“Using simulations and artificial life algorithms to grow elements of construction”

by

Przemyslaw L. Jaworski

This thesis is submitted in partial fulfilment of the requirements for the degree of Master of Science in Adaptive Architecture and Computation from the University of London.

Bartlett School of Graduate Studies
University College London
September 2006

Abstract:

'In nature, shape is cheaper than material', that is a common truth for most of the plants and other living organisms, even though they may not recognize that. In all living forms, shape is more or less directly linked to the influence of force, that was acting upon the organism during its growth. Trees and bones concentrate their material where thy need strength and stiffness, locating the tissue in desired places through the process of self-organization.
We can study nature to find solutions to design problems. That's where inspiration comes from, so we pick a solution already spotted somewhere in the organic world, that closely resembles our design problem, and use it in constructive way. First, examining it, disassembling, sorting out conclusions and ideas discovered, then performing an act of 'reverse engineering' and putting it all together again, in a way that suits our design needs. Very simple ideas copied from nature, produce complexity and exhibit self-organization capabilities, when applied in bigger scale and number. Computer algorithms of simulated artificial life help us to capture them, understand well and use where needed.
This investigation is going to follow the question : How can we use methods seen in nature to simulate growth of construction elements? Different ways of extracting ideas from world of biology will be presented, then several techniques of simulated emergence will be demonstrated. Specific focus will be put on topics of computational modelling of natural phenomena, and differences in developmental and non-developmental techniques. Resulting 3D models will be shown and explained.

Keywords: adaptive, architecture, growth, construction, simulation, virtual, self organisation

Word count: 9262

3Table of contents

Acknowledgements

I have big respect for Alasdair Turner for creating such inspiring course, Ava Fatah for her commitment and helpful hand whenever needed, also Sean Hanna for his deep knowledge of the subject, and Chiron Mottram for good dose of inspiration and 'sparks' of new ideas.
I'd also like to thank all course participants who shared their view and opinions on presented material.
Also big thanks to my flatmates, for their support during the academic year : Ida, Marek and Monika, and my entire family for encouraging me to develop my interests over years.

1. Introduction

When reading various books and articles about artificial intelligence nowadays, my impression is that technology becomes some kind of a vessel for us, reflecting our needs and desires in making our world better. The environment we live in, as well as ourselves are subjects to constant re-examination and improvement. And it is not due to some unclear circumstances in human reasoning, but just basic human nature - to seek the challenges, find the answers and keep the mind busy with the task of building cleverer solutions.

The role of technology has not less meaning than the sense of entire process itself, as the technology is essential part of it. Disassembling simple ideas into pieces, rebuilding and reconfiguring them, and reassembling again, has always been present in thinking of intelligent human kind. Starting from building simple shelters, inventing and using the wheel, blacksmithing, through entire parade of XIX-th century inventions - it seems it is a constant process, no matter how content people are with existing level of technological evolution, we keep inventing more and more.

However, not even so recently, subtle tendency is being observed in science, which focuses on how this already advanced technological phenomenon begins to resemble all the intricacies and complexities of living organisms. The most appealing thing that could illustrate it is quite famous part of "Koyannisqatsi' [1] movie, where satellite photograph of a city (which resembles a living organism) is slowly zoomed out, and then faded into magnified image of a microprocessor chip. Without us realizing this, we build highly complex systems (telecommunication networks for example), based on simple rules, which represent similar systems spotted in nature (neural networks in brains, traffic routes of ants carrying supplies etc). All of them share the same important quality – self organization.

More examples can be easily found in mathematics and physics, where programs and algorithms create beautiful, non-uniform shapes and curves which under closer examination remind of forms seen only in nature (Mandelbrot's fractal, Julia sets or Koch's stars etc).

Following this everlasting human desire for getting better and better in whatever is invented, its not hard to foresee that level of technological complexity, sooner or later, will closely match this embodied in organic carbon based lifeforms. It doesn't mean that organic lifeforms will be replaced by silicon chips and fiberoptic cables (however it is not so unlikely, as seen in 'Ghost in the shell'[2] movie or read in William Gibson's 'Neuromancer' [3]), but rather that artificially created constructions and creatures will be similar in efficiency, and competing with those organic ones.

All of this reverse-engineering of living organisms (wider known as biomimetics or bionics) is undoubtedly expanding our knowledge, and though as we know more and more about rules governing biological systems, it turns out that it is not too hard to understand and apply them in desired circumstances. That is how we've discovered (rather than invented), and successfully used in science, cellular automata, L-Systems and boids algorithms, all based on behaviour of natural systems.

Those and other, similar complicated algorithms are now being used by determining the geometry of constructions that could be assembled in traditional way, but very soon they will be used directly during the physical process of creating them. It's enough to mention nanoengineering and nanorobots that take single atoms and put them together, but also in bigger scale it's possible to manufacture small bots (spiders) that would carry and put together brick-like elements. Reference to spiders is not coincidental, as through their smart mobile capabilities they seem to be most likely to be used in assembly of high-rise constructions, where human labour would be otherwise put at risk. Following simple, nearly instinctive rules, like an 'algorithm' to plait a web, they will build complex structures.

Examples of intelligence emerging from simple rules in nature could be multiplied, like bees building hexagonal honeycombs, beavers building dams, termites - mounds etc. As they are not taught those skills by anyone (or at least as it appears so), they must have it somehow encoded in their DNA, which would mean that it is based on simple set of instructions or directions.

To underline importance of studying nature, it's worth citing one of the pioneers of this subject, Frei Otto, who in his book 'Finding Form' [4] writes :

' The process of seeking form for large engineering constructions can serve as a model for the future explanation of force and force transpositions in objects of an animate and inanimate nature. It is a first stage in the explanation of the origin of their form.

Processes occur in the development of extreme engineering constructions in which spontaneous ideas can be optimized gradually. These processes have a life of their own. They are dependent of the people who set them in motion, but often produce results that were not predictable, indeed are even surprising. (...)

'Self formation' and 'Natural constructions' are subjects that need a great deal of commitment'. (...) The most important, as yet still provisional, result is a new interpretation of life's origin and the acquisition of form. Future work requires insights into the formation of objects, of emergence from unordered state, of creation. It must occur through objective, level headed research with a clear aim. '

Additionally, even though by the time the book was being written, algorithmic approach wasn't well known, he recognizes significance of experimenting with natural processes in virtual environment:

' The 'reverse path' method makes it possible to recognize formation processes in animate and inanimate nature to the extent that such processes are set in motion artificially. This is done by experiment and the technical development of constructions. Technical developments driven forward at a high level of qualification permit better knowledge of nature's non-technical constructions. This is known as the reverse path. Nature is not copied but made comprehensible through technical developments. '

Following the latter, initial conclusion is that finding programmatic ways of solving complex problems, especially in design and construction domain, is quite important. It is now regarded as interwoven into entire design&build process of creating the buildings, from the start of the design activity to finishing stages of on-site construction, rather than just as a standalone tool or operation.

2. Simulating forces

But assuming that above is true, and can be applied in full extent to design process (and building process too), what is the main difference in presented (biomimetic), and traditional approach? The main question is, and always will be, what practical difference (apart from aesthetics and visual pleasure) will it bring to the end user?

This investigation will follow the question : 'How can we use methods seen in nature to 'grow' elements of construction? How can we employ self organizational capabilities of many artificial life systems to exhibit creativity, and use it to assemble innovative, but inspired by nature, structures?' This topic will be extended, covering some selected strategies in simulating natural behaviours in computational emergence, then resulting models will be compared.

To answer the above, let's first start with analysing briefly what ideas are available out there, in nature. How plants and animals adapt to get more resources (water, light, space, food), and how this adaptation translates to interactions with various forces. Its necessary to investigate and understand those issues before we start re-engineering them and introducing them to design process, and through that - to life. We must realize all aspects of natural growth and its responsiveness to the surroundings, not only during the time of growth, but also during the life of the organism. Analysis of this kind is the only way to make sure, that engineered simulations of both subject and its environment are well suited for particular design task.

