases.

This section introduces how to create your own optimization tools inside Grasshopper using Galapagos.

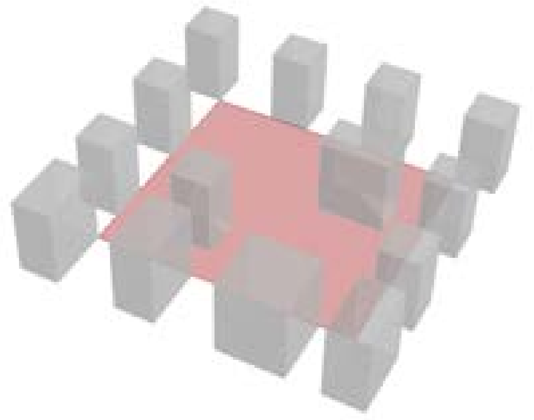


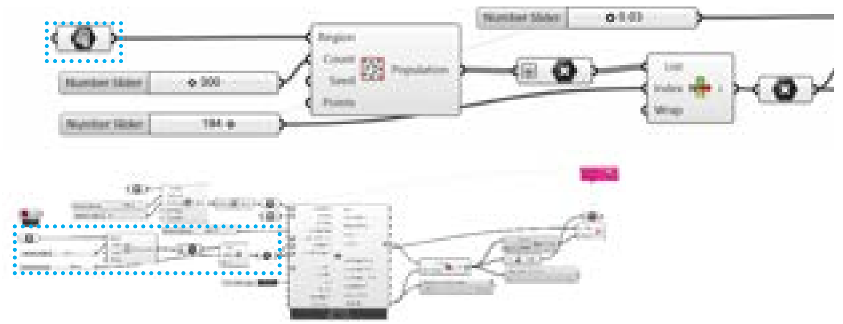


AP-14

View Optimization with Galapagos

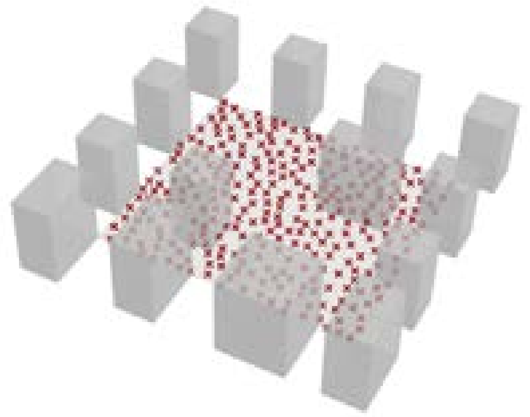
The View optimization process is an exercise that shows how to set the parameter, goal object by using Galapagos in Grasshopper. The view analysis component under Ladybug will compute the number of vectors at every identified random point to discrete points, and Galapagos's goal is to define a result that has the largest number of vectors at each point.

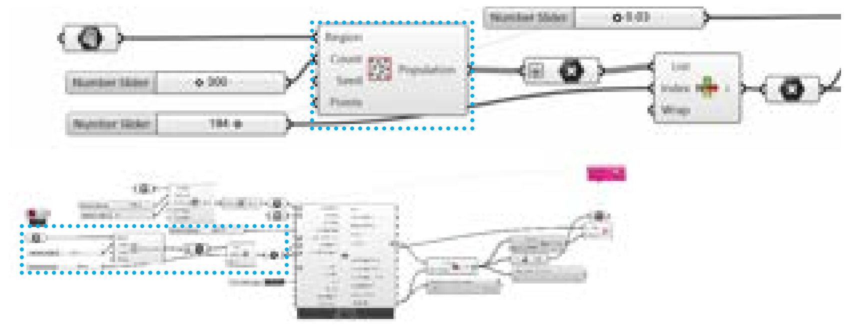




Target Area Import

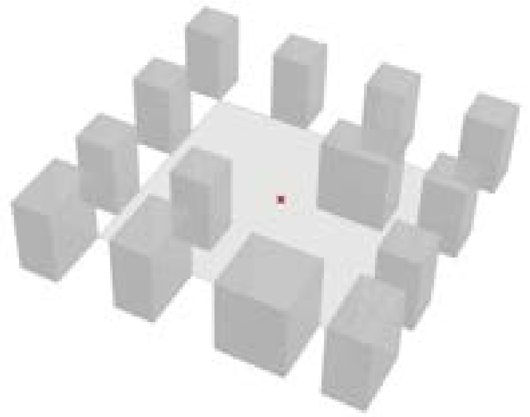
1. Set up a boundary surface for the region.

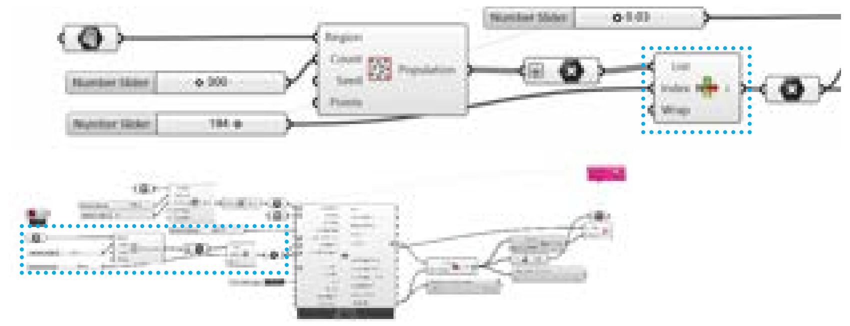




Origin of Object Point Set-up

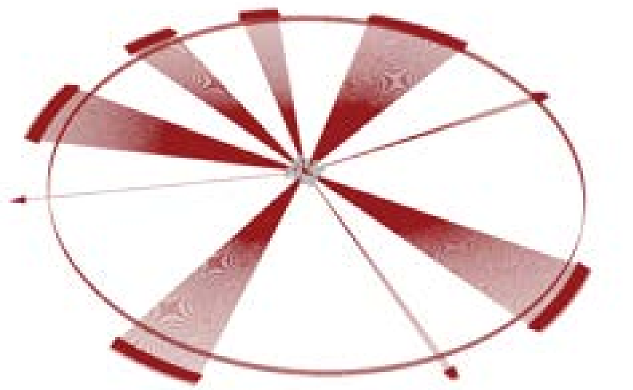
1. Connect Number Slider to the Count input of Populate2D.
2. Connect Mesh to the Region input of Populate 2D.
3. Move Number Slider to populate the number population of random points.

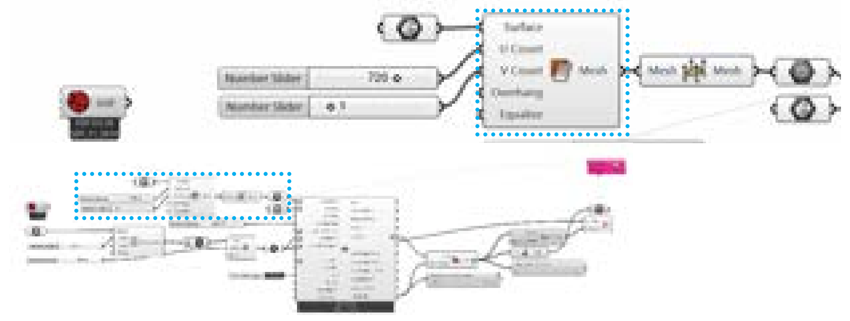




Simulation Parameter Set-up

1. Connect Point to the List input of the List item.
2. Connect Number Slider to the Index input of the List item.
3. Move Number Slider to pick one location of the point.



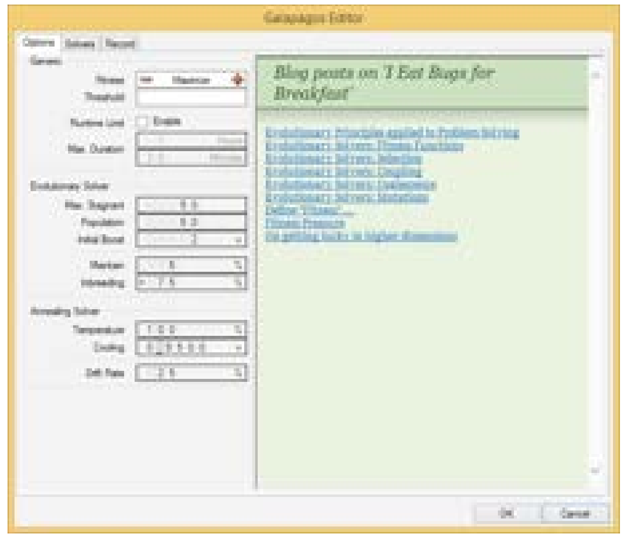


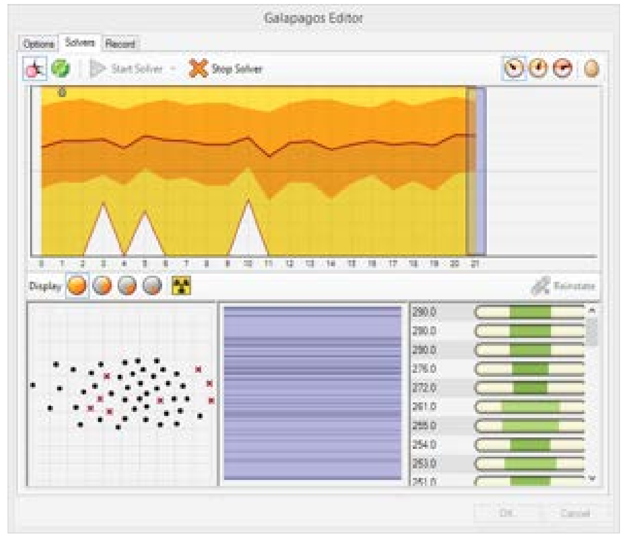
Target points Set-up

1. Create a Ring Mesh for the Surface input of the Mesh surface.

2. Divide Ring mesh by 720 (360*2 = 0.5 radian/meshface) in U direction.

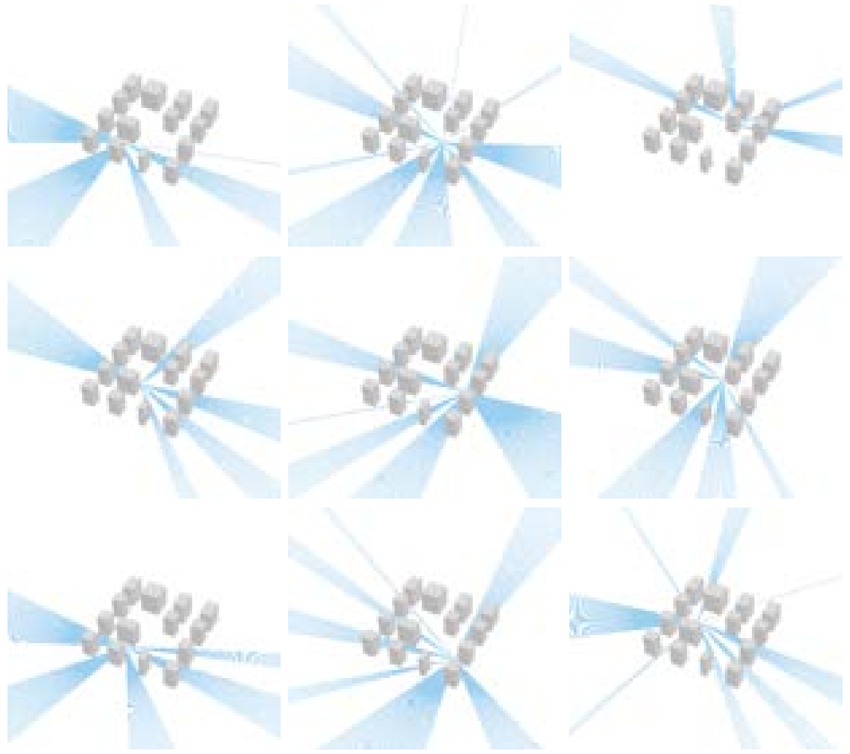
3. Divide Ring mesh by 1 for height 1.

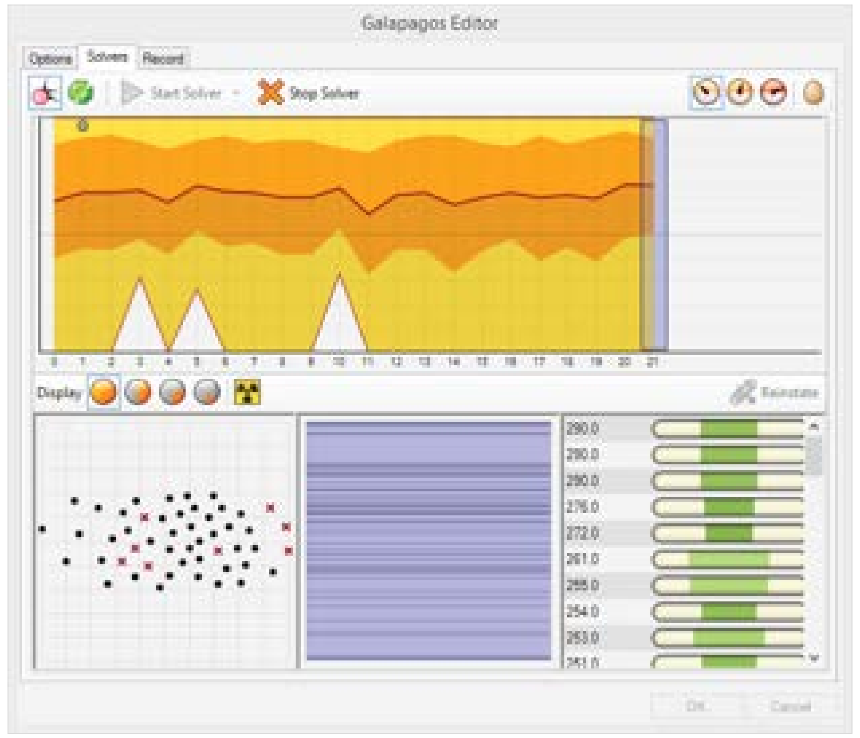




Fitness & Solver Set-up

1. Double click the Galapagos Evolutionary solver icon and access the Galapagos editor interface.





Simulation

1. Set Fitness to "minimize"
2. Click to display all genomes in the rhino viewport.
3. Click Start Solver to run.

Galapagos is still a very young product and hasn't really had time to position itself firmly in any work-flow. It seems to be capable of solving relatively small problems quickly, but it certainly needs a lot of work to make it more robust and usable. It is likely that the most effective applications for a solver of this type and capacity are small or partial problems. To try and evolve anything complicated will almost certainly result in frustration.

Advanced

Advanced

Energy & Form

Energy can be transformed into a specific form and the reverse process (that energy flow is computed from specific form) is also a common process in the architectureal practice. Data from environmental energy and physical forces have become invaluable design resources.

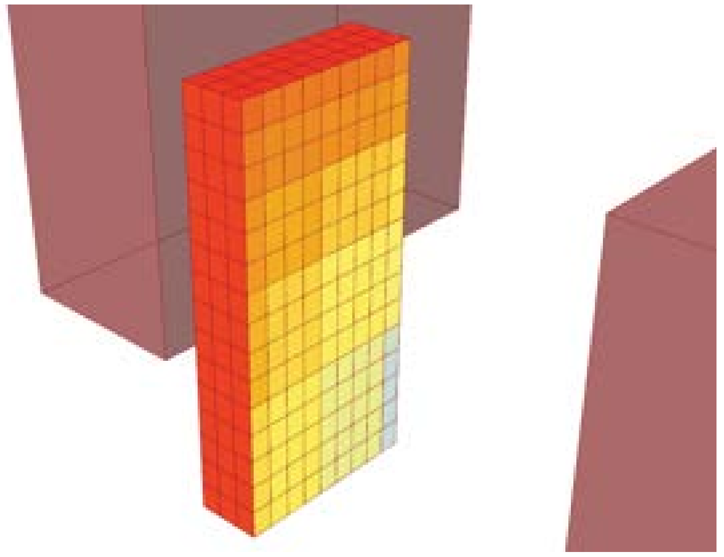
This session is a series of advanced applications of previous sessions.
The first section explains how post-management of thermal data allows us to achieve various types of facade developments. The remapped thermal data from the result of the original solar radiation can define various types of facade design opportunities, such as the range of an area, width, and execution of a window.

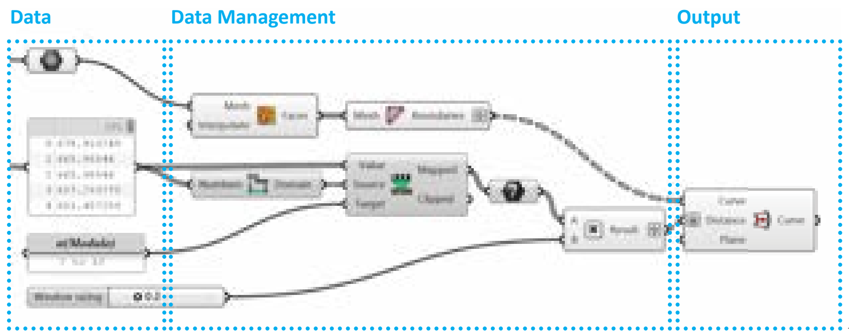
The application of multi-forces on a single object definition is the second exercise of innovative use of physic data in form generation. The uniqueness of morphological exploration can be the foundation for a design challenge.

