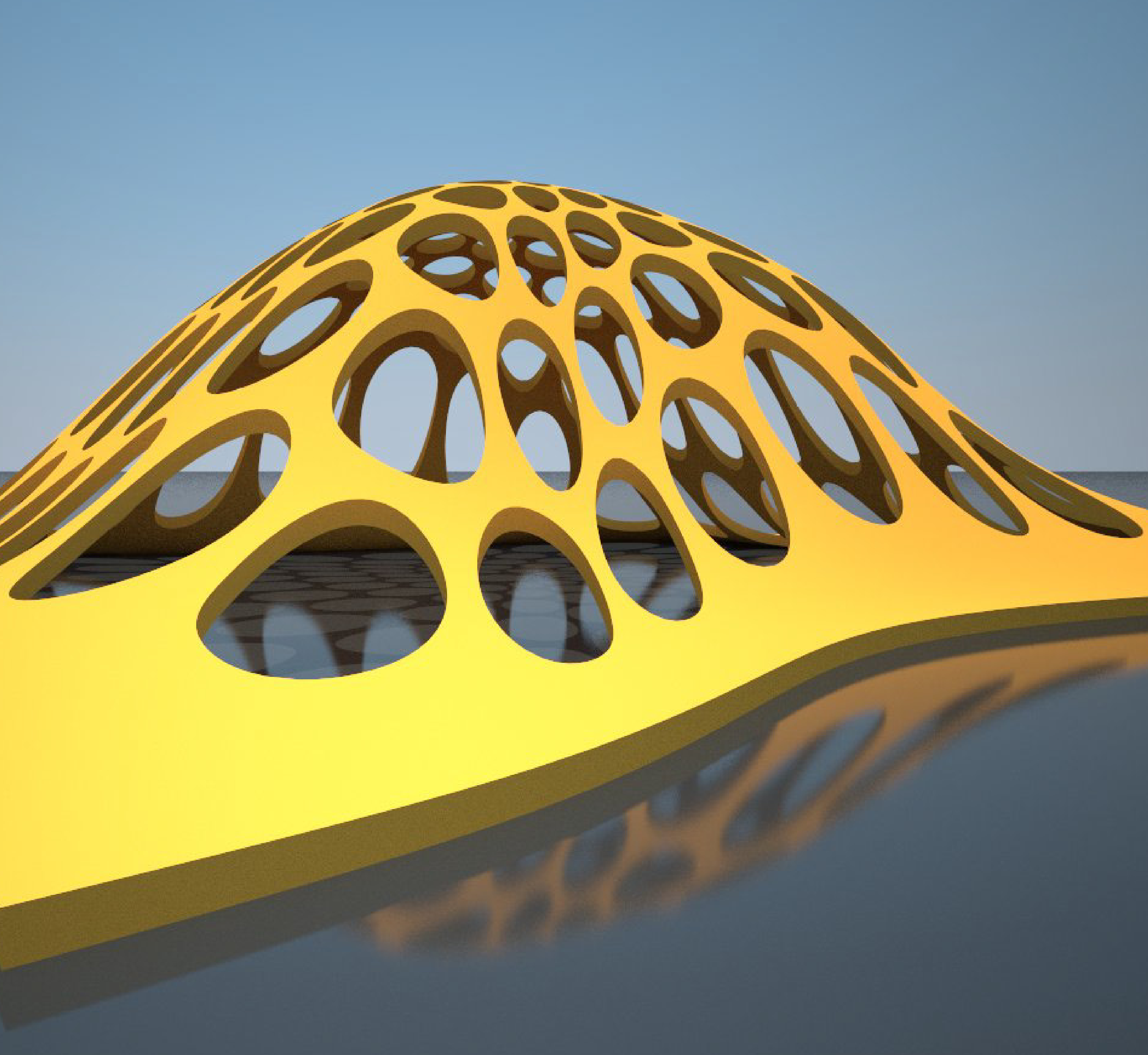


GENERATIVE ALGORITHMS

CONCEPTS and EXPERIMENTS: POROUS SHELL

ZUBIN KHABAZI

Generative Algorithms

Concepts and Experiments

2_Porous Shell

Zubin Khabazi

To see the latest updates visit my website or for enquiries contact me:

www.MORPHOGENESISM.com

zubin.khabazi@gmail.com

To **Michael Hensel**

Generative Algorithms, Concepts and Experiments

Introduction

The idea behind the 'Generative Algorithms, Concepts and Experiments' is to explore design and algorithmic solutions through experiments rather than pure geometrical/algorithmic subjects. Using current design problems or ongoing researches, here the development of such subjects pursued with a parametric approach in Grasshopper/Rhino Environment.

This book is a part of a research that was accomplished as a master thesis in Architectural Association, Emergent Technologies and Design course under supervision of Michael Hensel, Mike Weinstock and Achim Menges. In this research, full study of a natural system combined with algorithmic solutions in form-finding techniques to develop an algorithmic design tool. Here just a brief summary of research plus its algorithmic parts reflected. The research titled 'Algorithmic Morphogenesis' which generally investigated the field of form generation in algorithmic/digital spaces based on natural processes of form generation.

Considering the fact that parametric design is a fast growing subject and global experiments develop increasingly, here the aim of these series is to open up some of these subjects with detailed discussions on technique and method and also in each book, trigger some of the techniques that seem to be the point of concentration among those who share their interests, problems and ideas in Grasshopper Forum, weblogs or in global workshops as well. In this new book, with less description on Definitions and a global overview of the concept of algorithm progress, while all files are attached to be explored by readers, the idea is to bring forward these concepts. I hope it helps you in your design career.

Zubin Khabazi

Contents

1_Porous Shell System

1_1_Introduction

In a very general definition, architecture was aimed to inhabit human inside a space that covers him from wild extreme environmental conditions. During the centuries, it developed and achieved higher levels of complexity and advancements in technology, space organization, style, material assembly and so on. Buildings become complex products that have so many parts and have to fulfill different tasks. Among different building elements, building shells and envelops, as the border between interior and exterior space, developed with different ambitions to give beauty to outside, comfort and protection to inside. Building shells, from ancient domes, up to modern hi-tech envelopes have improved in technology, construction techniques, size, space quality and geometry and their beauty. But it seems that there are still more to explore.

_Looking at a specific line of research in building shells

Nature always seemed to be a great source of idea for human and for architecture as well. It inspired architects in so many different ways. In a very specific method, Frei Otto developed a technique, called Form-Finding, through which he tried to find the state of equilibrium of predefined material systems in predefined conditions while they become self-organized with natural forces. He used this technique to apply as a design medium in architecture. One of his researches in the field of building shells was studying the hard shell of some micro-organisms like Radiolaria and Diatom. In his research with biologist J.G. Helmcke, they carefully scrutinized how these organisms' hard shell get their form in nature and they tried to apply that process to a group of form-finding and design experiments.

This book is the further continuation of the methodology of Fri Otto in order to develop a shell system based on the formation process of the micro-organisms' hard shell. While using the methodology of form-finding, here it is pursued via current tendencies of design and technological advancements in building technology and the main aim is to develop the same methodology in digital space and the final product should be able to be fabricated with new machineries like Laser-cutters or CNC machines.

It should be mentioned here that the book you are reading mostly reflects the digital algorithmic part of a bigger research since the original text is way too much to be interesting to read as a design experiment summary.

1_2_what are Micro-Organisms?

Micro-Organisms are small microscopic organisms which are being studied in microbiology. There is a specific group of micro-organisms that share the same property which is a hard skeleton as a shell, which remain as fossils after their death. Some of them are: Calcareous Nannofossils, Conodonts, Diatoms, Foraminifera, Ostracods, Palynology and Radiolaria (www.radiolaria.org).

Radiolaria: Radiolaria are single cell organisms, part of marine planktons occur in all oceans. They are non-motile structures that float in the water and their body also prepared to enhance the buoyancy of the organism for this life style. Radiolaria are interesting for biologists and other scientists because of their silicified porous shell as one of the nature's most amazing structures. "Individual radiolarians are normally in the size range of hundredth to tenths of millimetres, but some reach the dimension of millimetre or more [...]" (www.radiolaria.org).

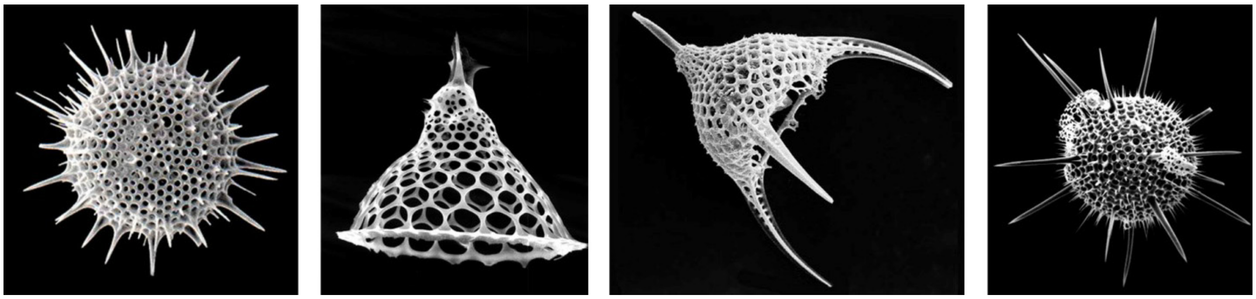


Fig.1.1. Samples of different species of Radiolaria.

1_3_Micro-Organism's Morphogenesis (process of form generation)

The process of form generation of micro-organisms like Radiolaria or Diatoms is among interests of different scientists up to architects and engineers. Since the result is a hard shell which has structural capacities, functional aspects of the cell and also includes the process of self-organization inherent in the process, it is useful to extract the concept of their form generation from the careful observation and study for further use.

The general concept of the form generation of micro-organisms is put forward by D'Arcy Thompson, revolutionary scientific thinker and author of the classic 'On Growth and Form', based on vesicular formation. This concept is as follows:

'The organism produces closely packed foam of vesicles around its cell nuclei. Then siliceous materials from seawater precipitate gradually in the interstices of the foam bubbles and start to shape the organisms shell. Since this deposition happens on top of these closely packed bubbles and vesicles, it produces a layer like an envelope over the cell (and not happens inside the cell), so that when this formation process were completed and vesicles reabsorbed, the final skeleton would be a latticework of holes'.

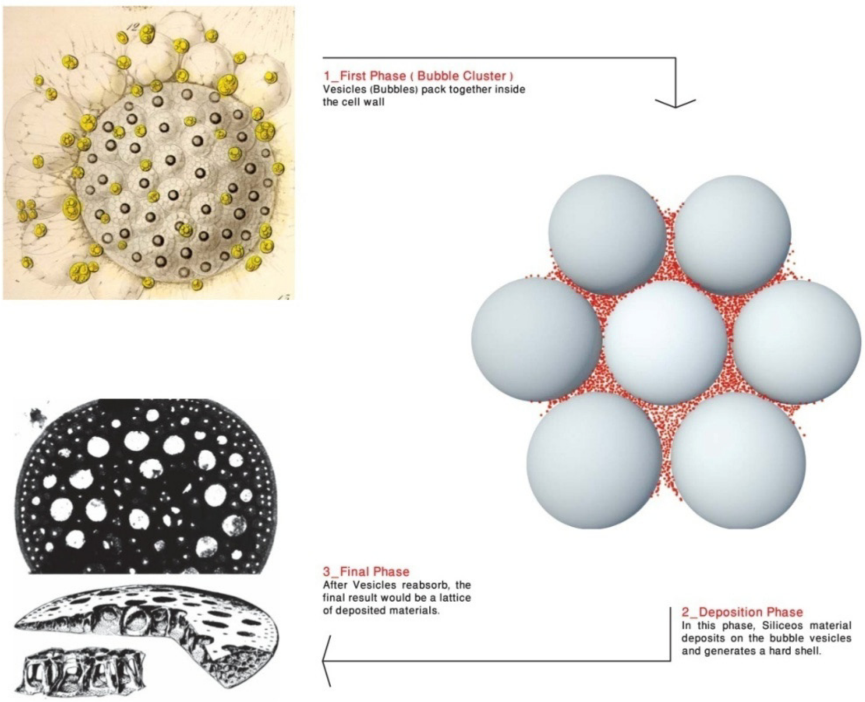


Fig.1.2. Formation process of hard shell in micro-organisms

So basically, **close packing of pneumatics (vesicles) and deposition of siliceous material over their interstices** is the concept of form generation of these micro-organisms which is believed by scientists.

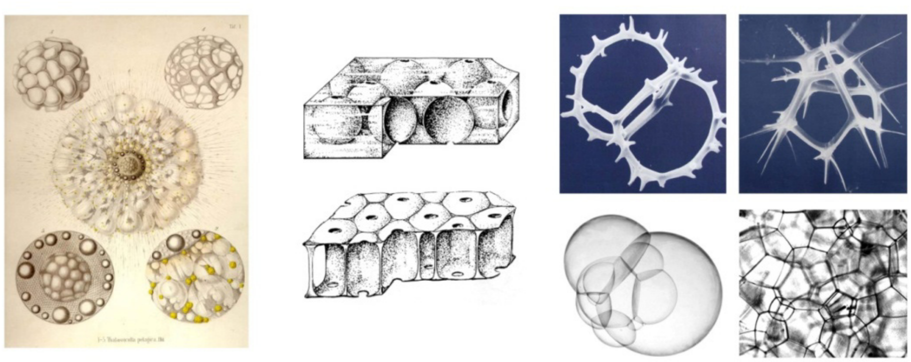


Fig.1.3. Close packing of Pneumatics as a concept of form generation

1_4_Micro-Organism's Forms: Ideal forms and Deformations

In an ideal situation, the hard shell which is like a lattice should have a regular hexagonal pattern. This is because of the mathematical laws regarding identical sphere packing. But perfect geometries rarely happen in nature. There are multiple reasons that cause the pattern not to be homogeneous at all times. If the differentiation of bubbles in the net happens (which happens almost always) then the result would be chambers like deviated polygons (not always hexagon, but it could be pentagon or heptagon as well) or deviated circles. The shape of chambers is usually convex which is related to concept of their formation from a pneumatic base.

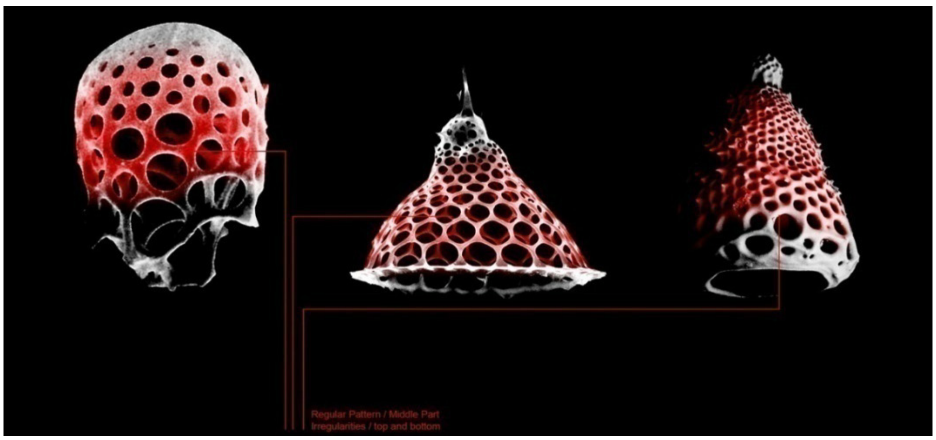


Fig.1.4. Regularity/Irregularity of the pattern

_Regional organizations of chambers

Having various global forms (especially double-curved and not flattened), hard-shell of the micro-organisms could be seen as an accumulation of chambers over a global surface. So based on the form, curvature and geometrical aspects of this global surface, general and local organization of chambers could be an issue of observation. This might show that how different types of chambers sorted or how they gradually change their size or pattern in order to cover the outer surface of the organism. Zonal organization of chambers could be simplified on a sphere as follows:

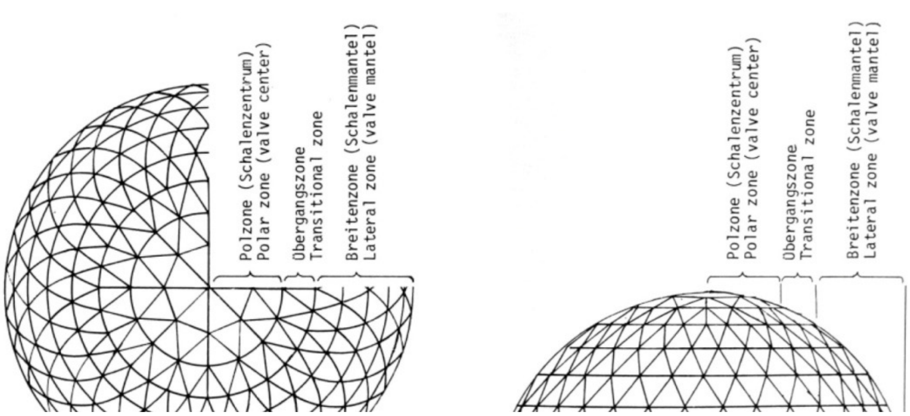


Fig.1.5. Zonal organization of chambers (Frei Otto's studies in IL)

1_5_Geometry and Topology Analysis

_Geometry

The general form of Radiolaria has different varieties. There are multiple cone shapes, spheres, and their deformations. Having this variety, in terms of geometry analysis, it is more useful to look at the lattice in small scale.

As mentioned before, the general shape of the lattice has hexagonal pattern. Looking closer at each chamber in the net, it is surrounded by six others. Chambers are not as homogeneous and regular as a mathematical model and a closer look would reveal that each chamber by itself has a deviated and differentiated hexagon or ellipse geometry. Especially when the net pattern converges towards the tip point, it causes irregularity inside the net. In most cases, chamber sizes and lattice thickness are bigger close to the edges and smaller in middle of the shell. The size differentiation is gradual across the lattice unless in marginal areas where irregularities happen.

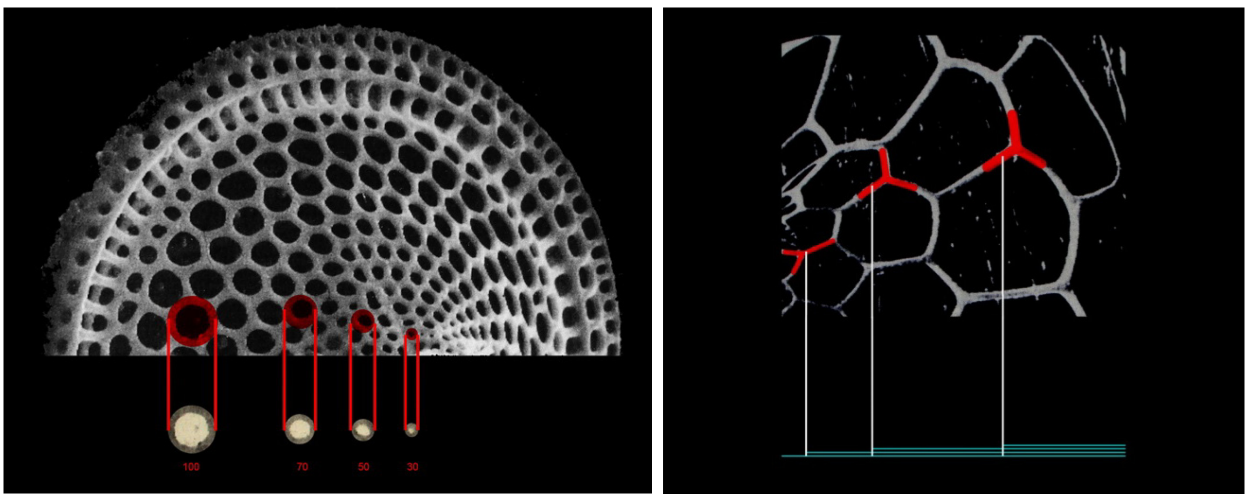


Fig.1.6. Geometry analysis in chambers and lattice.

_Topology

While geometry includes shape, form and logic of pattern, position and other related issues, Topology refers to the connectivity and neighbouring condition of chambers. Since pattern formation of shell always discussed as nets which have segments and nodes, so connectivity or 'topology' of nodes of the net is important. If each bubble's (sphere) volume centroid accepted as a node in the net, then connectivity would be defined as line segments which connect these center points. In a shell, these points would be situated in the middle of chambers in an empty space.

There are different methods of connectivity between these nods but a triangulated net seems to be the best representation of the connectivity pattern. In a mathematical point of view, a triangular mesh is the best way of covering curved surfaces by meshes.

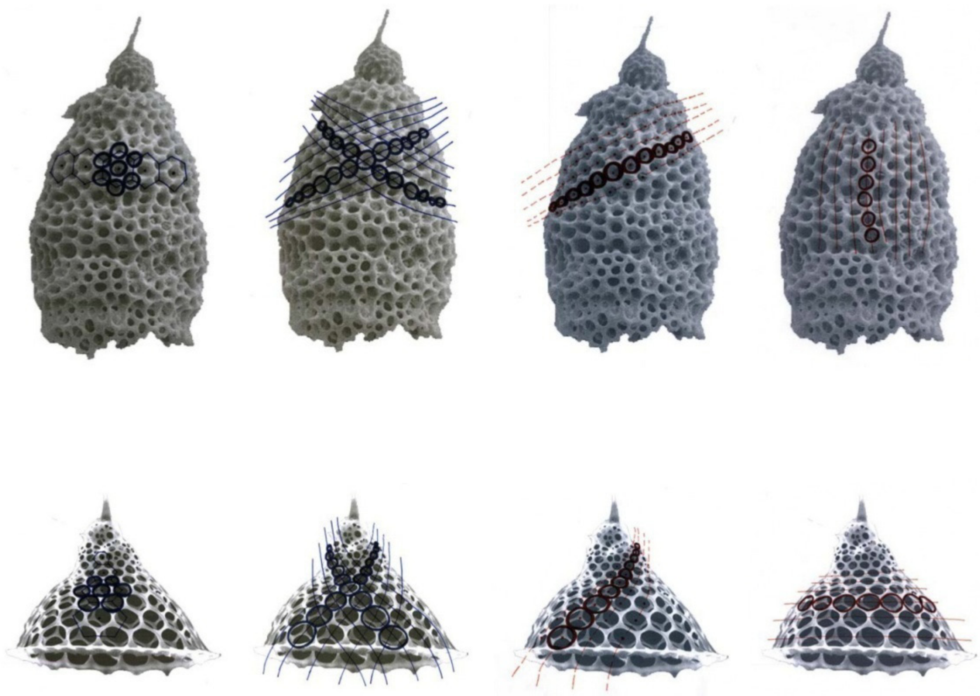


Fig.1.7. Different methods of Chamber Connectivity (Topology)

_Summary

Size Differentiation	Happens to control the relation between location of chambers and regularity of the pattern.	Related to general morphology
Chamber Deformation	Deformations happen when the number of chambers are not enough in one direction to cover the whole body and chambers are needed to extend towards that direction.	Related to local position
Pattern Deformation	when the general body (overall morphology) is not a regular one, some pattern deformations are needed to cover the shell.	Related to global body form and local irregularities
Zonal Organization	In general, even in a very symmetric geometry, zonal organization of chambers is needed in order to cover a spherical geometry with circles via hexagonal pattern.	Related to global body
Lattice Thickness	Usually thickness of the lattice is thinner, close to the centeral zone and thicker close to the edges.	Related to global body and local deformations and material availability
Differentiation in Form Elements	in order to fulfil the organizational, structural, and other performance capabilities of the organism, different form elements are being used in different species. It is believed that it is highly affected by the environmental conditions.	Related to global body and local changes
General Morphology	General morphology of radiolaria has been divided into two main categories of spherical and conical ones. other types of geometrical forms have been observed but they all could be classified under deformated ones.	Mostly affected by environment and other external factors

2_Algorithmic Morphogenesis

2_1_Computation: General Consideration

Studying the natural process of form generation in micro-organisms, now the aim is to start the digital process of form generation through computational techniques but first some general notes should be considered:

_Generic form

Considering the process of form generation and looking at the shell, the general form of the radiolaria are always respond to the pneumatic structures and in general it is mostly spherical or conical. But the idea in digital algorithms is a bit different. Here any generic double-curved surface should be considered as an input geometry to start the algorithm and generate the form. It is obvious that the result of the algorithm should not just limit to the catalogue of forms of radiolaria. This process is aimed to be used as generator of the same concept for any given surface which later on could be transferred into a physical entity that should perform as a shell system.

_Scale

One of the main issues in Biomimetic and using nature as a source of idea to generate something artificial is scale. Some great ideas in microscopic scales are not viable to transfer and mimic in the macro scale. That’s way any physical outcome of the biomimetic, undergoes lots of experiments and tests to check if the idea is working on the proposed scale or not.

Here the idea of scale has a general consideration. It is possible to generate the resulting model with the average scale of micro-organism’s chambers. So whatever is the scale of main surface, the resulting lattice would have the same ratio as for example radiolarian has. This would be satisfying for the algorithm in a digital space. In addition, any predefined scale, even far different from what radiolarian has, would be possible since the form generation is still in digital realm and up to the time that this scale is flexible, the process would be acceptable.

But if the process is aimed to end up in a physical outcome and designed for a product, then scale would be an important factor. In this situation, there are multiple factors that should be taken into account. What is the material? What is the scale of the project? What is the method of fabrication and assembly? What is the degree of self-stability? What is the established performance of the system? ….

So basically shifting the process towards any physical experiment and project, means further research that should be carried on in relation to that specific material and also criteria of performance that this experiment designed to fulfill.

_Ideality vs. Reality

As studied in micro-organisms, there are different cases of irregularities and deformations in the pattern of chambers or in the geometry of chambers by themselves. The question here is, to what extent should the lattice be regular and where should it be allowed for irregularities. Should the algorithm even refine the geometry in a limited degree to generate the ideal lattice or it should exactly follow the predefined situation and use irregularities in form generation.

Answer to these questions is not an easy one. There are multiple issues that affect the process and result and define criteria to evaluate the degree of regularity/irregularity depends on the material, scale, fabrication method and so on. For example using only complete circles as the main shape of the chambers might cause some parts of the lattice become thicker or thinner. This would further affect the general weight of the structure in a big scale experiment. But deviation in chamber's shape towards ellipses would retain the thickness of lattice almost the same with different modules.

It is also possible to refine the geometry in order to fix irregularities and generate a regular and ideal output, yet the degree of refinement depends on the preconditions of the material experiment (material, scale, fabrication, …). Since these refinements would change the general geometry, in a real project while there are connections between all elements of design, this type of geometry changes should be precisely observed. It is assumed that it changes the position of certain points or areas inside the general geometry that would cause disjunctions with other parts of the design.

_Final outcome

The process of computation has two main steps. The first step is the form generation process and the result would be a digital geometry which is associated with certain external data. This is the main aim of the algorithm which converts the natural morphogenesis of an organism to a digital one. But the final outcome of such an algorithm is not something that could be directly used as a product.

The next step is to convert the generated geometry to a real material product. This could be any type of material with suitable fabrication technique. Based on the physical necessities and structure, fabrication, assembly and any other consideration in the field of that specific material system, further steps of the algorithm should be designed and developed to meet those specific conditions. The second part of the algorithm also informs the first part in terms of size, numbers and so on.

2_2_Method

Based on all general issues discussed, the method of computation should respond to the form generation concept, studied in natural form generation phase. In this method, for any given surface, as the final state of the shell, organized groups of chambers should be generated over the surface to subtract and form the lattice. This lattice then should be prepared for desired method of fabrication.

In more detail, a reverse process of analysis of organism's shell, here the method of form generation in algorithmic space is to triangulate any given generic surface to make a net and then generate chambers over the nodes of this triangulation net. Removing the area inside each chamber from the surface would result in a porous surface as a representative of the lattice.

The main underlying part of the algorithm is the way this input surface triangulates. It is important that triangulation should encompass all geometrical/topological aspects of the organism's shell which is studied before. This triangulation should be center oriented, towards the center point of the surface and it should consider size differentiation from edges towards center point.

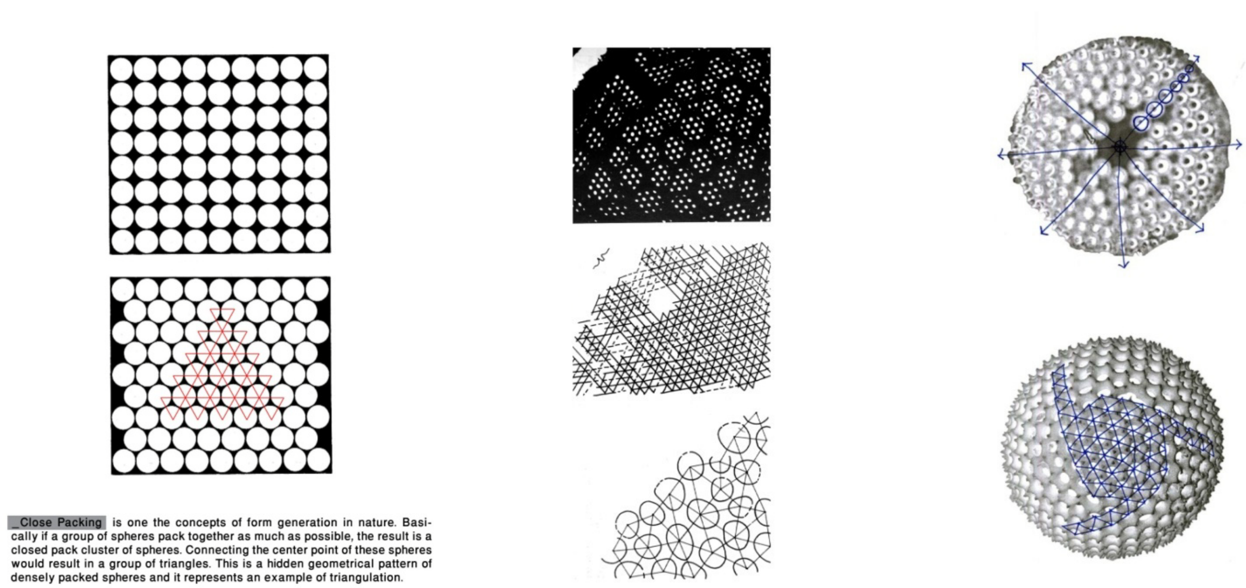


Fig.2.1. Analogy of the resultant net of close packing and triangulation.