In 'On Growth and Form' [5] D'arcy Thompson analyses energetic efficiency of many organisms, studying in depth not only the results of their efficient 'life policy', but also the reasons for which they are as they are (in terms of shape and size). One of the most interesting quotes I found, is as follows: *"The form then, of any portion of matter, whether it be living or dead, and the changes of form which are apparent in its movements and in its growth, may in all cases alike be described as due to the action of force. In short, the form of an object is a 'diagram of forces', in this sense, at least, that from it we can judge of or deduce the forces that are acting or have acted upon it : in this strict and particular sense, it is a diagram - in the case of a solid, of the forces which have been impressed upon it when its conformation was produced, together with those which enable it to retain its conformation".*

To know well, what form of an 'organism' we want, when designing a building or a part of it, it is necessary to know those forces first, even before we start initial simulations. How can we find out where they come from, and what load of energy will they bring with? Through careful investigation of the environment, surely, but also through use of intuition and common sense. Perhaps, there is no perfect solution to that, except of building a model in 1:1 scale and measuring those forces, readjusting it, and repeating this process for a fixed period of time - which unfortunately in most of the cases is impossible or too difficult. But without knowing what forces we're dealing with, its impossible to create smart architecture, that forms a dialogue with the surrounding and responds to it. Now, let’s try to classify them and explain closer, which ones are most significant.

First, and most important of those, is the gravity, and thus all tensions and stresses within the structural elements (and we’ll mostly focus on those). As all forms living on planet earth, any building has to deal with them too. Bones in legs, construction of spine, trunks of the trees and branches bearing loads of leaves and fruits, they all respond to forces caused by gravity. Dealing with the weight of itself, apart from the weight of the load, makes it quite uneasy to presume its final shape, until we make necessary tests. However, inter-cellular physical forces caused by gravity are only the most basic ones, and we mustn't consider them as those that define the design, but only as the ones defining the details (of the structure). Trees seek water and light first, and then fight against the weight of their branches.

Membrane (surface) tension forces will play significant role here too, as they're applicable to problems related to facade construction (single/double skin glazing, ventilation, difference in temperatures/pressures between inside and outside etc.). One, that is more of an 'influence' than a force, is access to light. Even though process of photosynthesis is much more important for plants than for people living inside of the building, it still means a lot. With direct, or indirect sunlight, building receives heat, sometimes in excess, sometimes too little, in both cases making it a significant problem. Another one, is air movement, and related wind pressure forces, that matter most to high-rise building (in the same way, as they matter to higher trees). Noise is an important factor too, related to stress and fatigue, especially in big cities, where finding a quiet place becomes a difficult task.

Moreover, forces that will have the greatest impact on the layout/spatial composition of the building are the least obvious. They are hard to find and and may be discovered by functional simulations (for example by releasing agents and letting them settle down with their activity/place of operation, thus marking some areas as of special purposes). Only by running those procedures many times we may find out what version is the most efficient, or most functionally 'beautiful' one. This process could be compared to finding a best position for internal organs in animal body, or 'designing' position of the limbs, so they're most effective.

Those, and tens, maybe hundreds of other influences, become very meaningful for a designer, in the same way as they're important in nature for growth of plants, and virtually any organisms. Shape, construction and manner in which elements of the organism interact, are all a response to the environment and its forces. But only the combination of some of them, most essential or most influential ones will give us (often unexpected!) image of the concept. Therefore, apart from just discovering them, it's necessary to know which of those, and in what degree will be considered relevant during the process. Keeping in mind, that 'quality of the input reflects the quality of the output ', let's imagine a big 'design support machine', to which a designer throws results of his investigations (what forces, where, how important) from which machine generates possible shapes, or 'design spaces' as an outcome. This wouldn't be a deterministic, definite solution crafting aid, this would be just a tool – as good as a pencil or a ruler – helping to inspire, or to confirm what we often feel intuitively.

This 'machine' will be the main subject of this investigation.

3. Natural growth

Phototropic (light seeking) growth

As there are many forces to choose from, let's concentrate on those that influence directly parameters of the construction – which are related to gravity. We'll look at the examples in nature, showing how much organisms strive to get the best and strongest construction for the smallest energy and material expense.

'In nature, shape is cheaper than material' writes Julian Vincent in his essay 'Smart by Nature' [6] , which is probably a statement well known to great designers and visionaries of our era. But, whereas it seems to be quite deep, nearly philosophical truth to us, humans, at other, micro-organic level it is just a usual thing. Self-organization capabilities of growing cells create complexity, as mentioned before, from simple rules. In most of the cases, those rules will relate to making connections to other emerging cells, or eventually, suggest direction of cell subdivisions (which is essentially the same, assuming that dividing cells stay connected). Cells will increase presence (through subdivisions or grouping with another cells) along the line of applied force. It happens, because the force needs to be balanced with counter-reaction from emerging tissue. That spatial and physical efficiency will be easier to achieve through clever distribution of elements, than by adding more cells everywhere.

What makes the phenomenon of growth so significant and relevant to this, design-oriented investigation, is that the emergence of material in construction happens during the process of growth. It means, that cells don't have a preconception about the final shape, nor any definition or target of what they will be. They don't know it, not try to find it out. They simply follow simple rules of increasing the size of the tissue, up to the reasonable volume (which is also subject of efficiency), and through that contribute to the process.

To find a reasonable way of introducing growth to a virtual environment, we have to touch the subject of accuracy and anatomy of computational approach. Then we'll try to use self organizing-systems to grow adaptive constructions, that will find the most appropriate shape to fit design purposes.

4. What is a computational model

Subject of growth simulation in academic community has grown rapidly in recent years, and became quite popular among scientists. It's very easy to find books and articles concerning mathematical principles of living organism development, especially plants (for example 'Algorithmic beauty of plants' A.Lindenmeyer, P.Prusinkiewicz [7]). Trees are most common, but some researchers also tackle issues like bacterial growth, ice crystal growth, black hole growth etc.

Modelling of natural processes seems to enable us to enter very promising research area, greatly expanding and diversifying our design space. We have to bear in mind though, that translating biological mechanisms to language of mathematics and physics is not an easy task, sometimes impossible to perform with good accuracy. However, sometimes lack of perfectness in description of the mechanism may be an advantage rather than a downside. Philip Ball in his book 'Self made tapestry: pattern formation in nature' [8], describes this problem as follows: *' The point is that scientific descriptions of phenomena in all of these cases do not fully capture reality - they are models. This is not a shortcoming but a strength of science - much of the scientist's art lies in figuring out what to include and what to exclude in a model, and this ability allows science to make useful predictions without getting bogged down by intractable details.'*
This would fit previous assumptions, that we don't have to include all the forces interacting with possible construct, subject of our design process.

For the purpose of this work, we may assume that a computational model is an algorithmic description of phenomena, where behavioural properties of systems inspired by nature are simplified to fit designer's needs, and though treated as tools. Computational models are used for simulations of forces in action, producing outcome that without them would be impossible to achieve, as they only exist in virtual reality and are kept alive by computational power.

One of interesting examples of computational models that I've found, shows how combinations of different forces influence topologies and geometries of trees :'Structural simulation of tree growth and response' (John C. Hart1, Brent Baker, Jeyprakash Michaelraj)[9]. Few of the major findings include for example the fact, that by producing phototropic (light seeking) effects in the tree throughout the development, tree kills branches that are in the shade, though being able to

collect the same amount of light with only 25% of the branches, compared to purely ‘genetic’ model.
In further simulations of this ‘developmental’ model (being grown bit by bit, as opposed to non-developmental ones, where tree is generated to its mature state), authors included many additional influences, like gravity (geotropic models), angle of incoming light (planartropism), or wind, producing very realistically looking structures.

Some approaches combine simulations of growth with evolutionary algorithms, like this by Martin Hemberg in Genr8 plugin for Maya program ('Integrating generative growth and evolutionary computation for form exploration' Martin Hemberg, Una-May O'Reilly[10]). Program accomodates multiple techniques which allow to grow surfaces with use of L-Systems (here extended to Map L-Systems), then to deform them gradually with use of attractors and repellers. After initial configuration we're allowed to breed a generation of surfaces, from which program will promote some of them, depending on predefined fitness function. Promoted ones become 'parents' to a new, improved generation of geometries. This step, repeated even hundreds times, produces fantastic, surprising results.