The later part of this section will introduce a combinatory application of environmental and physic data. The exercise incorporates the physic and thermal data as interdependent design processes.

A series of examples in many different type of energy data usage will be the foundation for more practical uses of data in the field of architecture.

1. Solar Radiation & Form

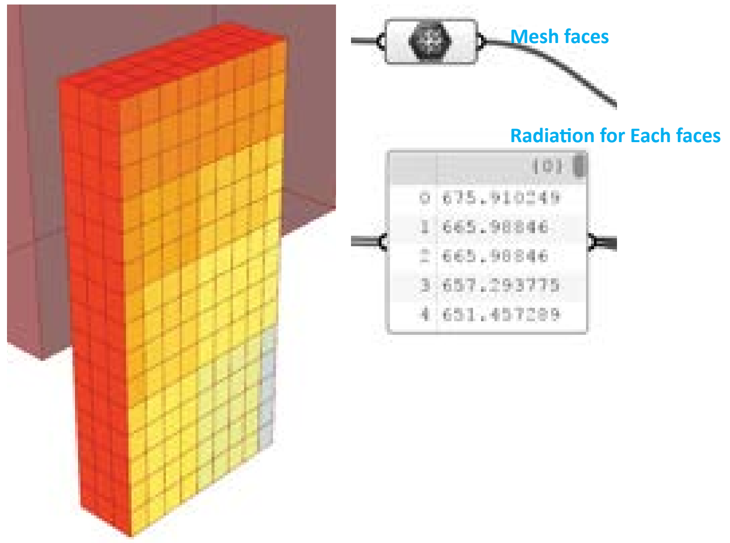




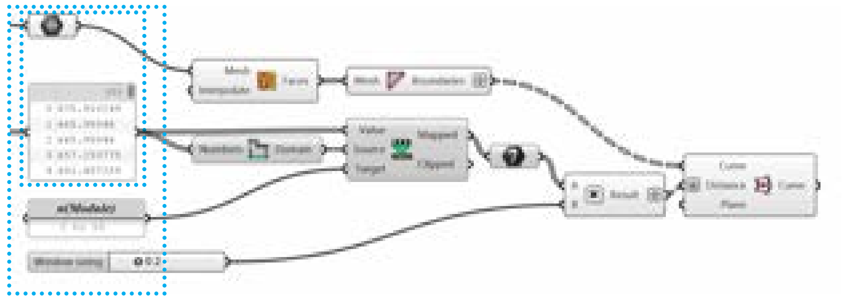
AP-15

Ladybug Environmental Simulation

This tutorial starts from the end of previous session about ladybug environmental simulation. The base geometry is from the 'radiationMesh' output and the numbers are from the 'radiationResult' output.

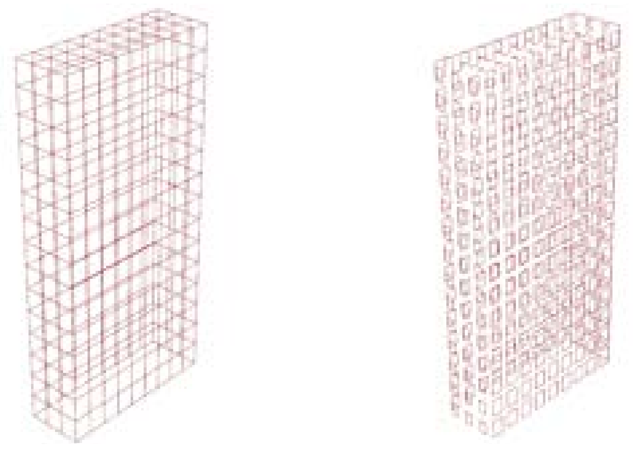


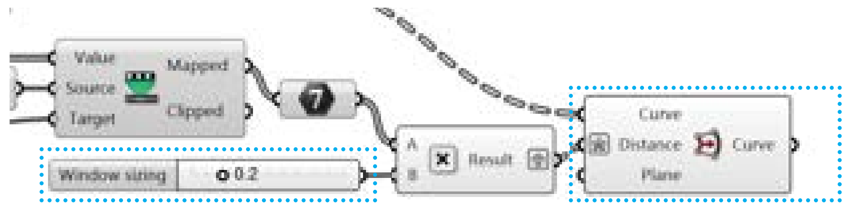
```
{0}
0 675.910249
1 665.98846
2 665.98846
3 657.293775
4 651.457289
```

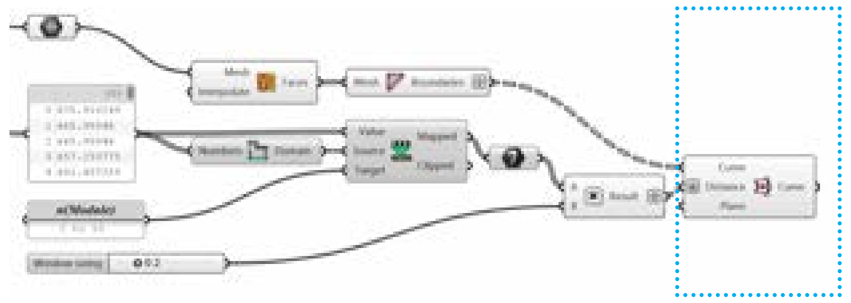


Data Matching & Treatment

Mesh can be exploded into individual faces with 'explode mesh' component and the result of radiation color can be simplified to be optimized for a Modular system.

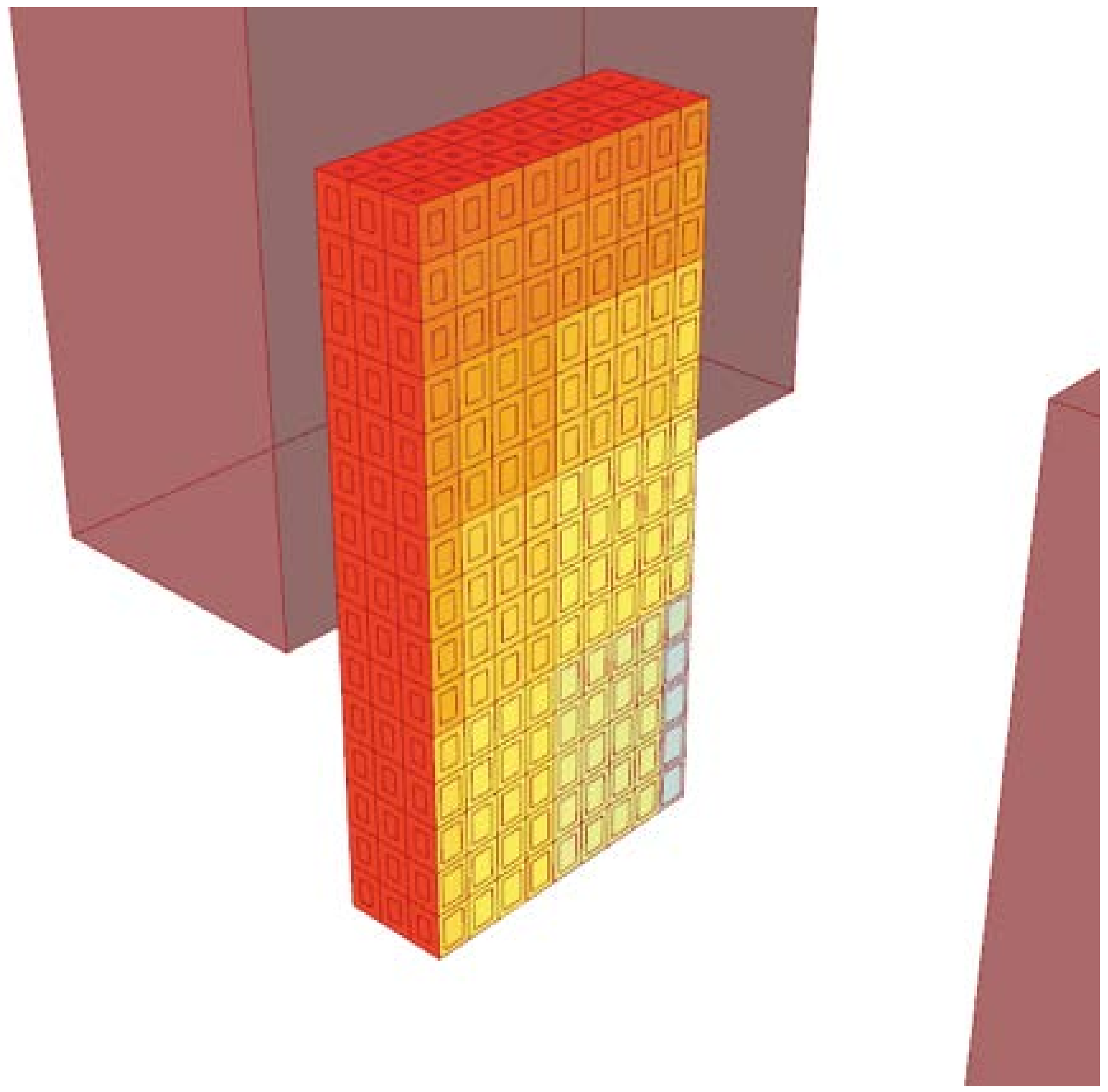






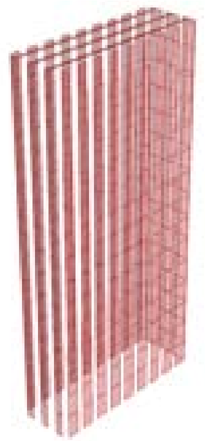
Option 1 - Punchings

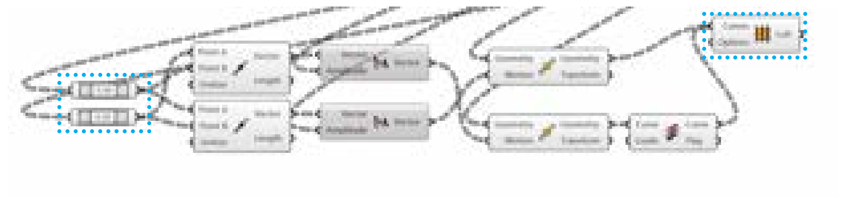
By using the 'Face boundaries' component, extracting Mesh edges and offset the boundaries inward according to the offset degree which is originated from the remapped radiation number. the remapped number should be scaled with the 'multiplication' component to identify the optimal size of an opening.



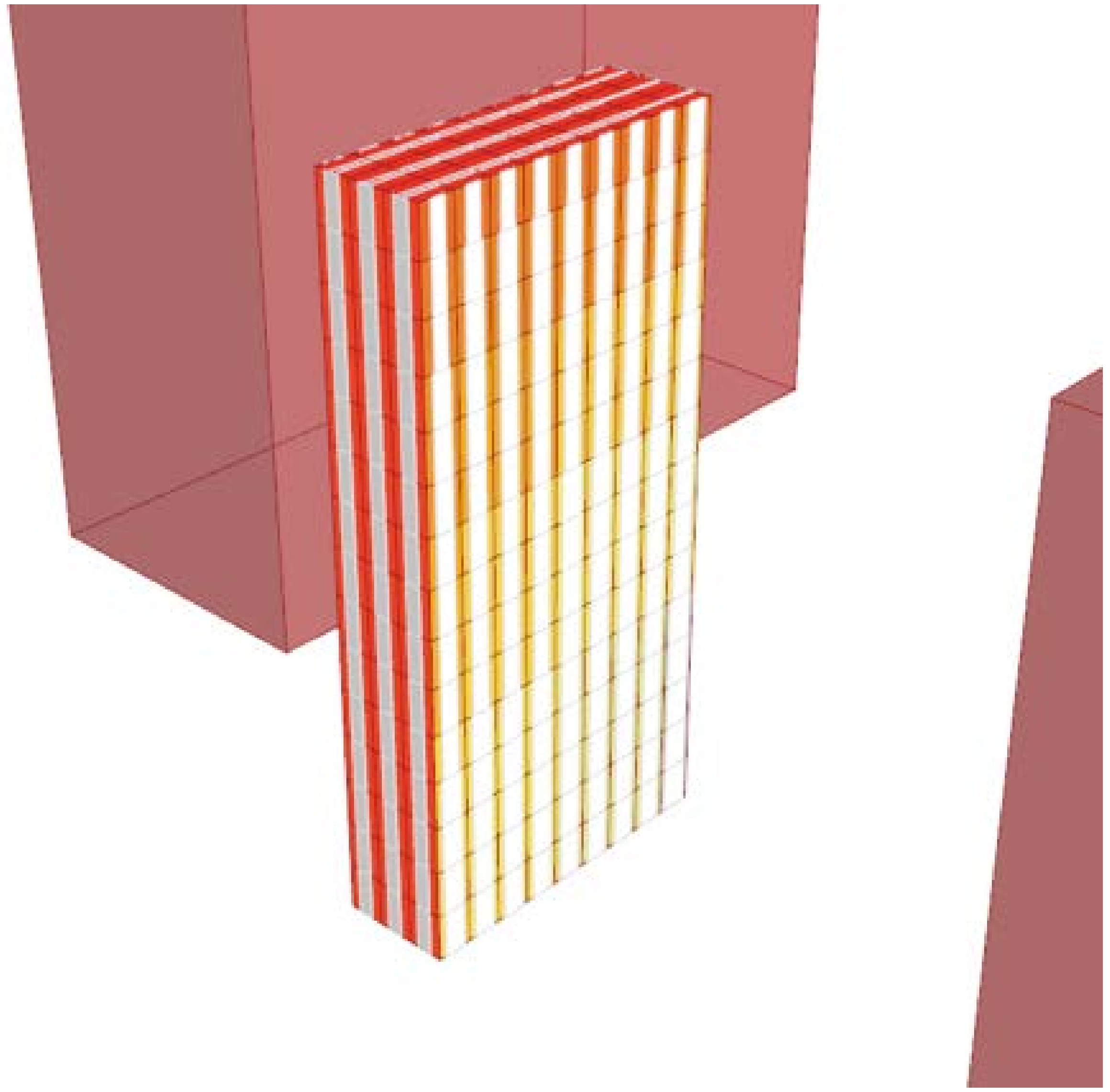
Result & Contribution

This is the directive method of using environmental data from ladybug's environmental simulation. This method of design is great for the decision making process in both concept and design development phase.



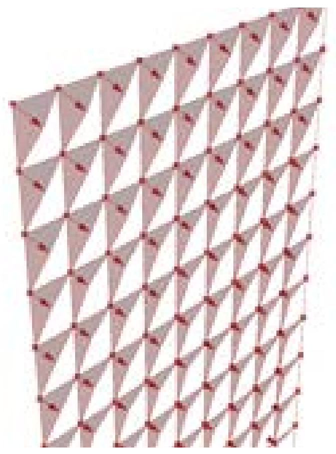
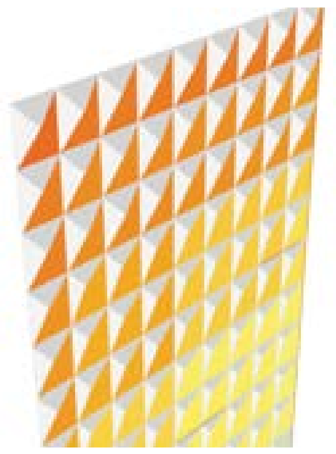
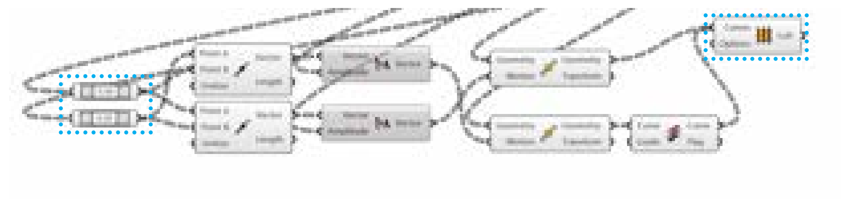


Loft 2 curves and create windows that reach to the top & bottom of the floors. Adjust the distance by regulating the scale parameter which is attached to the 'multiplication' component.