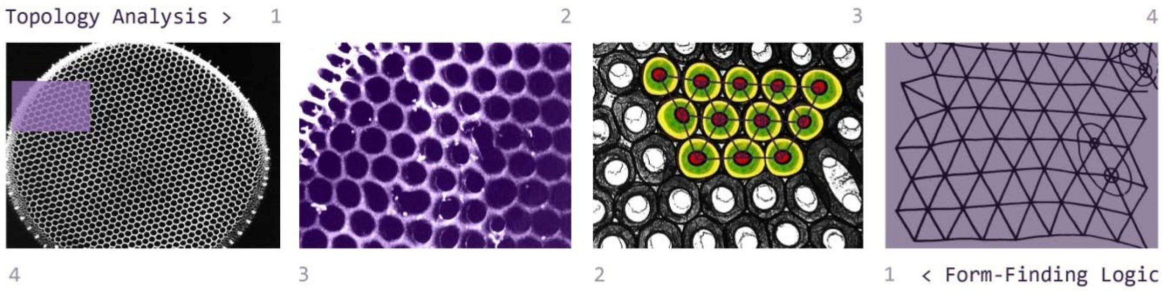
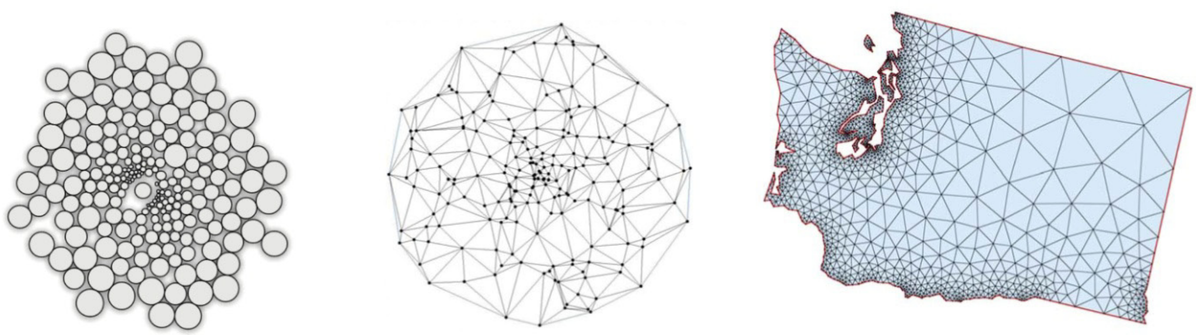


Fig.2.2. Process of form generation



_Triangulation is not a simple formula or technique with a unique result that could be applied to any given geometrical problem. There are different possible ways of triangulation under the main title of Tessellation. Polygon Triangulation, Delaunay Triangulation, Mesh Triangulation, Surface Fitting and Segmentation are among them. There are also many techniques and algorithms that manipulate them like Relaxation, Mesh Refinement, Optimization, Reduction, Subdivision and so on.

Basically Triangulation is a technique which was developed in survey and geometry. It is a technique to locate objects or study the terrain condition (usually in large scales) and so on. Recent technical developments in computational geometry bring new tools and softwares that could calculate and generate triangulation.
Most useful toolkits and algorithms that generate triangulations like Delaunay Triangulation, are working with point sets. Despite the origin of triangulation that comes from survey and terrain studies with no predefined locations, here in computational geometry, it is assumed that there is a point set and the aim is to find the triangulation that meets the condition of Delaunay for example. Or the concept in mesh triangulation is to find the minimum amount of triangles with which a mesh could cover a geometry and there is not any restriction in the angles and so on.
So if triangulation is the geometrical analogy of the sphere packing that happens in natural morphogenesis, here the main issue in front of algorithmic morphogenesis is to generate the proper triangulation. This means that the algorithm should be able to receive a given geometry (a generic surface) and in the first step tessellate it in triangles. Since most algorithms of triangulation working with predefined point sets, and there is not any predefined point in this case, it is important to position the points and then find the connectivity which generates a net of triangles.

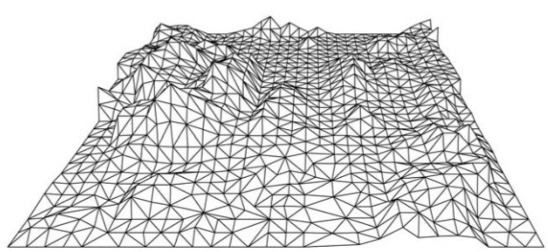
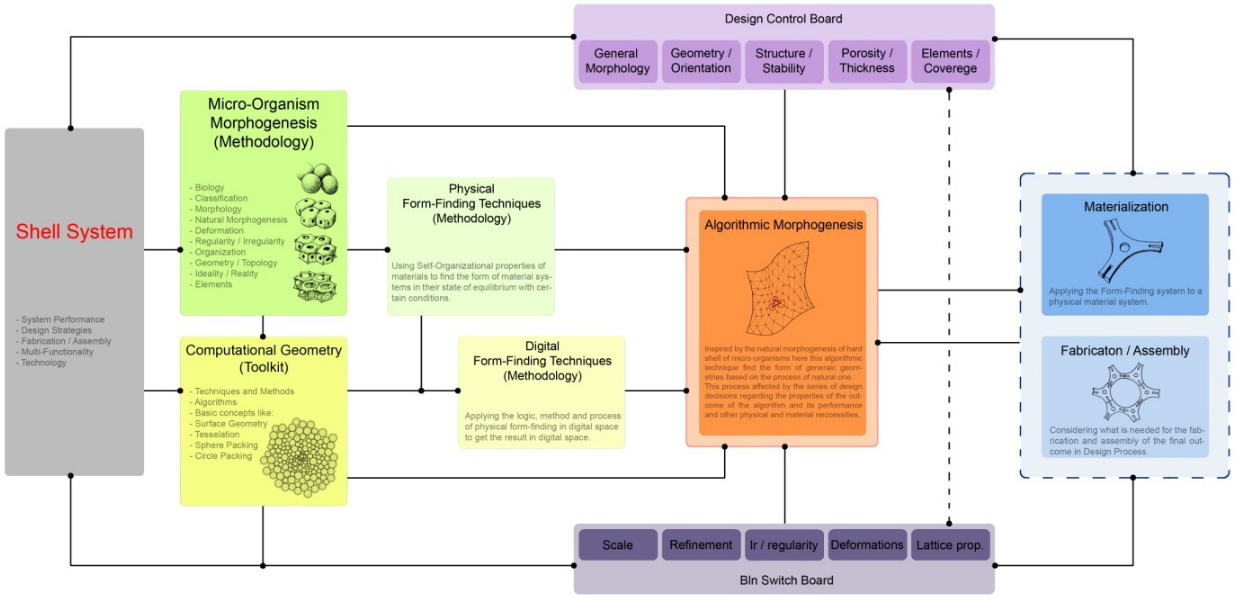


Fig.2.3. Triangulation

2_3_Concept of the Algorithm



As outlined in the main research diagram above, the core of this research is the design algorithm, the algorithmic morphogenesis tool that works based on the natural morphogenesis of the organism being studied. This algorithm as described in the algorithm diagram has different inputs, variables, switches and manipulations also in multiple phases. Here these phases are described as an overview:

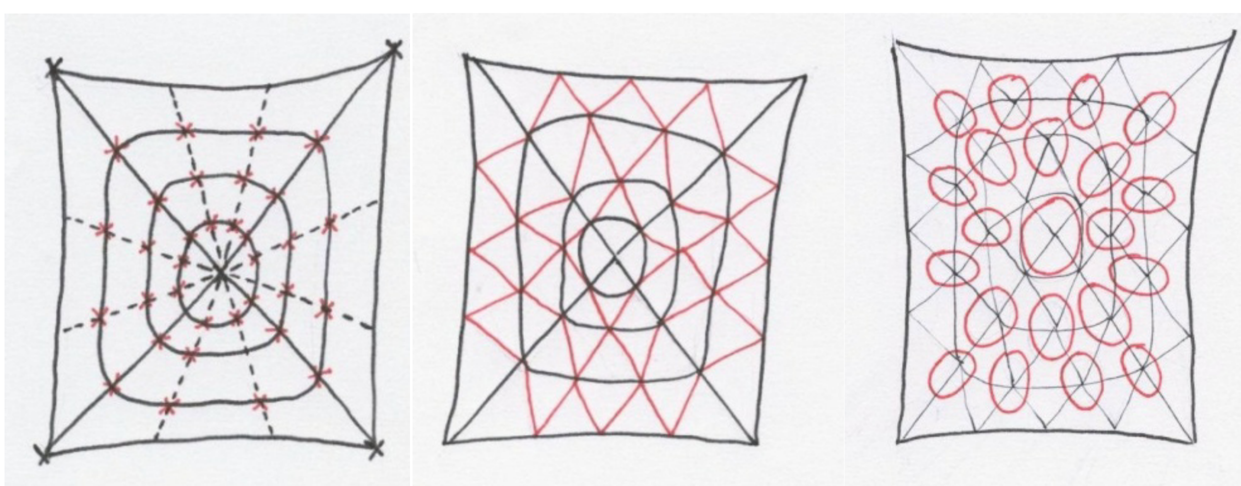


Fig.2.4. Form-Finding Algorithm progress

_Algorithm Phases:

1_Phase One_Center Point and Contour lines

The organization of chambers in radiolaria is center-oriented and radial. This type of organization in any given space needs a center point that should be found. After locating center point, several closed curves as contour lines should be generated around it up to the edges of the surface. The general scale of the surface and desired size of chambers define the amount and distance of the contour lines. This phase of the algorithm corresponds to the general geometry of the input surface.

2_Phase Two_Surface Subdivision and Triangulation

After finding the center point and generating contour lines, the surface is ready for triangulation. As studied before, this part corresponds to the logic of sphere packing. Basically the algorithm divides the contour lines into pieces and connects them to generate the base lines of triangulation. The number of base lines and the distance between them are all adjustable. Manipulation of them depends on the chamber sizes and local surface curvature. The result of this phase is the virtual position of all spheres (bubbles) as if they packed together. This phase corresponds to the physical behavior of the natural morphogenesis of the organism.

3_Phase Three_ Chamber Generation

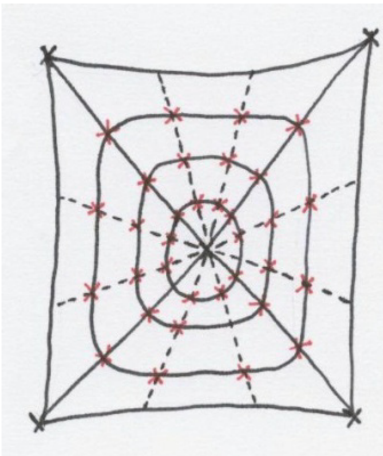
In this phase all preconditions for chamber generation is met and process should focus on the single task of extracting data and drawing chambers. Each node of the net is a center point for a chamber to be drawn around it, using parameters of all net segments connected to that point. After all chambers being generated, these curves and their inside area should be subtracted from the main surface which makes a lattice shape surface which is porous shell.

4_Phase Four_Fabrication

Although did not mention before, all issues regarding size and number of chambers, thickness of lattice, etc. should be defined by the values coming from the product, design, material and fabrication necessities. While the design phase has been informed by such information, now the porous shell is ready to go to the process of fabrication and assembly. It should be decided which fabrication technique going to be used and what is the material and how the final material product would be assembled.

2_4_Algorithm_Grasshopper Definition

Algorithm_Phase One_Center Point and Contour lines



As mentioned, this part of the algorithm designed in order to find the center of organization to generate closed curves as contour lines from center towards edges (from small sizes towards bigger ones) which would be base lines for triangulation. These curves aimed to be generated using radial subdivision of the main surface (by lines from edges towards the center point).

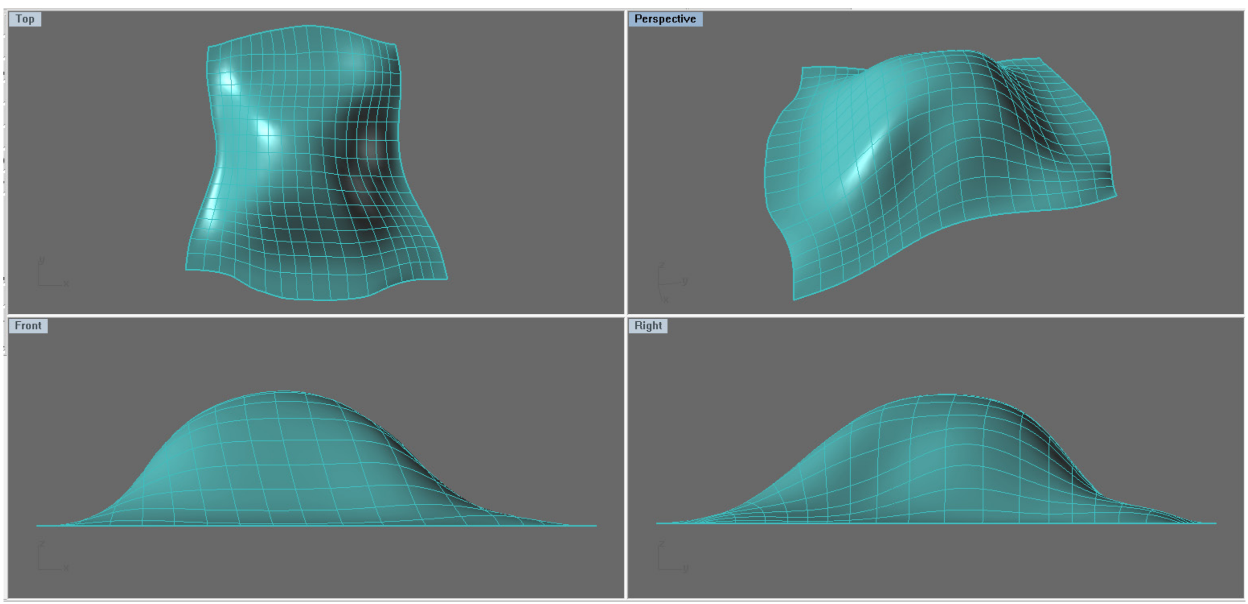


Fig.2.5. Algorithm's input; A generic NURBS surface.

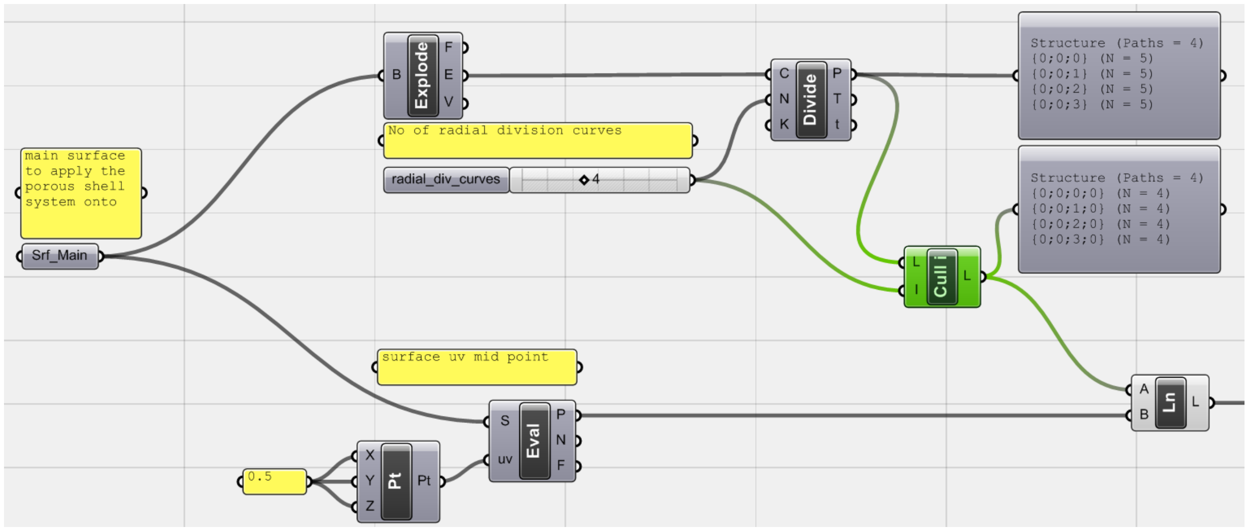


Fig.2.6. The main surface introduced to canvas by <Srf_main>. Components of this surface as a BRep <Explode>d to get its edges. These edges are <divide>d in order to generate start points for radial subdivision lines. At the same time, mid-point of the main surface also <evaluate>d as the second point for radial subdivision lines. Since start points and end points of edge curves are overlap and produce two points at each corner, end point of all division point lists omitted using <Cull i> with the index number equal to the number of divisions.

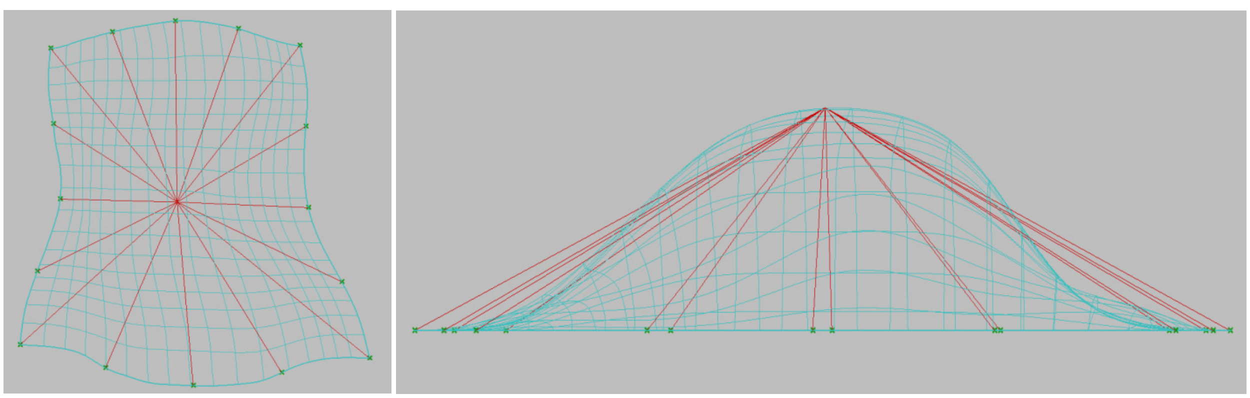


Fig.2.7. Radial subdivision lines. Although these lines are not lay on the surface, for this level of algorithm they would be fine.

To generate contours, these radial subdivision lines should be divided in order to provide points for curve generation (as contours). The next step is to prepare parameters to evaluate subdivision lines.

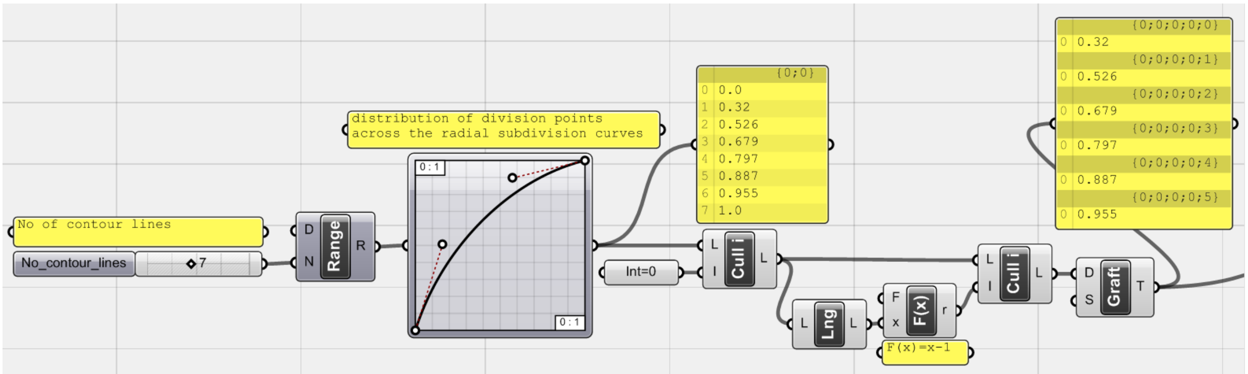
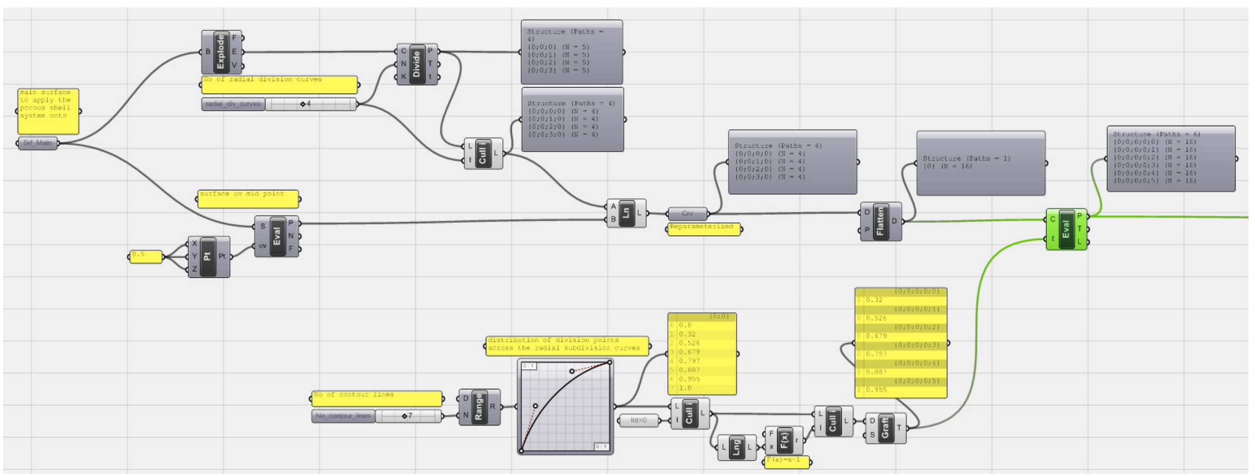


Fig.2.8. Non evenly distributed values between 0 and 1 generated and first and last numbers are also excluded from the list.



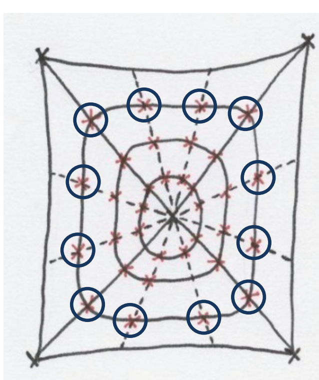


Fig.2.9. Evaluation of subdivision lines. Here by using <Flatten> for subdivision lines and <Graft> for evaluation parameters (t), the way that data (points) branches, reversed so it could be used directly to generate separate closed contour lines.

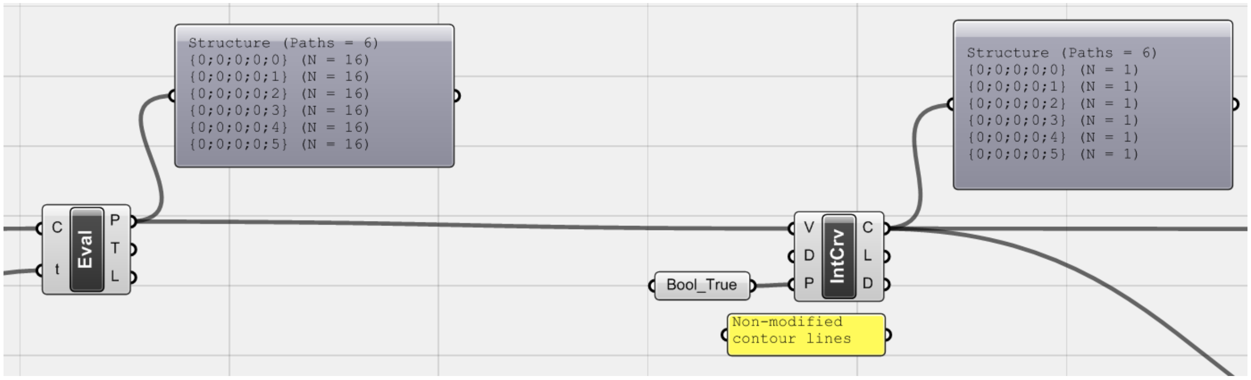
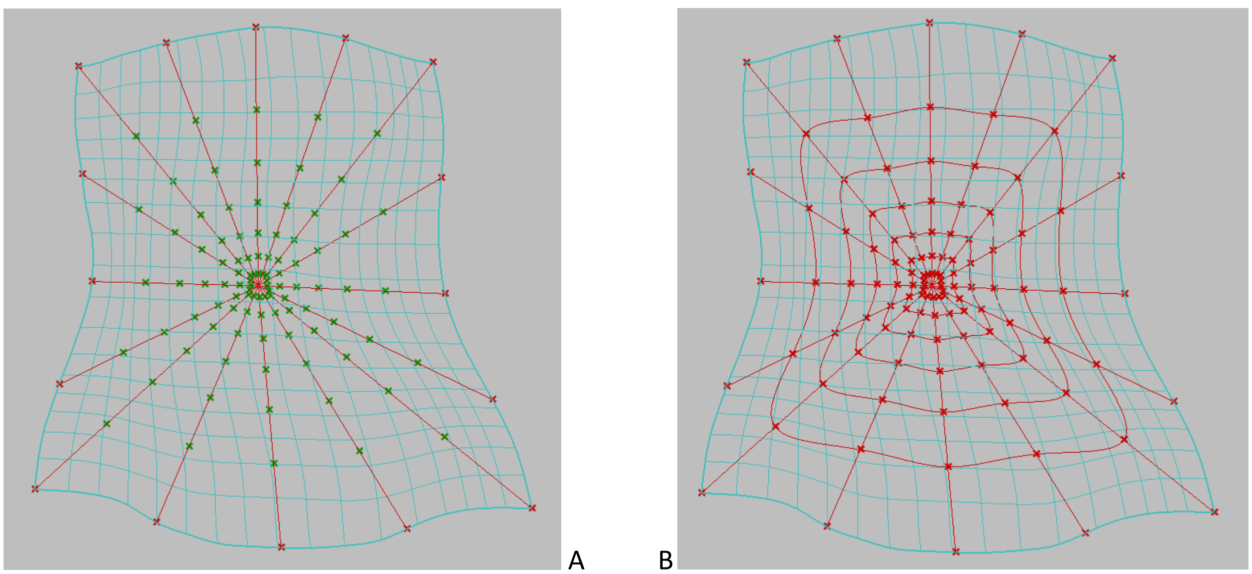


Fig.2.10. Using all <Evaluate>d points as vertices to <interpolate> as contour lines.



A B

Fig.2.11. A. Radial subdivision lines with evaluated points on them. B. Contour lines

Based on the way Contour lines generated, they are generally affected by the geometry of the edge curves of the main surface which shifts the irregularity of the boundary of surface to the central part of it. But Organization of chambers in the organism is a bit different and more regular at the central part. So it seems that a level of modification is needed, in order to achieve more regular geometries in central part of the surface.

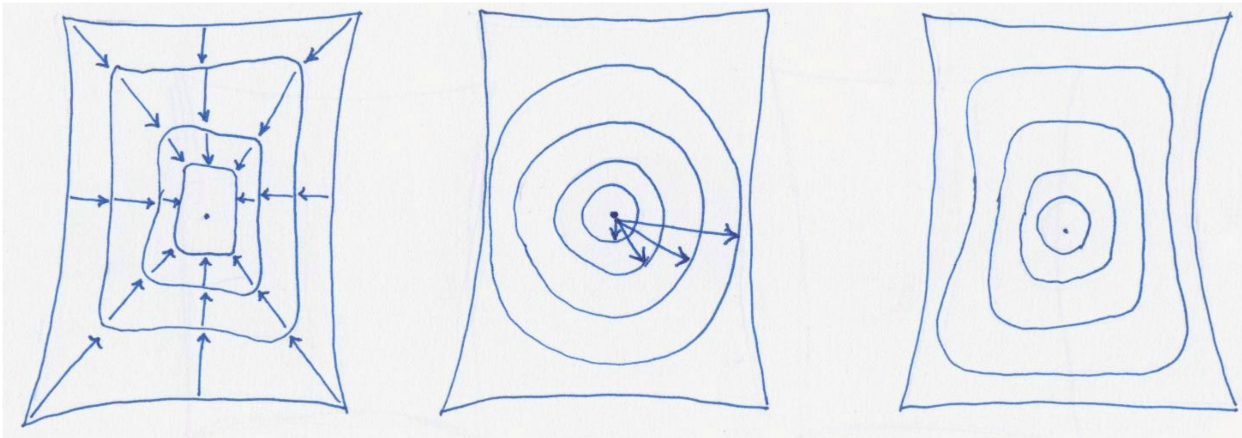


Fig.2.12. In the above diagram, the first one from left shows how curves are generated under the effect of boundary condition of the surface, and the middle one shows if they wanted to be circular around the center point. In the third one, curves started with the circular shape but gradually changed to adapt to the shape of edge curves. This manipulation is the aim of the next part of the algorithm.

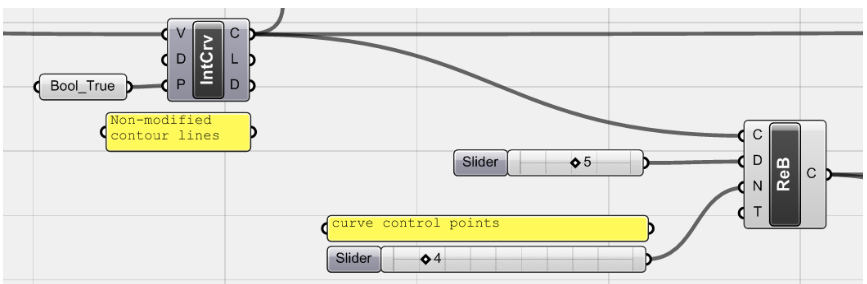


Fig.2.13. In order to achieve smooth curves, here they <rebuilt> with less control points. The result would be circle-like curves.

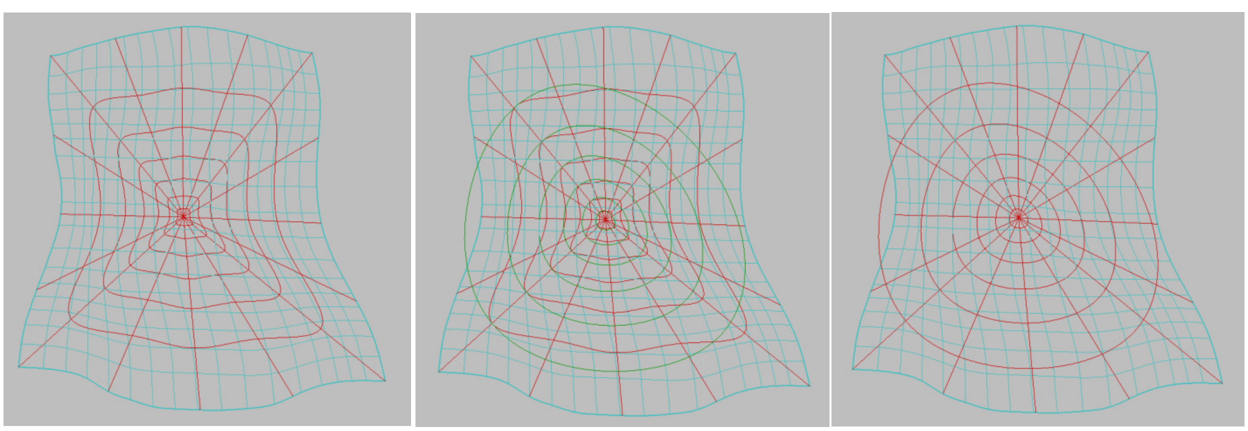


Fig.2.14. looking at figures above, in the first one from left, non-modified contours seem to be affected by surface edges too much. In the third one, rebuild curves seem to be affected by the smoothness of the central-circular shape of curves too much. Looking at the middle figure and comparison between two sets of curves, it seems that a solution between these two sets of curves could be the desired one.

The technique is to use rebuild curves as the final state which non-modified curves tend to metamorphosis towards them with the maximum amount in center and the minimum amount close to the edges.

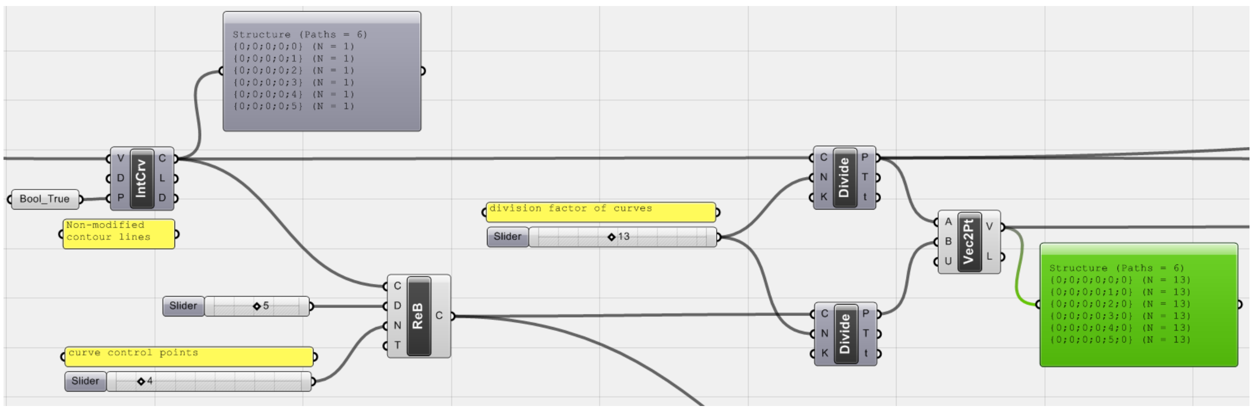


Fig.2.15. In order to literally ‘morph’ contour lines towards rebuild curves, sufficient amount of sample points across both of them are needed. These sample points would be used to pull points of original curve towards the rebuild one. That’s why both curves are <divide>d and series of vectors are generated from sample points of contour lines towards rebuild ones. U part of the <Vec2Pt> is set to True.