Some researchers actually don't use any of previous concepts (generative growth and evolutionary algorithm), and create models through use of mathematical functions of complex variable. Dr Chris JK Williams (University of Bath) models beautiful organic forms resembling bone-like structures through folding complex meshes and enveloping desired volume with them, thus producing information for digital fabrication process ('The generation of bone-like forms using analytic functions of a complex variable' Miss EAO Nsugbe and Dr. CJK Williams, School of Architecture & Civil Engineering, University of Bath') [11]

In this investigation though, we're going to leave genetic algorithms behind, and concentrate on growth and adaptation procedures.

5. Methodology

Methodology developed for the purpose of simulating forces and generating responsive growth of structures, should be divided into three stages:

1. Designer sketches first concepts of what he thinks is the solution to the design problem. After that, data about environment from around the construct must be gathered and understood. Only then, we may decide which influences from it should be included in simulation. Generating ‘seeds’ initiating the process is next important task, defining starting conditions for the virtual model. ‘Seeds’ will often mean literally where first elements start to grow, that may be defined by a special ‘map’, which may be arbitrarily sketched or also generated by special functions.

2. Simulation is being run with different parameters, experimenting with seed location takes place, fine-tuning, and adjusting input values. Each influential force in the process has to be represented by specific, often different algorithm. Generally for physical forces, like tensions and stresses, its enough to use techniques based on Finite Element Analysis

(FEM), wind for example may be simulated with Computational Fluid Dynamics (CFD). All other algorithms may be approximated with use of agent based systems, because they're most versatile and it's easy to create many types of agents interacting with other types, which increases accuracy and variety of behaviours. One of the most intriguing and smart algorithms are Cellular Automatons, who work best with growing cellular structure of the construction (especially in initial parts of simulation). Occasionally neural networks and genetic algorithm may also be helpful.

3. Final simulation session runs, allowing the system to build it’s final state, define most efficient geometry, responsive to external influences. Data is being exported to modelling programs, producing final 3d model for further rationalisation.

6. Developmental and non-developmental simulations

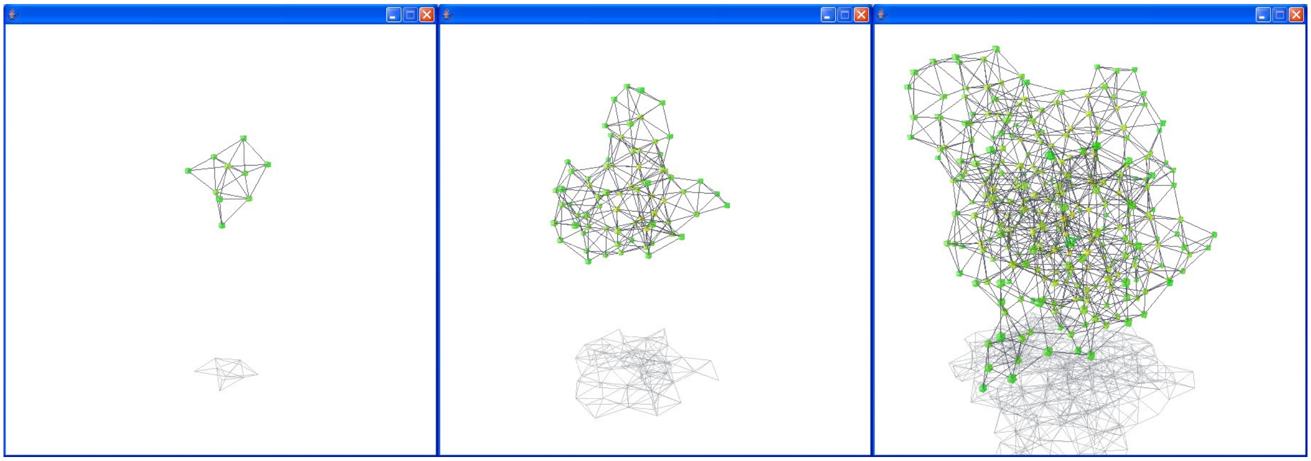
As said before, significant difference has to be made between those two approaches. Developmental models produce elements by additions, or subdivisions of existing elements. This is suitable for processes where spontaneous results are welcome, or where self-organizational capabilities are favoured, for example due to efficient material distribution. This however requires more time and effort to develop, as it happens bit by bit, element by element.
Non-developmental models are advantageous over the previously discussed, when it comes to generating   time. They are often inserted into simulated environment as they are, 'prefabricated', and then undergo adaptation and deformation processes driven by virtual influences. Their disadvantages are mainly related to simplified geometry of forms and non-flexible topology.
It's worth to say that most of the techniques using word 'growth' are in principle developmental. 'Adaptive' may mean both, however it is suggesting more non-developmental attitude.

As this work focuses on growth, this will be main subject, but later on, non-developmental approach will be also be briefly presented.

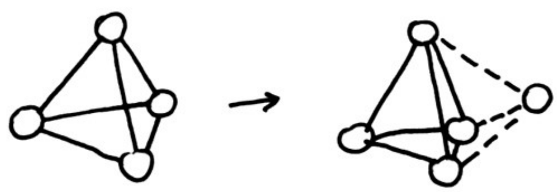
7. How to assemble growth simulation – developmental approach.

7.1. First step - growing bubbles

One of the first experiments I carried out, involved dynamic distribution of forces method (closely resembling Finite Element Method, just simpler and less accurate). Although gravity wasn't included here, mathematical description contained a procedure simulating molecular attraction and repulsion forces, which are actually slightly more interesting from mathematical point of view. The most similar thing seen in nature would be a foam made of simple, equally sized bubbles. First a 'seed' consisting of four points was placed in simulation space, connected as regular tetrahedron (form enclosing space with four equally edged triangles).
Even though I say this experiment creates bubbles, the graphic outcome doesn't look like foam, and this is because each bubble is treated as a point in space, with lines showing connecting forces working between them. Firstly, it simplifies graphical interface, secondly it shows only things that matter, which are spatial arrangement and distances between units.

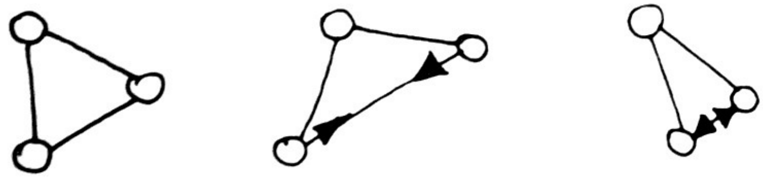


After placing the seed (4 connected points), program defines random point in space, neighbouring with one of the points. After that, it checks what other points are in close proximity (defined by threshold variable), and counts those. If it finds three or more, and randomly chosen point is not too close to other, already existing points, then it adds the node to entire structure, connecting it with selected nodes. This happens every displayed frame.



Adding new node to existing structure

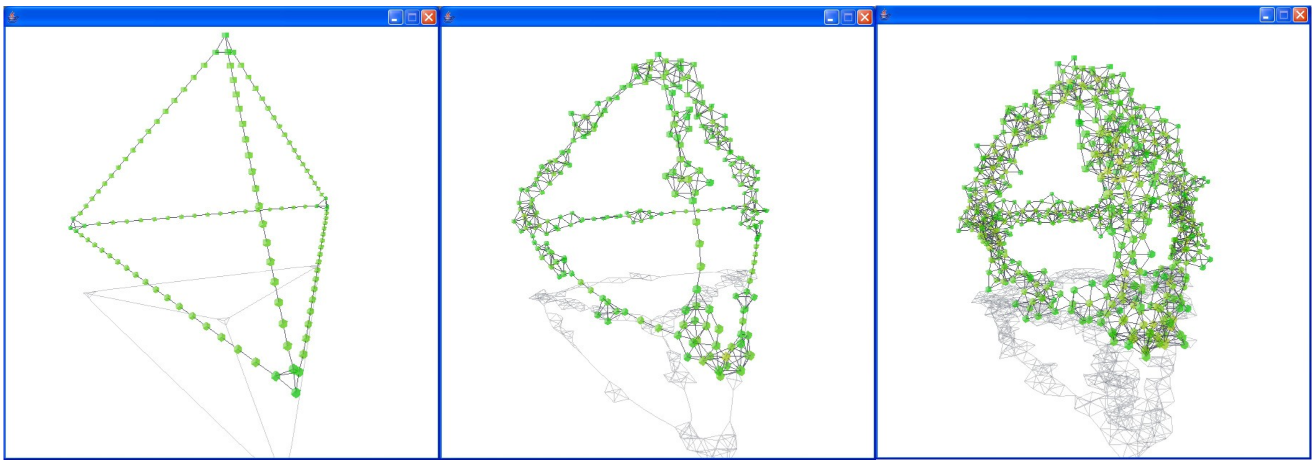
Additionally, each point knows to what other points it is connected to, so each time it is displayed, it also calculates distances to neighbours. If they're unequal (too close or too far), it works out a movement vector, which when applied to the point, equalizes the distances.