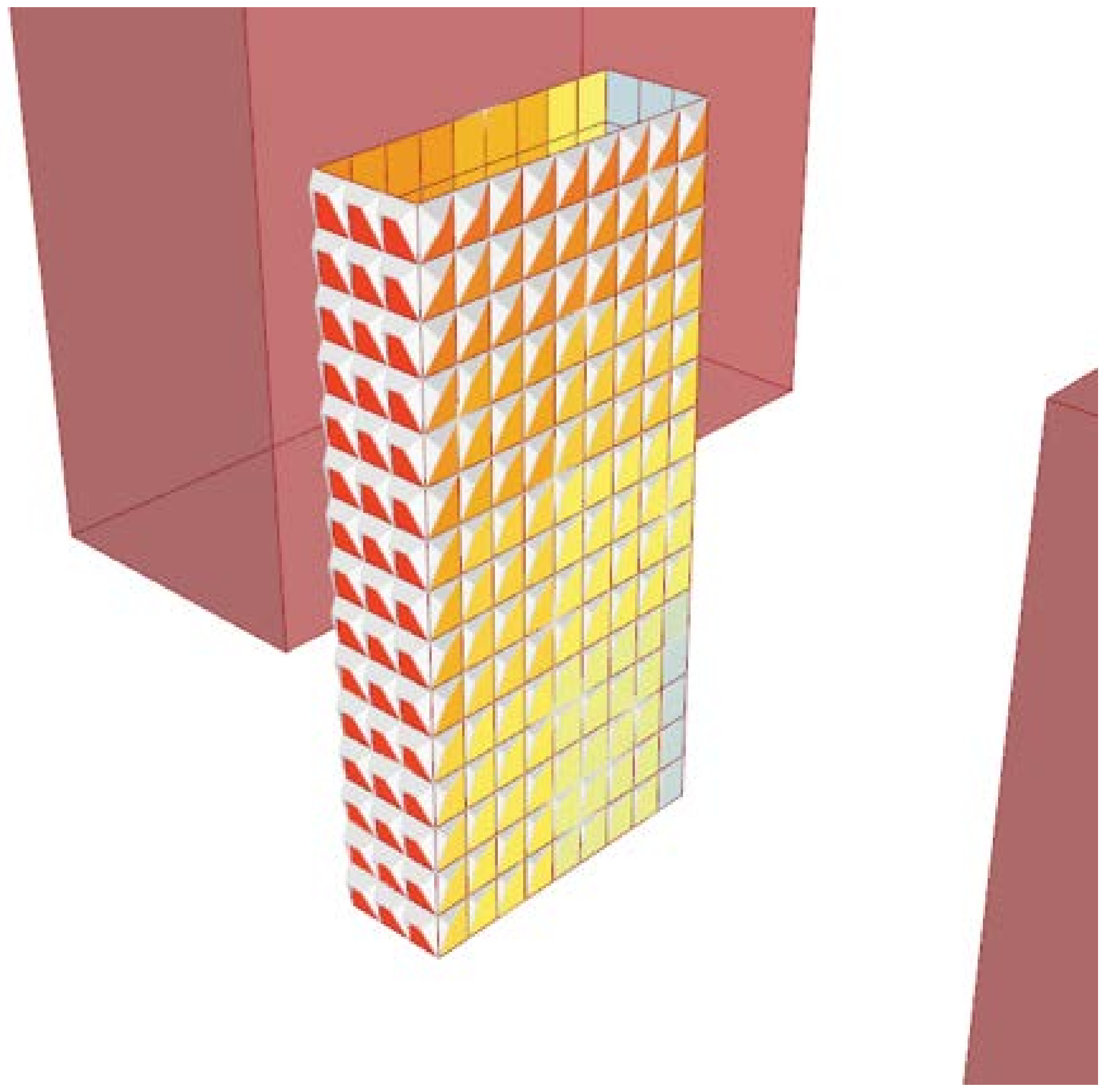


Result & Contribution

This is a directive method of using environmental data from ladybug's environmental simulation. This method of design is great for concept design phase.

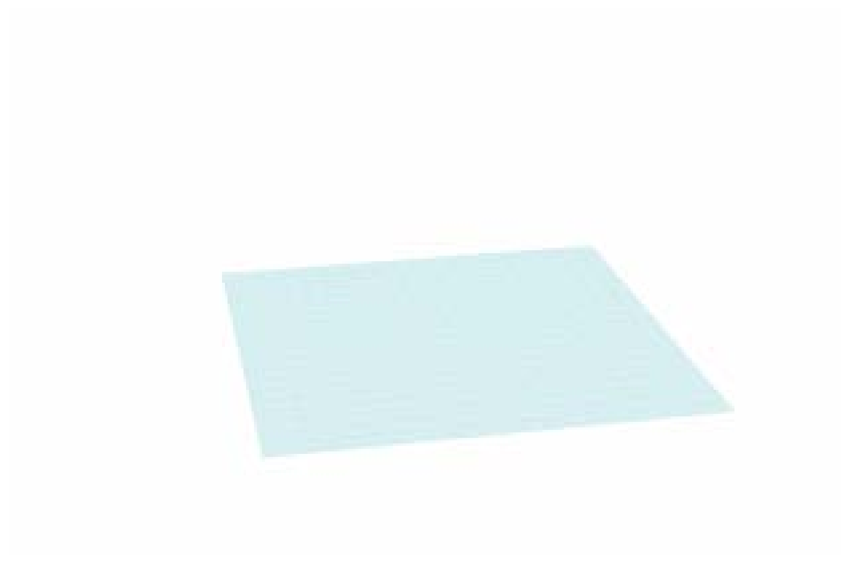
By using the 'Addition' component, merge the two vectors and apply it to the point picked from 'list item' component. Create 2 polylines. One polyline from the three vertices of each mesh faces and another polyline from the two vertices of each mesh face and one vertice moved by merged vectors as a middle point of the polyline. By lofting them, the basic shape of the louver can be created.

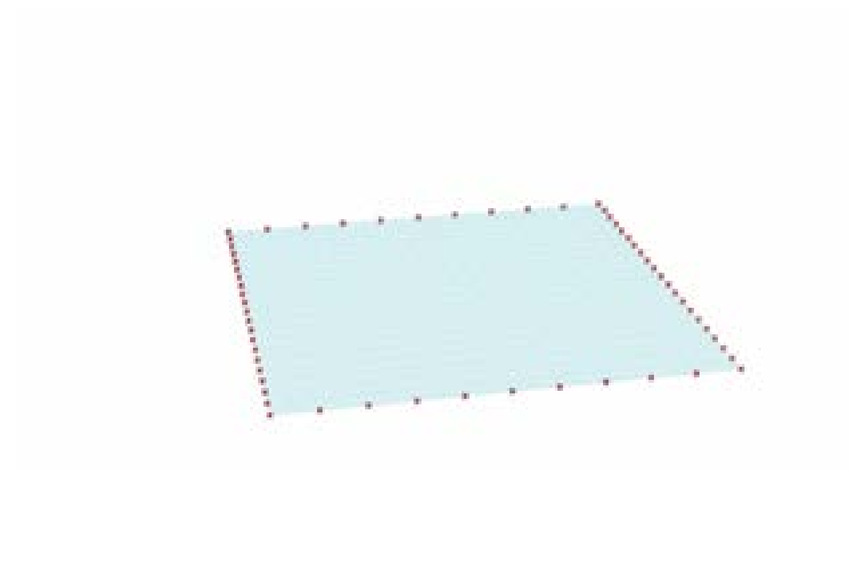


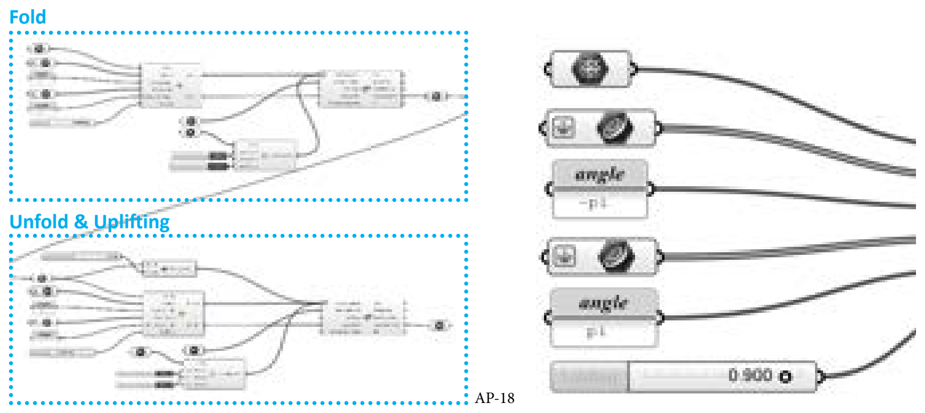
Result & Contribution

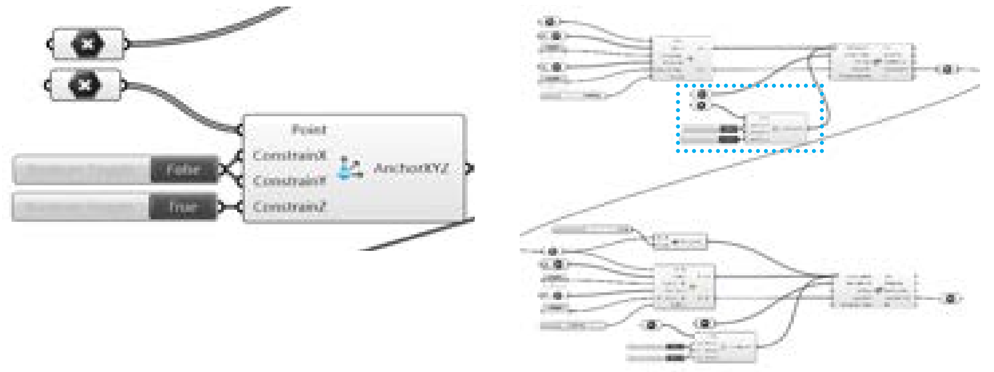
The system can be applied with the same logic on the other surfaces. Since every face has different normal vectors, the main vertice that drives the shape of the louver should be carefully and manually chosen face by face in the code.

2. Force & Form







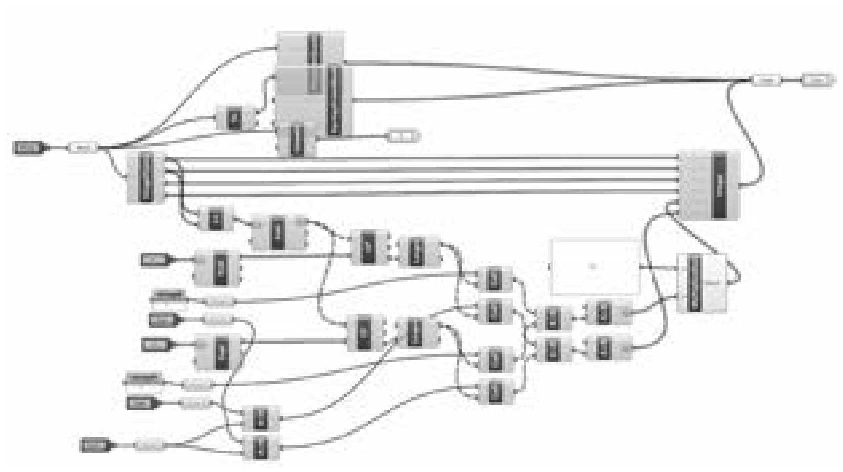


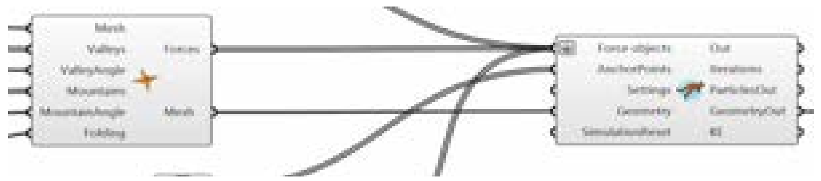
Double time Fold 1

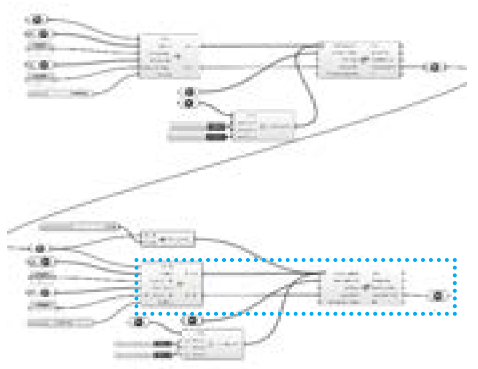
This session is about how to create a complex shape with a complex or repetitive simulation technique. Shapes which requires the specific form cannot be created easily with imagination. By applying the physic simulation, designer's can accelerate their experiment on form. This session will focus on how the folding technique of one surface can be used to create a covered area.

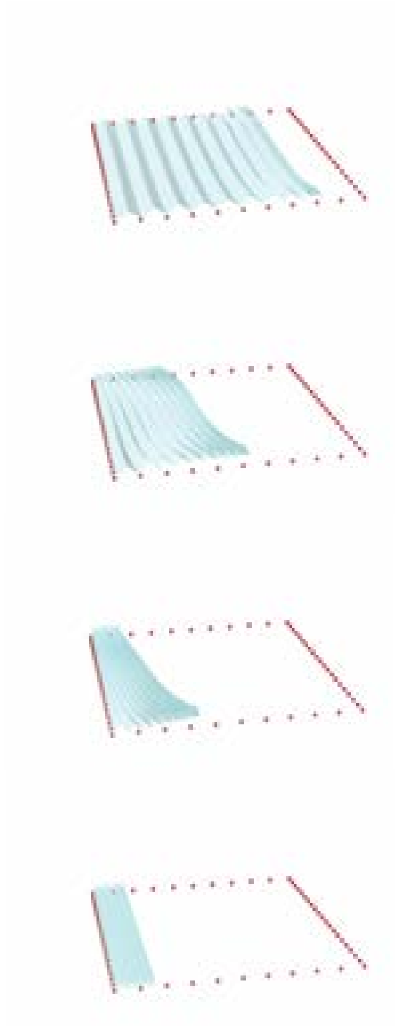
Anchor Points

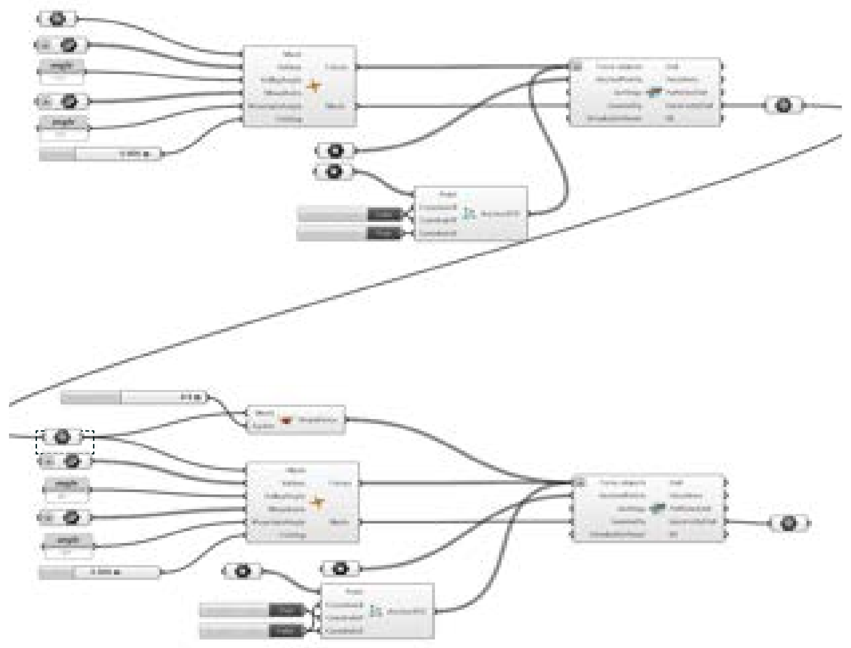
1. Set Points from one edge for the Anchor points input of the Kangaroo Physics.
2. Set the Rest of the point from 3 side for the point input of the Anchor XYZ.
3. Set Boolean toggle False on Constrain X and Constrain Y input of the Anchor XYZ.
4. Set Boolean toggle True on Contain Z input of the Anchor XYZ.











Folding Line & Angle Set-up

1. Construct mesh surface.

2. Set Curves for valleys input of the origami

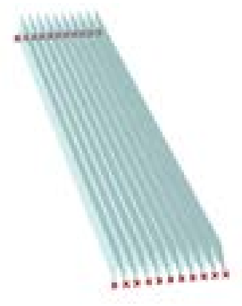
3. Set Curves for mountain Angle input of the origami.

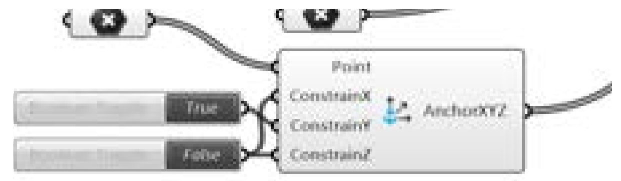
4. Set an angle of "pi" and "-pi" and to the dedicated concave angle.