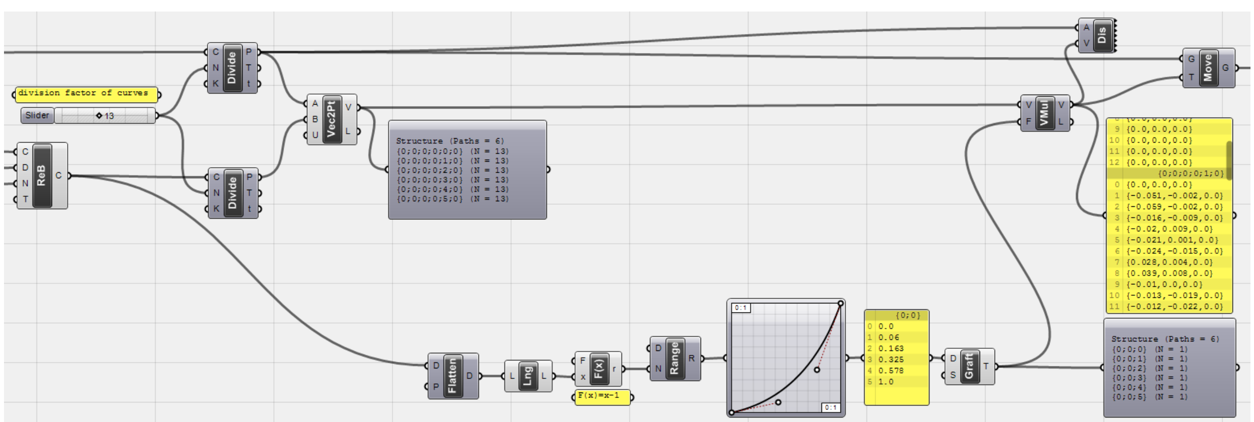


Fig.2.16. While the idea is to move sample points of contour lines, using these vectors, the power of these vectors should be modified. So at the end, vectors close to the center would have maximum power and those close to the edges would be minimum.

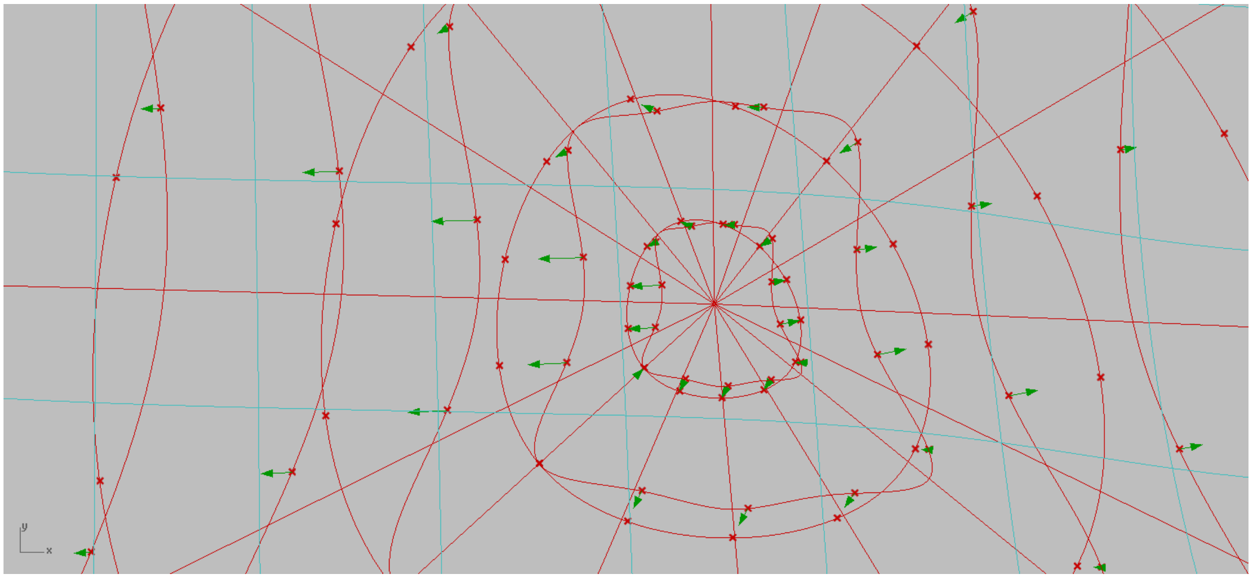


Fig.2.17. All modification vectors are visualized.

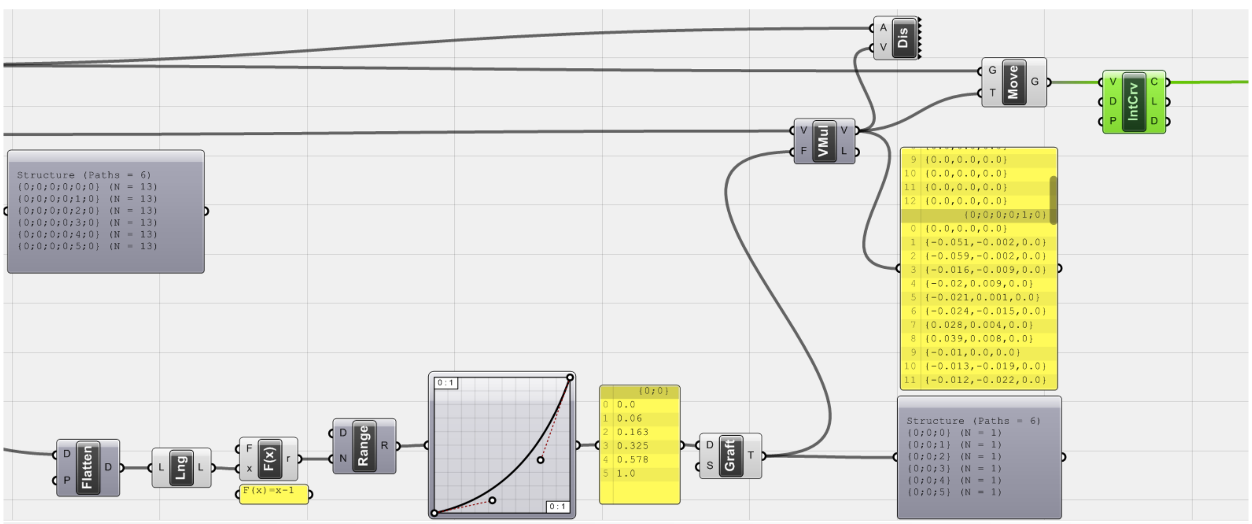


Fig.2.18. If these <move>d sample points <interpolate>d again, the result would be modified contour lines.

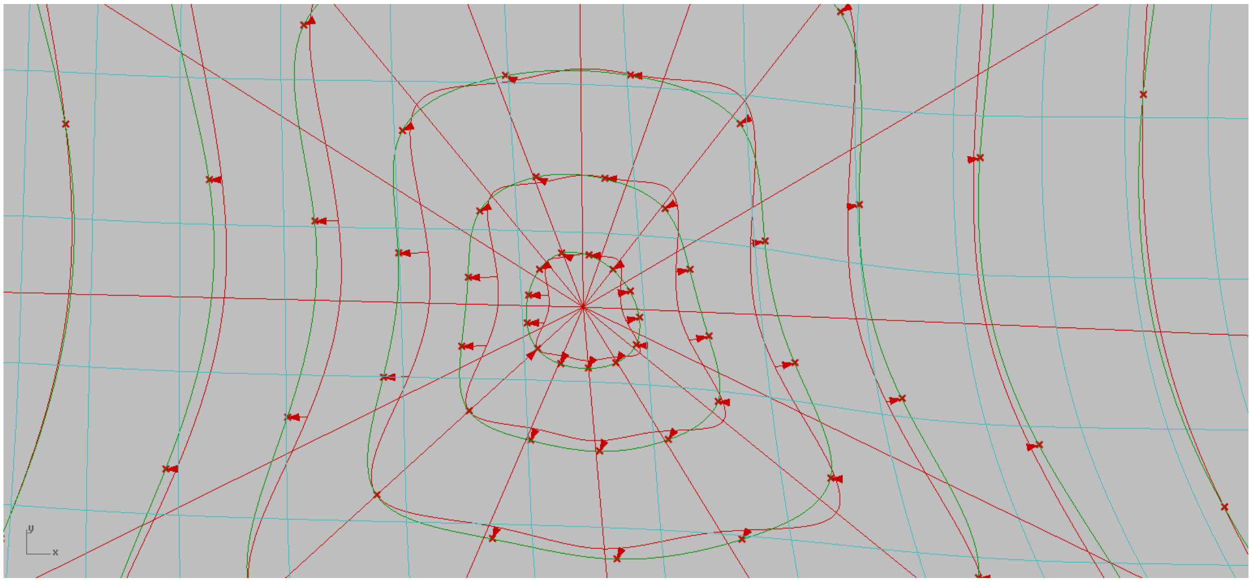


Fig.2.19. Selected curves in green are modified contour lines.

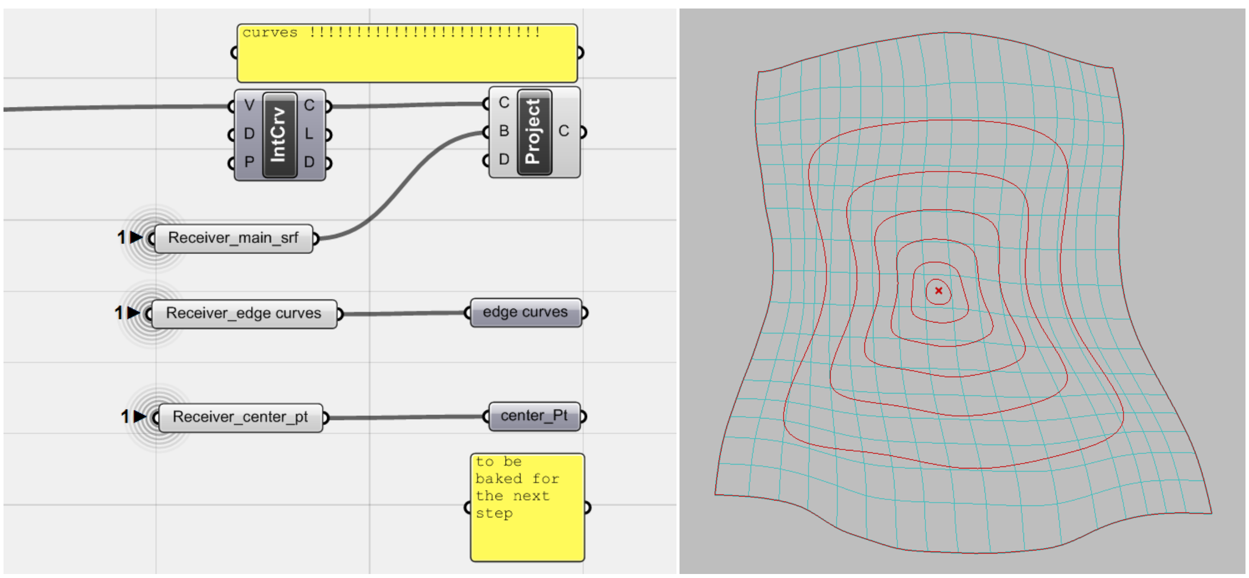


Fig.2.20. Now if these modified contour lines being projected onto the main surface, the result would be the desired contour lines on the main surface. Here edge curves and center point are again brought forward to the end part to be 'Baked' for the next step of the algorithm.

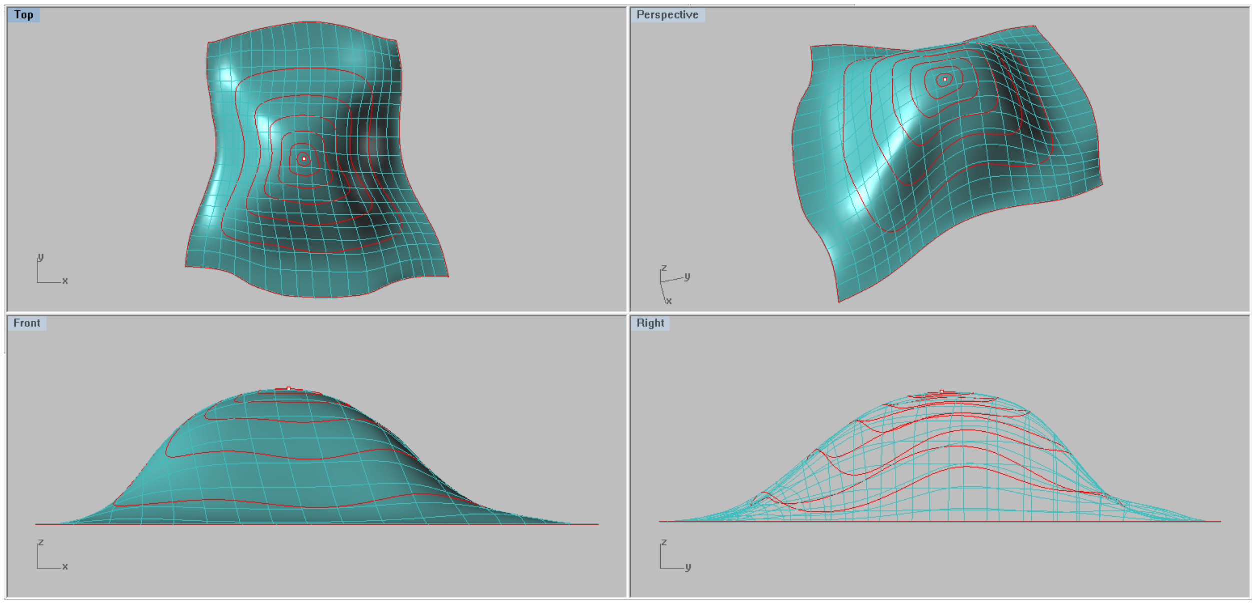
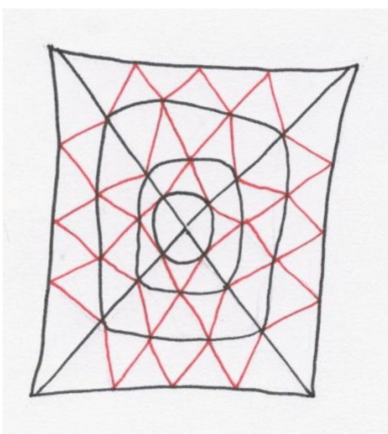


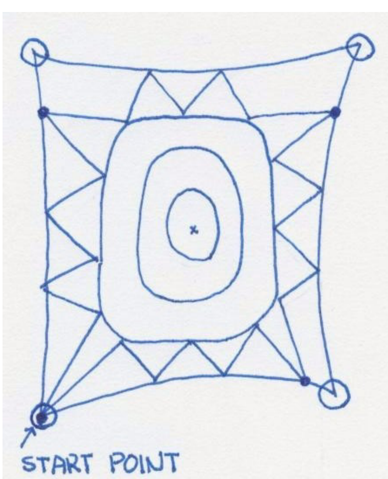
Fig.2.21. Baked contour lines, Edge curves and surface Mid-point.

Algorithm_Phase two_Triangulation

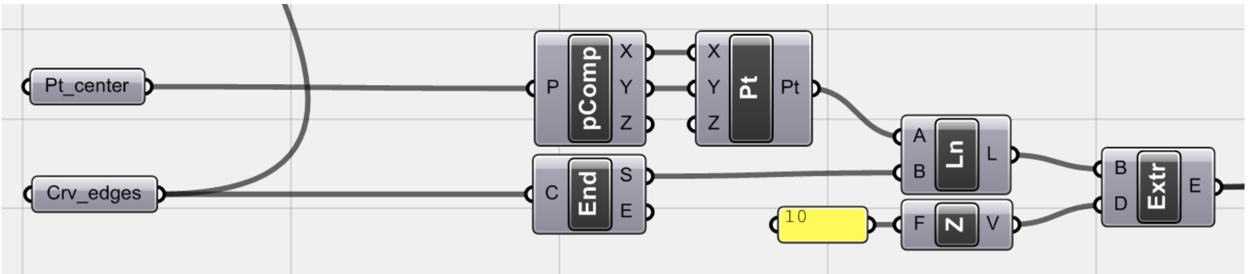


This is the second part of the algorithm with a new file to proceed. This part is designed in order to triangulate the surface in the space between contour lines. The idea is to divide contour lines and get some points, and in the space between any two curves, connect these points, one from each curve with a zig-zag pattern to generate triangles in between. But the strategy to design the algorithm should be clarified first.

It is possible to divide edge and contour curves and in each pair of adjacent curves, connect every other point to get a triangulated net. But let's look at the diagram below to check what might happen in this technique:



If edge curves assumed as one closed curve and then this closed curve and contour curves are divided and connected, it would be impossible to control the relation of triangles with the corners of the surface. So when triangulated, position of triangles would be irrelevant to the boundary conditions and final position of chambers would be unorganized in the surface. Here in order to avoid this situation, division of points should be made upon contours which are splitted by diagonal subdivision lines of surface (which were visualized in the next figure).



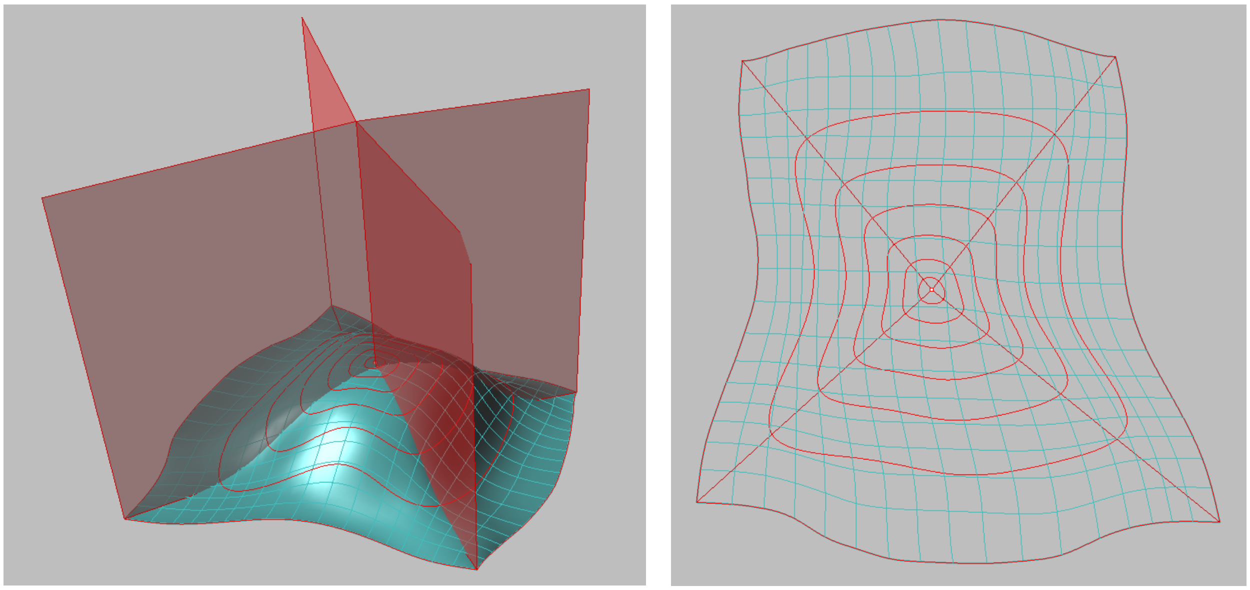


Fig.2.22. Using Four extruded surfaces, contour lines splitted into pieces.

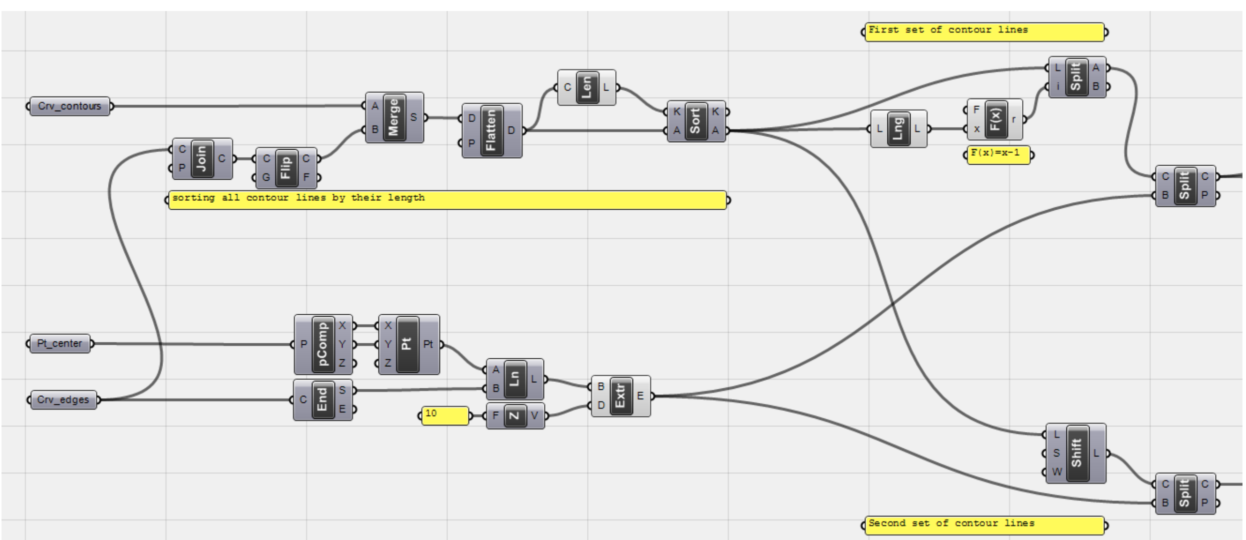


Fig.2.23. In order to generate a zig-zag pattern of connectivity, two sets of points are needed. That's why here two lists of curves are prepared to split and divide for the next steps. These two lists are: 1.list of sorted contour lines without the last one. 2. list of sorted contour lines without the First one.

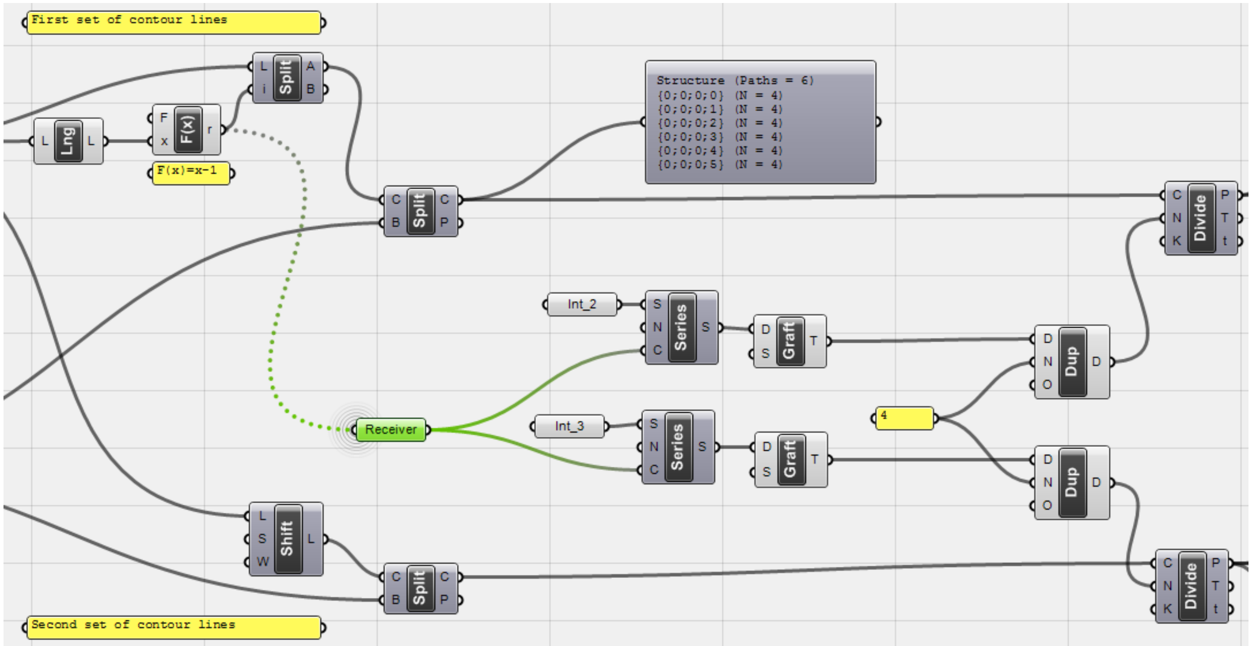


Fig.2.24. The next step is to prepare number of divisions for splitted curves. Here it should be noticed that number of divisions increase one by one, so when the first curve divides into two parts the next one divides into three, the next four, … .

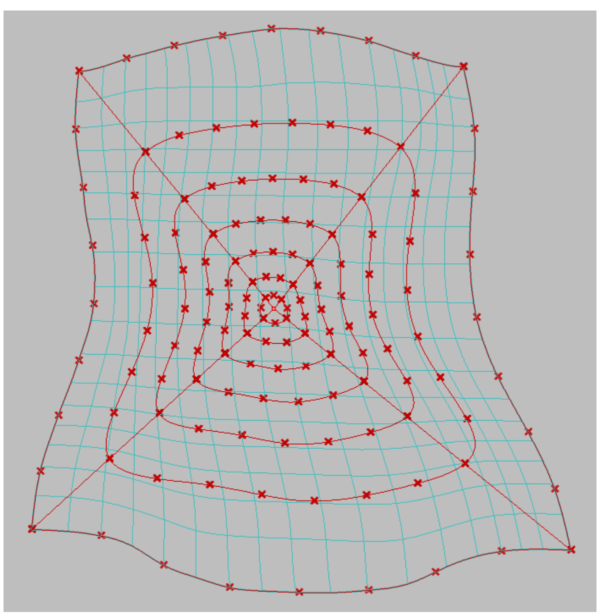


Fig.2.25. Checking the resulted geometry, it is clear now that curves are divided in an incremental pattern and there is always one division point in each corner of the surface and its associate division line up to the center of the surface.

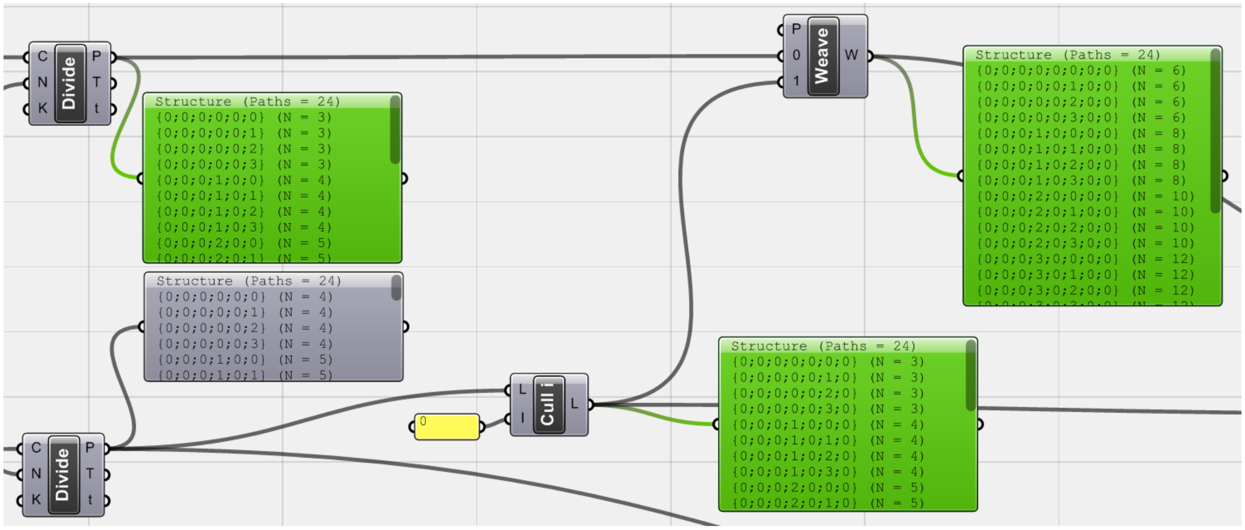
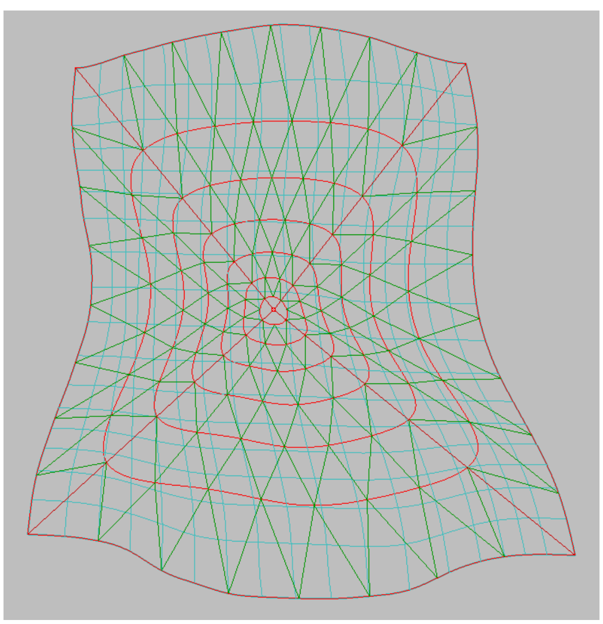


Fig.2.26. While division points are all available, it is now possible to generate the list of points to generate zig-zag lines. Just a small bit is that again here last part of each zig-zag line would overlap and lay on the first part of the adjacent zig-zag line and that's why here by using a <cull i> component, all first points of second lists are omitted. The result is a <weave> component with data prepared for line generation.



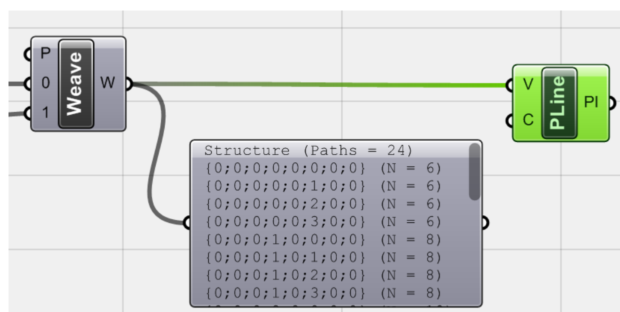


Fig.2.27. Woven data list used to feed a <Pline> component to generate polylines as triangle edges.