Static, attracting and repulsing,
three interactions between 'molecules'

Result of this simple operation repeated over time, is an equally distanced in space network of points, resembling structure of the foam. Nodes, connected to each other pass tensions and stresses, to other connected units, though distributing anomalies across entire structure. What is interesting here, is that the system comes up with efficient solutions by itself, without pre-conceptualizing them, for example structure consists mostly of regular tetrahedrons, which are the most rigid of all primitive solids.

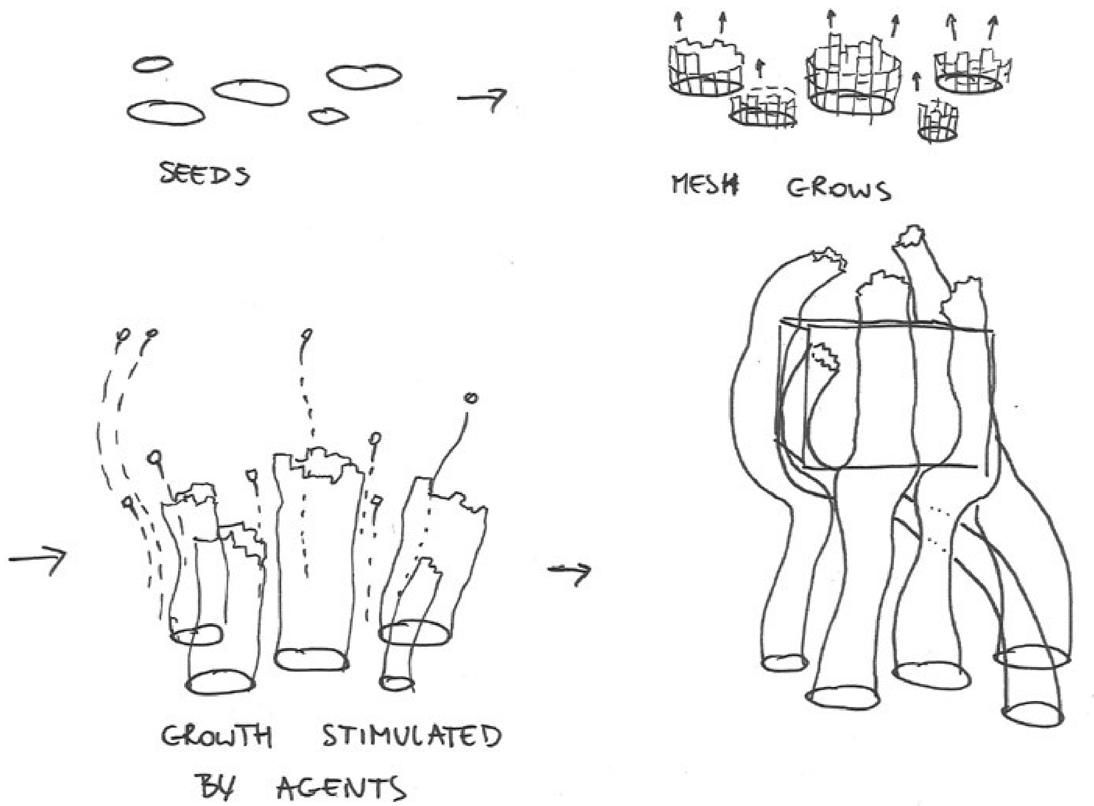
Idea may be applied to any pre-existing seed, upon which new structure is built (crystallised). At image presented below, seed was prepared as regular tetrahedron made with 120 nodes. In actual simulation, as in previous example, positions of new nodes were selected randomly.



Experiment with this technique proved very useful with defining behaviour of the system based on interconnected nodes, especially when it comes to distribution of forces. As a first step it provided good framework for further experiments, and in reality entire 'virtual playground' remained almost unchanged during rest of the work, only ways of adding and connecting nodes have been modified.

7.2. Growing supporting structure

New direction involved growing a structure that supports a building. Idea is, to accept some initial input data (seeds and position of blocks/volumes to be supported), and produce output as a 3d model, showing geometry and topology of appropriate structure, which will be specifically adapted to this particular situation. As 'adapted' I understand well organized, or actually 'self-organized' with use of material. There was no detailed vision of how it will look in result, only a clear question and expectation that system will provide an interesting answer.



proposed strategy of simulating growth

At this moment, realising what algorithms are available, system is going to assemble a piece of software generating supporting structure for volumes enclosed in cubes. To do that, it needs :

-data structure (node positions plus connections to other nodes),
-positions of cubes
-seed initiation procedure,
-algorithm with dynamic distribution of forces (for data spatial self organisation, presented above),
-growth procedure, and a
-'driving force' showing the direction of growth, which comprises an agent based system.

All those elements are components of the simulation, simply assembled together, or 'designed' in no other way to those used in ordinary design process, whether architectural or mechanical.

Data structure

It's a simple array of encapsulated data, consisting of : node's Cartesian coordinates, numbers of points to which the node is connected, and few other variables. Class structure also encloses procedures for displaying nodes.

Seed

After executing the program, a 'seed' image is read from file, thus determining initial conditions. Black fields on the ground, or rather boundaries between them and white area, are being used as starting points for nodes growth. Special procedure searches the image, and places first nodes in space basing on the position of the seed.