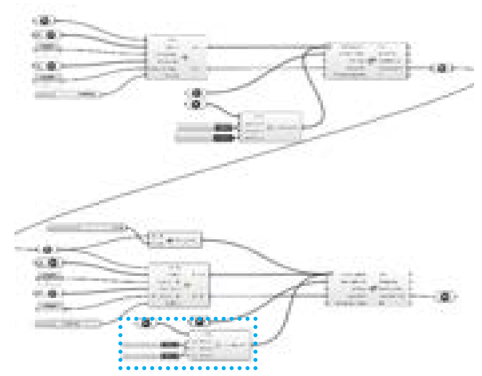
5. Set a Number Slider to the folding input of the origami. (0-1 Domain)

Mesh Data Transfer

1. Connect the Folded mesh to the Mesh Input of the origami.

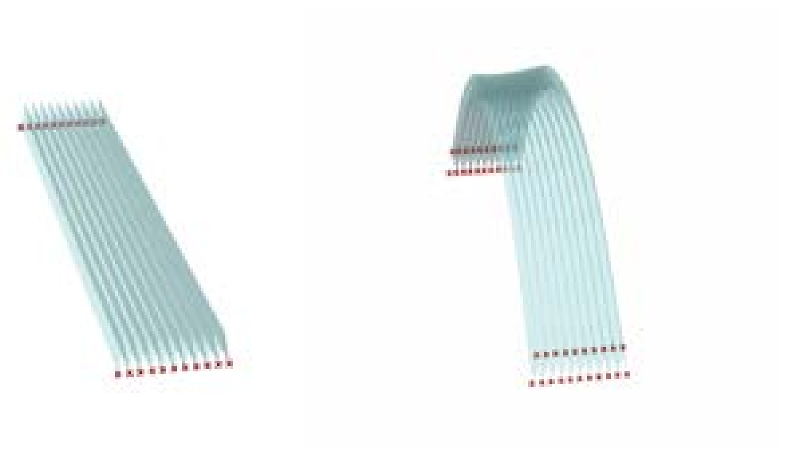


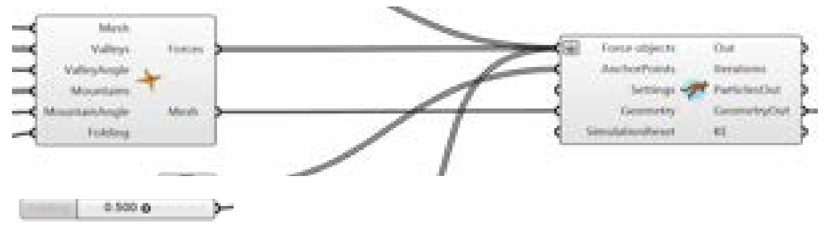


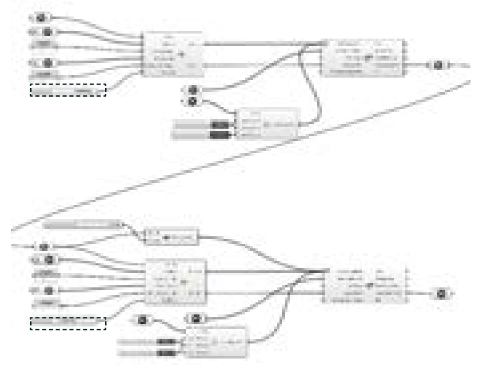


New Anchor Points Set-up

1. Set Curves for valleys input of the origami
2. Set Curves for mountain Angle input of the origami.
3. Set an angle of "pi" and "-pi" to the dedicated concave angle.
4. Set a Number Slider to the folding input of the origami. (0-1 Domain)

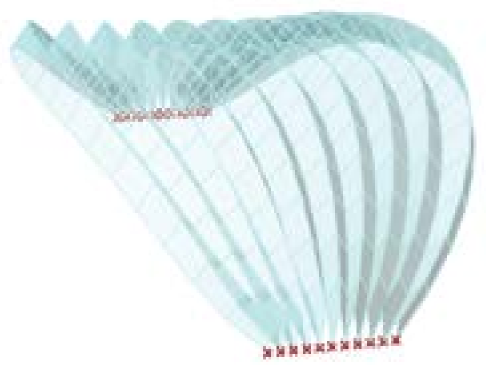
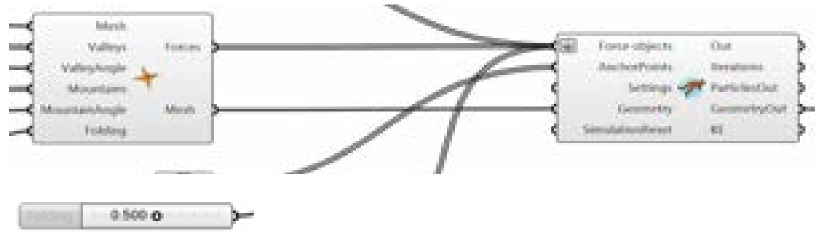


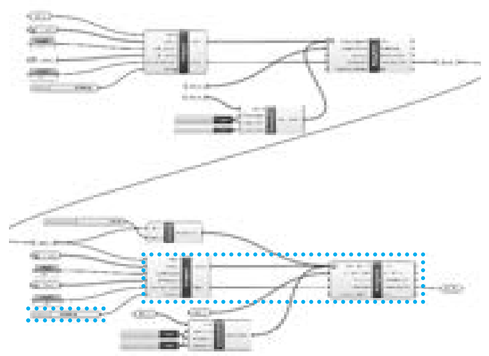
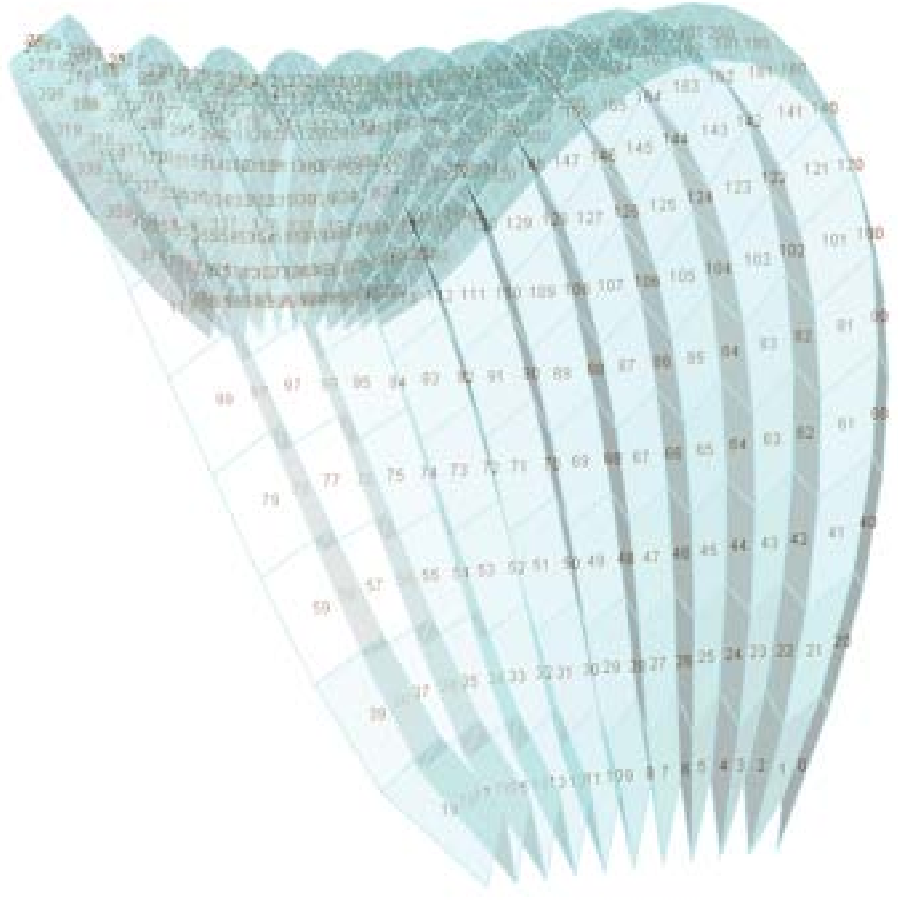




Force Iteration (Gravity)

1. Connect the folded Mesh to the Mesh input both origami and gravity component

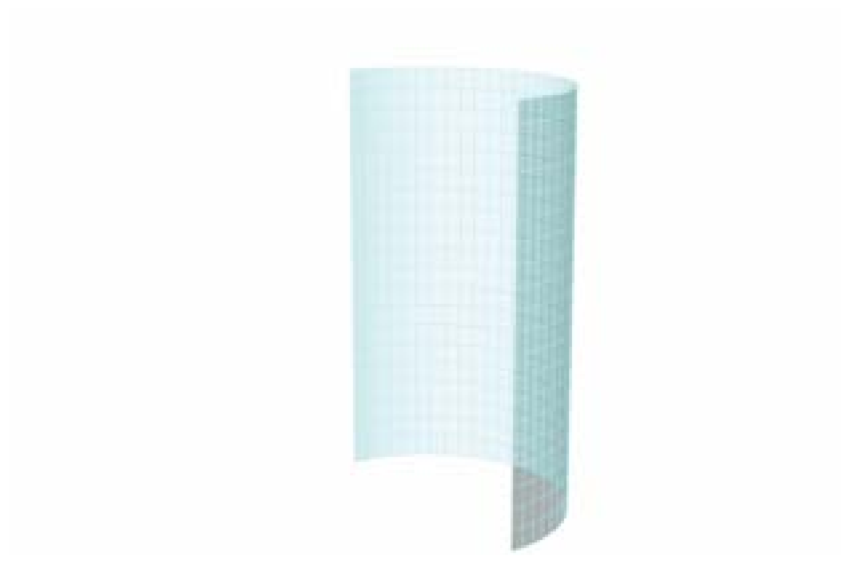



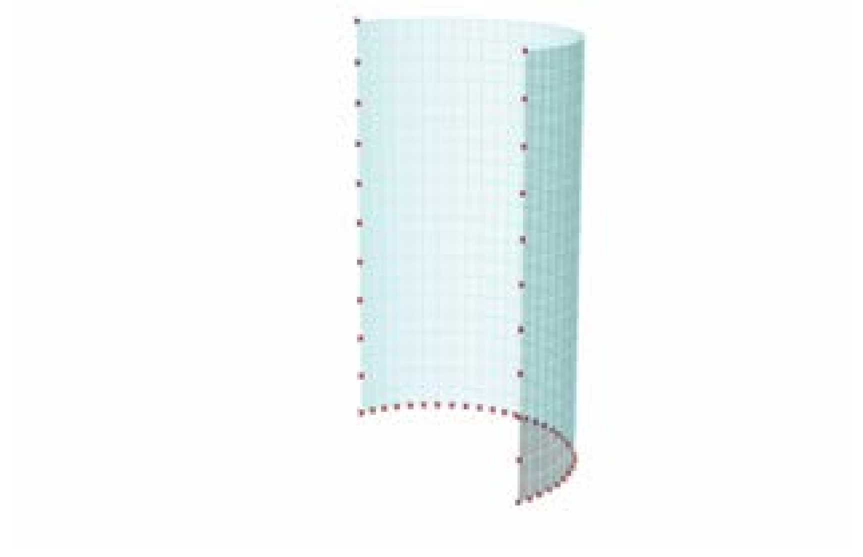
Secondary Iteration (Unfold)

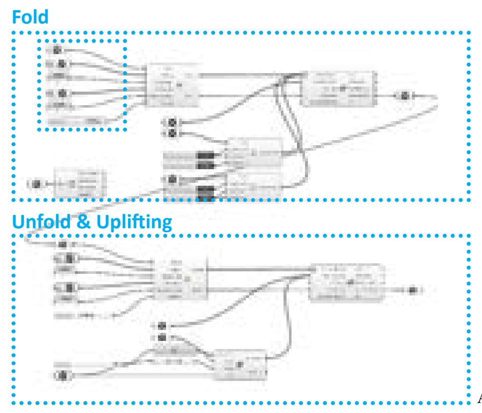
1. Slide mountain and valley angle value to identify geometry.

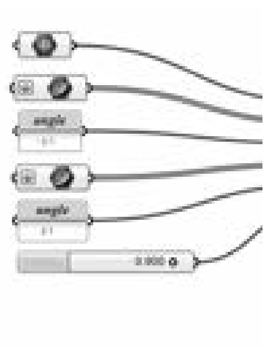
Result & Contribution

The character of the folding and usage of the anchor point with gravity created this form. Similarly to the previous basic Kangaroo sessions, the forces used in this code is Gravity, unfolding force and Anchor points. However, by regulating the order and setting, the results can vary.

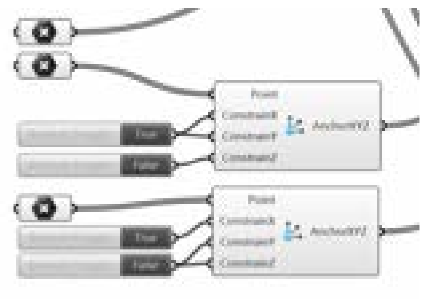


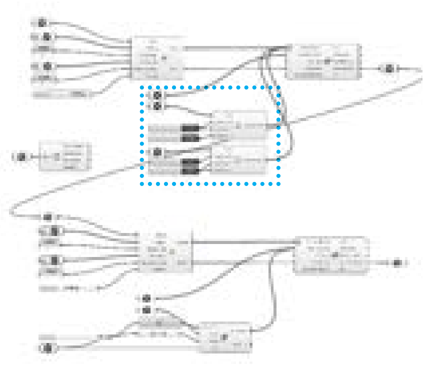






AP-19





Double time Fold 2

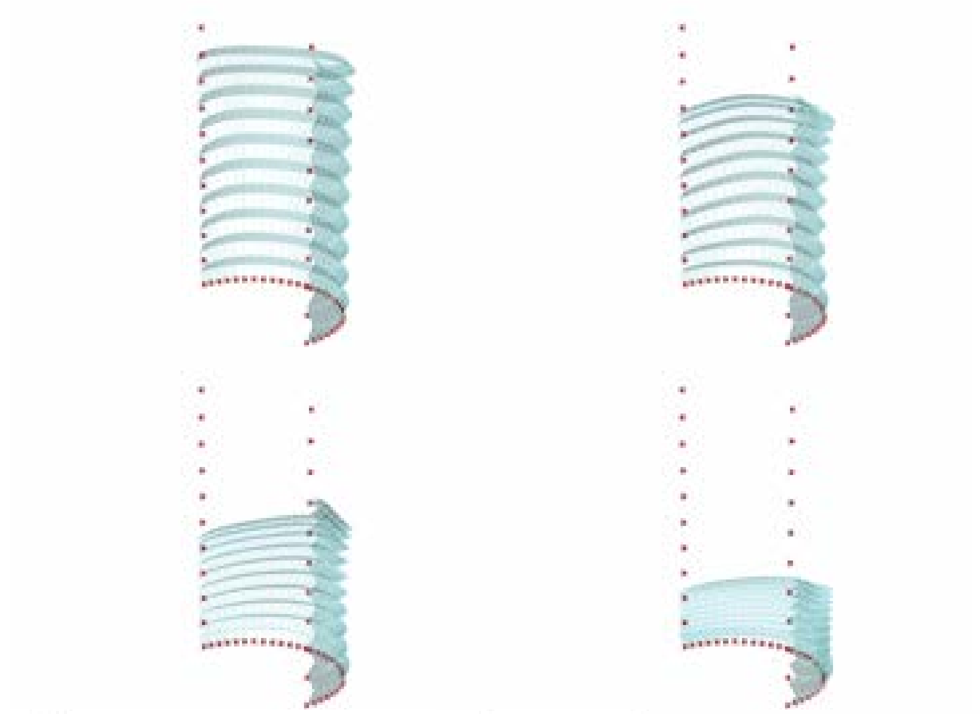
Folding the geometry and unfolding it while partially holding the geometry can create unexpected results. These forms are not easy to imagine without simulation and the designer can find the meaning of the Physics simulation on this part of complex form making process.

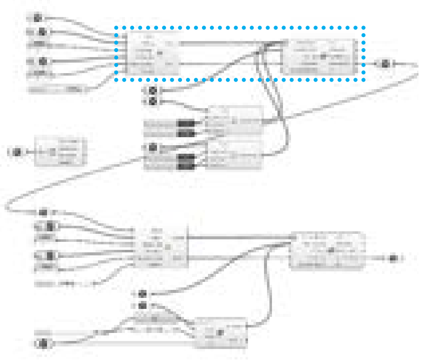
Anchor Points Set-up

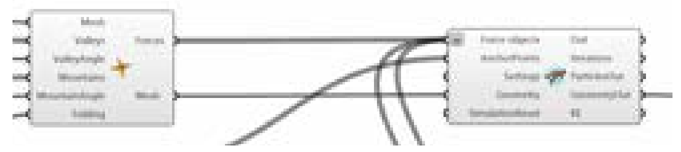
1. Set Points from the geometry.