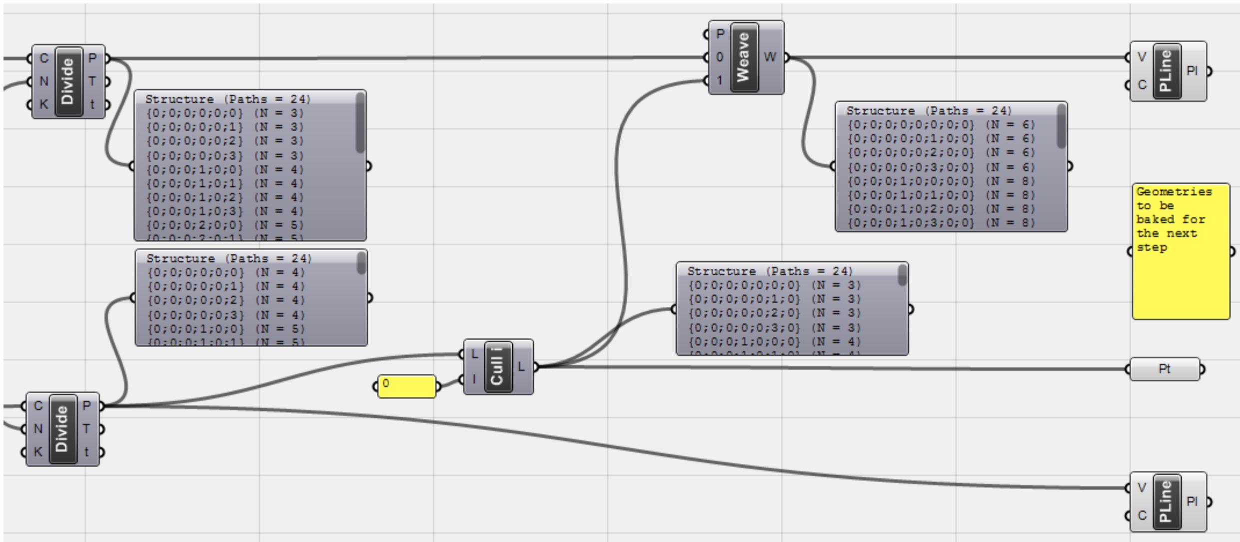


Fig.2.28. Again <Pline> as triangle edges, <Pt> as nodes of net and a new <Pline> which connects sides of triangles across contour lines, are brought forward to be ‘Baked’ for the next stage.

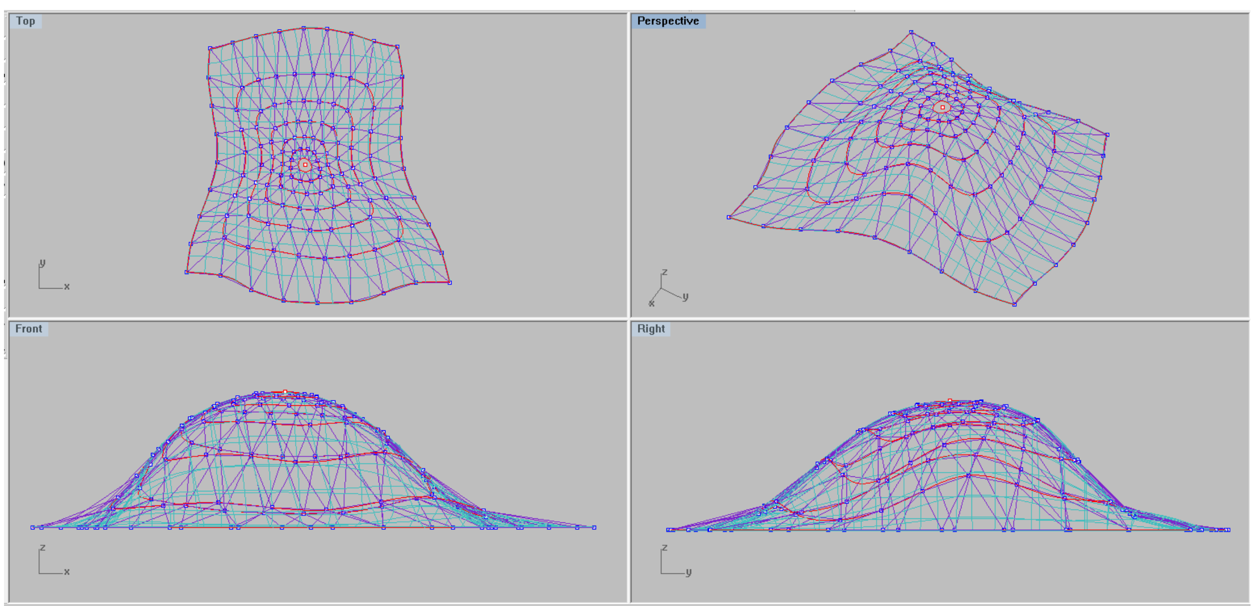
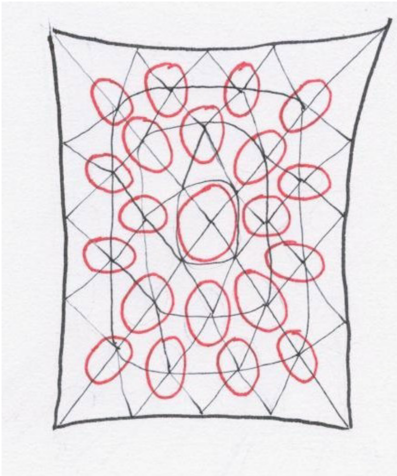


Fig.2.29. Final Triangulated surface with baked net lines and nodes.

Algorithm_Phase Three_Chamber Generation



In terms of form-finding, this is the last step of the progress. Here a closed curve as a chamber should be generated on any node of the network of triangulation.

The strategy is to check each node and list all lines which are connected to that node. Then a closed curve would be generated through the midpoint of these selected lines.

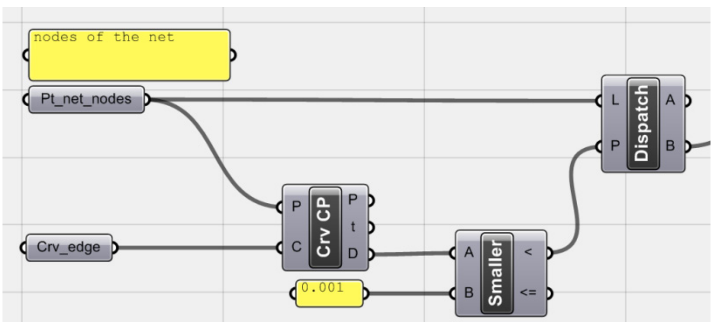


Fig.2.30. Polycurves are exploded in Rhino and introduced to the canvas by <Crv_triangles>. In the first step marginal nodes (those which lay on the edge curves) should be removed from the list. That’s because in this specific case it is not aimed to generate any chamber at the edges.

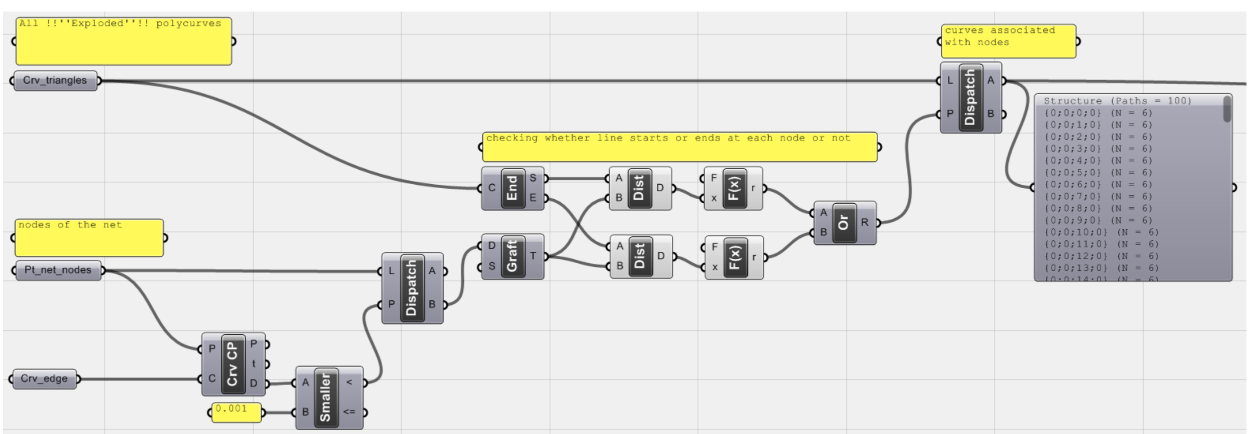


Fig.2.31. In this part of the algorithm all lines of the net which associate with each node should sort in one separate data branch. So nodes are <graft>ed and the <Distance> of each node is calculated from **Start** and **End** point of net lines to check whether the start/end point is close to (almost lay on) the node or not.

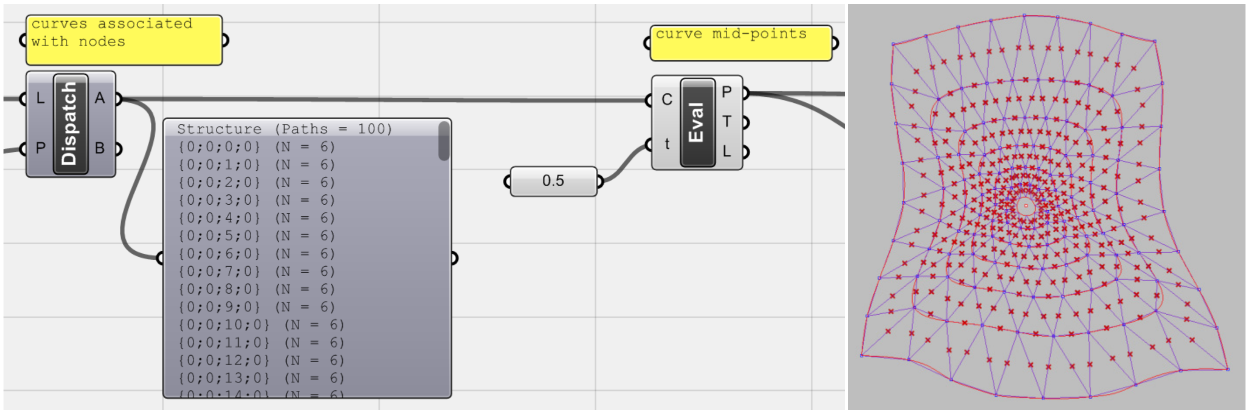


Fig.2.32. Sorted net lines are being <Evaluate>d in their midpoint at t= 0.5 (it is possible to use other parameters to evaluate curves but then all curves starting from each node should have the same direction. This would happen later on this book).

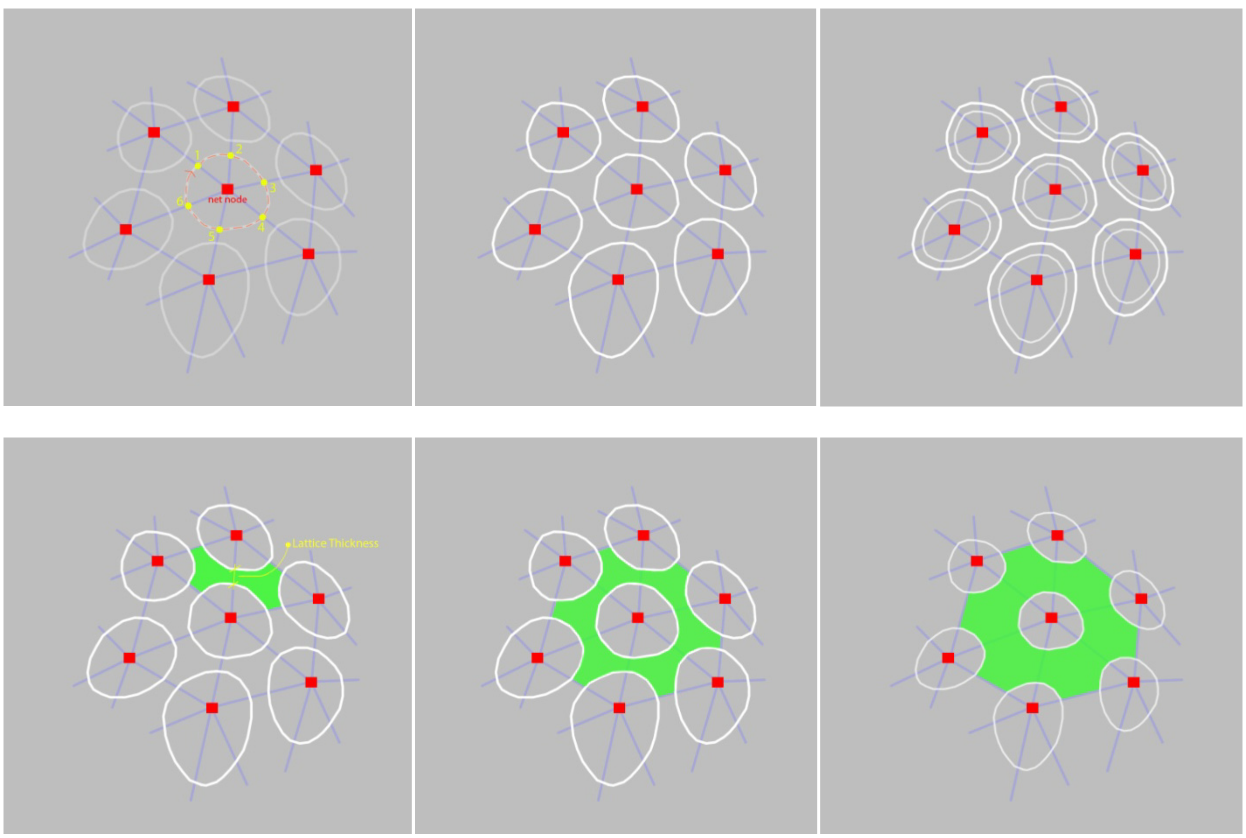


Fig.2.33. The above diagrams show how chambers are going to be generated and how the final shell's porosity could be changed according to the chambers' sizes. As lines evaluated in their mid point at the previous step, this would result in almost maximum size of chambers and thinnest lattice size for shell.

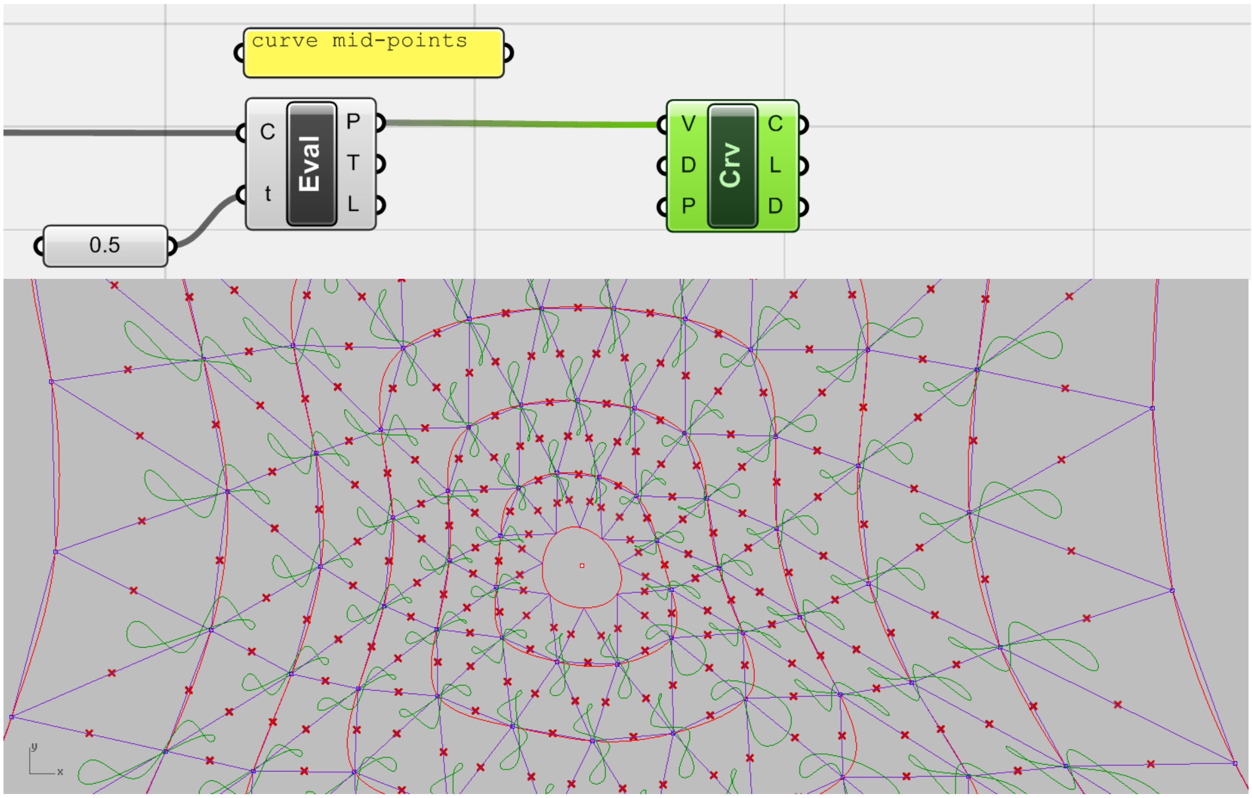


Fig.2.34. Now if all evaluated points feed a <Crv> component with **True** value for its ‘Periodic Curve’ option, the resultant geometry would be something like this figure. It is now clear that all points are listed in groups of six around each node but they are not sorted.

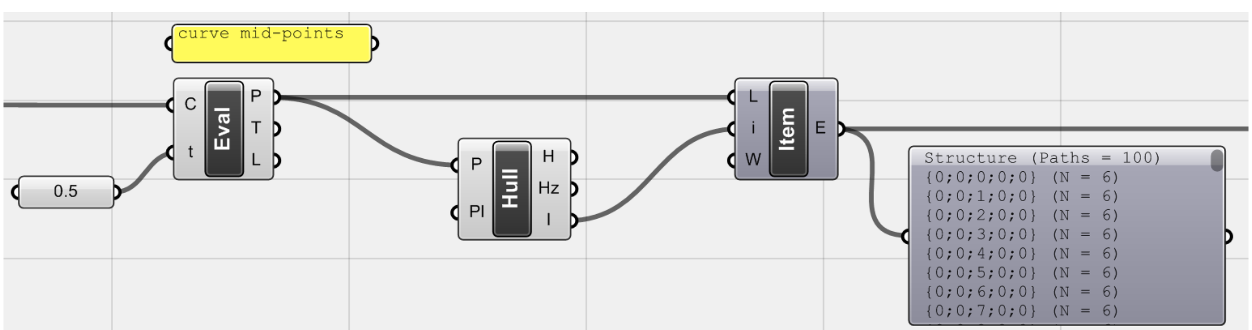


Fig.2.35. In order to sort points in their lists, the best possible option is to use Convex Hull of any set of points. The Convex <Hull> component sorts points and provides ‘Index’ numbers of them as well, so points could be selected in this fashion.

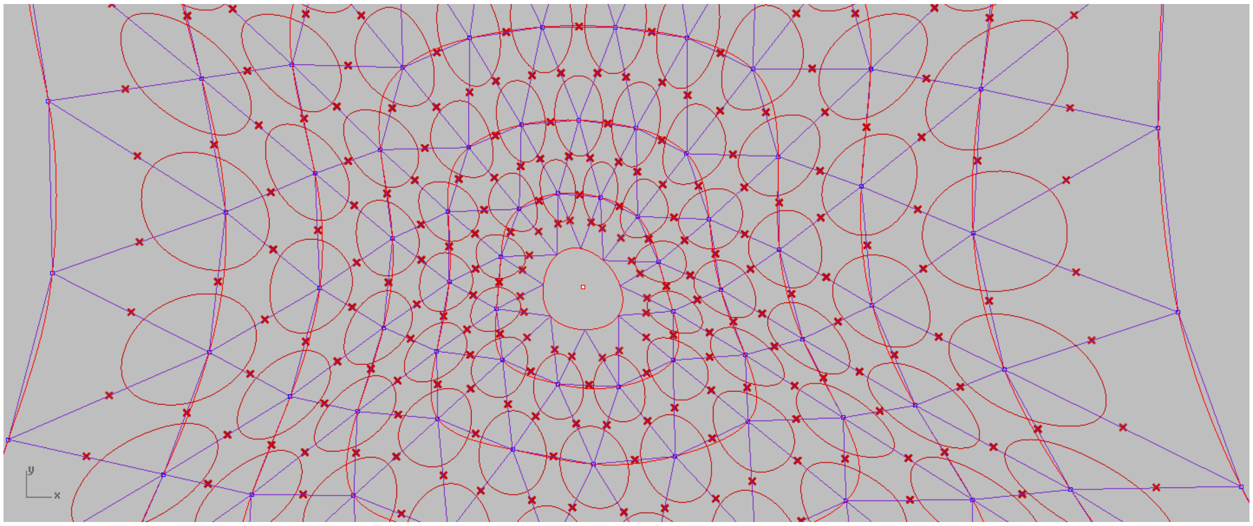


Fig.2.36. Feeding <Crv> with sorted points. The ‘Periodic curve’ option is set to **True**.

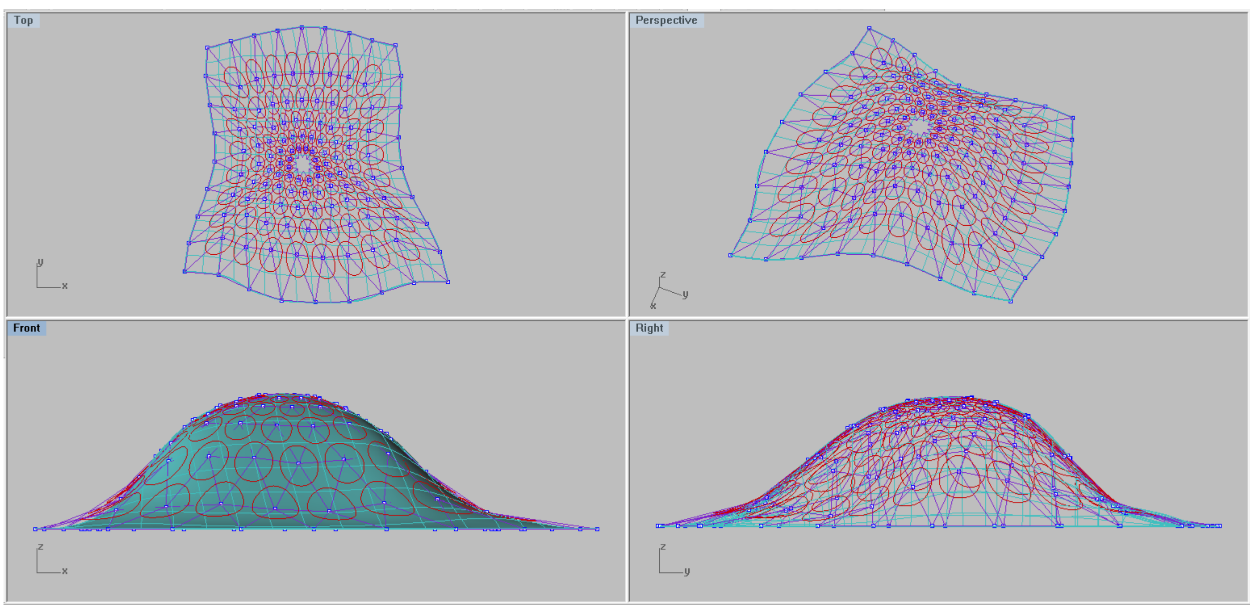


Fig.2.37. The main surface with its chambers created on it in a radial organization, with desired size differentiation form center to edges.

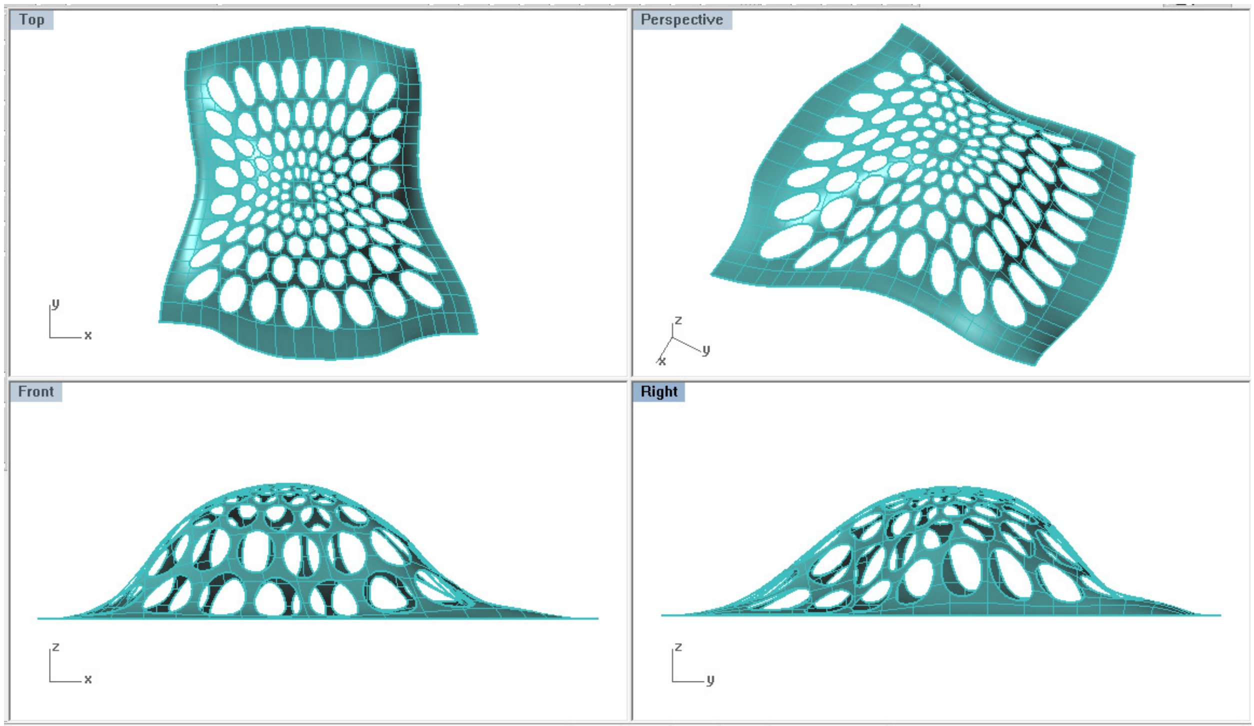
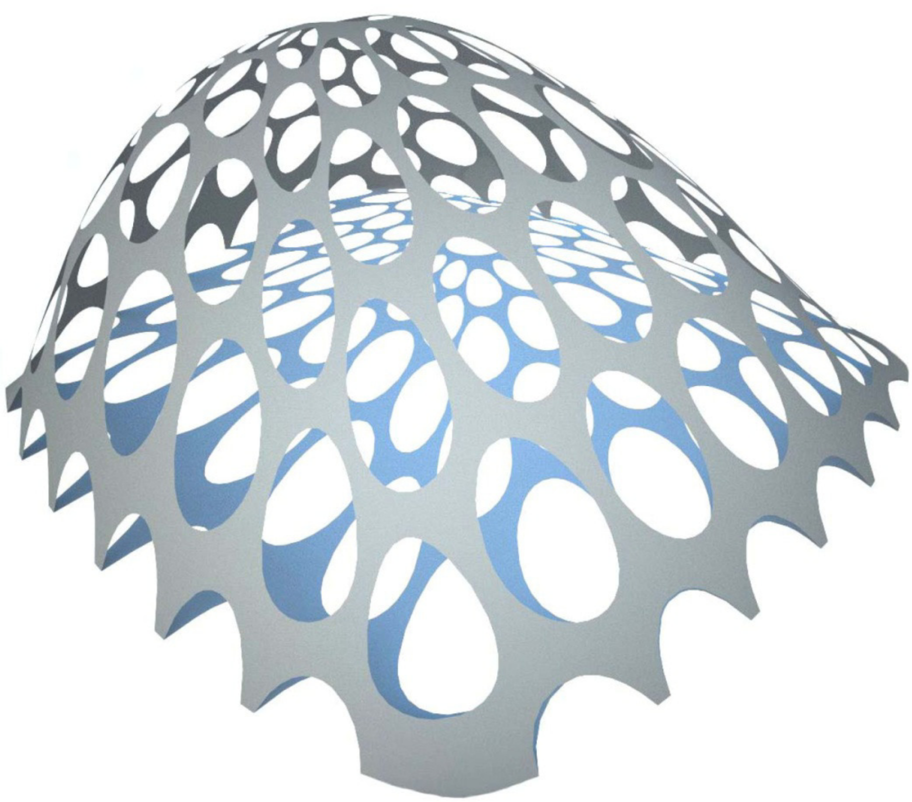


Fig.2.38. The final step. All chamber curves are 'Baked' and the main surface splitted by these curves in Rhino. The main surface is also splitted by the smallest contour line in its center. Surface pieces inside chambers are deleted. The result is a lattice, as a form found geometry of the '**Porous Shell**'.



3_Fabrication

3_1_Fabrication Techniques

The Algorithm that developed so far is a general form-finding algorithm with several variables. In order to set these variables it is important to subject the form finding process to a material outcome. In such case it is possible to set different values like scale, size of chambers, thickness of lattice. There are different possible avenues in front of this research to materialize the final outcome. Here three possible methods are discussed.

1. Porous Cast

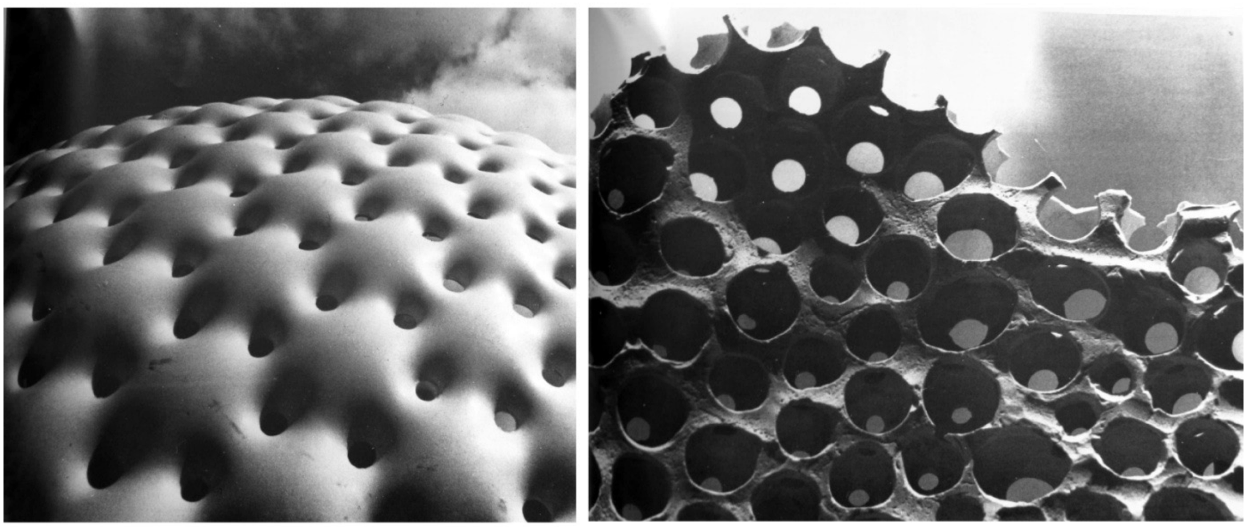


Fig.3.1. Frei Otto's experiments on porous shell systems.

After Frei Otto and Helmcke developed the concept of deposition of siliceous materials over a cluster of pneumatics as form generation process of hard shells, they worked on the 'Porous Cast' as a form-finding technique to materialize the form of these shells in experiments. This method was quite the same as the natural form generation process.