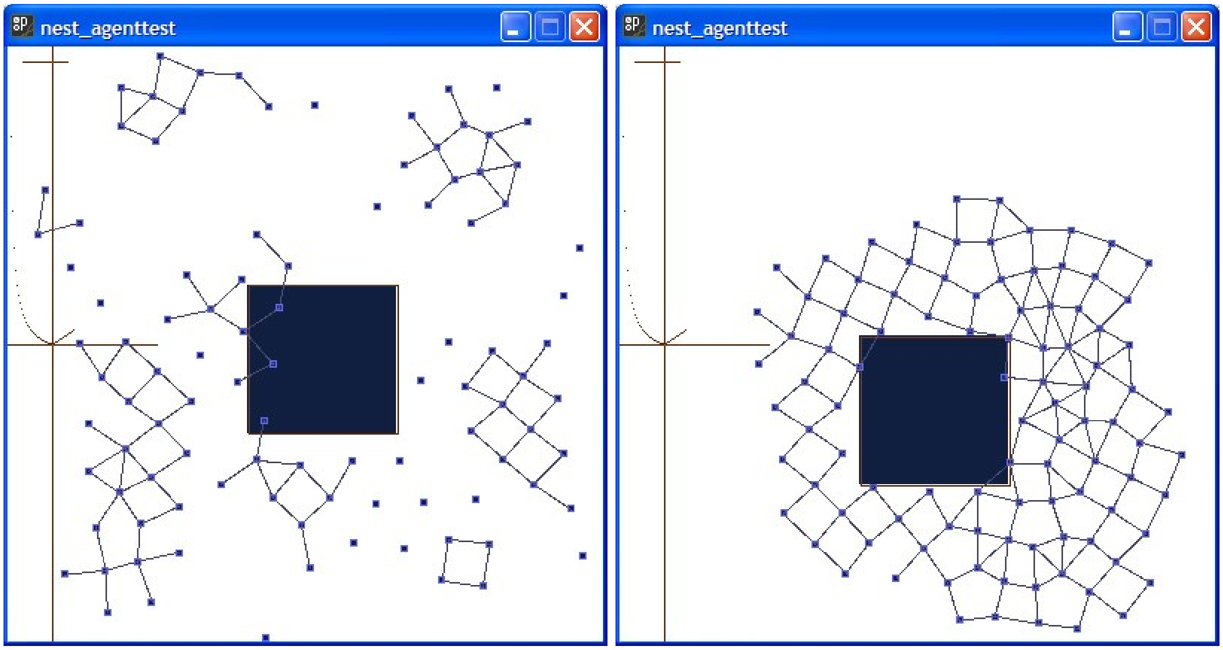
Dynamic distribution of forces

Dynamic distribution of forces (similar to FEM- finite element analysis) equalises distances between points. It, simply put, treats surrounding points to which calculated node is connected, as vectors. Adding all of them to current node position (where distance is proportional to attraction or repulsion strength), results in such position, that all forces are being equalised. Here goes the pseudo-code :

```
for each point i
{
 v = 0;                                                   // initial vector of resulting forces
 find number of connected points j
   for each connected point j
   {
     v1 = point[ i ].position - point[ j ].position;  // calculate vector between two points
     d  = length ( v1 );
    if  ( d<threshold )                                  // if distance is below threshold
         {
                v = v + v1* (threshold/d – 1)       // add vector to repel points
         }
    if  ( d>threshold) and (d<range)                 // if distance is greater than threshold
         {
                v = v - v1* (d-threshold) / 10       // subtract vector to attract points
         }

   }
   point[ i ].position += v;                            //finally, add vector sum to point's position
}
```

This procedure, shows in greatly simplified way, that it is not necessary to know physics, nor include complex mechanics of molecular interactions, to construct a model built of equally spaced cells, which maintain themselves without our guidance. To deepen understanding of this technique, special program was written to analyse relations between nodes. It was very helpful to have it first in simpler environment (two-dimensional instead of three-dimensional), as fine-tuning of parameters, setting proper threshold of forces is very important in any self-organizing system – otherwise equilibrium that arises may be unstable or may not form properly at all.



Nodes, or molecules, exchanging forces. System equalizes the energy state aggregating them in one mass after few seconds (through attraction), but still maintaining minimal distances between (through repulsion).

On image above we see randomly placed nodes (not built upon the seed), floating around the square (which also repulses them), who organize themselves in spatial structure following simple rules.

Illustration of strengths of forces acting between nodes is shown on following diagram.

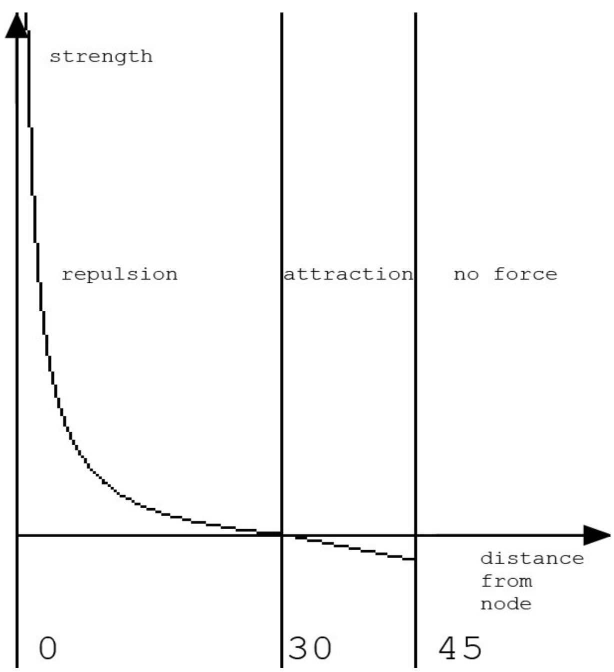
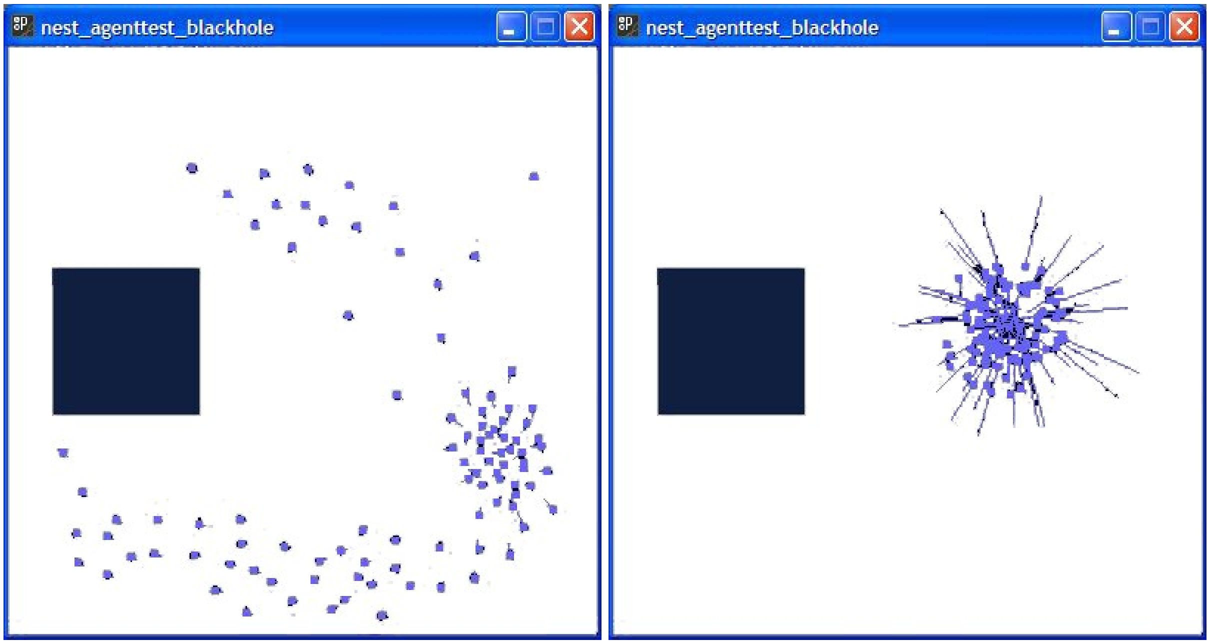


Diagram of magnitude of forces between nodes.
Horizontal axis shows distance between the nodes, vertical one strength of force. If nodes are from 0 to 30 units apart, they are being repelled from each other, with force decreasing with distance (they repulse with strongest power when very close). When distance is equal 30 units, nothing happens. When it rises to 45, nodes attract each other with power proportional to the increasing distance. When it gets greater, bond breaks and node is set free.

During this experimentation few surprising behaviours have been spotted, for example when attraction forces were too strong, nodes aggregated in few 'buzzing' swarms, which 'sucked' other free nodes inside, then joining with the other ones, and finally collapsing inside, performing it in very similar manner as black hole.