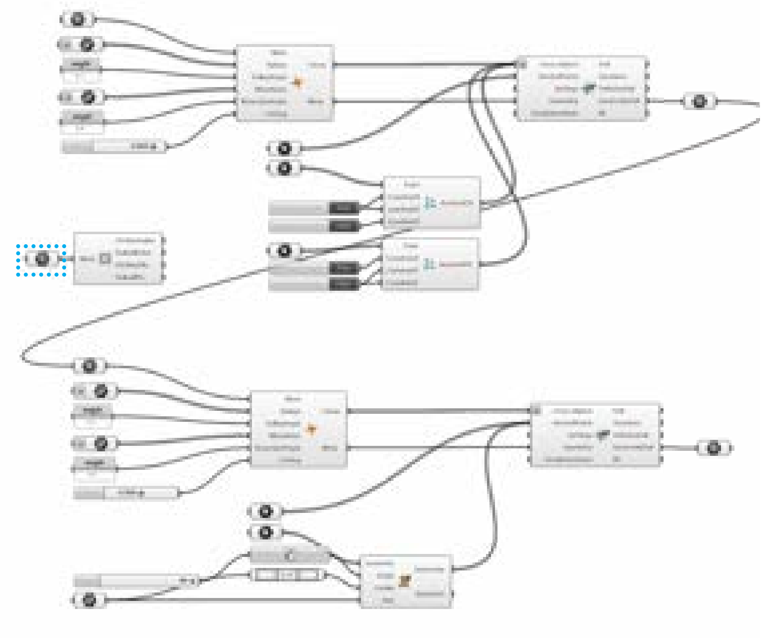
2. End Points from valley curves are the point input of Anchor XYZ true on X and Y, false on Z.

3. End Points from mountain curves are the point input of Anchor XYZ true on X and false on Y and Z.







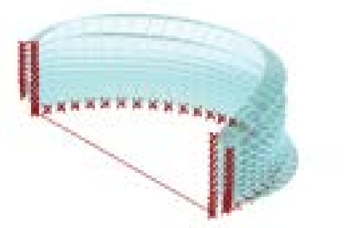


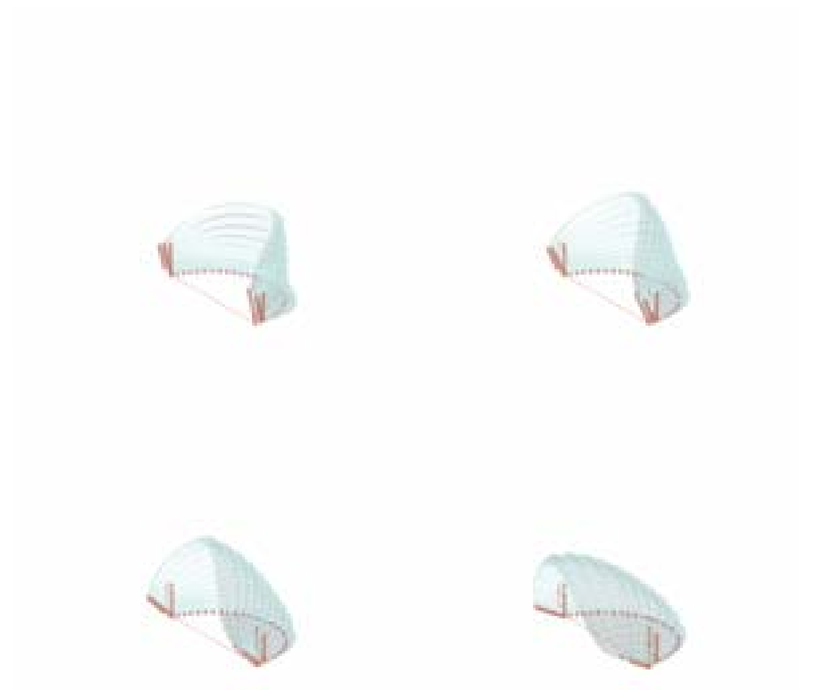
Force Implementation (Folding)

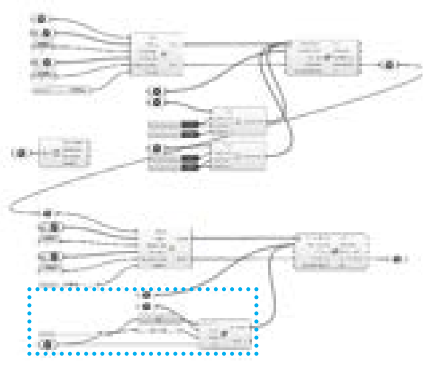
1. Run Kangaroo simulation.
2. Connect Points from bottom points of geometry to the Anchor points of kangaroo.
3. Connect 1 of 2 vertical naked vertices of the mesh to anchor XYZ.
4. Set ‘True’ for the XY constraint and ‘False’ for Z constraint.

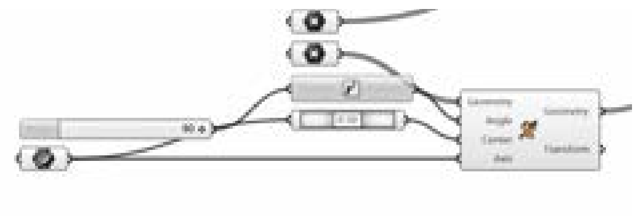
Mesh Data Transfer

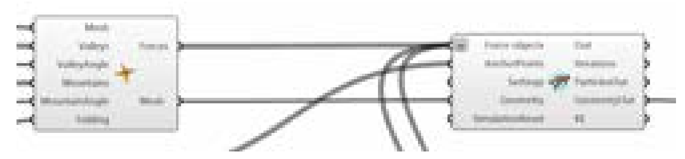
1. Take the folded Mesh and connect to Mesh input of origami.
2. Set curves for valleys and mountains from the geometry.

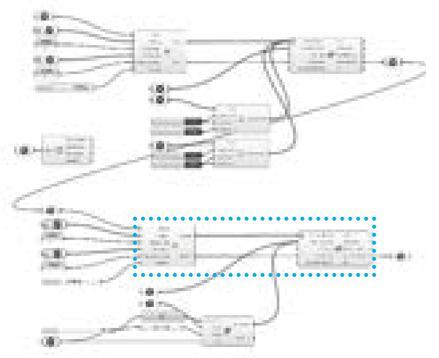












Anchor Point Set-up

1. Set a mid point for axis vector from the curves.

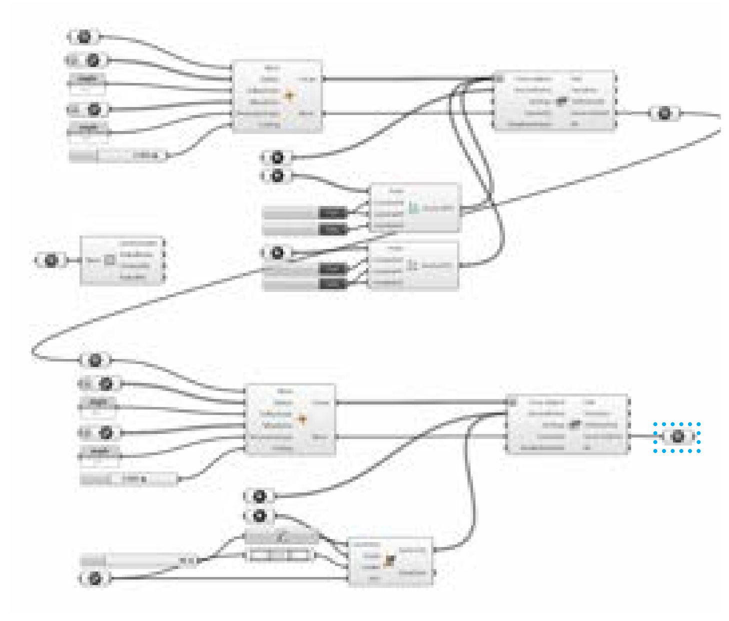
2. Set a Number Slider to rotate the geometry.

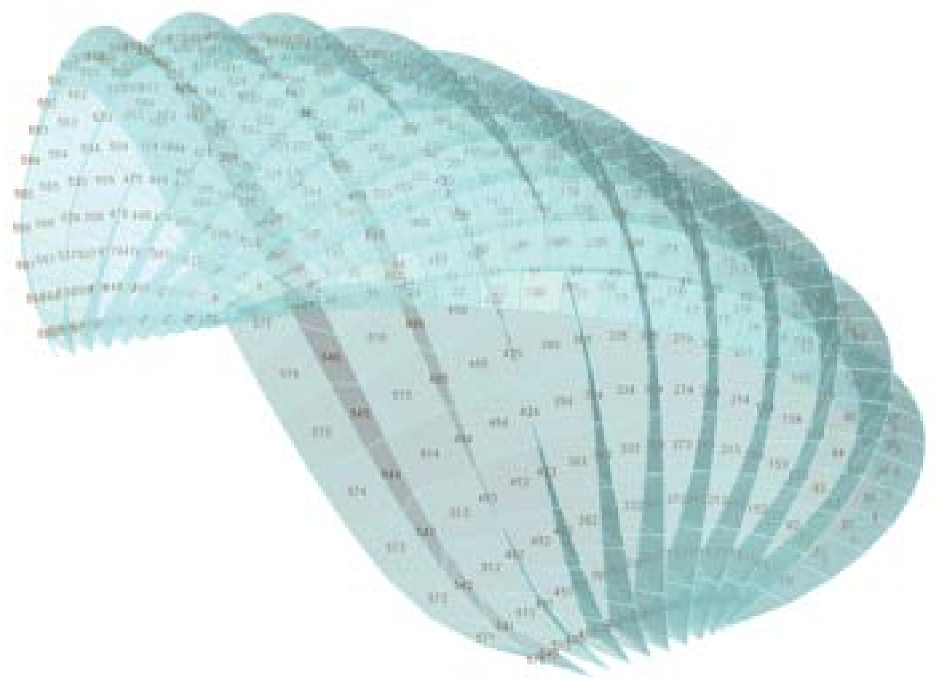
3. Set points for anchor point input of Kangaroo Physics.

Anchor Point Adjustment

1. Simulate Kangaroo.

2. Change the Number Slider of degree of rotation while Kangaroo simulates to adjust the degree of unfolding.



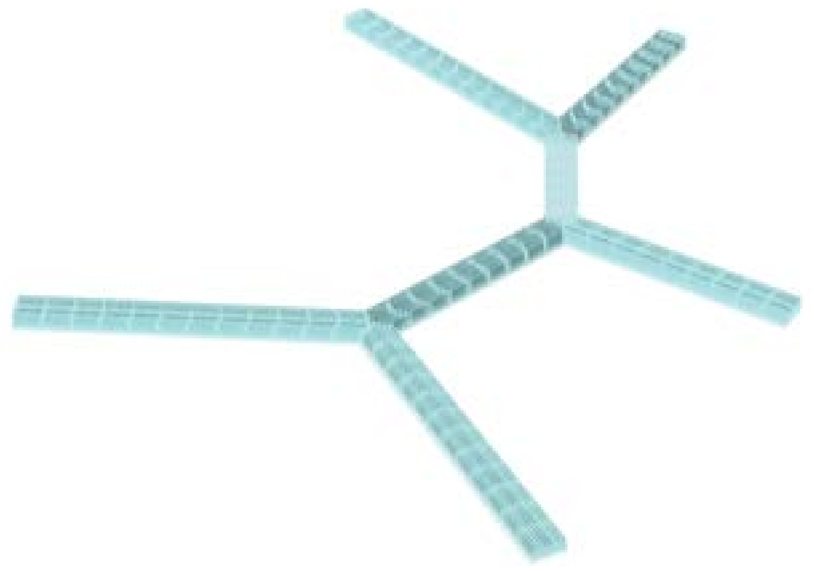


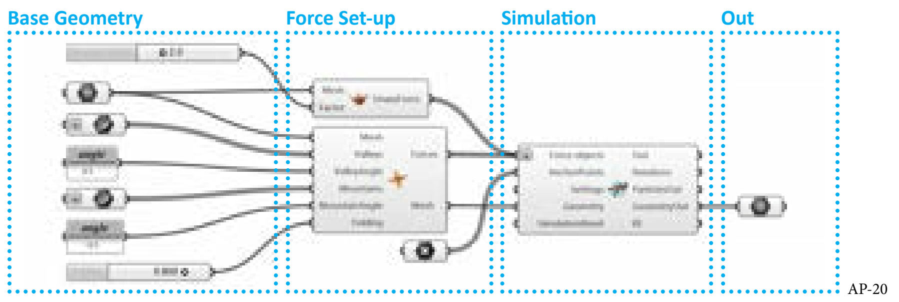
Force Implementation (Unfolding)

1. Set a Mid point for axis vector from the curves.
2. Set a Number Slider to rotate the geometry.
3. Set points for anchor point input of Kangaroo physics.

Result & Contribution

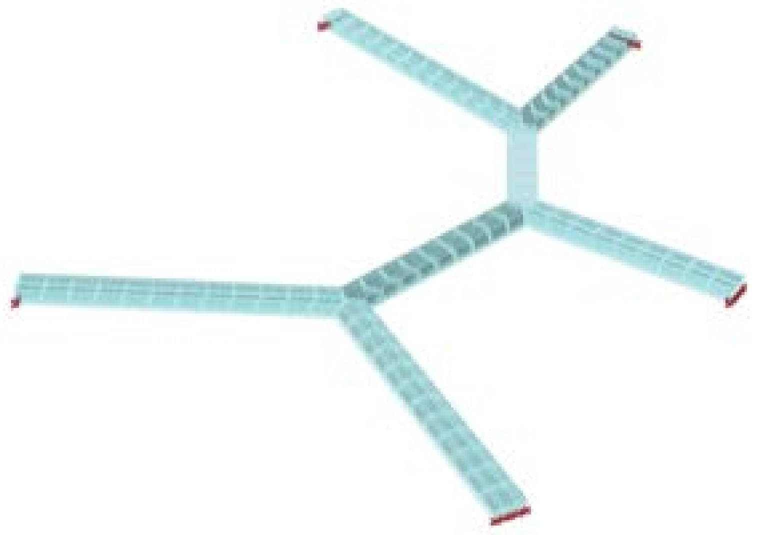
The character of the folding and usage of anchor point with gravity created this form. Similarly to the previous basic Kangaroo sessions, the forces used on this code is gravity, unfolding forces and anchor points. However, by regulating the order and settings, the results can be varied.

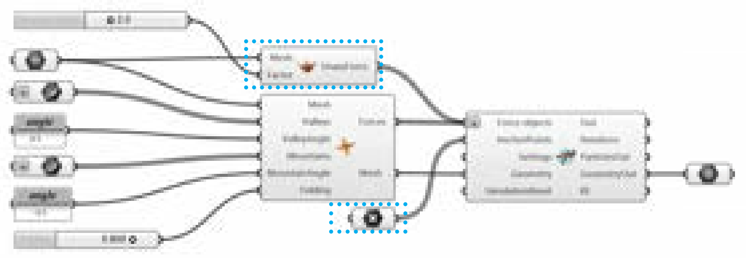




Folded Vault

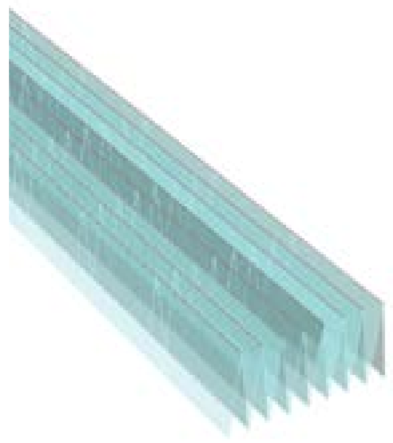
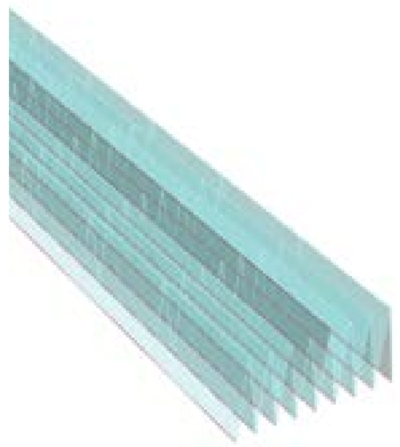
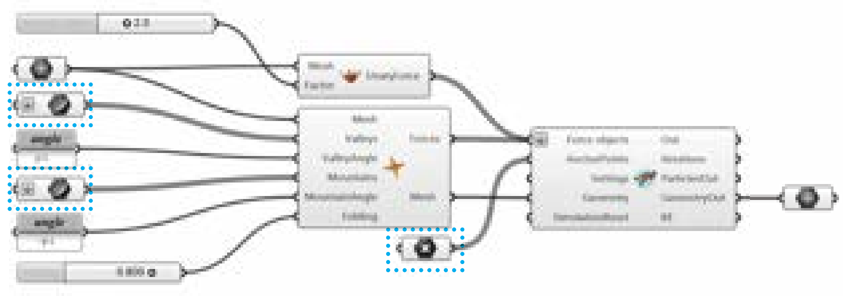
This session is designed to create a folded vault form which typically appears in Gothic architecture with a vault and ribs. This is not the exact same form but shares a similar logic of form making in terms of force driven design. Base geometry can be made by joining several folded meshes which is explained in the previous sessions.