2. Component based - Geodesic Dome

Geodesic Domes invented by Richard Buckminster Fuller in the 1940 and the aim was to cover large spaces. Fuller developed different techniques and geometrical configurations for geodesic domes. Since making a complete sphere was geometrically complex, he tried to find methods in which triangulation of a sphere could be organized in a way that all parts become the same size. One of his projects was Fly Eye dome which he developed for affordable housing. It was a modular based system with a porous shell but covered with transparent pieces.

Fig.3.2. Buckminster Fuller's 'Fly Eye' dome.

3. Sheet Material /Cutting based Fabrication

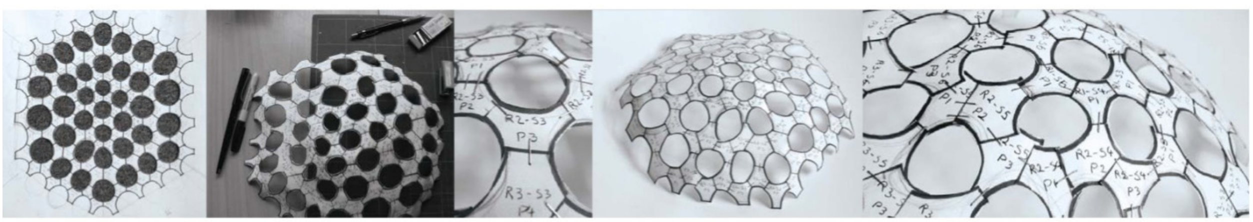


Fig.3.3. Flat-Piece Modulated geometry for sheet material fabrication (Mock up).

The third option of the materialization is to use current technological equipment and machineries for fabrication, like Laser-Cutter or CNC-Machines while using flat sheet materials. In this technique, geometry of the porous shell should be splitted into pieces to be cut and then assembled together, to form the desired geometry.

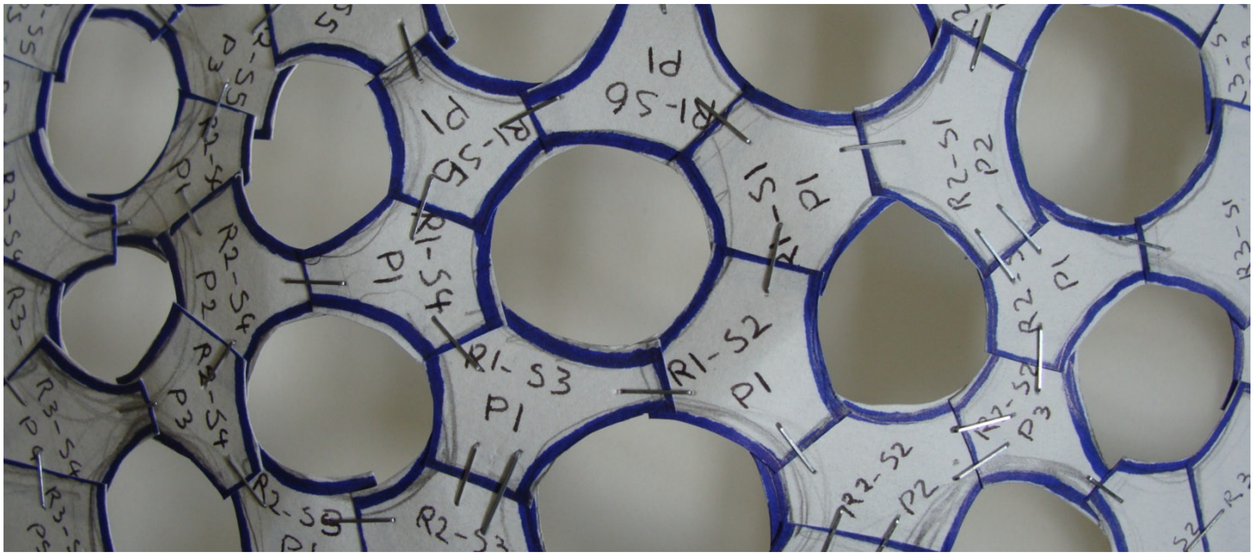
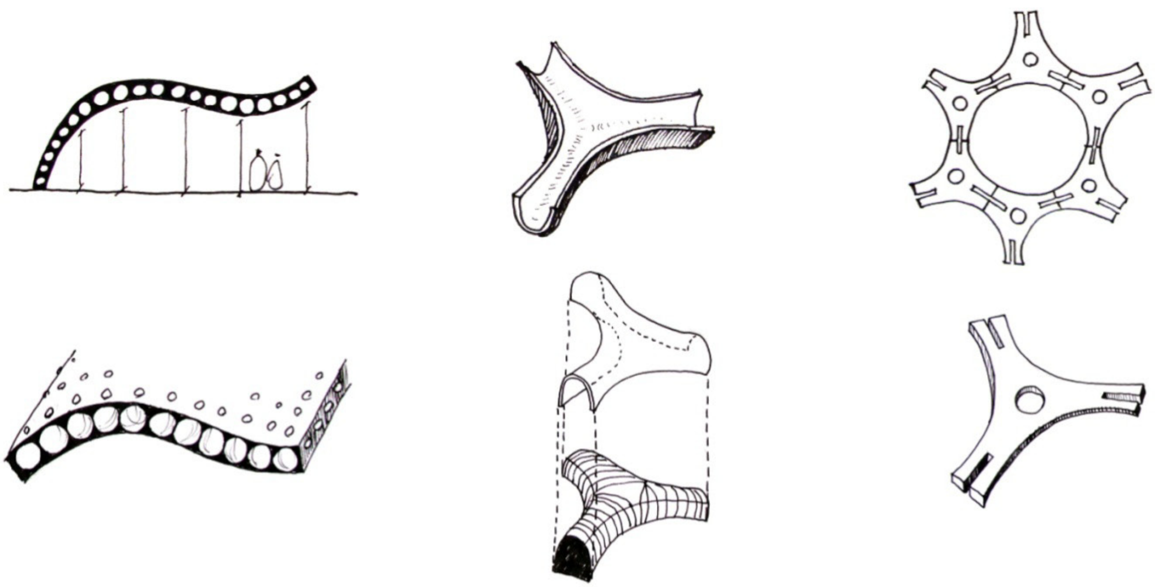


Fig.3.4. Mock up



_Porous Cast:

- The cast material could get any shape that mould has so the flexibility of the mould determines the flexibility of the final outcome of the project

- The porosity of the shell depends on the bubble cluster which is quite flexible

- The chamber sizes depends on the bubble sizes

- Since the resultant geometry is continuous, the pouring and casting should be done in one phase, so fabrication in big scale is limited to the moulding and casting technique

_Geodesic Dome / Bendeding Technique:

- The geometry is made by pieces not a one piece project

- The geometry and form, sizes of chambers, thickness, etc. depends on the design and could be varied accordingly

- The overall geometry is spherical and differentiation is almost imposible, because:

- Bending of metal pieces by itself needs machines or moulds and jigs or they might be generated in different parts which being welded together, but all together, fabrication of pieces is a difficult process

- There is a possibility to cap the geometry and fill it with a casting material

_Sheet-Cut Fabrication:

- The geometry is made by modules and pieces but not a one piece project

- Since it made by pieces, it has joints where these pieces meet. These joints could be structurally weak points of the system

- The geometry and form, sizes of chambers, thickness, etc. depends on the design and could be varied accordingly

- Since there is not any moulding, the geometry could be very flexible

- Cutting sheets has its own restriction like material, machinery, bed-size, etc.

Fig.3.5. Comparisons and Evaluations

_Selection of the Technique

Since the idea is to use algorithmic tools and digital machineries, the third technique suits the project in different ways and sounds more viable for this stage.

Fig.3.6. One piece of splitted porous shell as a mock up, cut out of flat sheet.

3_2_Fabrication Algorithm

_Fabrication necessities

1. The geometry should be splitted into pieces to project onto a flat sheet to cut.
2. A system of joints should be developed to attach and assemble pieces.
3. All pieces should be labeled in order to address them in assembly phase.
4. A proper material should be selected for fabrication. Properties of this material would affect the algorithm in certain ways.
5. Assembly instructions and plans might be needed and should be provided by the algorithm beforehand.
6. Any post-production action, if affects the final piece or assembly, should be considered beforehand.
7. Preparation for fabrication usually needs data management, file exchange between softwares, technical manipulation of drawings to match the requirements of machineries and so on. It contains manual jobs and non-parametric! computation as well.

Algorithm_Phase Four_Fabrication and Assembly

_Fabrication Preparation Process

Based on the sheet material fabrication the rest of the algorithm is designed.

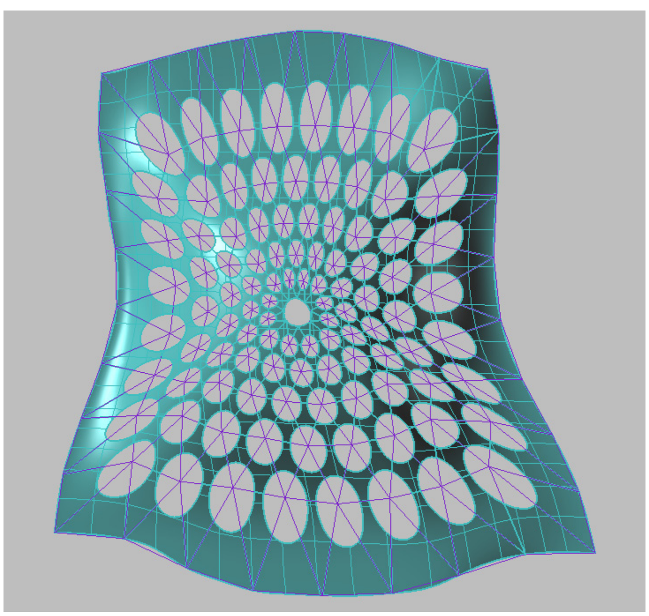
In order to prepare the porous shell/surface to fabricate by flat sheet material with the bridle joints to assemble, there are five main steps that should be taken:

1- The porous surface should split into pieces (components) which are able to be cut through sheet material
2- Joints should be generated
3- Sluts for bridle joints should be cut from all pieces and joints
4- Pieces and joints should be labeled and nested into sheets.
5- All prepared files should be transferred for Manufacture and Assembly

Fabrication_Step_1_Splitting the surface into pieces

In order to fabricate the porous shell, the first step is to split the form-found geometry into pieces. Generally for fabrication purposes for any project, there are multiple steps, some of them might be in Grasshopper, some of them in Rhino, and even in other softwares.

To generate all pieces, the porous surface of the previous step has been splitted in Rhino, using '**Triangulation**' net lines as cutting objects. As visualized in following figures, the surface divided into pieces, each one is inside one triangle of the triangulation net.



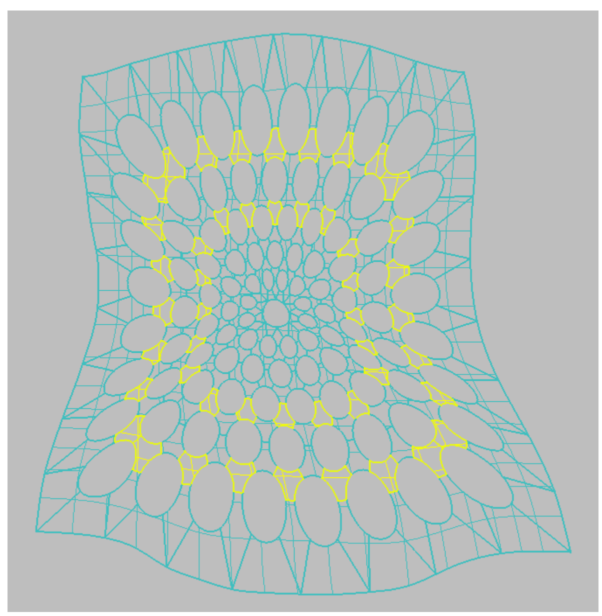


Fig.3.7. Using all lines of triangulation (exclude edge lines) as 'cutting objects', the porous shell (surface) splitted into pieces in Rhino. As clear in the second image, now pieces can be selected individually.

Fabrication_Step_2_Joint Generation

2-1- Sluts/Joints Baselines

Since the surface of the porous shell divided into pieces, it needs joints to connect these pieces together. These joints are made from the same sheet material, in the shape of small cylinders with two sluts on them in the direction of a diameter, that any of the surface pieces should slide into these sluts gently and by three connections for each piece, they would remain fix in their position. So the next step would be joint generation for all pieces in their division line, adjacent to the next piece.

In the first stage of joint generation, position of these joints on the surface pieces should be extracted from the model. These positions are on the connectivity lines of surface-pieces mid-points. So if all adjacent surface-pieces mid-points connect together, the baselines for joints would be there. Lets look at Grasshopper definition:

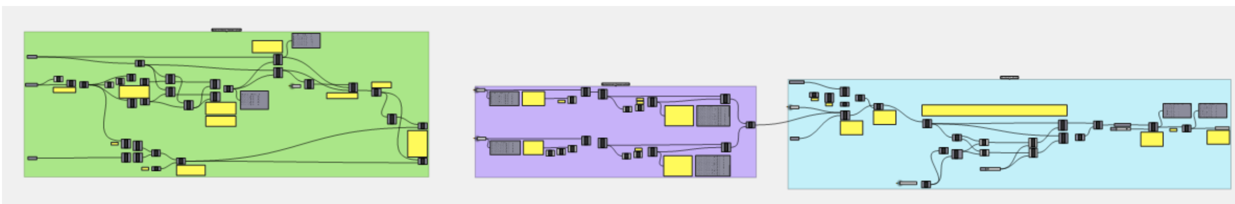


Fig.3.8. Overall view of the Grasshopper Definition: “4_Slut_Baseline_Generation”.

Working with component based systems, it might encompass hundreds, sometimes thousands of relatively small pieces and it is crucial for any fabrication algorithm to sort them efficiently, usually based on the assembly logic. So for instance if pieces should assemble row by row on top of each other, then the sorting mechanism should be in the same fashion.

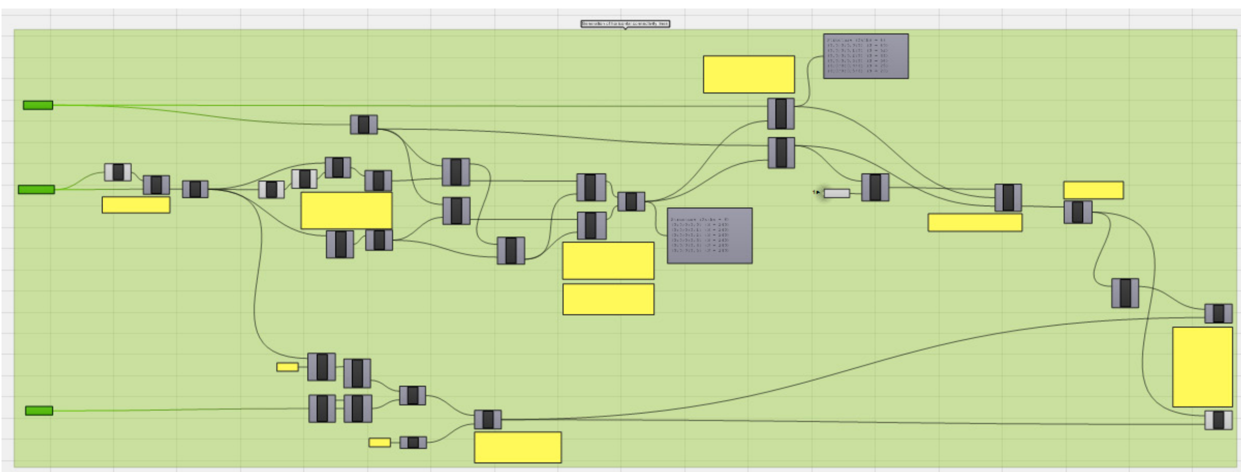


Fig.3.9. The first part of the algorithm sorts all pieces in separate rows and then connects their centers to each other to make the first level of connectivity lines. There are three inputs for this part <srf_pieces>, <crv_contours>, and <pt_srf_midpoints>, all baked geometries from previous parts.

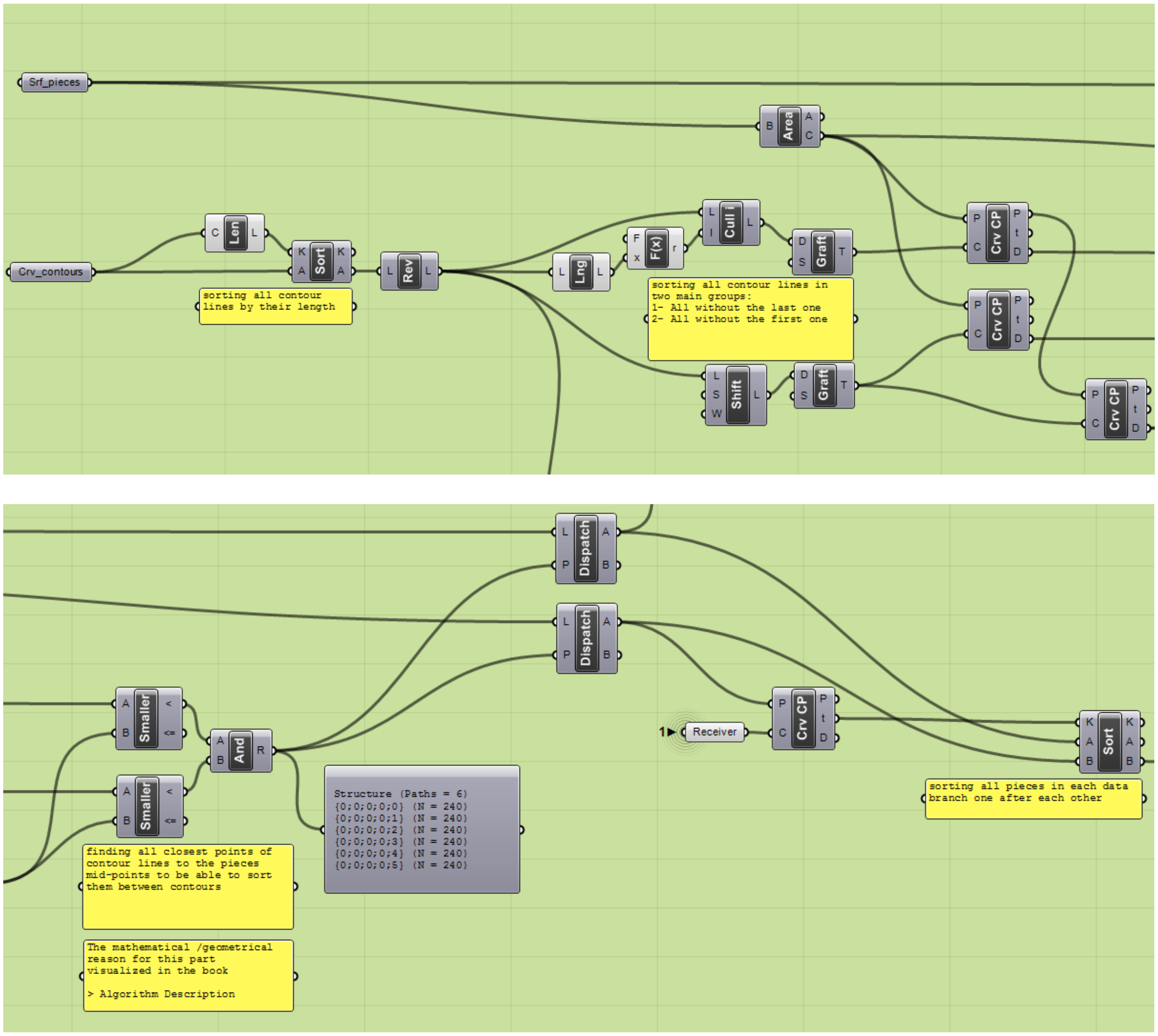
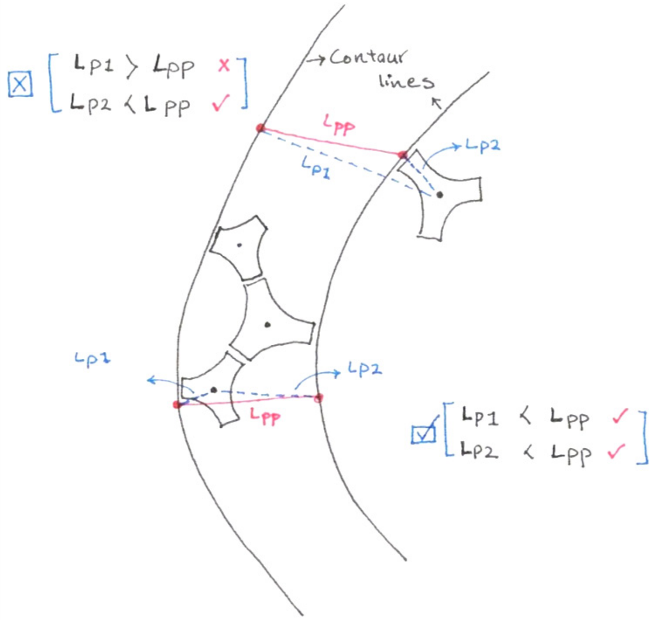


Fig.3.10. The main challenge in this part is sorting all pieces which is being described in more detail. In the very first step, all components are introduced to canvas by <Srf_pieces> and all contour lines are also, by <Crv_contours>. Contour lines are sorted by their <length> and their list reversed to have a list of all contour lines from longest one to the shortest.

In order to sort all pieces, first they should sort in rows from bottom to top of the shell. That's because assembly of them would be in the same order, row by row, from bottom to top. In this case, the helping geometry to sort these pieces is contour lines. Pieces between any two adjacent contour line should be sorted as one row.

The next step would be sorting pieces in each row one by one in a circular fashion. So at the end there would be separated groups of pieces sorted in rows and in each row, sorted pieces by their connectivity.

The way that these pieces are sorted is as follows:



A triangle has been used to check whether the piece is inside the area between two adjacent contour lines or not. First, the closest point of the piece's mid-point has been found for both contour lines. So there would be two lines that connect piece's mid-point to these new founded points (Lp1 and Lp2), and there is also another line that connects these two points to each other (Lpp). To check if the piece is inside the area between two contour lines, this third line (Lpp) should be always longer than other two (Lp1 and Lp2), otherwise the piece would be outside and should be omitted from the list.

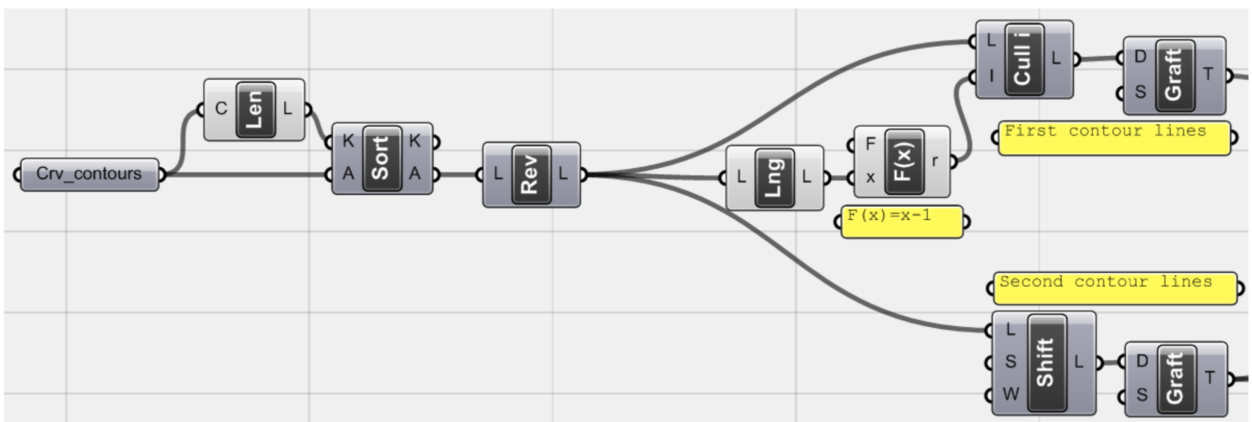


Fig.3.11. In the first step, pieces should be sorted between two adjacent contours, and here two lists of contour lines needed. The first one is a list of contour lines (without the last one), and the next one would be shifted list (contour lines without the first one).

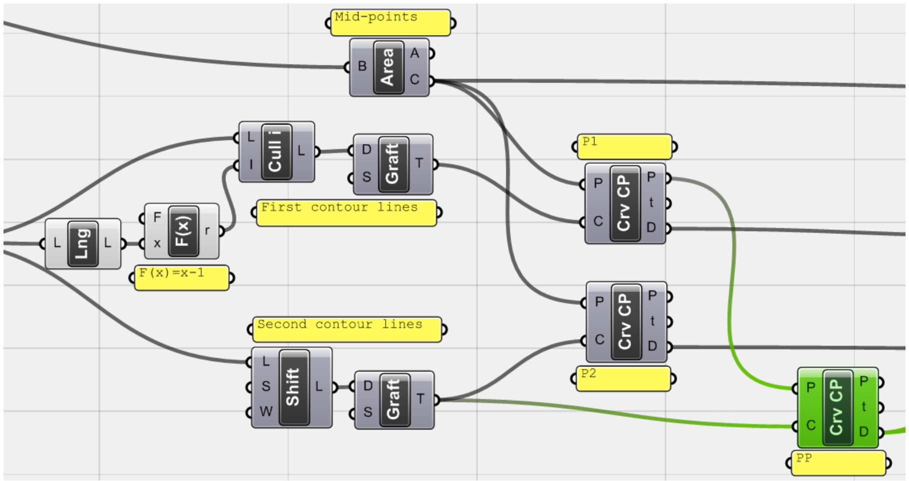


Fig.3.12. Now all surface mid-points are calculated by <area>, and the closest point to both lists of contour lines are also founded (P1 and P2). The same component gives the distance between any mid-point and its associate closest point on curve as well (Lp1 and Lp2).

Closest points of the first points list are also calculated to get the distance between two contour lines at that position.

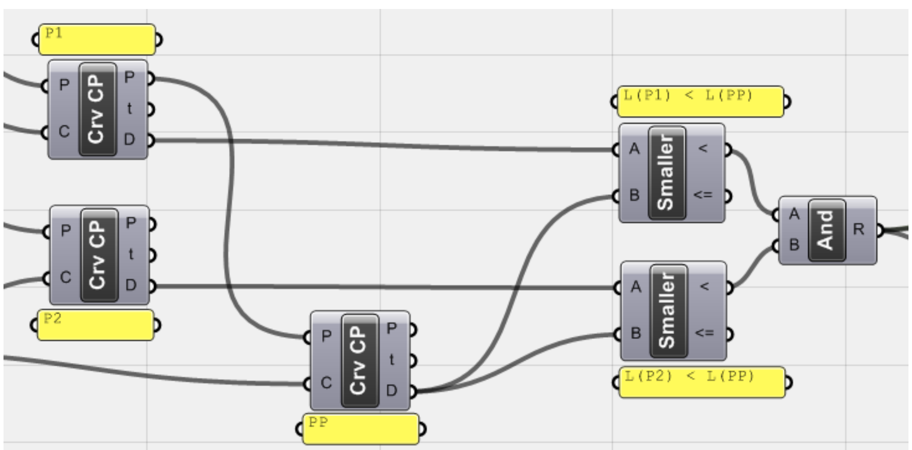


Fig.3.13. Since the <Crv CP> gives the distance as well, here the length of Lp1 and Lp2 are used to make comparisons with the distance between contour lines Lpp. By a gate <And>, it is now possible to find those distances which are both smaller than the distance of contour lines (Lp1<Lpp and Lp2<Lpp).

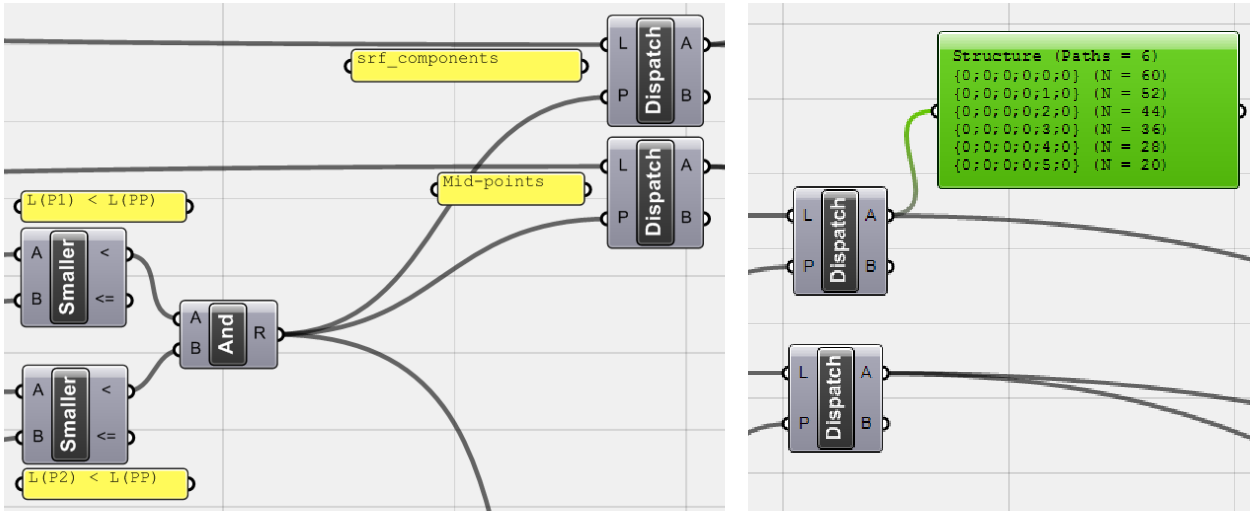


Fig.3.14. Providing Boolean data lists, now it is possible to <dispatch> both lists of components and their mid-points. Components and mid-points are sorted row by row in different data branches.

Now all pieces are sorted between contour lines, it is time to sort them in each row in a circular fashion. Although it would be better to use polar coordinates, here the idea is to use contour lines again to sort pieces one after each.

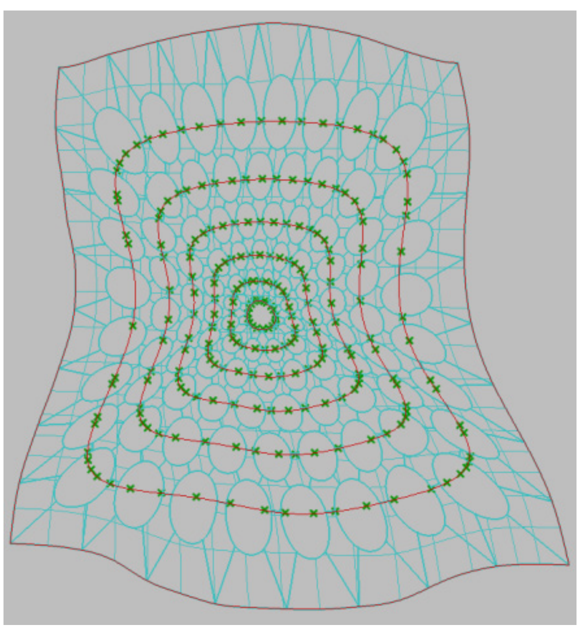


Fig.3.15. All sorted mid-points are used to find their closest points on <Receiver_shifted_contours>. Since the components gives the parameter (t) of that closest point, this parameter as a numerical value used to sort all components and mid-points across contour lines.