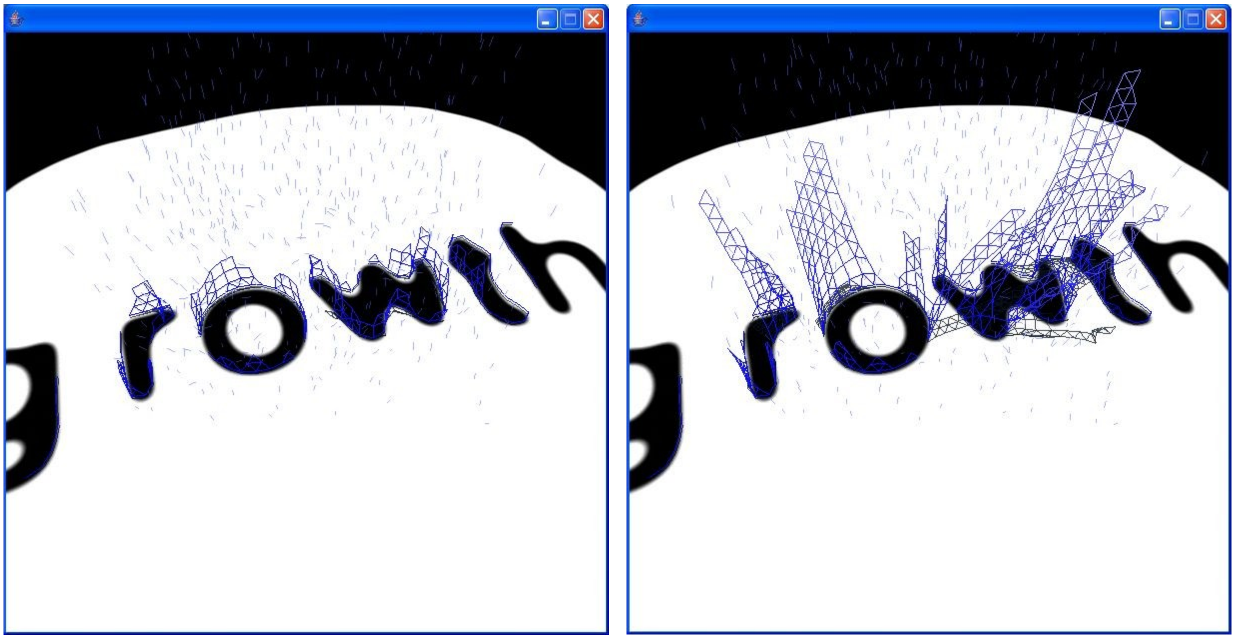


Unplanned formation of 'black hole'. Agents start gathering, thus creating huge magnet for other agents (left), next all of them stick together and strive to get out (right).

That shows, that it is essential to fine-tune values of the forces, thresholds and other rules, as otherwise unexpected things may occur. Also testing parts of an algorithm in different environment is necessary. Here, attractions and repulsions were tested in two-dimensional world, and after proving successful were plugged into three-dimensional mechanism.

Growth procedure

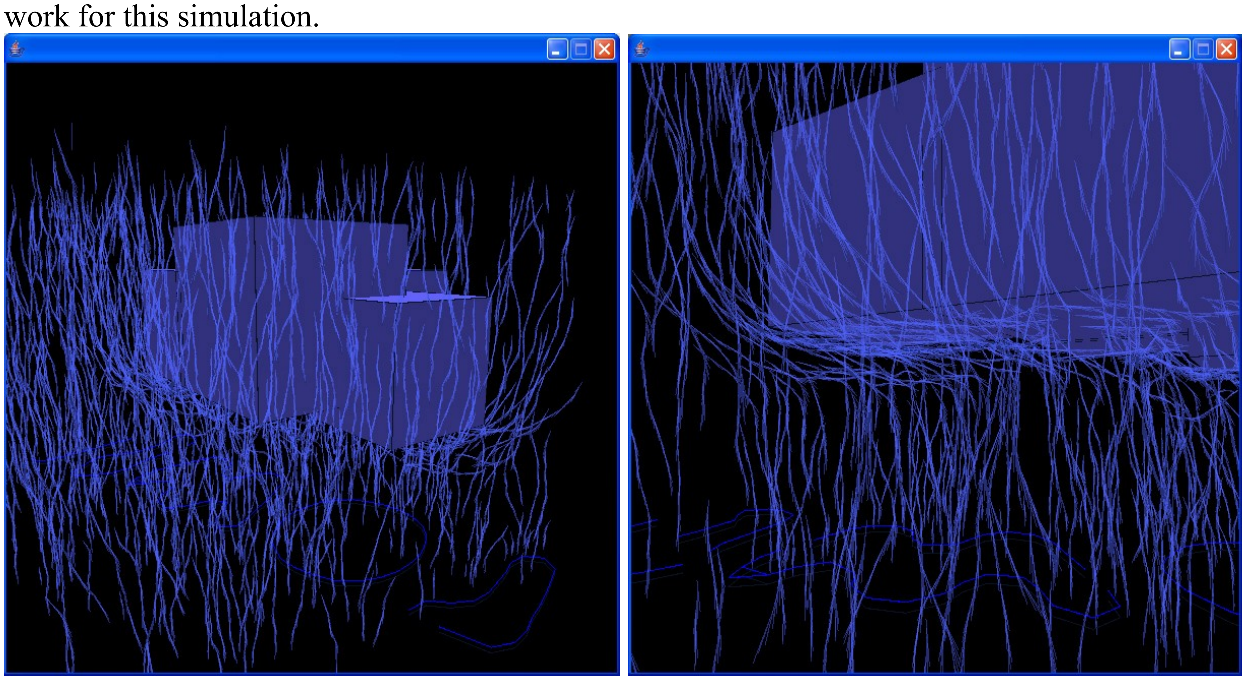
Having first few nodes set up upon the 'seed' as an initial structure, a sort of 'stem cells' of our construct, we can start adding new nodes, and thus growing the proper construction. The procedure consists of few steps. For each frame of simulation it checks all nodes, determining number of connections to other nodes. If this number is smaller than threshold (4 in this case), then it assumes that there may be not enough support for structure in this position, so another node should be added nearby. Then, it searches space around examined node and finds best position to fit a new one. It has to fulfil few requirements, for example it must not be too close to other nodes, nor too far from them, and it has to find connections to at least two nearby points. Therefore, with itself and three other points (1+3) it forms a tetrahedron, with two (1+2) a triangle etc.
Those steps are repeated over time, resulting in new nodes being added and connected with existing structure. It's significant to mention, that in most versions of this algorithm, positions of the nodes were determined using current positions of existing elements, and also vectors given by 'driving force'.



Structure grown upon a seed

Driving force

As each organic structure has some forces that drive its growth (branches of trees grow towards light stimulated through photosynthesis, roots grow towards water sources, stimulated by dampness), this structure has to have one too. In this case, solution that was selected doesn't actually resemble any natural force, but is based on some principles of hot air movement, so is related to fluid dynamics. Simply put, it is an agent based system, where agents move vertically upwards, avoiding obstacles, and thus forming paths of possible growth directions, which 'embrace' avoided object. This force is inspired by natural world, but was specially adapted to work for this simulation.