Anchor Points Set-up

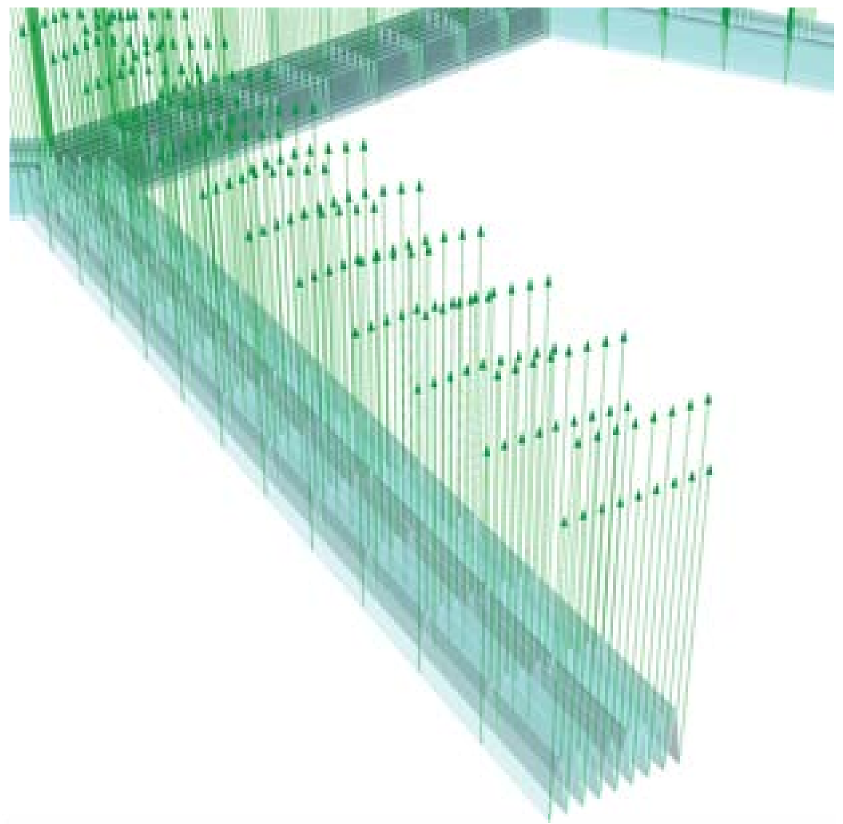
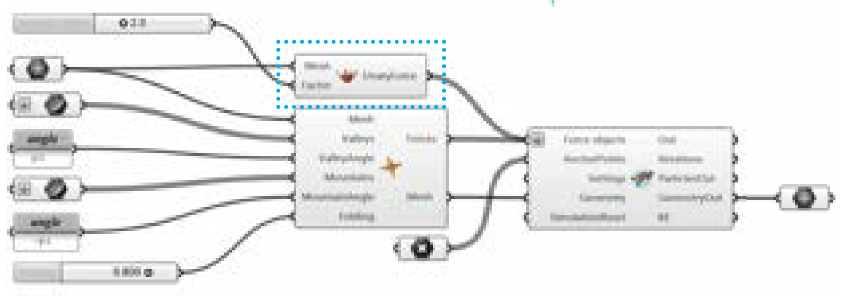
1. Connect point to the anchor point input of Kangaroo.

Folding Line Set-up

1. Valleys curves for in, and mountains curve for out.

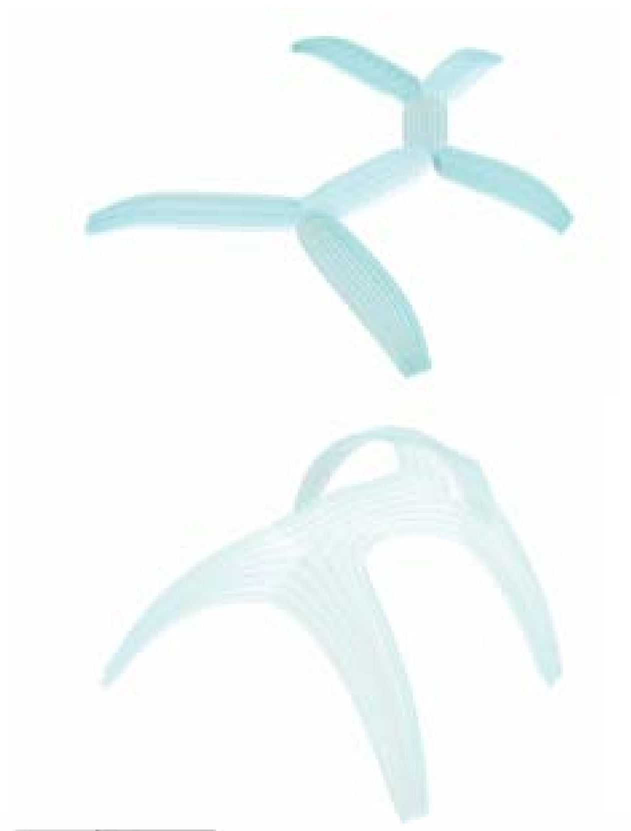
2. Extract curve from geometry by valleys and mountains.

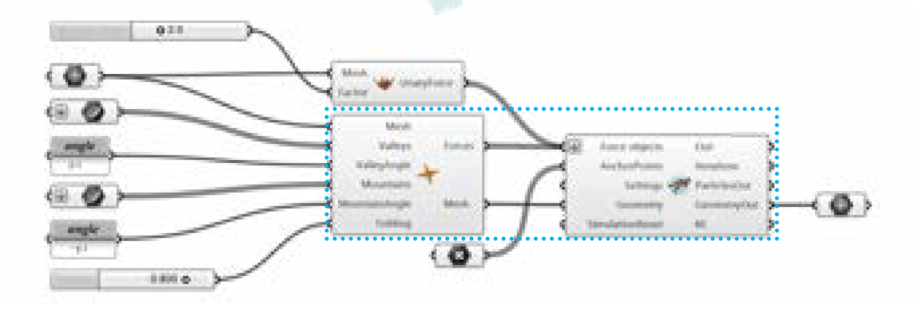



Force Implementation (Gravity)

1. Connect the Mesh to the mesh input of gravity.

2. Connect the Number Slider to the factor input of gravity.





Force Implementation

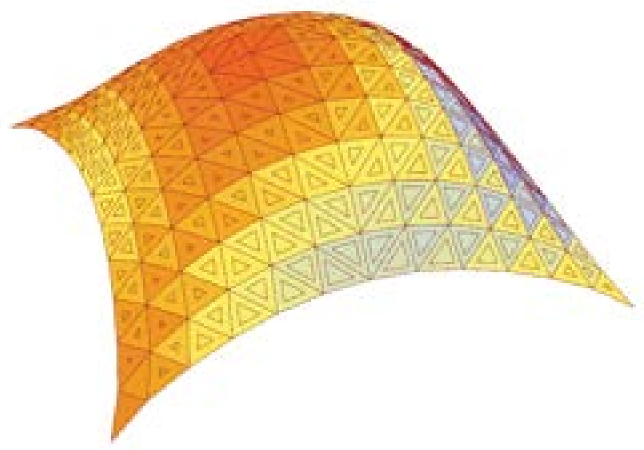
1. Connect the Origami force to Force object of the Kangaroo physics.
2. Run the simulation and adjust the folding degree.

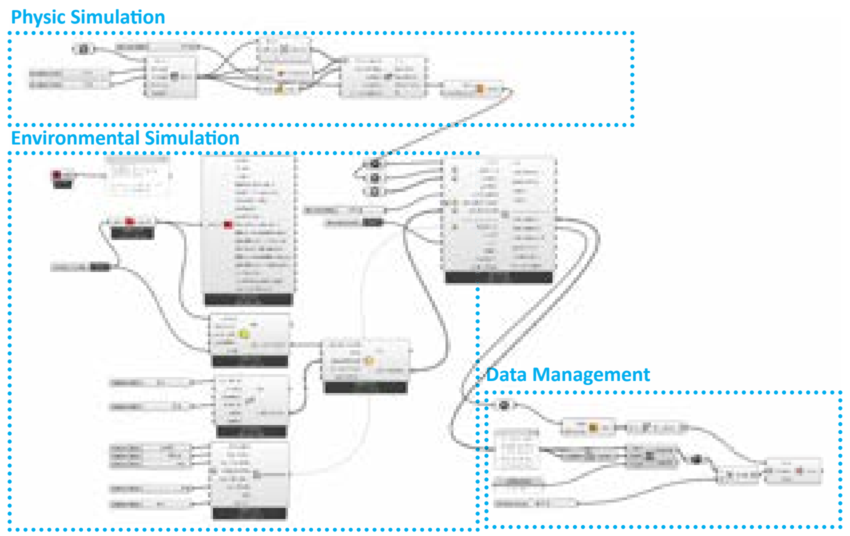
Result & Contribution

Catenary surface form is used with a folding force in this session. Even though this type of work could be done with an intuitive imagination, collaborative application of the forces cannot be accurate without physical simulation. Since the form is originated from gravity setup, this form can be more rigid than other forms without the consideration.

3. Combinatory Design

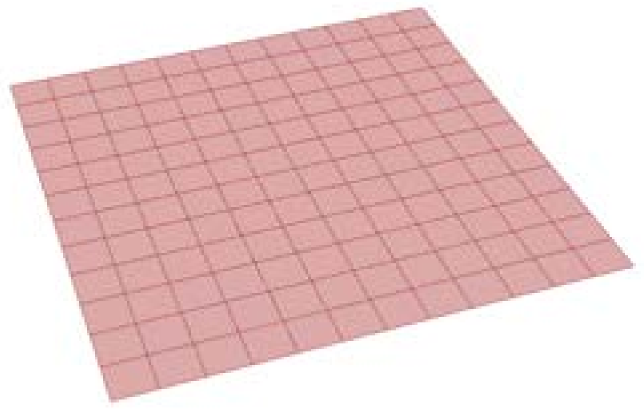
3.1 Energy + Force = Design

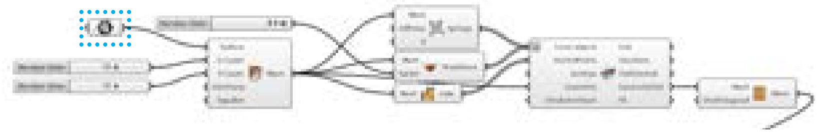


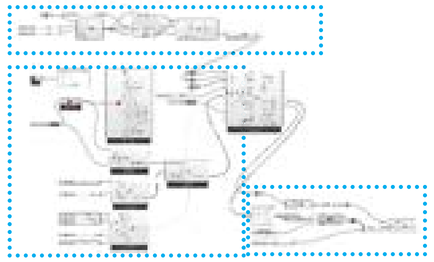


Combinatory Process

This exercise is to incorporate the two different type of data into one process. Physic Forces and Thermal energy informs the geometry and pattern. In this case, the catinary surface generates the overall geometry, environmental data is overlaid onto the surface to identify the size of inner pattern.

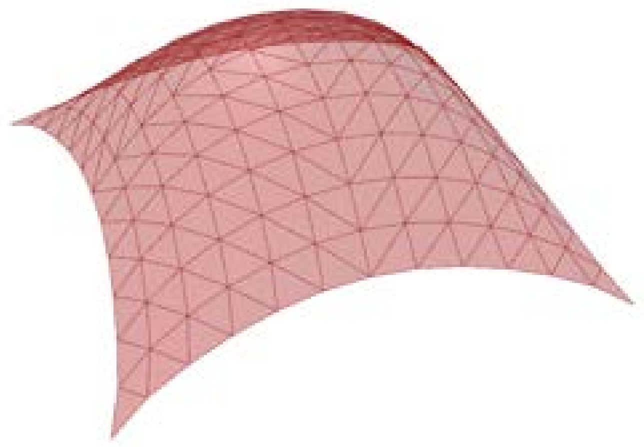


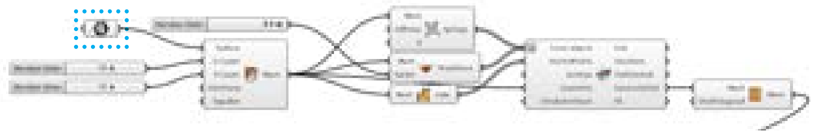


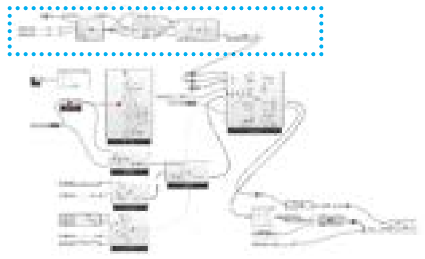


Mesh Set-up

1. Create mesh surface.
2. Connect the Mesh to the Mesh input of the Spring from mesh, Gravity, and Mesh Corner.





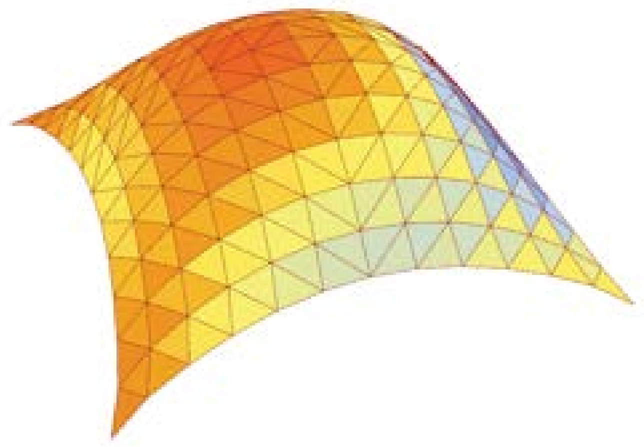


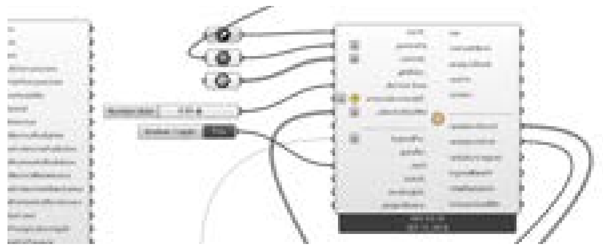
Force Implementation (Gravity + Spring Force)

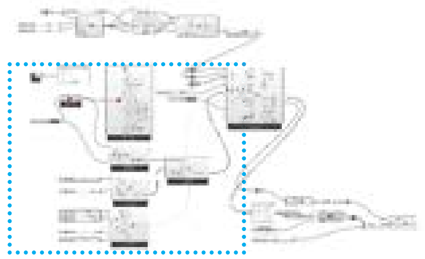
1. Make Gravity negative to invert the force direction.