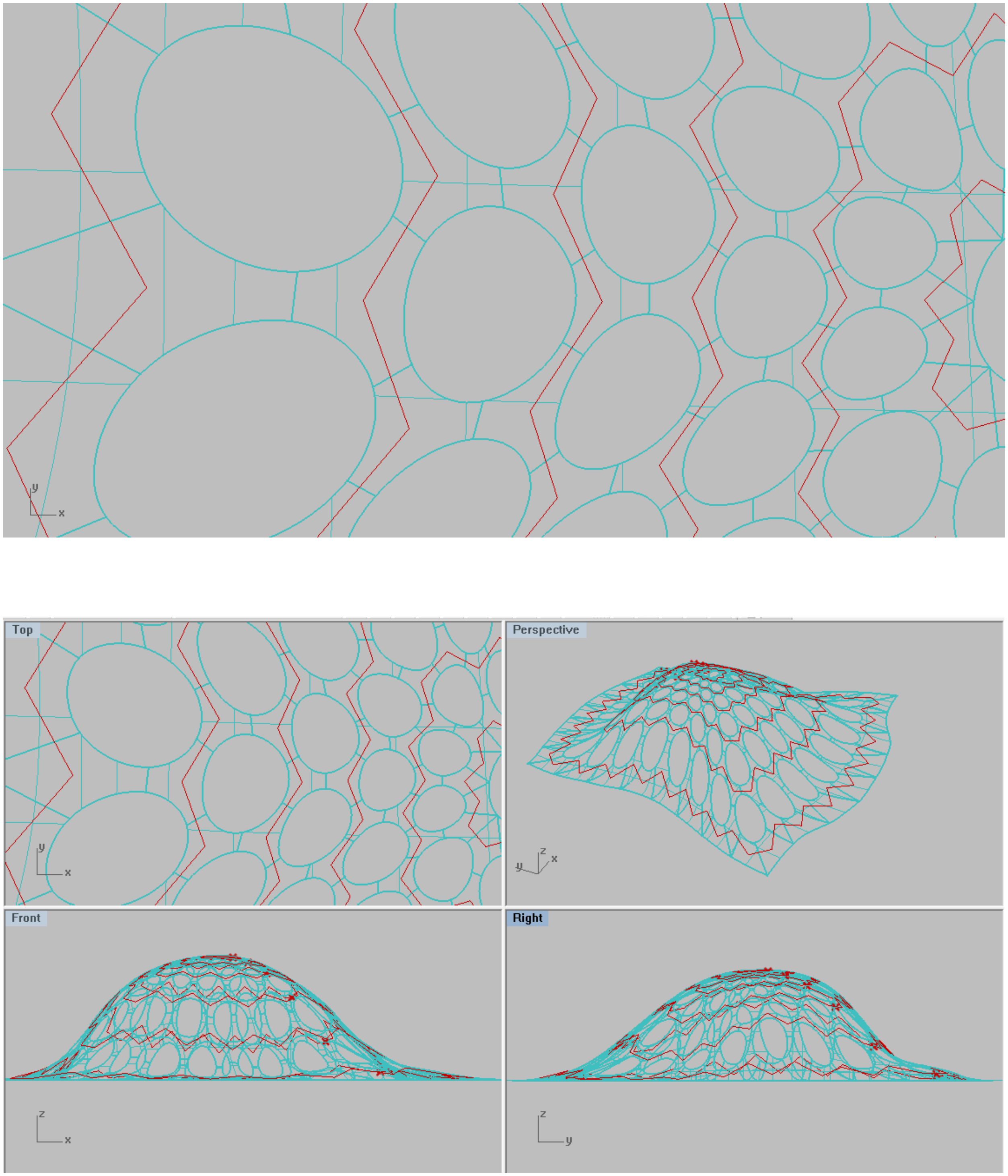


Fig.3.16. First set of connectivity lines in horizontal direction.

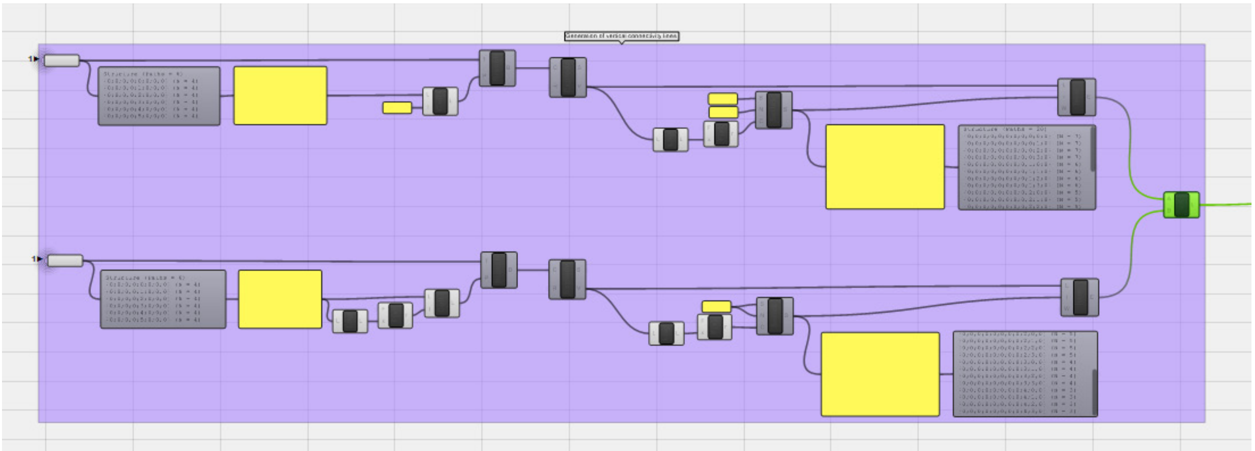


Fig.3.17. In the next part of the algorithm, all adjacent surface-piece's mid-points should be connected to each other in vertical direction. The main problem in here again is the way that points should be sorted to connect via a <line> component.

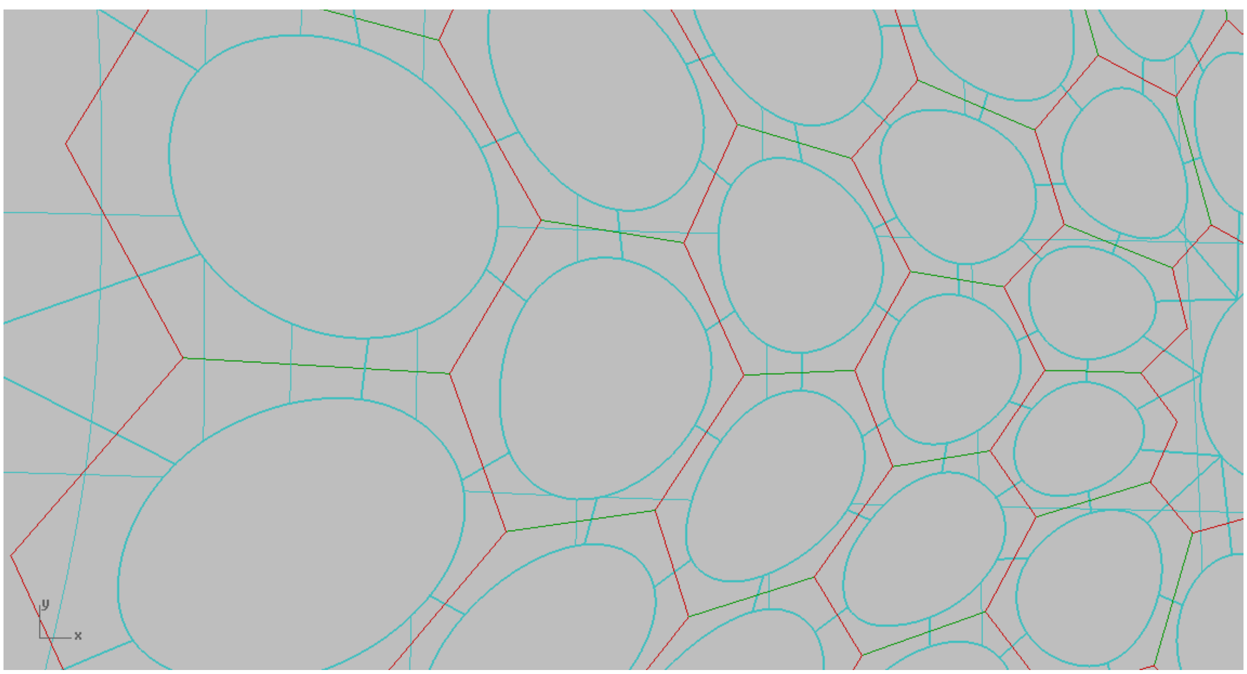


Fig.3.18. The result of the line connectivity would be a group of lines that generally converts the slut baselines into a pattern of deformed hexagons.

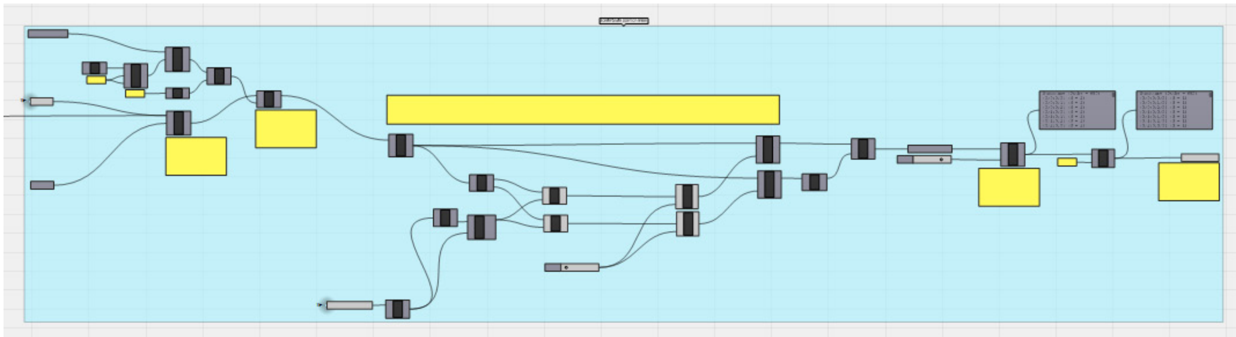


Fig.3.19. In the last step, all connectivity lines are being sorted in groups, each group consists of lines which attached to a surface mid-point. This has been achieved by controlling the distance of their end-points to the surface-mid-point. For each point, those lines whose start or end points are almost has the 0 distance from the mid-point are sorted into a group.

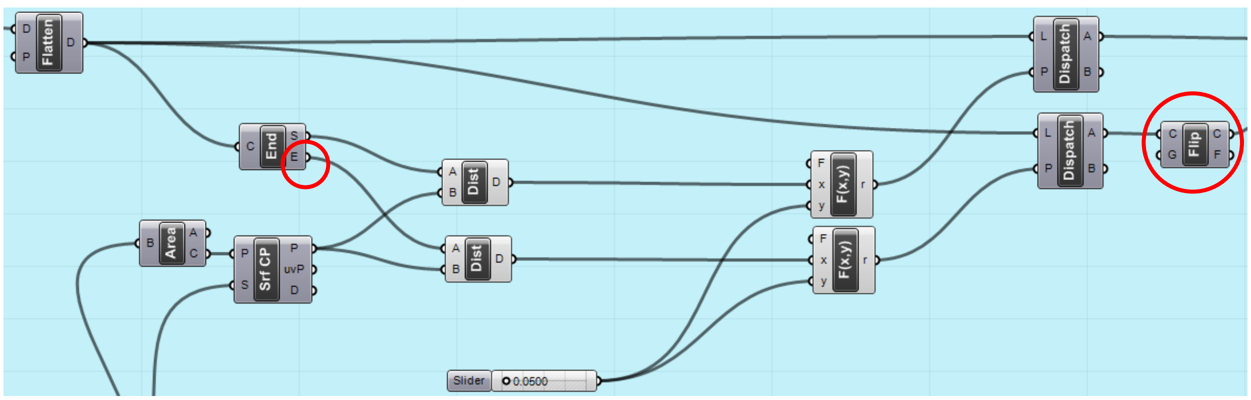


Fig.3.20. In order to select part of each connectivity line as the position of sluts, all lines whose end points are close to the surface-mid-point are <flip>ed to have the same direction as others, so at the end, all lines have their start point, close to the surface-mid-point.

In the last bit, as the position of joints (and sluts), just a portion of each line is selected.

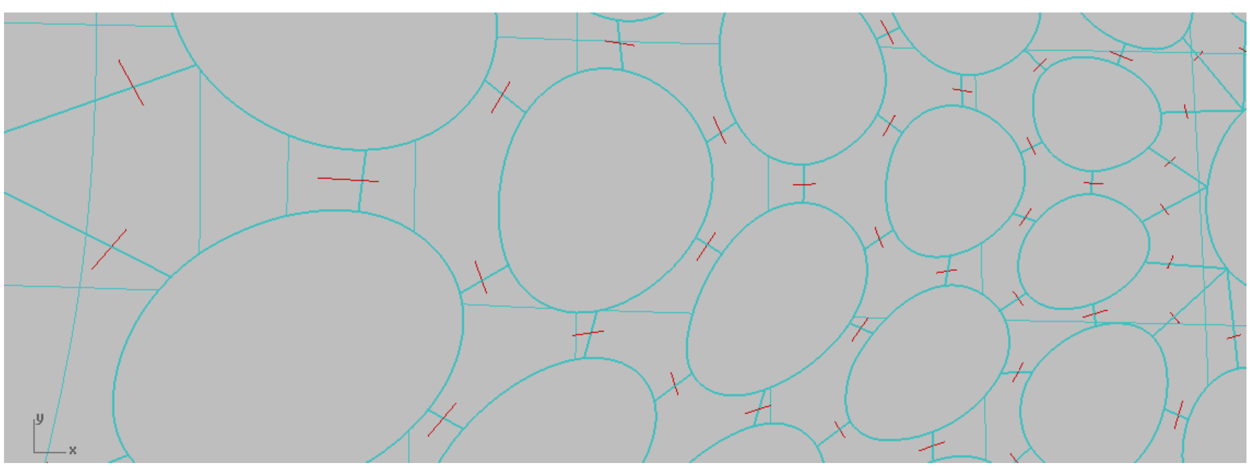


Fig.3.21. Baselines for Sluts/Joints.

2-2- Joints

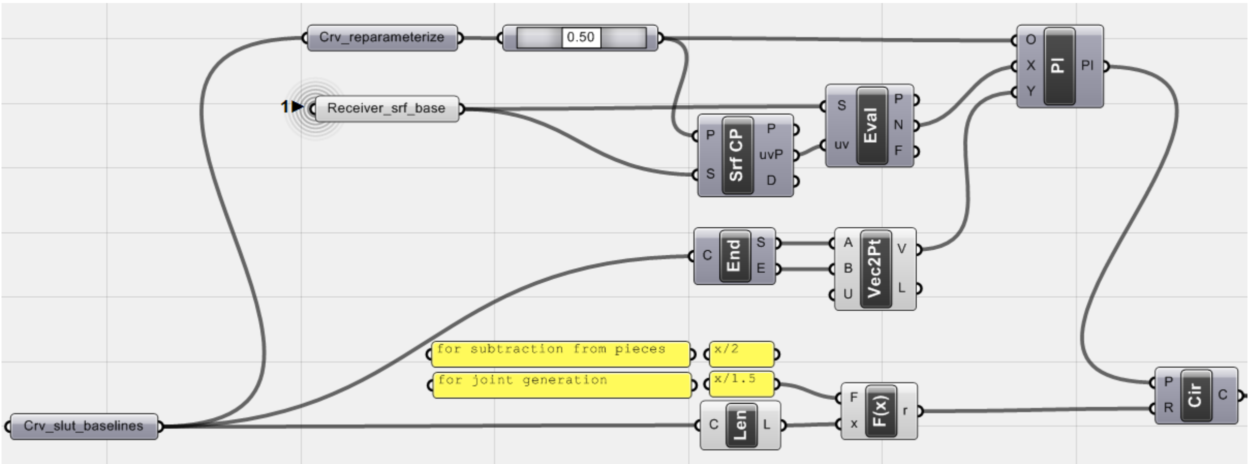


Fig.3.22. Since all baselines are ready, it is time to generate couple of planes at each surface edge and perpendicular to the surface and generate basic circles of the joints on them. The size of the circles is relative to the size of the surface piece and two different formulas could be used, one for generation of joints, and one for sluts which are smaller than joint a bit.

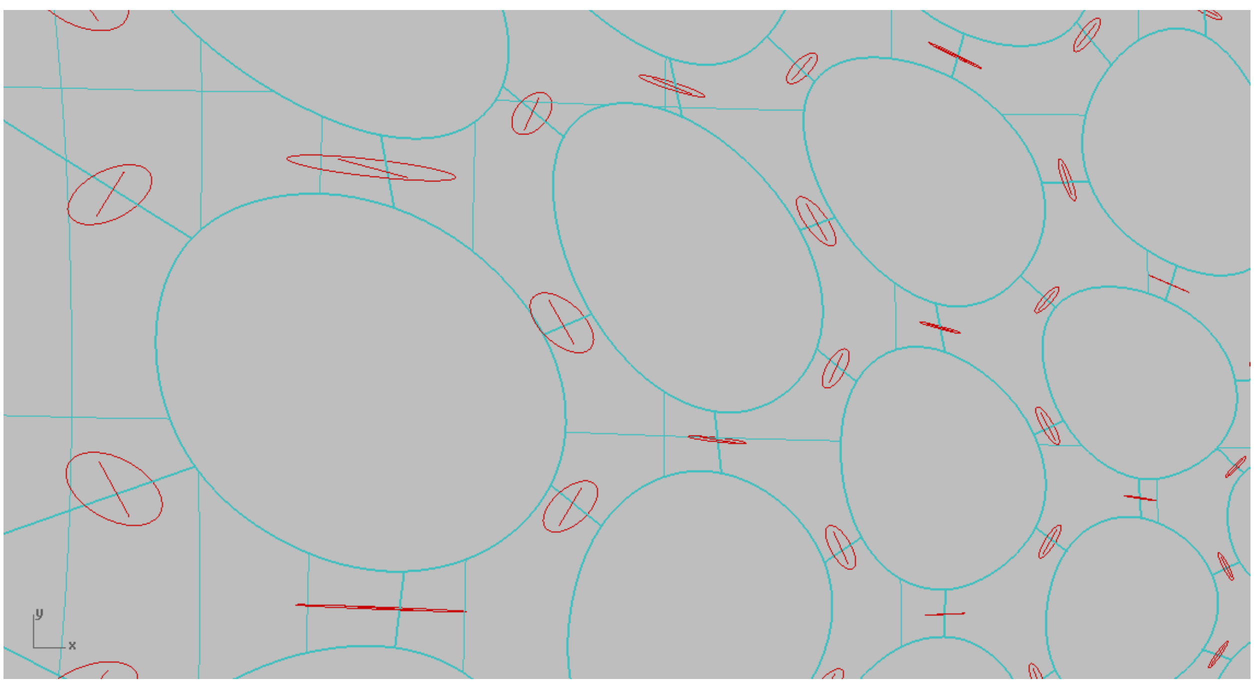


Fig.3.23. basic circles for joint generation.

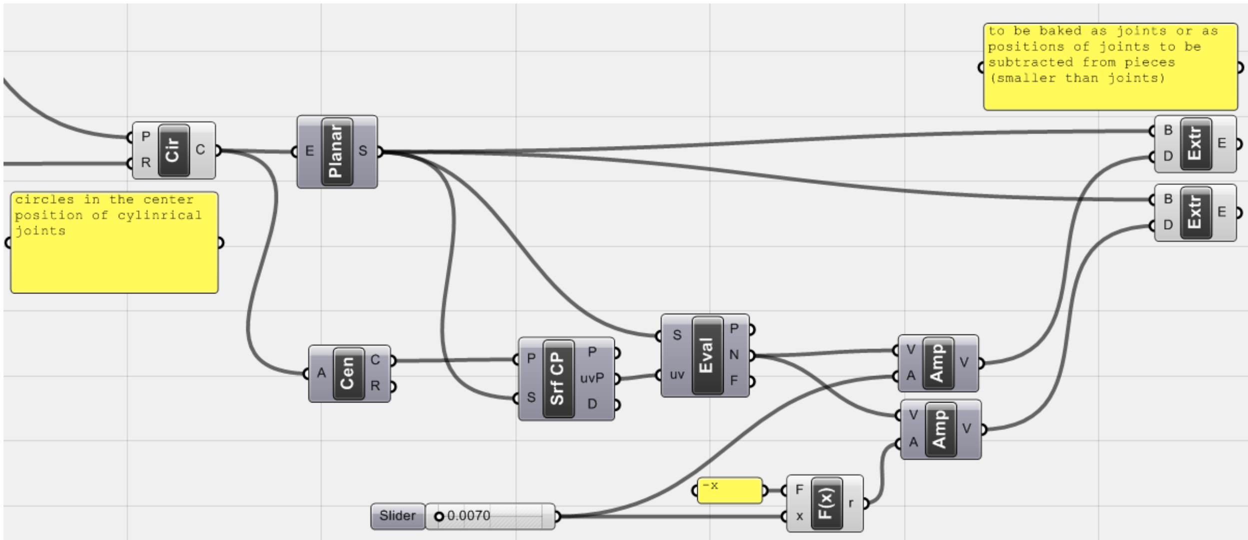


Fig.3.24. Circles are being <extrude>d for this next step to become cylinders in both sides. The amount of extrusion should be extracted from the material thickness that aimed to be cut from.

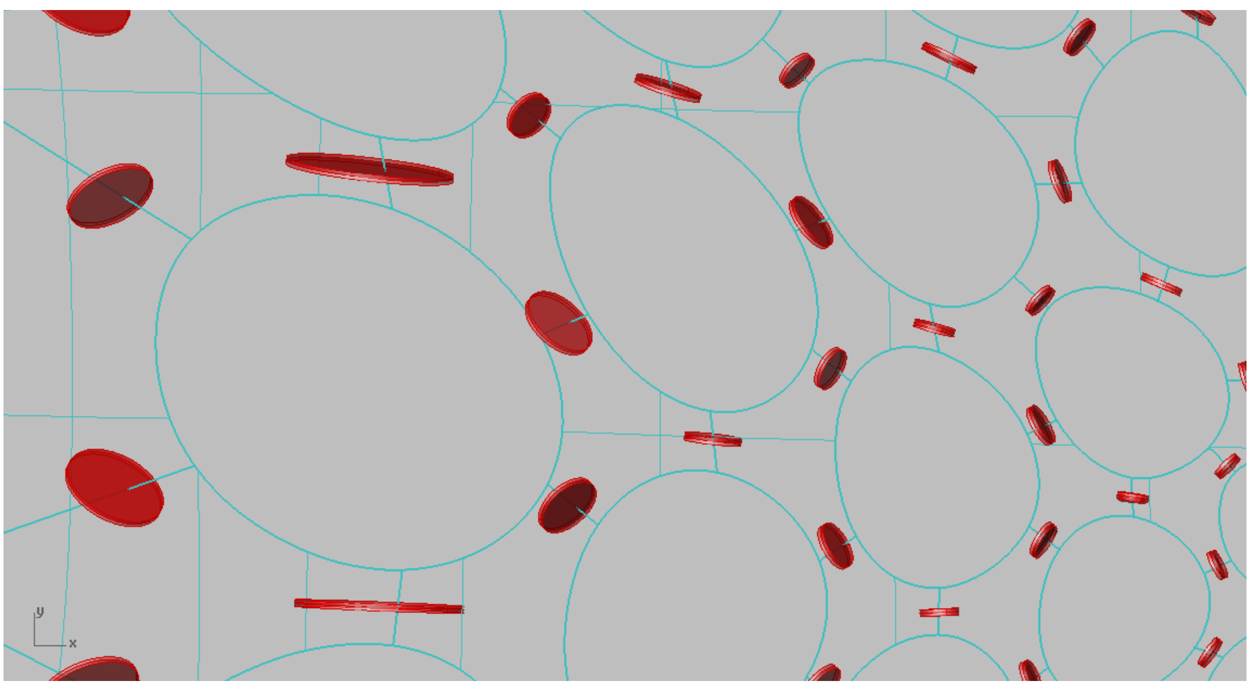


Fig.3.25. Joints.

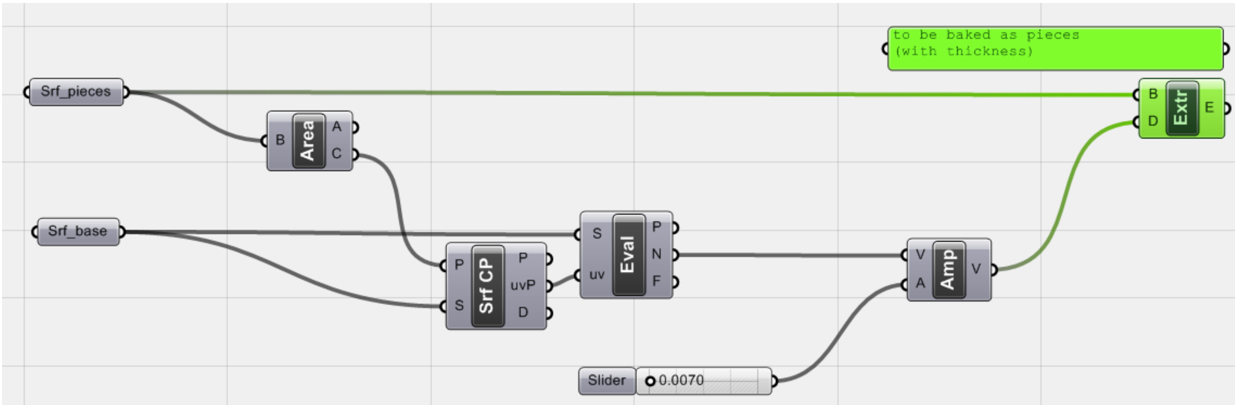


Fig.3.26. In this small bit, all surface pieces are also extruded a bit to perform Boolean subtraction on them later on.

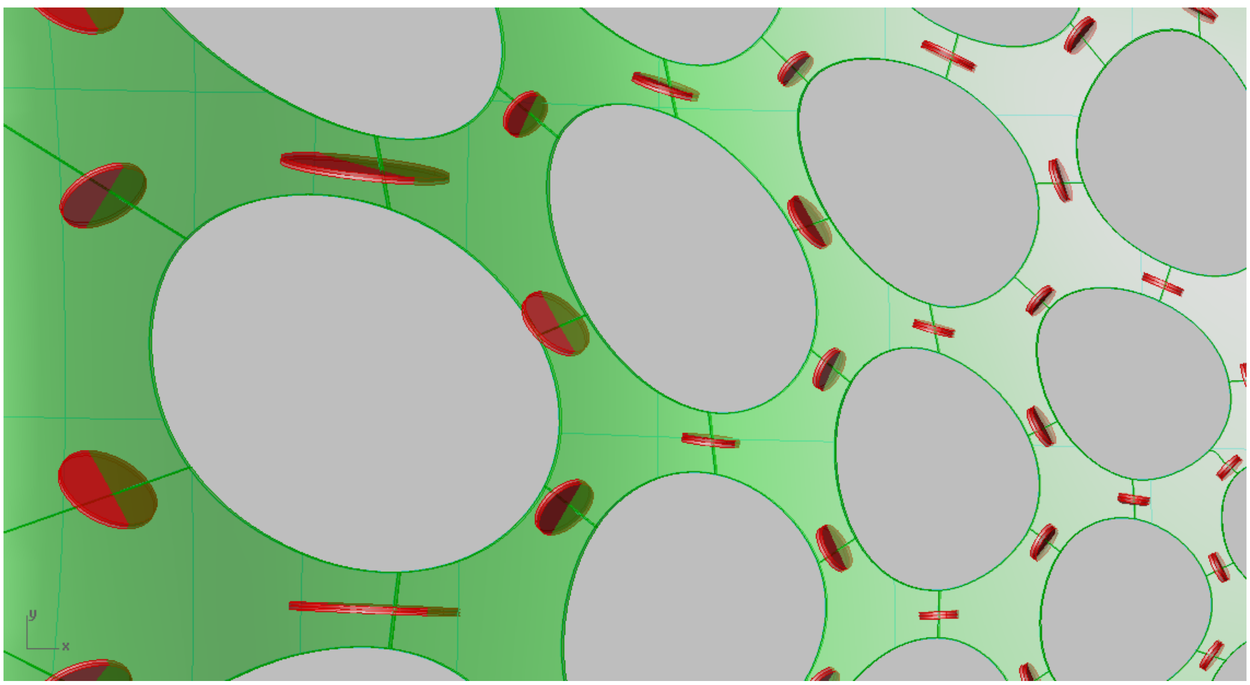


Fig.3.27. Surface pieces and their associate joints. All these pieces and joints should be baked in order to continue with later stages. Here the (x/2) as the size of sluts is used.

Fabrication_Step_3_Slut Generation

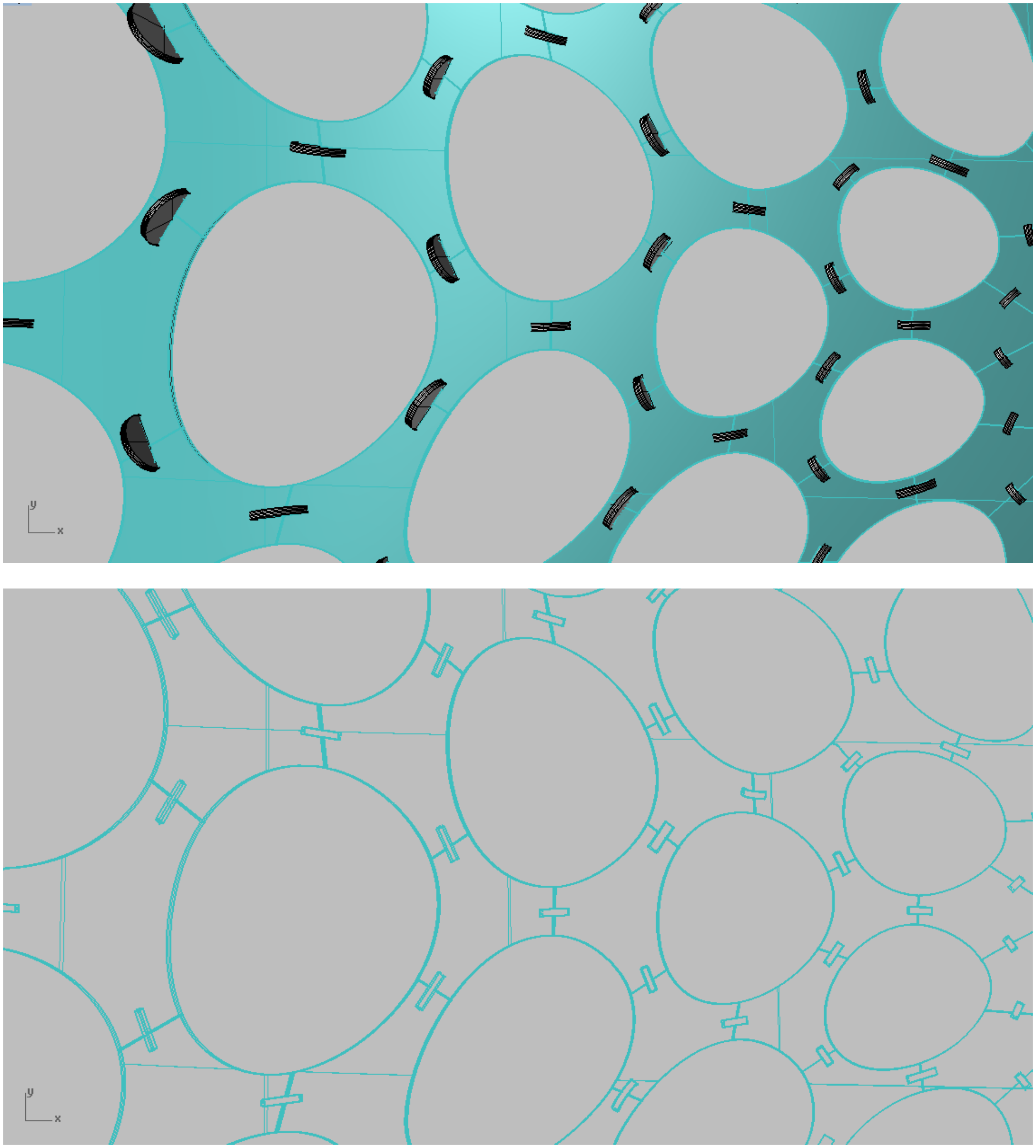


Fig.3.28. All cylinders (as representative of sluts which are smaller than actual joints) are being subtracted from the surface pieces in Rhino. The result is surface pieces with sluts on them.