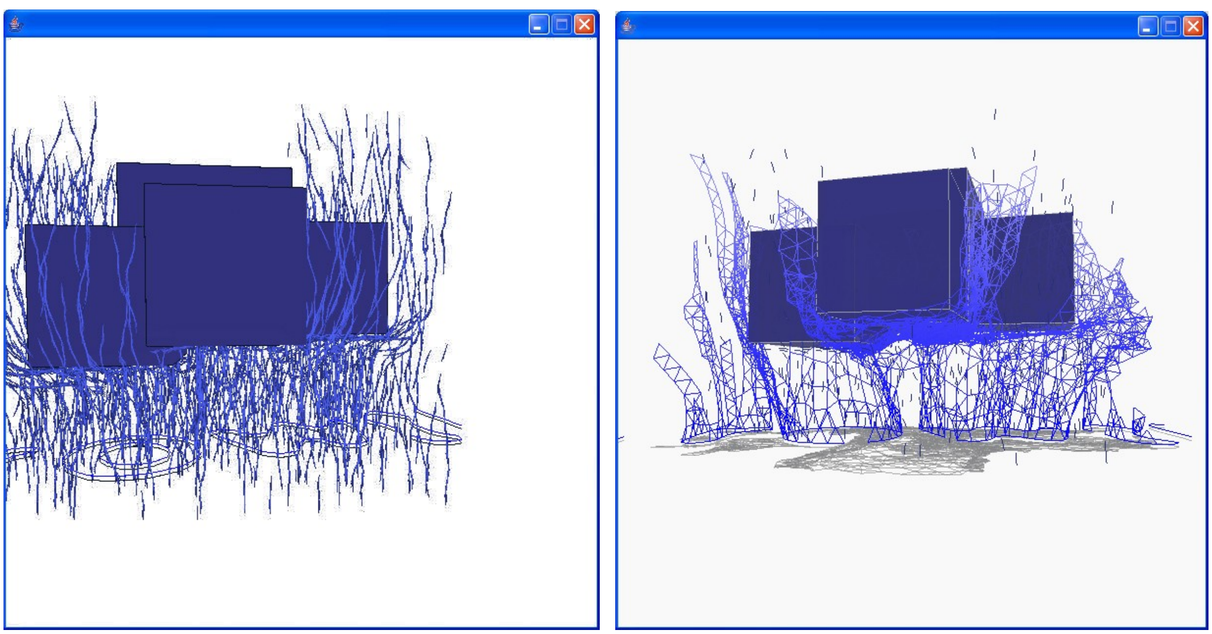


trails of agents moving, stimulating growth vertically

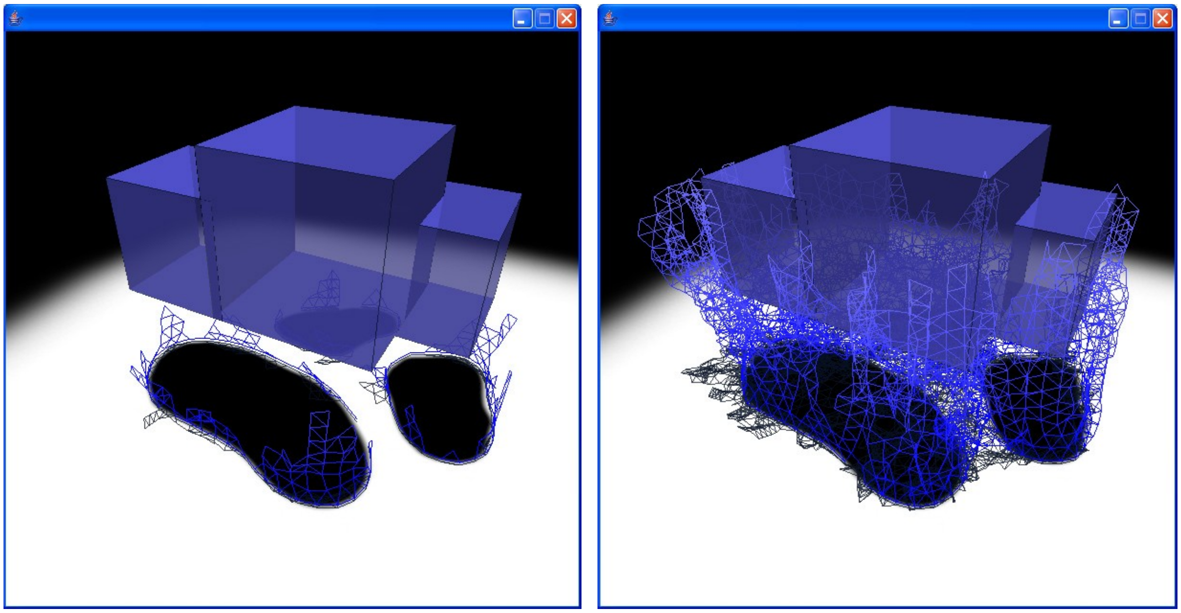
When growth procedure is connected with a 'driving force' algorithm, which in reality is an agent based system, we get a direction-driven growth, another words, a growth that follows trails of agents, 'embracing' preprogrammed shapes (in this case cubes). Each agent passing nearby the node, 'drags' it for a while, changing its position, thus stretching entire structure. Nodes counter-react to this, reorganizing themselves, and overall shape changes gradually.
Second important dependency is that new nodes are added to existing ones, by moving cursor from specified node following direction of growth. Through that, they literally follow the agents determining this direction. As agents avoid obstacles on their way, structure does it too.

This juxtaposition of two techniques wasn't selected in relation to specific behaviour of carefully studied organic creatures, but simply because it works similarly to some natural phenomena and it gives the results that satisfy the needs of this project (directional growth, avoiding the boxes). In this case it could be compared to 'sketching' with algorithms, which would be directly related to what Paul Graham writes in his essay 'Hackers and Painters' [10], comparing coding (programming) to painting (with code). In the same way designers may paint with algorithms.

The outcome of the program would look like this :



Trails of agents (left) stimulate growth (right)



Direction driven growth

Technique may be used in many variations, mainly regarding usage of different seeds and various configurations of boxes. Apart from that some variables may be adjusted, but this causes only minor differences (for example attraction-repulsion threshold, which influences size of basic unit).

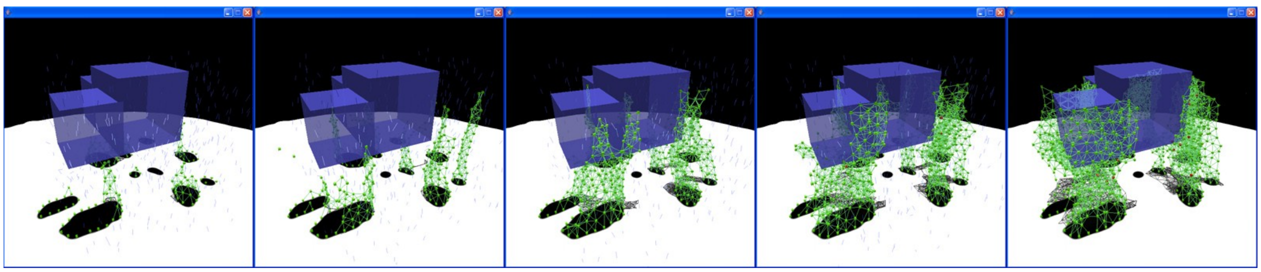
Very powerful addition could comprise a special tool for generating the seed itself (in presented examples, it is hand made). Mechanism could be established to vary its location, size and shape according to the positioning of cubes – and that would probably be the best way to fine-tune system to produce best performance. It could also be breed by genetic algorithm, in which fitness function would assess the level of compatibility between seed and boxes. I presume, that after few hundred generations we could witness 'birth' of absolutely innovatory properties in those constructions, or perhaps program could invent completely new arrays of shapes that we'd never expect to arise here.

Generally speaking, machines can surprise us, but we have to remember that whatever outcome we get from it, mechanisms that created it were made by humans. That's why it is still up to us, not artificial intelligence, how new generative architecture will look and perform.

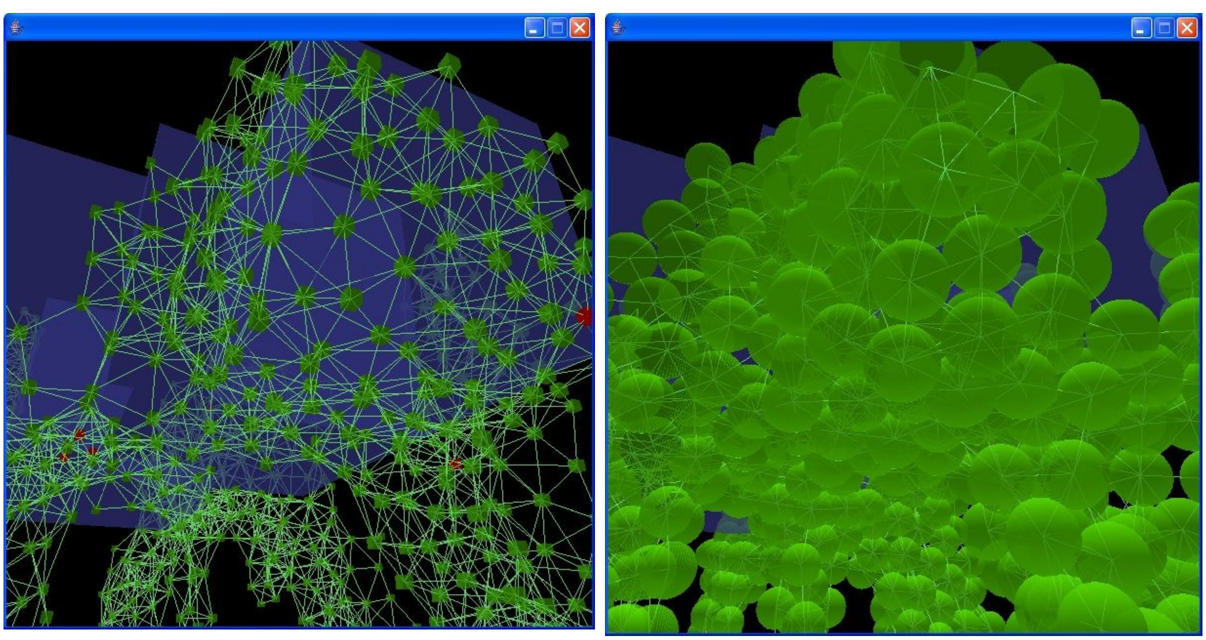
7.3. Further development – repositioning the nodes.