2. Run the simulation.

3. Convert quads to triangles.

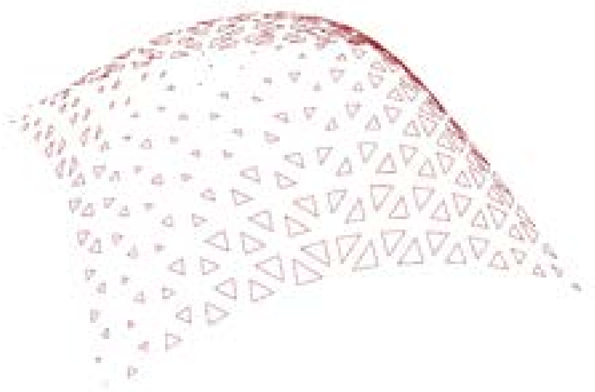


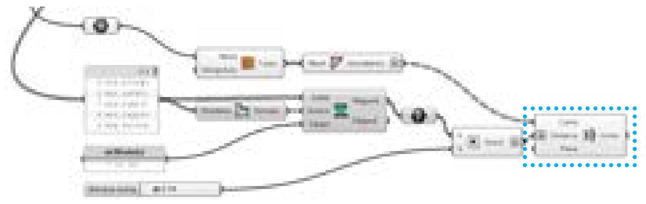


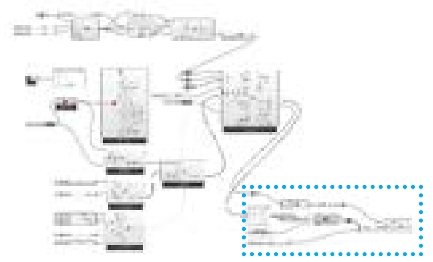


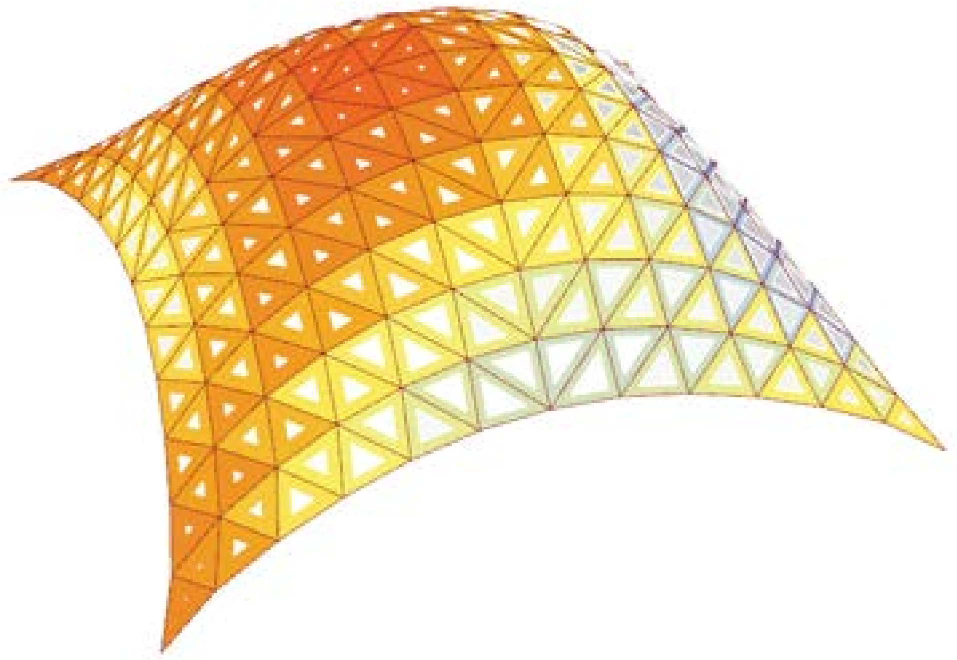
Thermal Data Implementation

1. Connect the Triangulated Mesh to the Environmental simulation.









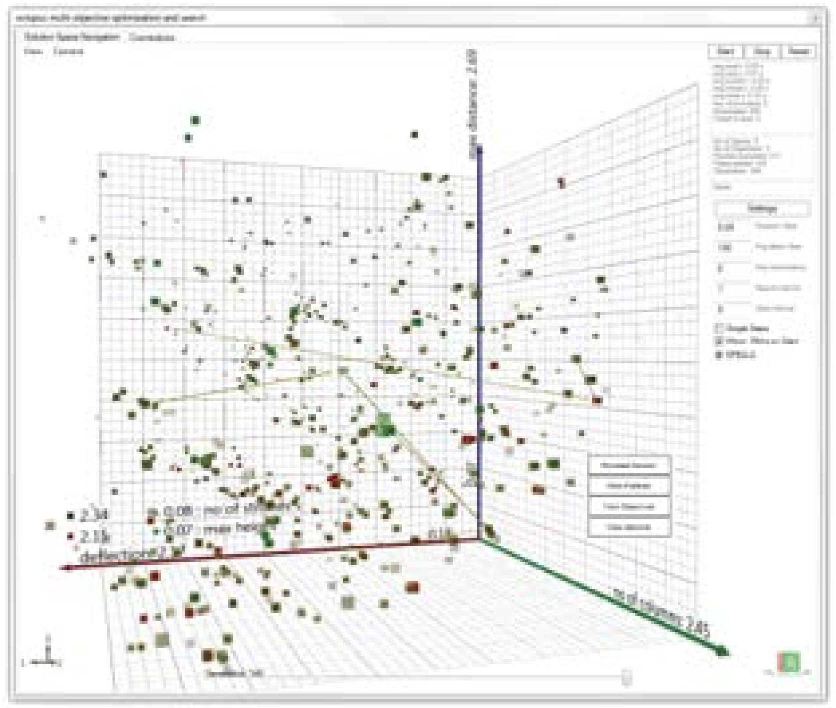
Face Mapping

1. Each mesh surface explodes and becomes one individual surface.
2. Each individual surface converts to a polyline and takes the boundary line of the surface.
3. Offset the boundary curves according to the result coming from the environmental simulation.

Result & Contribution

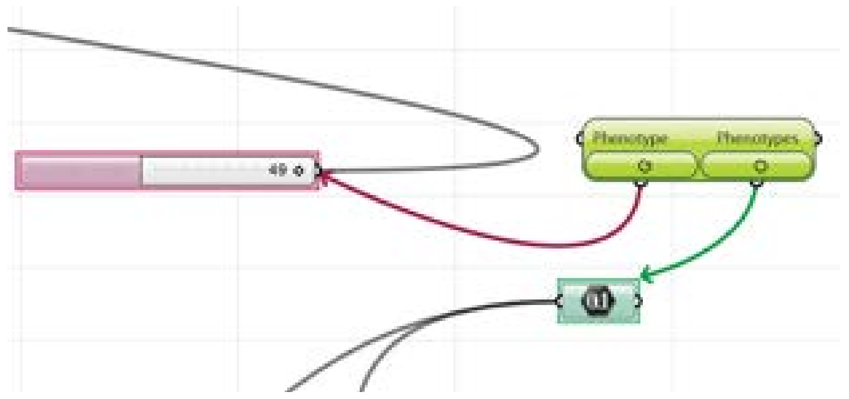
This Combinatory data implementation has proven the potential use of mixed matched codes in different hierarchy and an order that creates the different outcomes.

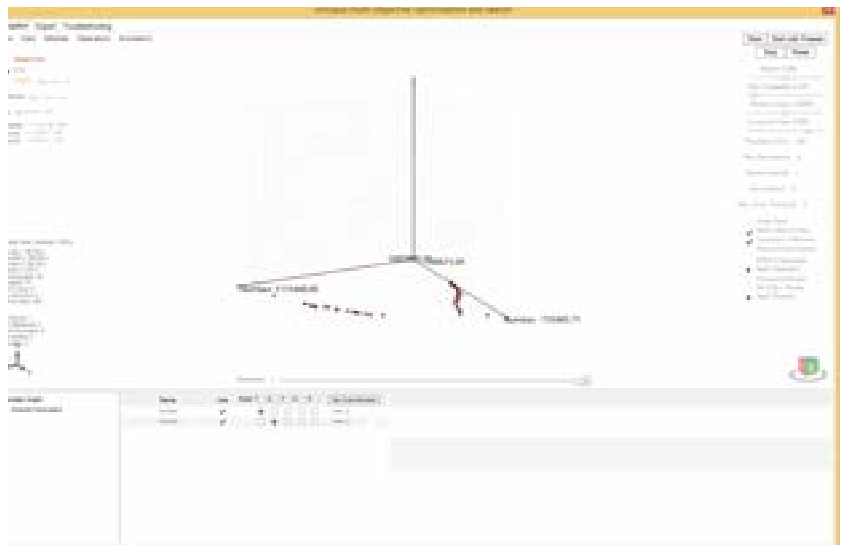
4. Evolutionary Optimization



Multi-Criteria Optimization

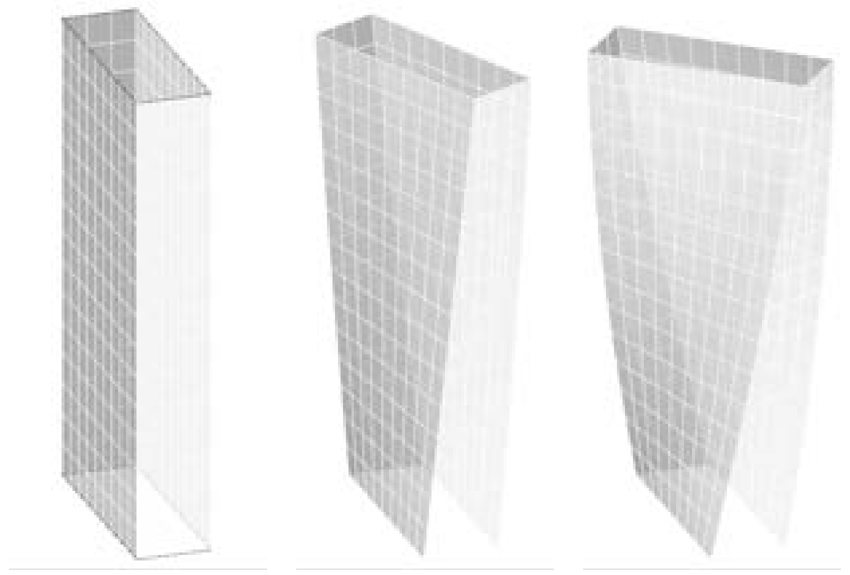
Octopus is a multi-criteria optimization program which uses the evolutionary logic as the base of the program. First and foremost, the difference between Octopus and Galapagos is whether it can treat multiple parameters or not. By using this tool specifically, environmental simulation tools can get an environmentally optimized result.

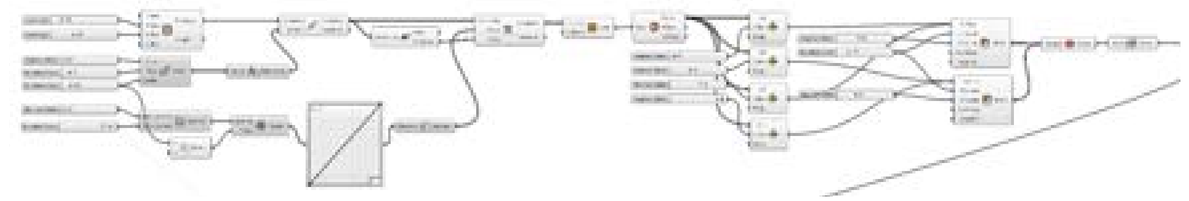




Gene pool & Objective

Octopus can receive the very first parameter of the code and also the result of the simulation. By using these start and end parameters, it tries to optimize the modeling by analyzing it back & forth. The next session will introduce the basic logic of the evolutionary theory with a specific example.

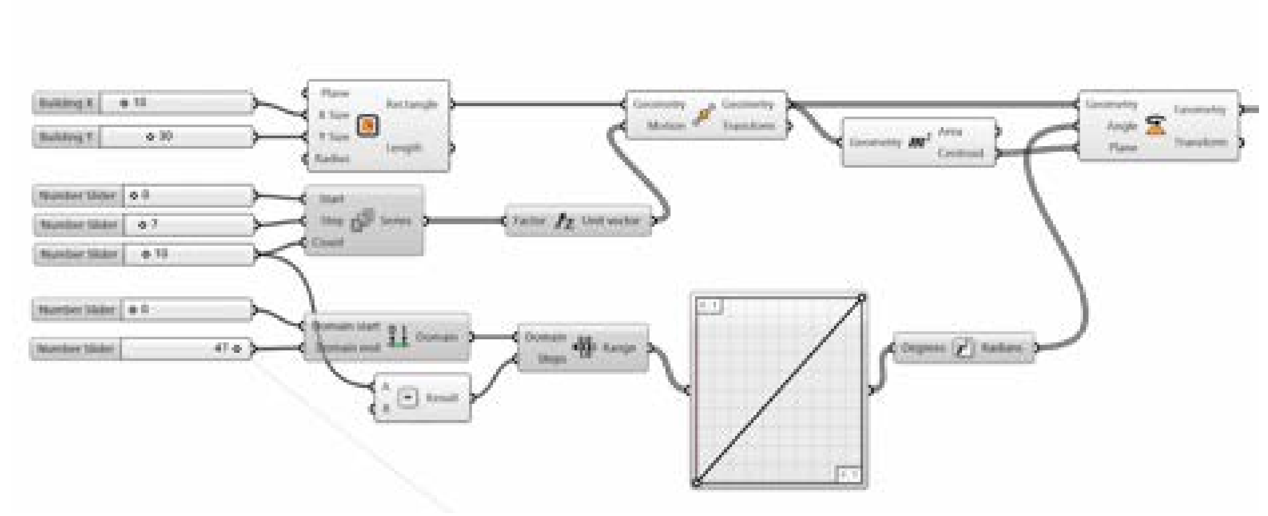


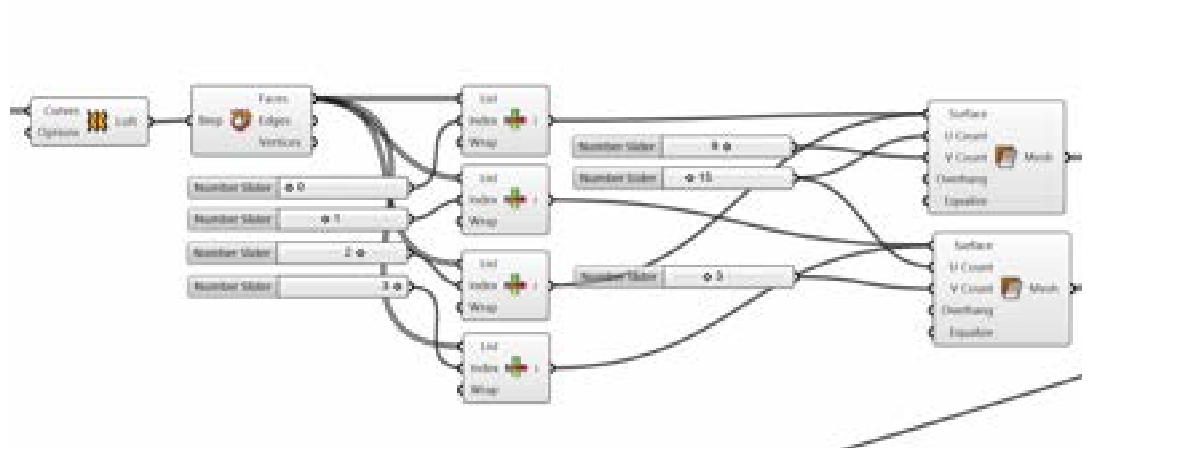


AP-21

Design parameters

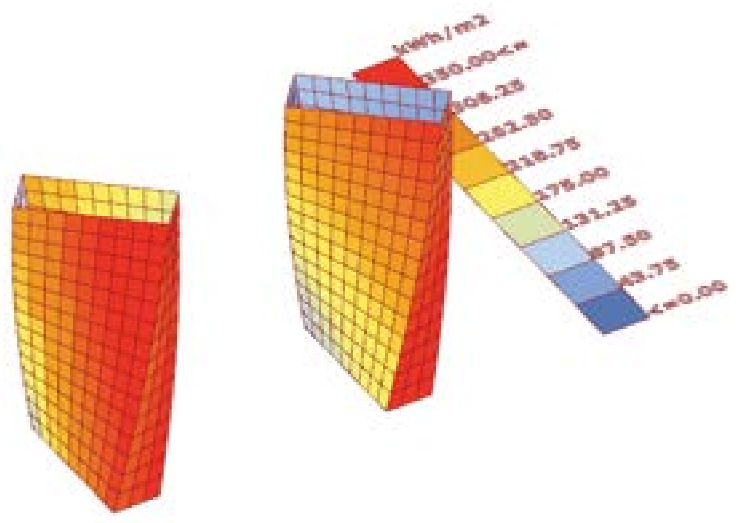
The left diagram above is the same mesh used for the former ladybug simulation. At this time, this mesh will be designed to twist a little with a single number parameter which represents the twisting angle.

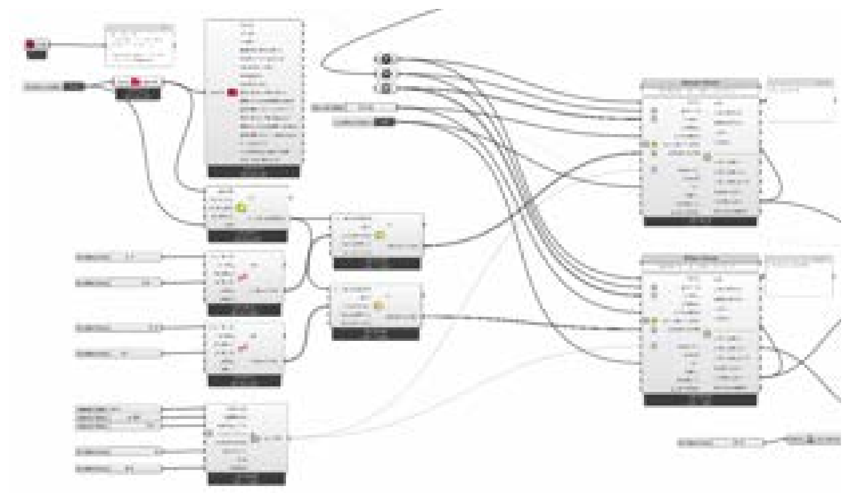


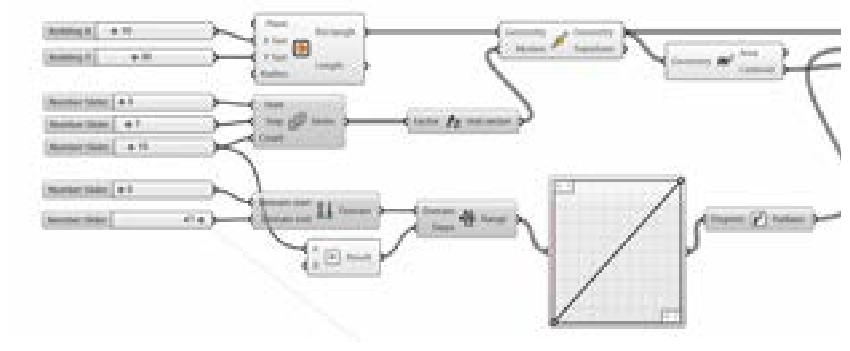


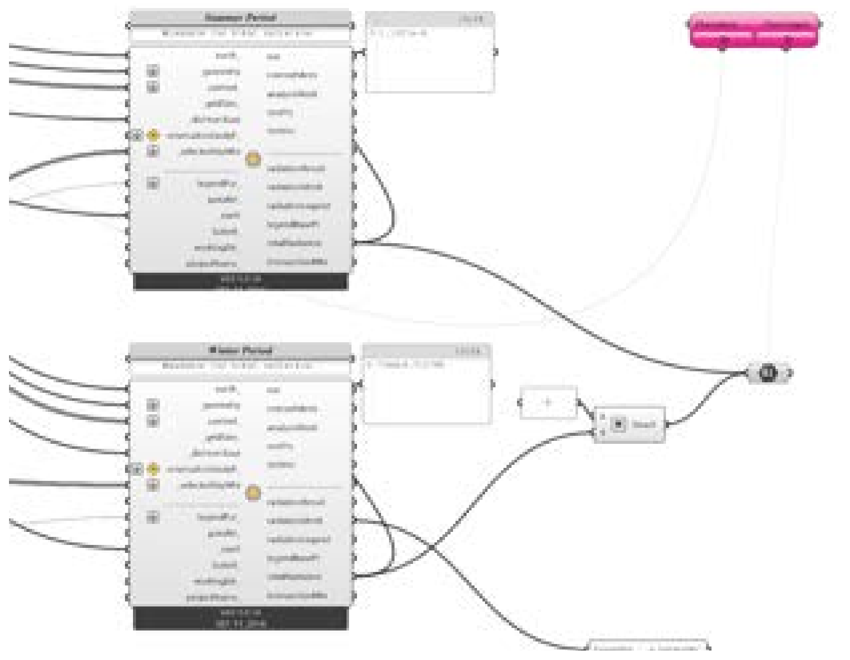
Basic Geometry

There could be numerous ways of making geometry depending on an individual's perspective in parametric design. The code above is one of the ways that can be create a twisting mesh geometry. In this session, it is strongly recommended that one develops their owen code to make the mesh geometry change its shape with a single parameter.