Fabrication_Step_4_Nesting and Labeling

_1_Nesting Pieces

All components are being generated, now it is time to label them and nest them into sheets to cut out from desired material.

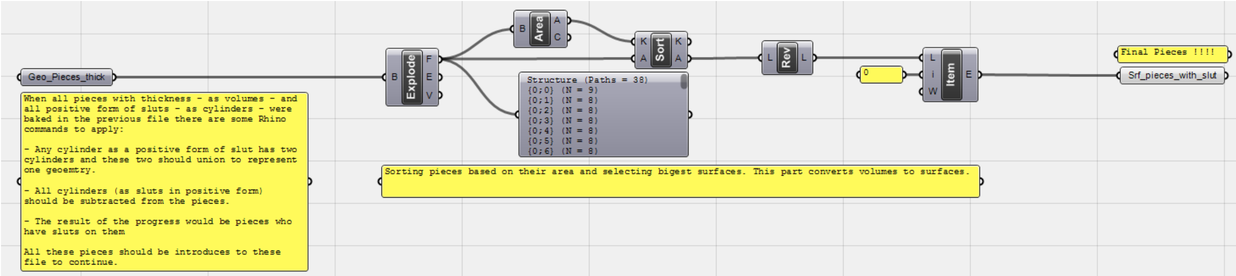


Fig.3.29. The very first step is to convert pieces from a volume to a surface to project only a surface into nesting sheet. Here by a single algorithm all surface volumes are exploded and sorted by their size and then their biggest faces has been selected. Since these files are potentially having loads of calculations they are separated into multiple files in order to prevent memory/computer crashes.

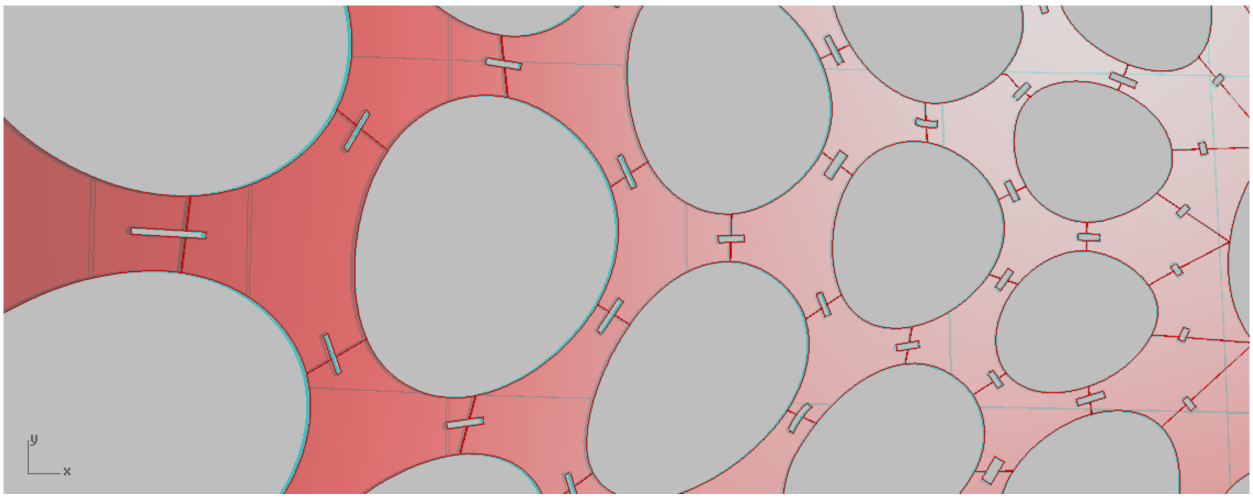


Fig.3.30. Surface pieces should be baked for the next step of the algorithm.

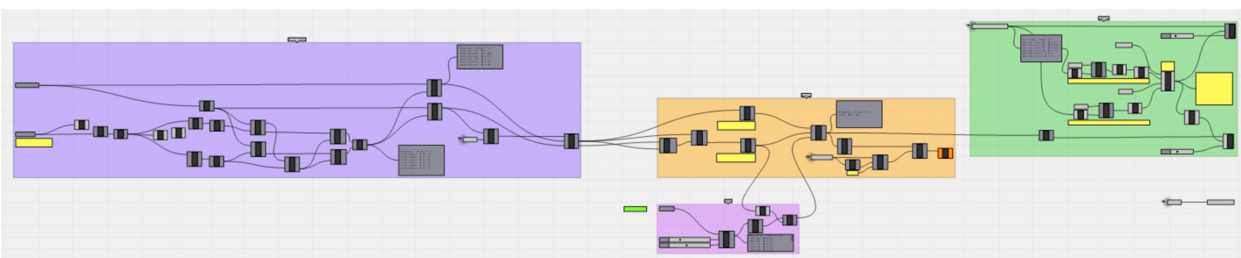


Fig.3.31. The algorithm for nesting has four main parts. The very first part is exactly the same as the sorting part of the previous section when all pieces were sorted. Here the new pieces with sluts are used as <srf_pieces>.

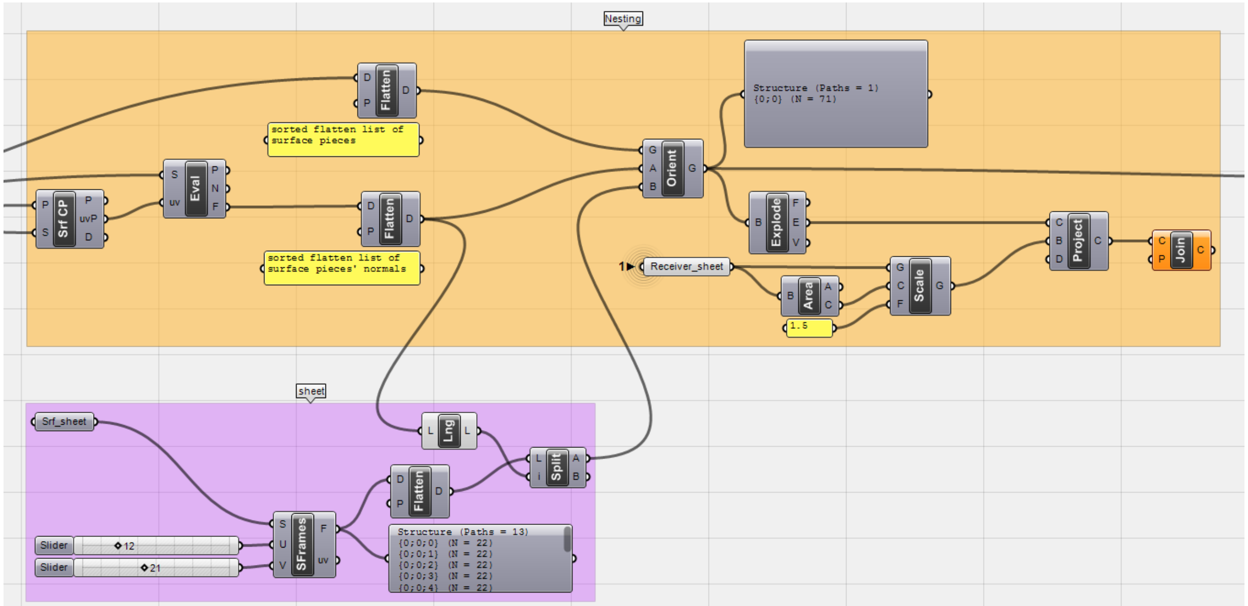


Fig.3.32. After sorting all pieces again, surface midpoints are evaluated to find their associate surface Normals, used to <orientate> pieces onto a sheet. The sheet is a rectangular surface, divided to have bunch of frames across it. Since there are more frames on this surface than needed, these frames are splitted by the number of pieces. All surface pieces are <orient>ed, then <explode>d and their edges are <projected> into a surface the same as the previous sheet but a bit bigger. These projected lines are <join>t again to give some closed shapes, nested in the sheet, ready to cut.

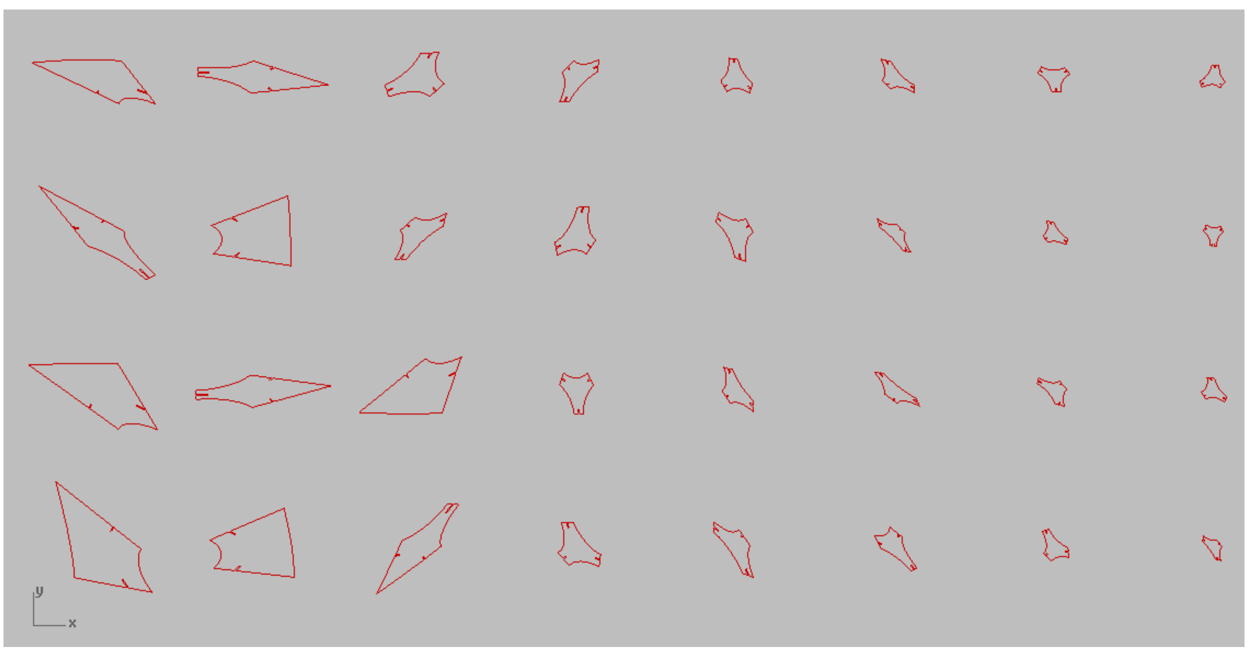


Fig.3.33. Nested pieces.

4_2_Labeling pieces

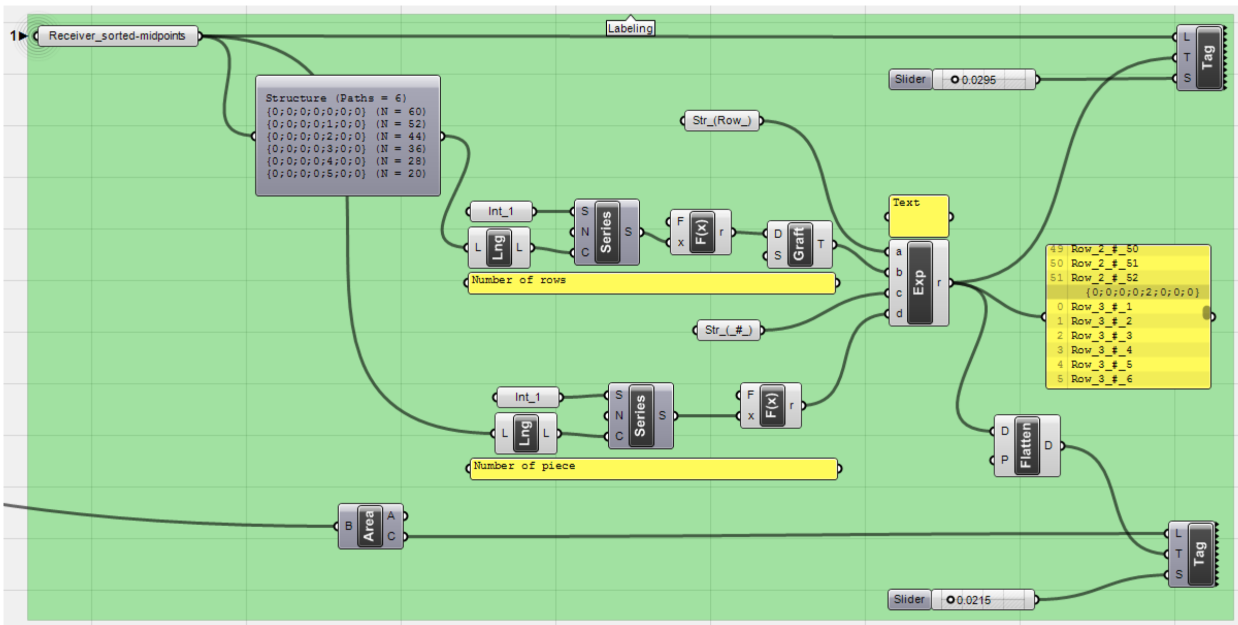


Fig.3.34. In order to label pieces, a list of text tags should be provided. These text tags should show in which row each piece is posited and the number of each piece in the row. With the same fashion as nested pieces, all pieces of the porous shell in their original positions are also labeled which help in later assembly stages to check the correct positions of pieces with the digital model.

HELP

In order to label pieces, a string should be generated that shows the position of the piece in rows and its index number in that row. So for example ‘Row_2_#_1’ will show that the piece is in the row 2 of the shell and in that row, it is the first piece.

So here two different numbering systems needed. To generate row numbers, the algorithm should know how many data branches the list of components has. Here using a <param viewer> and a <list length> component calculates the number of branches. This would be the number of values that a <series> component should generate numbers starting from 1. Here the ‘Fix’ part of these numbers has been used.

The second number has the same concept but here the number of values inside each branch should be the value to tell the <series> how many numbers needed.

The rest is two fixed string values (‘Row_’ and ‘_#_’) to complete the string. Using a merge component, all pieces are attached together to generate the labels needed.

Here by a <Tag> component, and using pieces mid-points as the location of text tag, all pieces labeled.

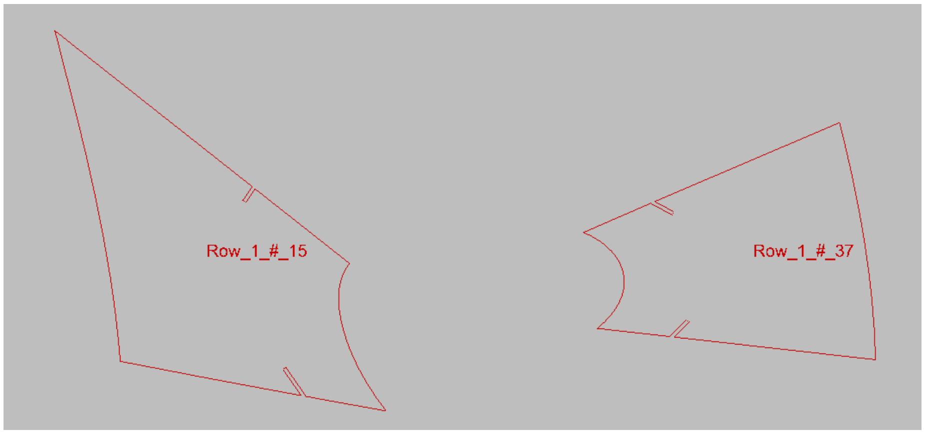


Fig.3.35. Nested-Labeled pieces.

4_3_Nesting and Labeling Joints

The last bit of the nesting is to prepare all joints to nest and label to be ready to cut. This has an algorithm almost the same as the pieces nesting algorithm with some minor changes.

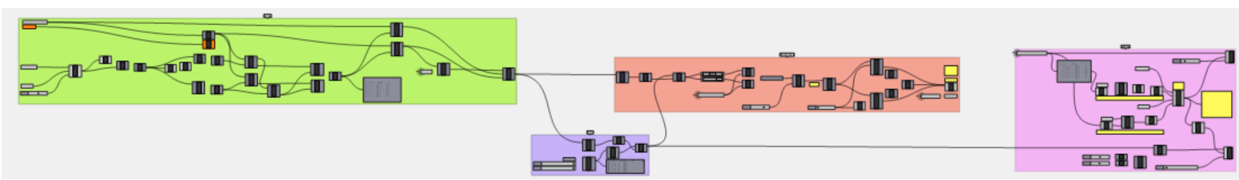


Fig.3.36.Overall view of the algorithm

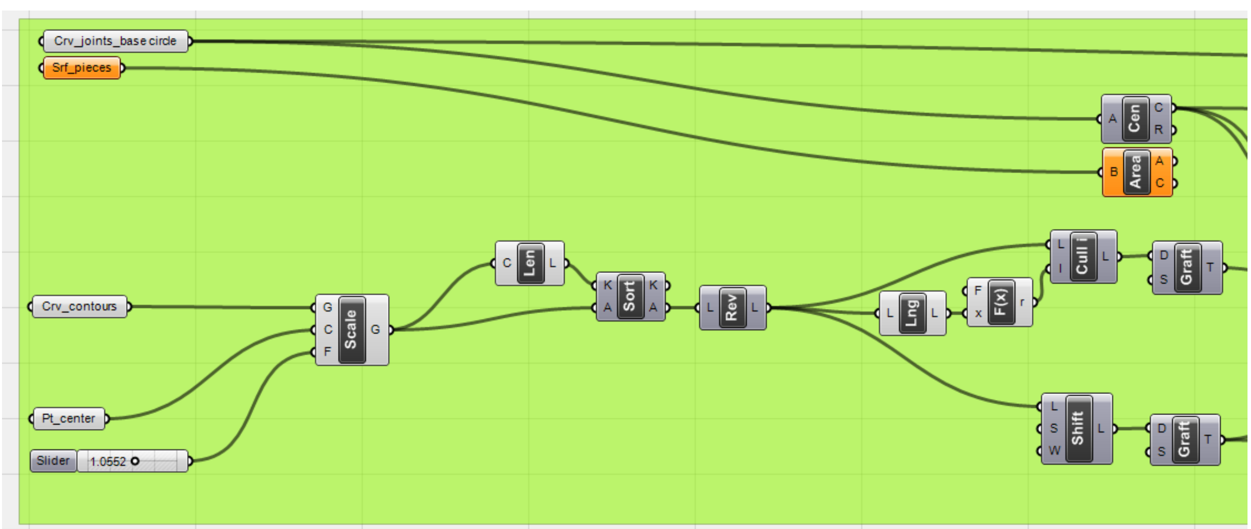


Fig.3.37. The first part of the algorithm is again sorting but here with small changes. First of all here, instead of surface pieces, the main input of the algorithm is circles that generated before as base circles of joints. These circles should be baked through the joint generation algorithm.

The next change is that since some joints are posited on the contour curves that has been used to sort them, here these contour lines has been rescaled to hold all joints in.

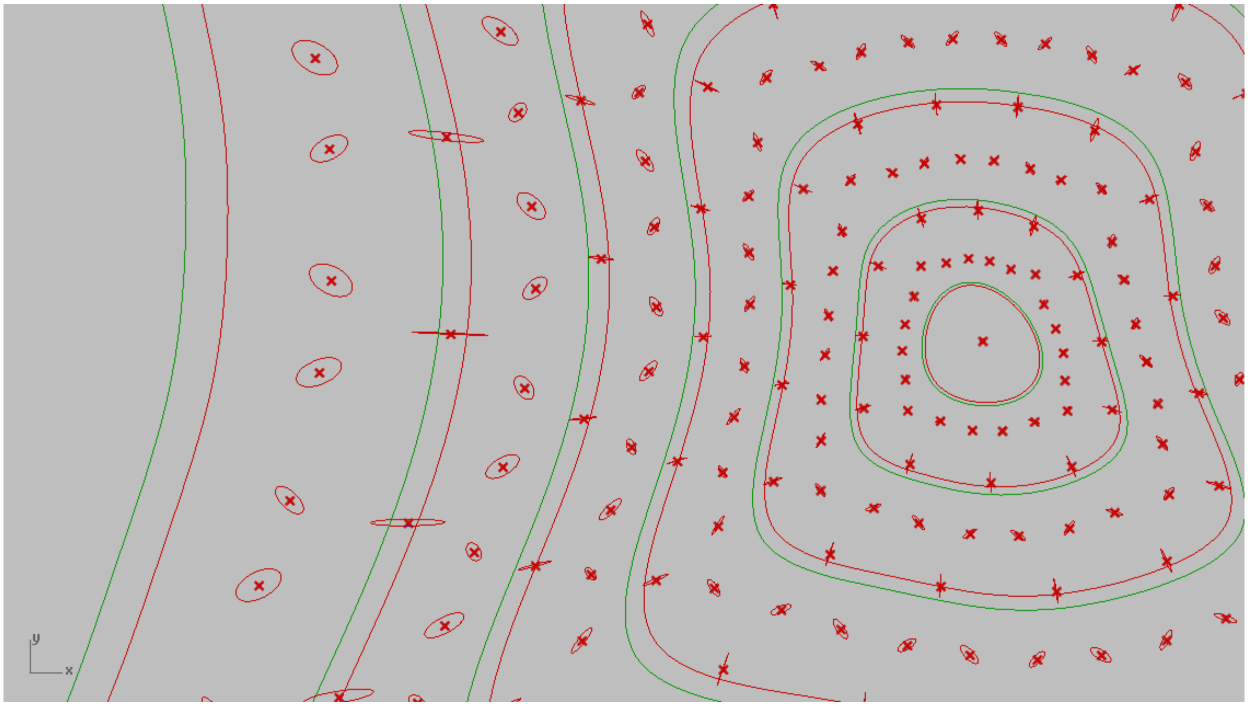


Fig.3.38. Rescaled contour lines.

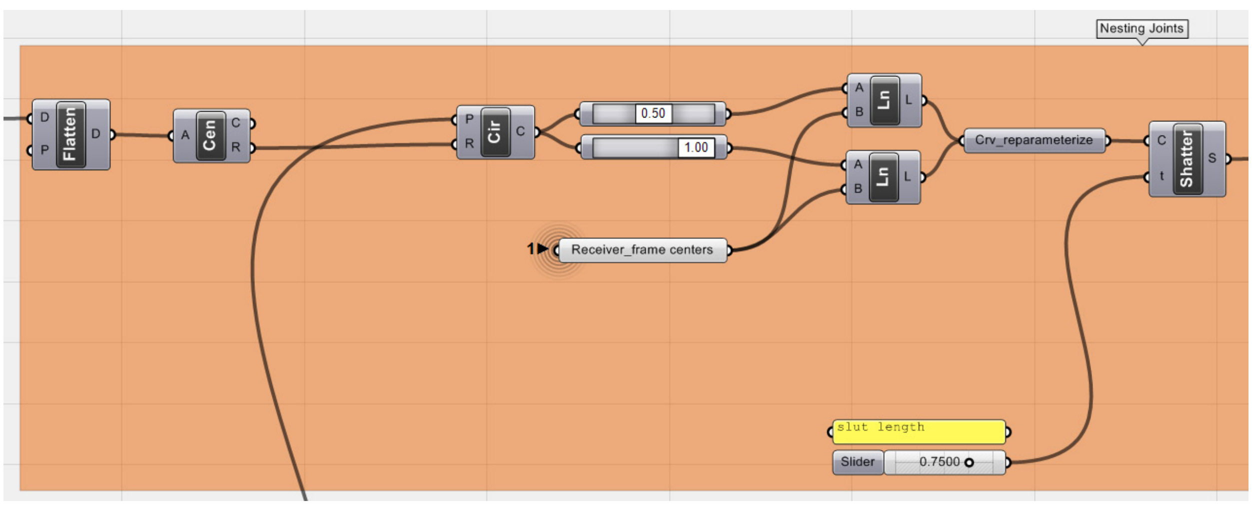


Fig.3.39. In order to nest joints onto a sheet, The <Center> point and the radious of circles extracted and all of them have been generated again on the sheet. Slut's base line has been drawn on a diameter of the circle and part of it trimmed as well.

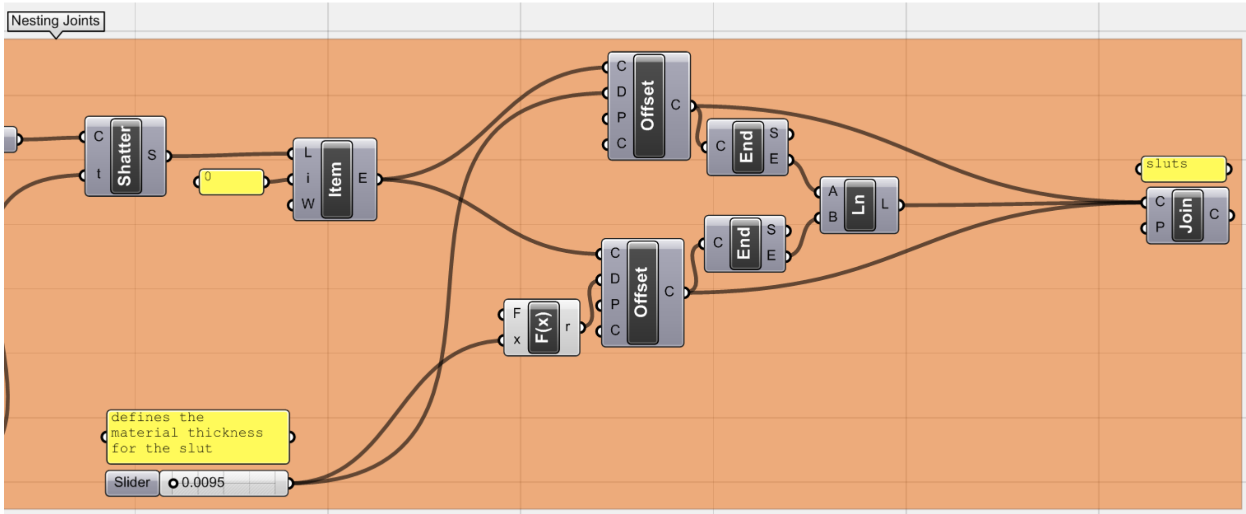


Fig.3.40. All base lines are <offset> to give thickness to the slut, proper to the selected material and all lines are <join>t to be baked later on.

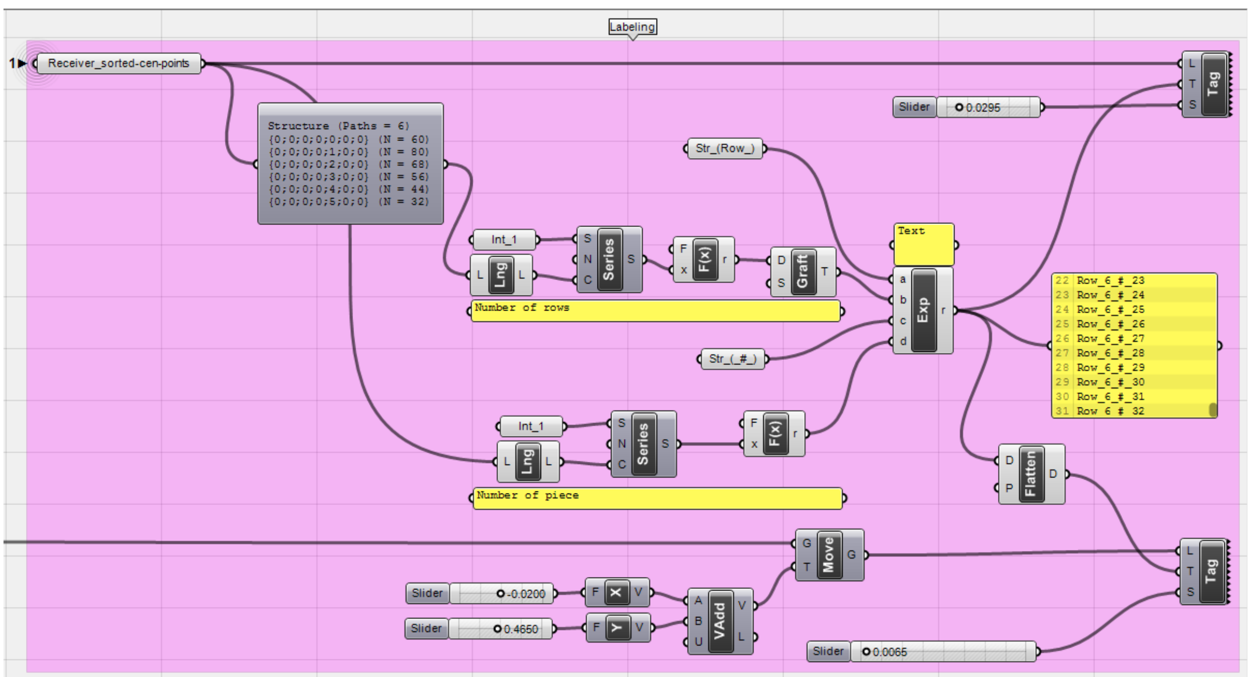


Fig.3.41. The Labeling part is again the same as the previous algorithm, here the position of text tags are also <move>d a bit in order to move them out of the slut area of the joint. Here sheet frames used as the position of the text tag which is the center of the circles.

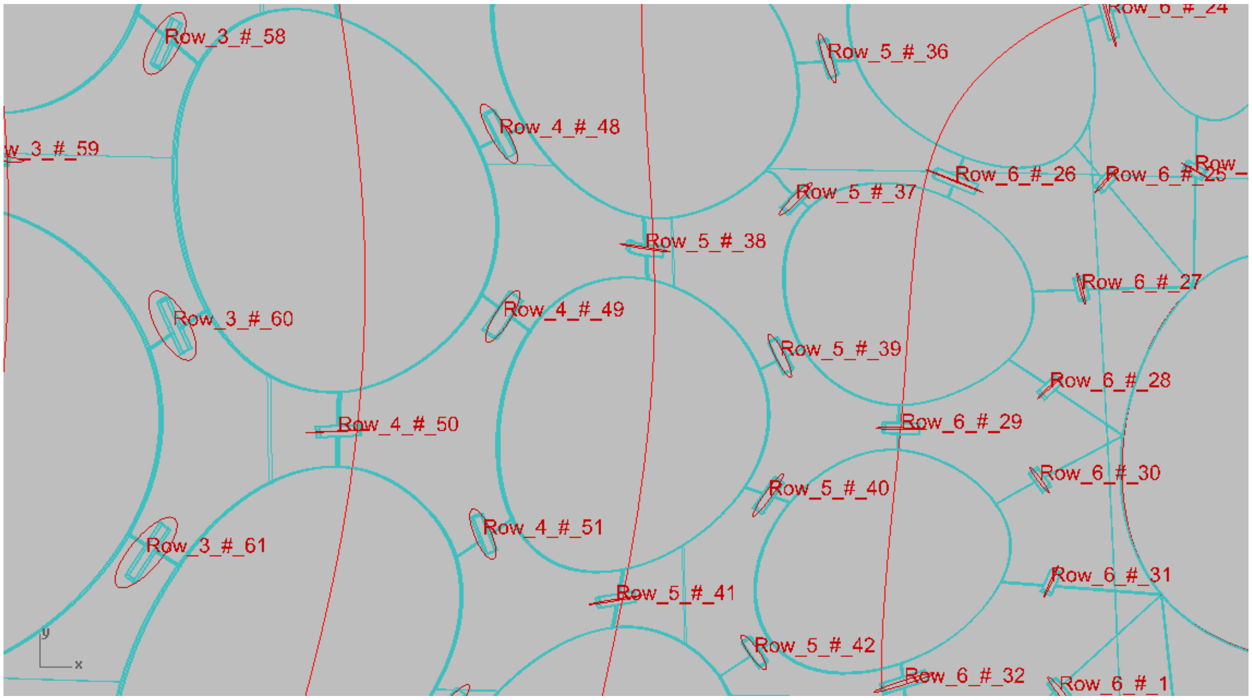


Fig.3.42. All labeled joints in their original position.