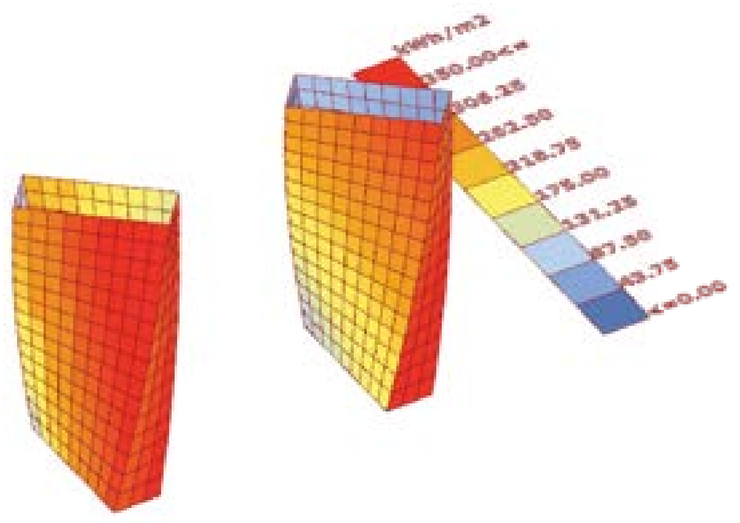


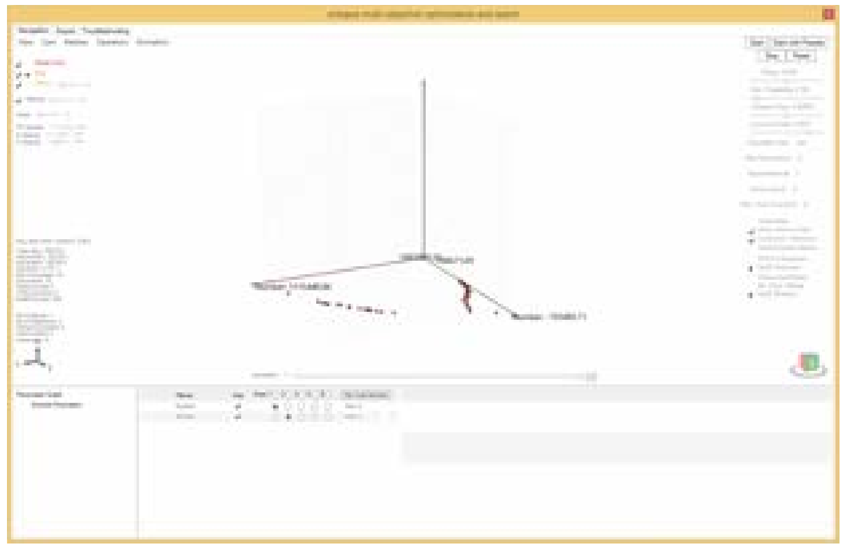
Environmental Simulation on Basic Geometry

Once the base mesh geometry is created, it has to be simulated with the ladybug plug-in. In this case, the simulation ran twice. Once for the summer period radiation and another for the winter period radiation. The setting is to maximize the winter radiation in order to gain more heat during the winter period and vice versa for the summer period.

Gene pool & Simulation Set-up

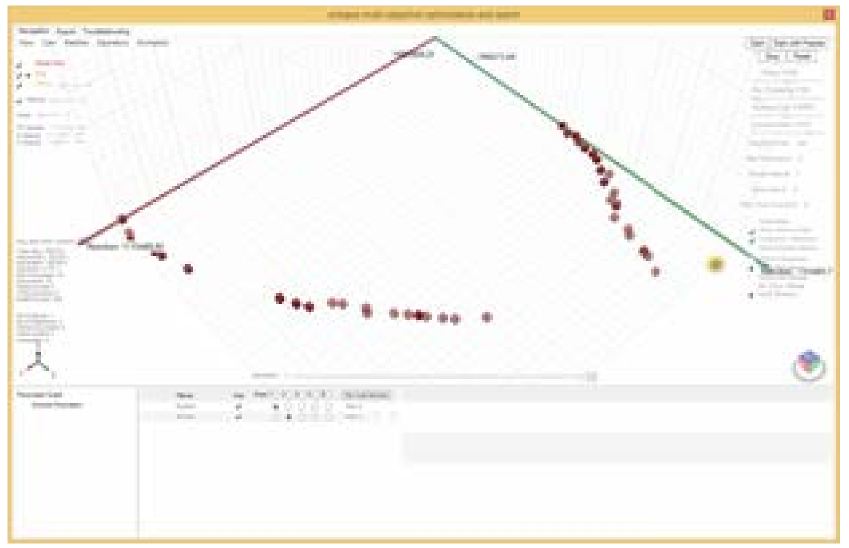
Evolutionary solver's basic setting is to minimize the result and the way to convert the 'minimize' function to 'maximize' is by multiplying '-1' to make the engine conceive it as a counter direction of evolution. In this code, to maximize the winter radiation, (-1) is multiplied to the winter total radiation and plugged into Octopus.

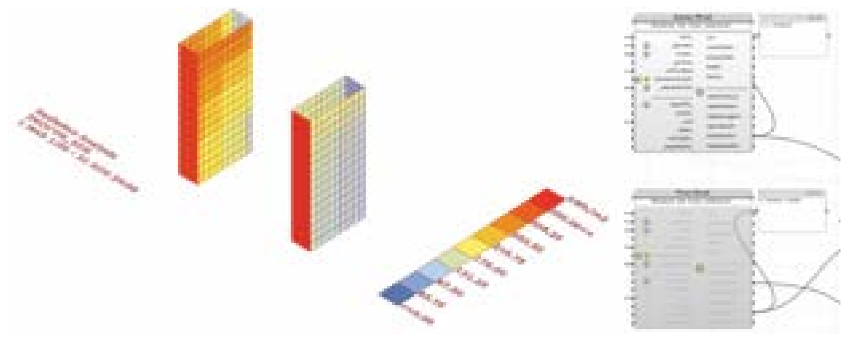




Evolutionary Solver

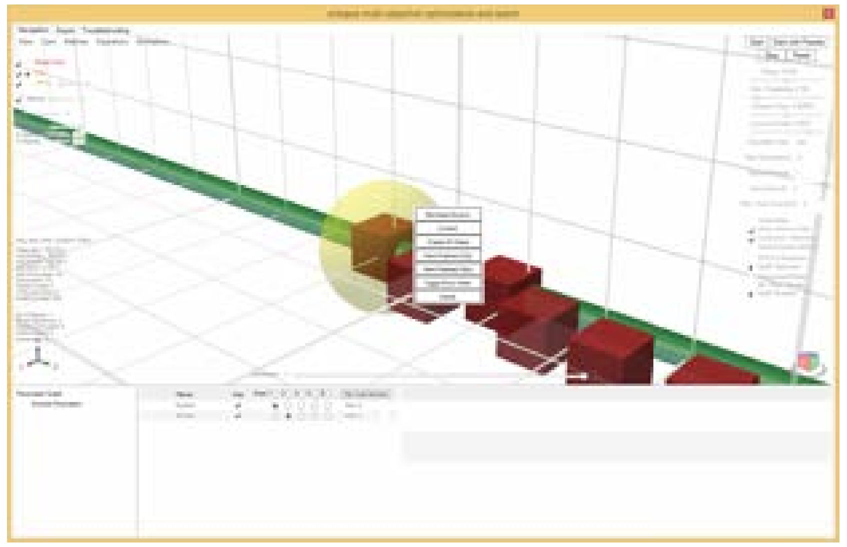
Once the solver starts to run, pop-up window with 3d graph appears to juxtapose the evolutionary options. Each option is represented as a small box which is conceived as a point on the image above and the shape of the 3d model that appears in the Rhino viewport is constantly changing according to its volatile input parameters.

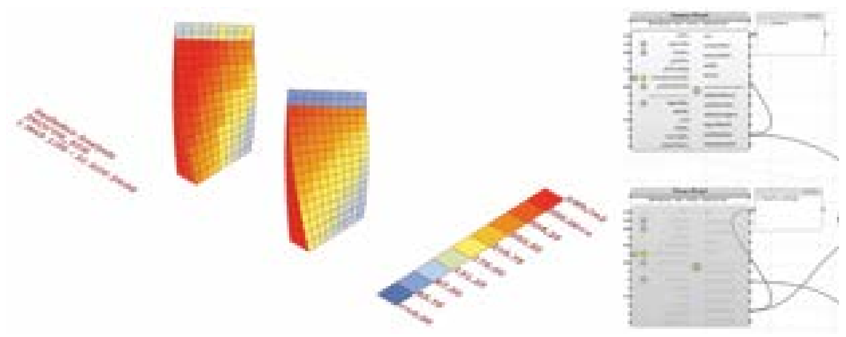




Generation

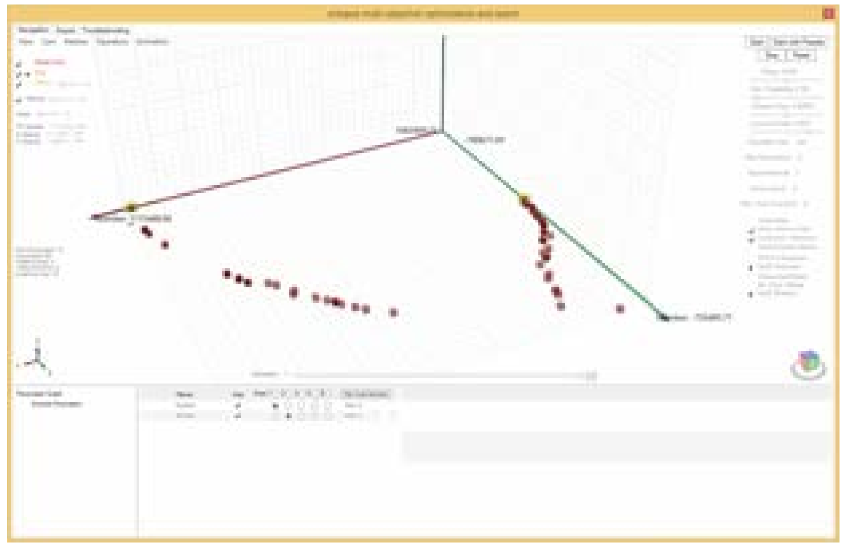
Since the Octopus simulation takes a long time, it is recommended to just simulate the one or two generations at the schematic design level. Once the job is done, the outputs can be checked in the 3d graph. The graph is showing each design output with boxes and as it is clicked, the input parameters and geometry in the rhino viewport is represented as the data in the box.

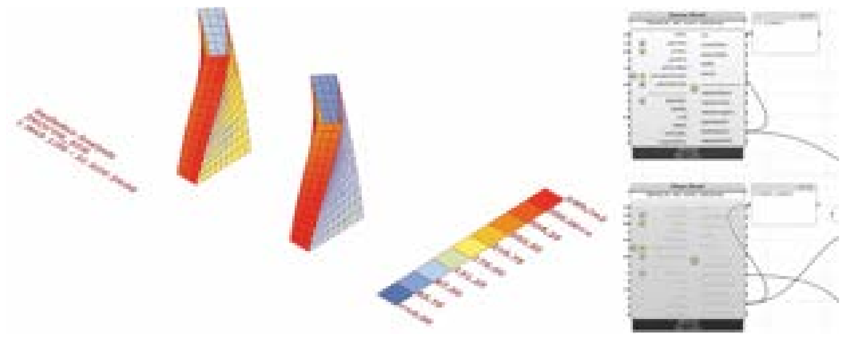




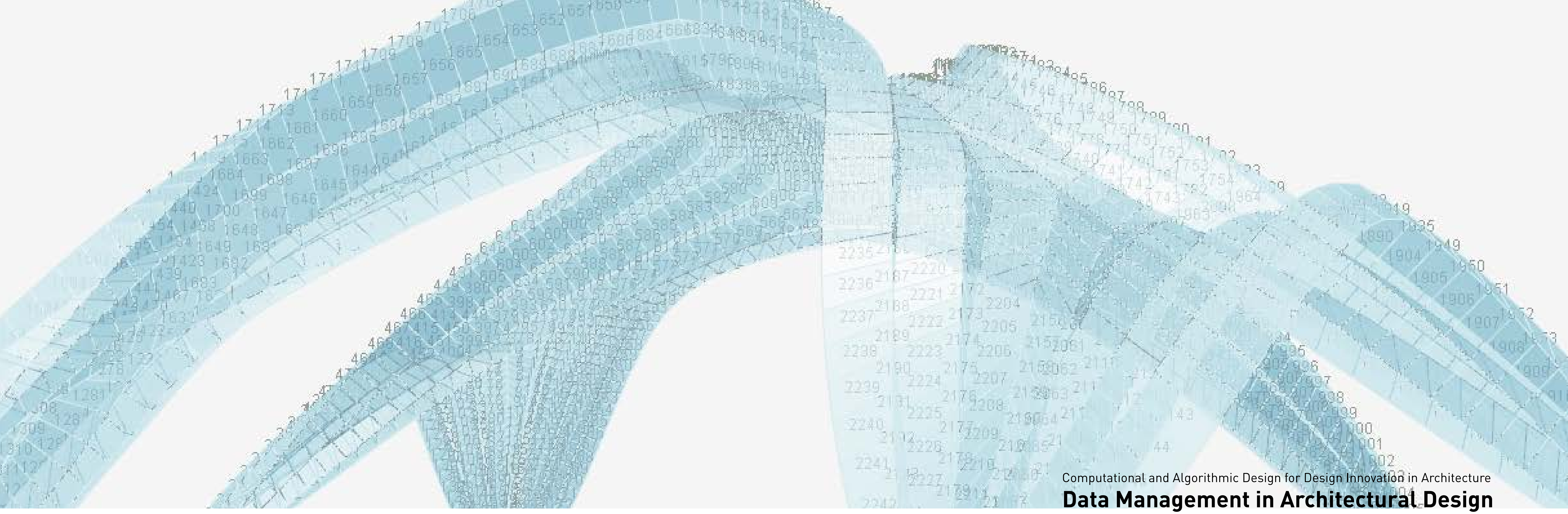
Options

Each option can be marked as a key option which can then be experimented in relation to other aspects of the project. The most optimized option can be shown at the very end of the box groups. In this case, since 2 criteria - Summer radiation / Winter radiation - are given as an output to be minimized & maximized, 2 possible optimizations were detected on the graph.





As it is described above, the two groupings of boxes mean optimization can be more focused on minimizing the summer radiation or more focused on maximizing the winter radiation. The choice between two options can be done with a design consideration. For example, in this case, the EPW file is weather data for Incheon, in South Korea, the region requires more cooling cost than heating cost. Thus, selecting the option that optimizes to minimize the summer radiation could be better.



Computational and Algorithmic Design for Design Innovation in Architecture

Data Management in Architectural Design

H Architecture

Computational and Algorithmic Design for Design Innovation in Architecture
Data Management in Architectural Design
Part 1

H Architecture

This research paper was written and edited by Dongil Kim and Jake Jaekyung Han at H Architecture. 2017.