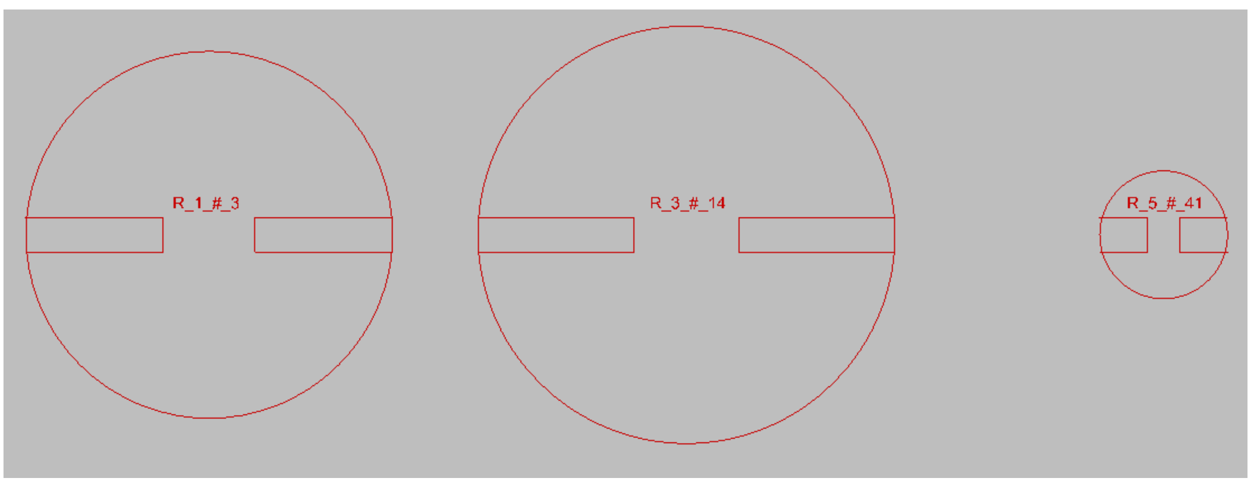
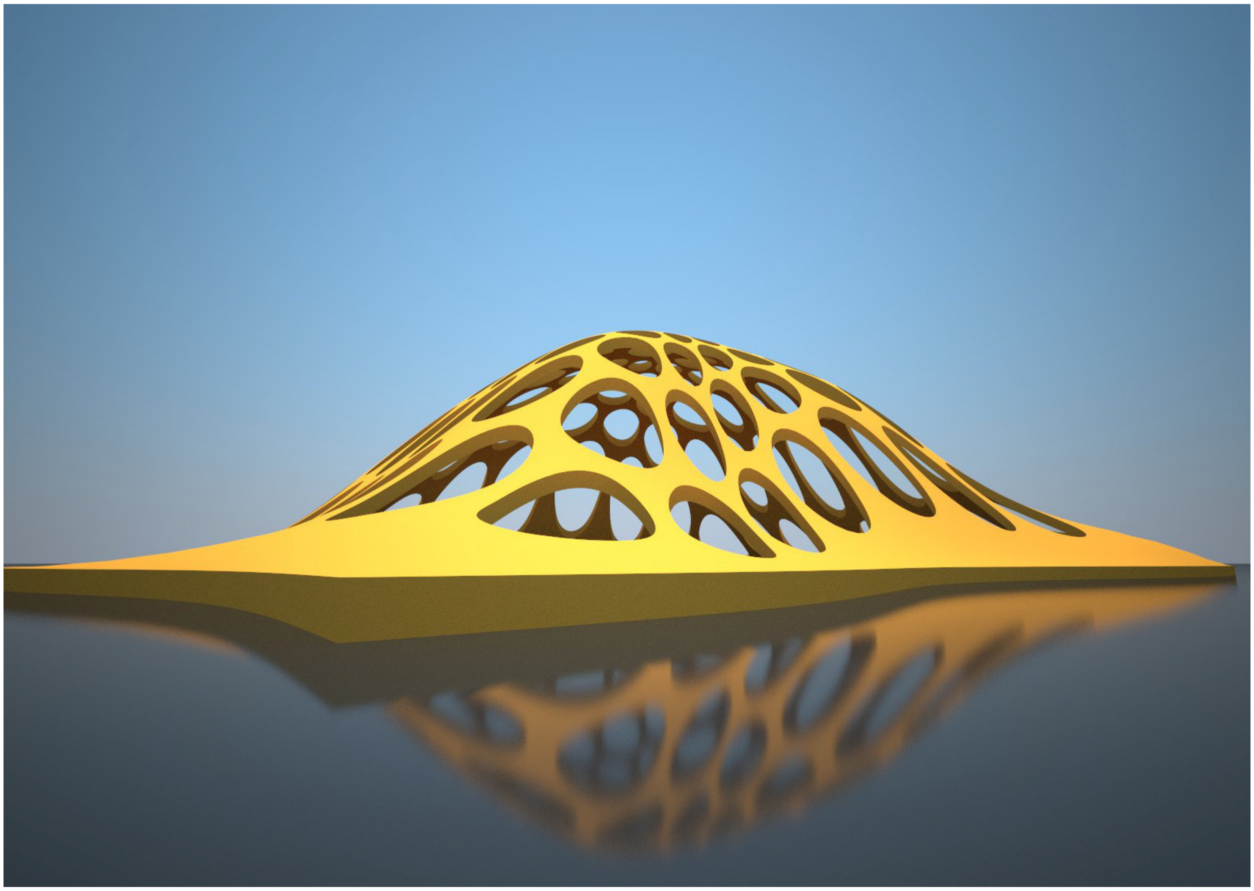
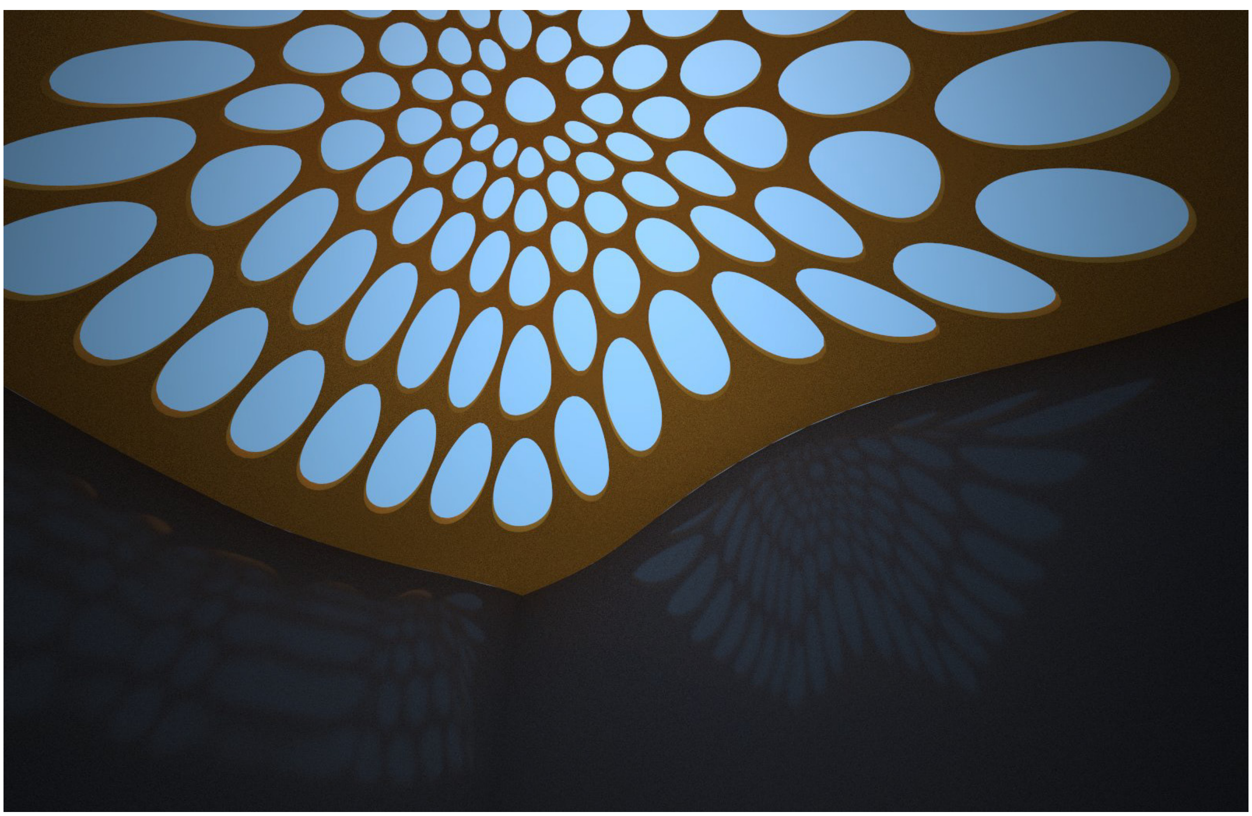


Fig.3.43. Nested Labeled Joints.

This is it. All pieces and joints are now ready. There are some manual manipulations needed to fit them best into sheets and then they have to be saved with necessary file format to send to the laser cutter and get the pieces back. All other issues are related to the machine and its arrangements to suit the material.

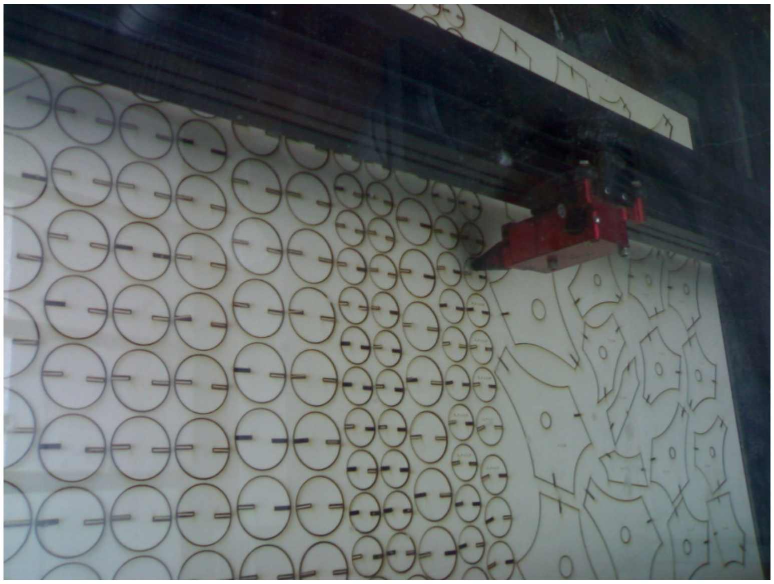


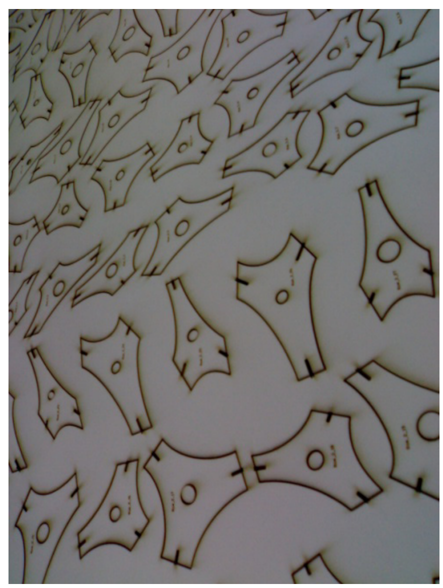


Fabrication_Step_5_Manufacturing and Assembly

Note: The project which is presented here is a little bit different from the one that proceeded before and tested with an earlier version of the algorithm.

_1_Manufacturing





_2_Assembly

_3_Final Product

3_3_Outlook

_Further development of the system

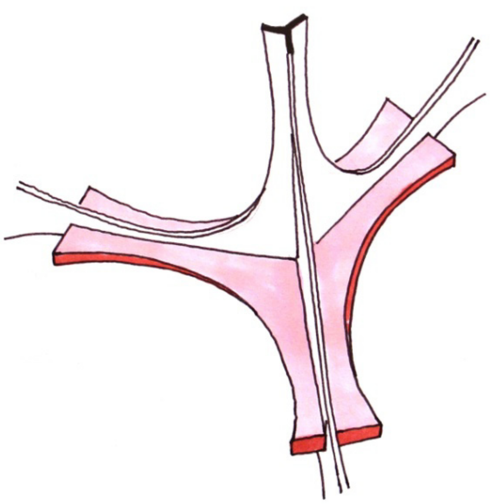
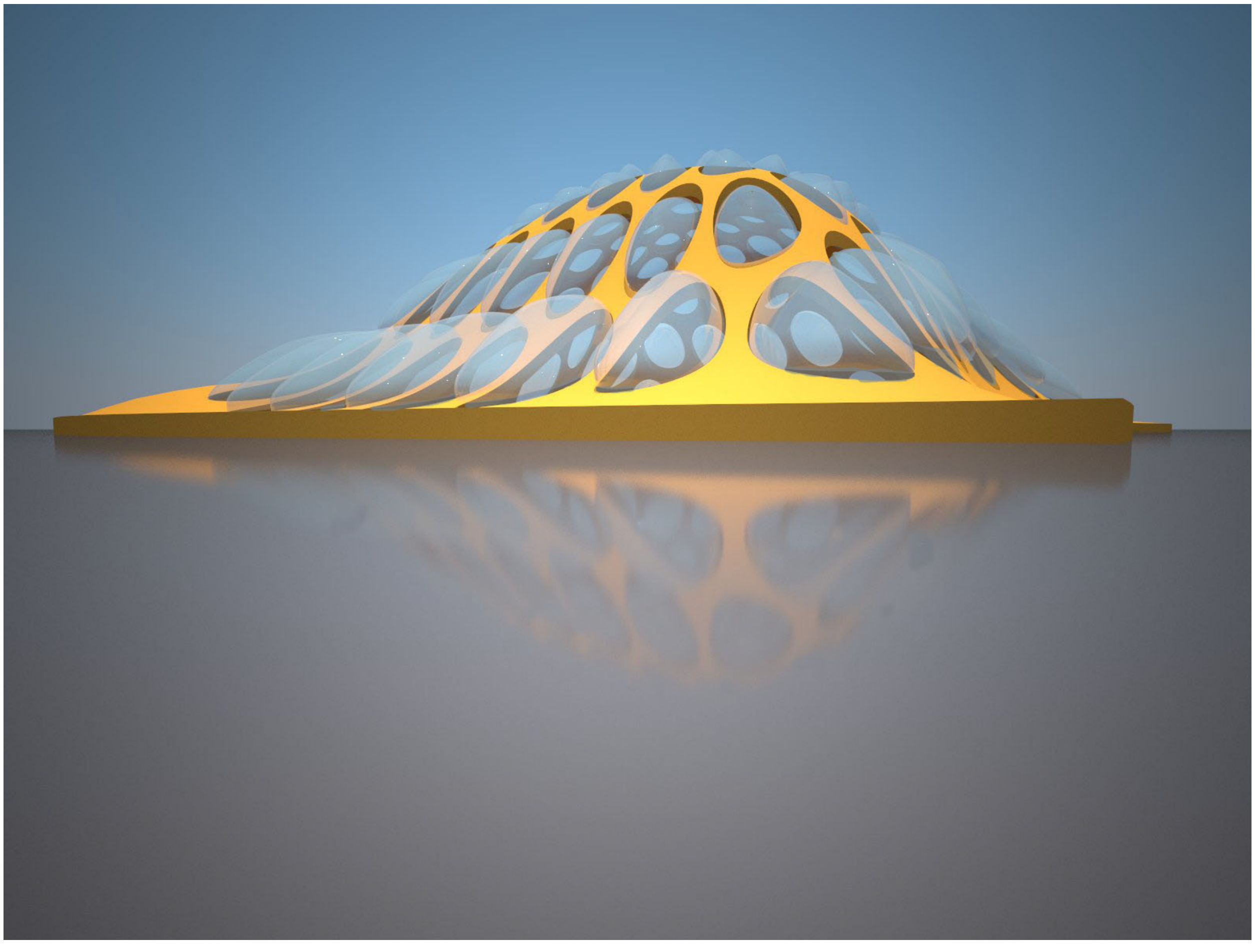
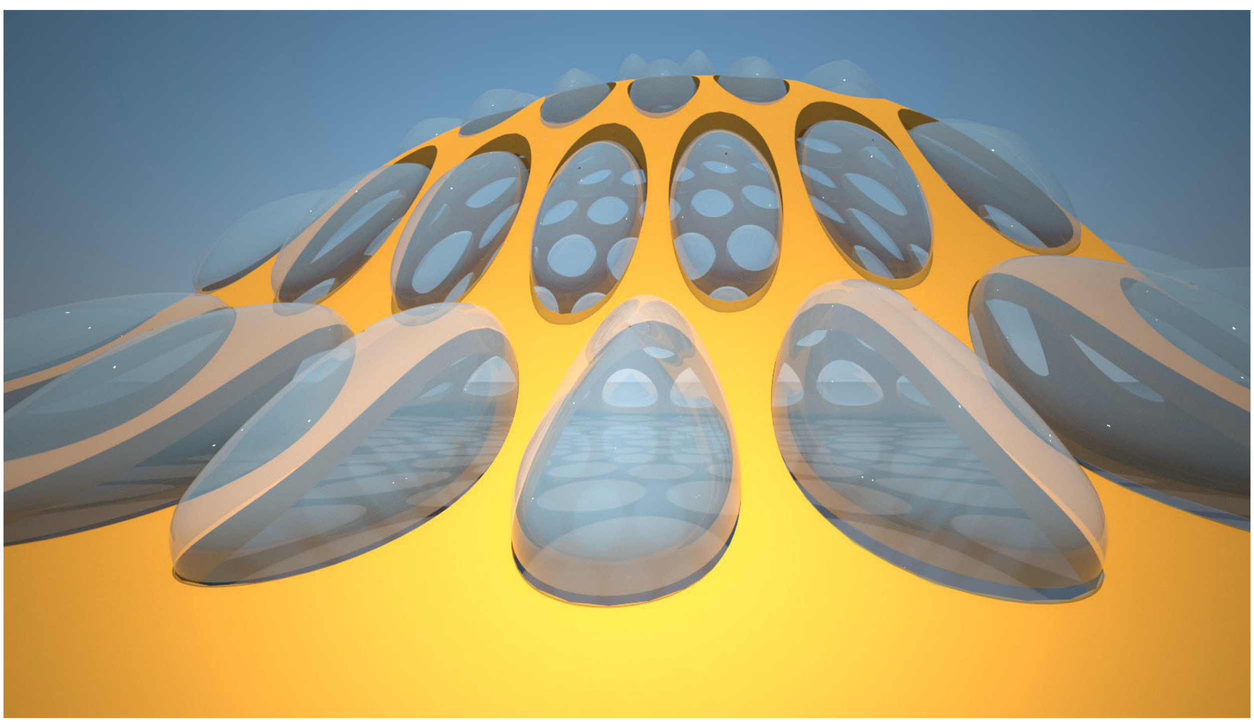


Fig.3.44. Further development of the system with new joints, here as an interlocking system of elements which makes a more stable connecting system with other possibilities for the system like shell height for more structural capacities or extra points for coverage connections.





_Bibliography

1_Pottmann, Helmut and Asperl, Andreas and Hofer, Michael and Kilian, Axel, 2007: “Architectural Geometyr”, Bentley Institute Press, Exton.

2_De Berg, Mark and Van Kreveld, Marc and Overmars, Mark and Schwarzkopf, Otfried, 2000: “Computational Geometry, Algorithms and Applications”, Second Edition, Springer,
Germany.

3_D’Arcy Thompson, 2004 (Sixth Printing): “On Growth and Form”, Cambridge University Press, Cambridge, UK.

4_IL, (journal of Institute for Lightweight structures), No.28, 1984: “Shells in Nature and Techniques I: Diatoms”, Krämer, Stuttgart.

5_IL, (journal of Institute for Lightweight structures), No.33, 1990: “Shells in Nature and Techniques II: Radiolaria”, Krämer, Stuttgart.

6_IL, (journal of Institute for Lightweight structures), No.33, 1999: “Shells in Nature and Techniques III: Diatoms II”, Krämer, Stuttgart.

7_Hensel, Michael and Menges, Achim, 2008: “Morpho-Ecologies”, Architectural Association, London.

8_Nerdinger, Winfried and Otto, Frei, 2005: “Lightweight Construction-Natural Design”, Birkhauser, Basel.

9_Otto, Frei and Rasch, Bodo, 1995: “Finding Form”, Edition Axel Menges, Deutscher Werkbund Bayern.

10_Otto, Frei, 1972: “Tensile Structures, Volume One: Pneumatic Structures”, The MIT Press, Cambridge, Massachusetts and London, England.

11_Otto, Frei, 1972: “Tensile Structures, Volume Two: Cables, Nets and Membranes”, The MIT Press, Cambridge, Massachusetts and London, England.

12_Dent, Roger N., 1971: ”Principles of Pneumatic Architecture”, The Architectural Press, London.

13_Hensel, Michael and Menges, Achim and Weinstock, Michael, (Editors), 2004: “Emergence, Morphogenetic Design Strategies”, Architectural Design (AD) Journal of Architecture,
Wiley Academy, London.

14_Hensel, Michael and Menges, Achim and Weinstock, Michael, (Editors), 2006: “Techniques and Technologies in Morphogenetic Design”, Architectural Design (AD) Journal of
Architecture, Wiley Academy, London.

15_Hensel, Michael and Menges, Achim and Weinstock, Michael, (Editors), 2008: “Versatility and Vicissitude”, Architectural Design (AD) Journal of Architecture, Wiley Academy,
London.

16_Ball, Philip, 2001: “The self-made Tapestry, Pattern formation in nature”, Oxford university press.

17_Ball, Philip, 2009: “Shapes, Nature’s Patterns, a tapestry in three parts”, Oxford university press, Newyork.

18_Abraham, Ajith and Grason, Crina, Ramos, Vitorino, 2006: “Strigmergic Optimizaion, studies in computational intelligence, volume 31”, Springer, Germany.

19_Bonabeau, Eric and Dorigo, Marco and Theraulaz, Guy, 1999: “ Swarm Intelligence: From Natural to Artificial Systems”, Santa Fe Institute Studies on the Sciences of Complexity,
USA.
20_Engelbrecht, Andries P., 2005: “Fundamentals of Computational Swarm Intelligence”, Wiley, UK.

21_Brodie, Christina, 2005: “Geometry and Pattern in Nature 1: Exploring the shapes of diatom frustules with Johan Gielis’ Superformula”, http://www.microscopy-uk.net/.

22_IL, (journal of Institute for Lightweight structures), No.19, “Pneus”, Krämer, Stuttgart.

23_IL, (journal of Institute for Lightweight structures), No.22, “Form”, Krämer, Stuttgart.

24_Gebeshuber, I. C. and Crawford, R. M., 2006: “Micromechanics in biogenic hydrated silica: hinges and interlocking devices in diatoms”.

25_Garin, Gabriel Sanchiz, 2007: “Pneumatic Form-Found Concrete Structures”, Emergent Technologies and Design, Thesis Submision, Architectural Association, London.

26_Patrick Shoumacker: ‘Paraemtricism as a style, Parametricist Manifesto’, http://www.patrikschumacher.com/, 2008

27_http://www.designexplorer.net/designtooling , Axel Kilian.

28_http://www.morphographics.com

29_http://www.ucl.ac.uk/GeolSci/micropal/index.html

30_http://www.radiolaria.org

31_http://www.microscopy-uk.org.uk/mag/indexmag.html?http://www.microscopy-uk.org.uk/mag/artfeb05/cbdiatoms.html

32_http://www.micropress.org/micro_journal_special.html

33_http://www.michael-hansmeyer.com/projects/project4.html

34_http://earthsci.org/fossils/microfossils/microfossils.html

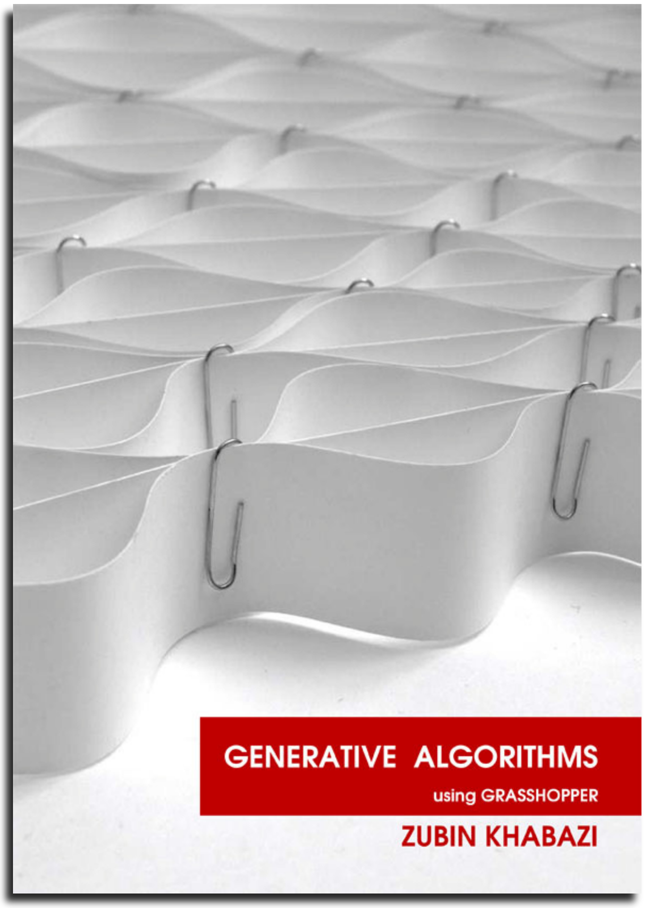
35_http://caliban.mpiz-koeln.mpg.de/haeckel/radiolarien/

36_http://www.pirx.com/droplet/radiolaria.html

37_http://www.wolframscience.com, Stephan Wolfram.

38_Zubin M Khabazi: 'Generative Algorithms (Using Grasshopper)', On-line publication by www.Grasshopper3d.com, 2010 listed at: www.morphogenesism.com under Grasshopper tutorial page.

_Notes

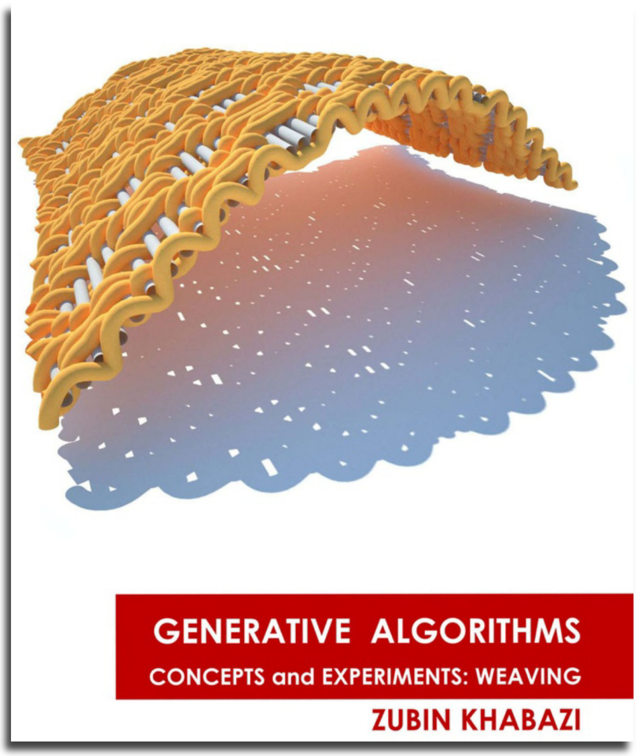


Generative Algorithms (Using Grasshopper) / Zubin Khabazi

Previously published on-line by www.Grasshopper3d.com, 'Generative Algorithms' as a mixed tutorial on geometry and grasshopper was the first of these design experiments series which focused on basic topics. These topics gathered together various fields of Generative Algorithms and Parametric Geometries include:

1. Generative Algorithms
2. The Very Beginning
3. Data sets and Math
4. Transformations
5. Parametric Space
6. Deformations and Morphing
7. NURBS Surfaces and Meshes
8. Fabrication
9. Design Strategy

The book comprised of design experiments in order to examine subjects and concepts in a practical way and would be useful for beginners.

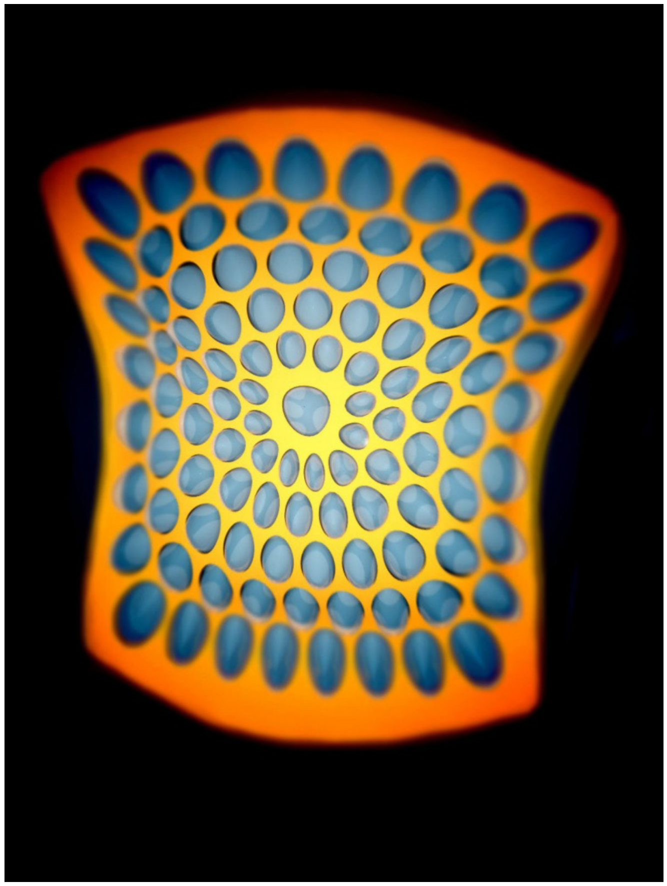


Generative Algorithms_Concepts and Experiments_Weaving / Zubin Khabazi

From the Introduction of the book: "In this series, 'Generative Algorithms, Concepts and Experiments', I am trying to search for concepts and methods, combine physical and digital experiments on topics that seem 'Generative' as prototypes for possible applications in architecture, to extend the catalogue of available systems and methods in design. I hope these experiments benefit your design issues or line of research, as mine."

The first one of the series was focused on the **Weaving structures** and two main algorithms developed through experiments. First one was a simple weaving pattern called simple **Loom** while in the second experiment, a more advanced and complex pattern system of fabrics called Jacquard, developed through experiments again, and the algorithm called **Jacquard Loom**.

The book is available to download as an on-line publication (pdf format) at www.morphogenesism.com in the Grasshopper tutorial section.



Generative Algorithms

Concepts and Experiments

2 _ Porous Shell

By

Zubin Khabazi

www.MORPHOGENESISM.com

zubin.khabazi@gmail.com