Program in this shape produces quite interesting and also, inspiring results, however after many experimentations it was made clear that few things need improvement. What seemed most limiting for this kind of emerging structure was that nodes after being placed in space were constantly connected to the same neighbouring nodes. However, as the structure was inspired by foam, they should be made free to reposition themselves in the places where they're needed more. For example, when a situation occurs, that we have very close packing of elements, and they even pack more through multiple attraction forces, strong compression happens, sometimes even causing 'the black hole' effect in the model. Therefore, it was greatly needed that the elements change positions releasing the tensions, and strengthening the construct where it needed strength.

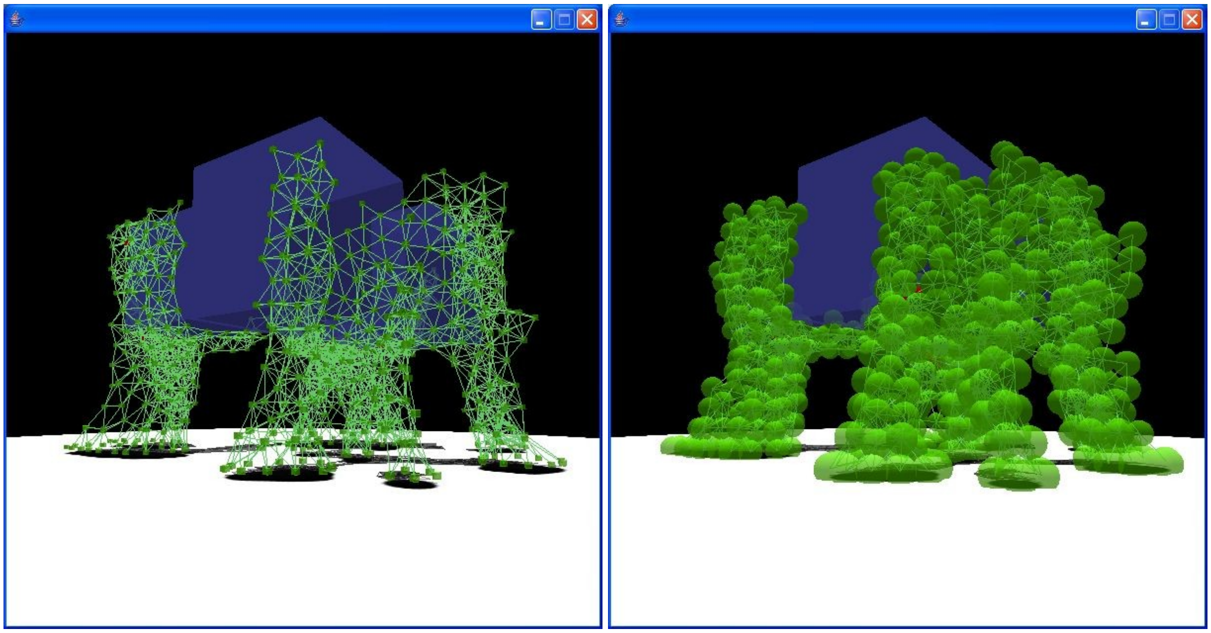
Two important changes were introduced. One releasing them from fixed connection idea, instead relying on dynamic interactions (like in molecules of physical substances, for example water). Molecules don't allow too close packing, and 'catch' other molecules in distance, though 'gluing' construction together and keeping it in one piece. Second change involved removing node from the model, when it's too closely or too loosely packed with others, and putting it into other parts of it (often those, that need more elements).



After that was introduced, generated models were more compact and natural. Interestingly, even after growth process was stopped, and simulation only kept distributing forces, nodes still disappeared and reappeared in different places, though making irresistible impression of being the real soapy foam.



Foam-like structure during simulations, nodes disappear and reappear in different places



Constructions embrace boxes. Spheres visualise equal distances between 'molecules'

This kind of modelling of physical or organic phenomena always gives unexpected results, as if the system wants to show us the best possible solution for formulated problem. We have to remember though, that to get a good answer, we have to articulate the question very clearly.

7.4. Gravity

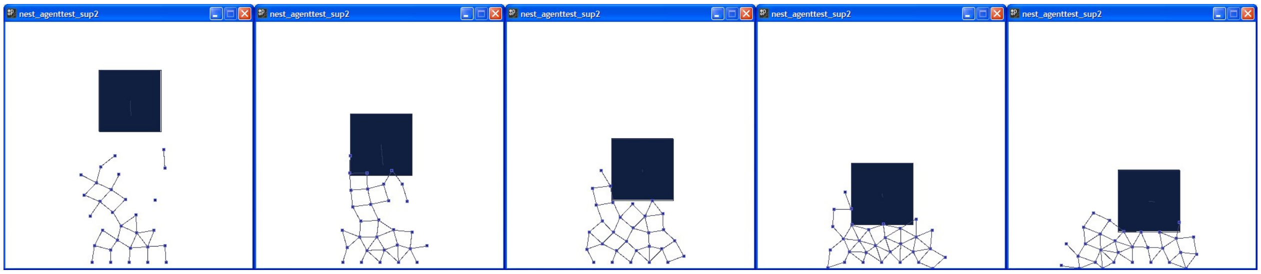
Finally, system has to create real construction, so it should embody gravitational force. Simulations shown before embodied stresses and tensions occurring within structure, which prevented model from disintegration caused by agent movements (agent drag nodes with them) and reaction to blocks, which repelled them.

To make the structure more convincing, gravitational force will be added as another movement vector to each node. Each node will bear load of itself, therefore a small vertical vector will be summed with it first, before other vectors are added to it (from stress and compression forces). Sequence calculated each frame for each element will look like this:

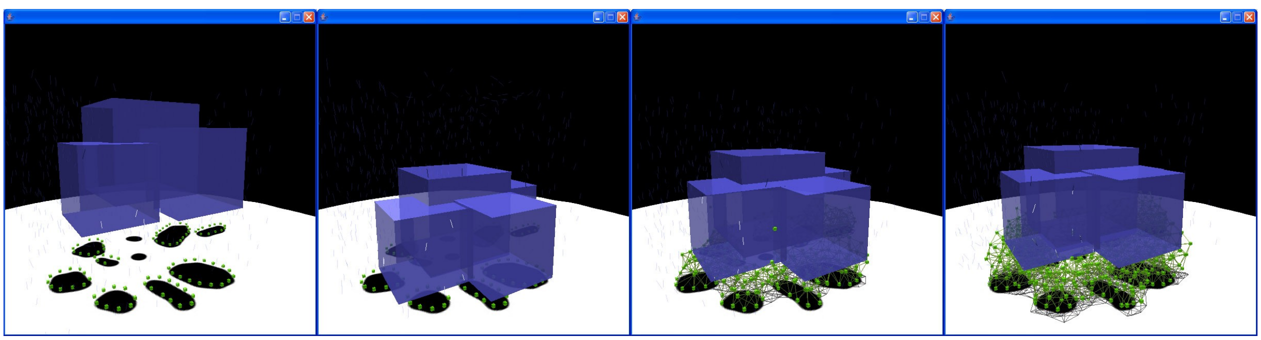
```
for each node j
{
        v = 0;                          //vector of movement for the node
        v = v + weight of the node itself; // drive node down with its weight
        v = v + internal forces;  //add other forces from surrounding nodes
    if (touches)
        v = v + weight of the carried load; // add weight of the box if they touch
        position = position + v;   // finally, move the node
}
```

We see that at the end of the sequence, weight of the load is added as another vector, which is the

most important operation.
Again, special program was written to perform simulation with gravity. Load (box) was set in the air and released, falling on the construction. On following image subsequent frames show construction's reaction. It was compressed to a certain degree, but then repulsion forces reacted to the weight and didn't allow it to be crushed, keeping the box above the ground.



After introducing this idea to 3d environment, it became apparent that surprisingly, it is not important whether we drop the load at growing structure, or just leave it on the ground and let the structure arise from beneath it. In both cases boxes got lifted to certain height by the power of emerging cells. On illustration below we see phases of this happening, starting from blocks falling at the floor (first image shows them in original position). Then we see them being lifted gradually.



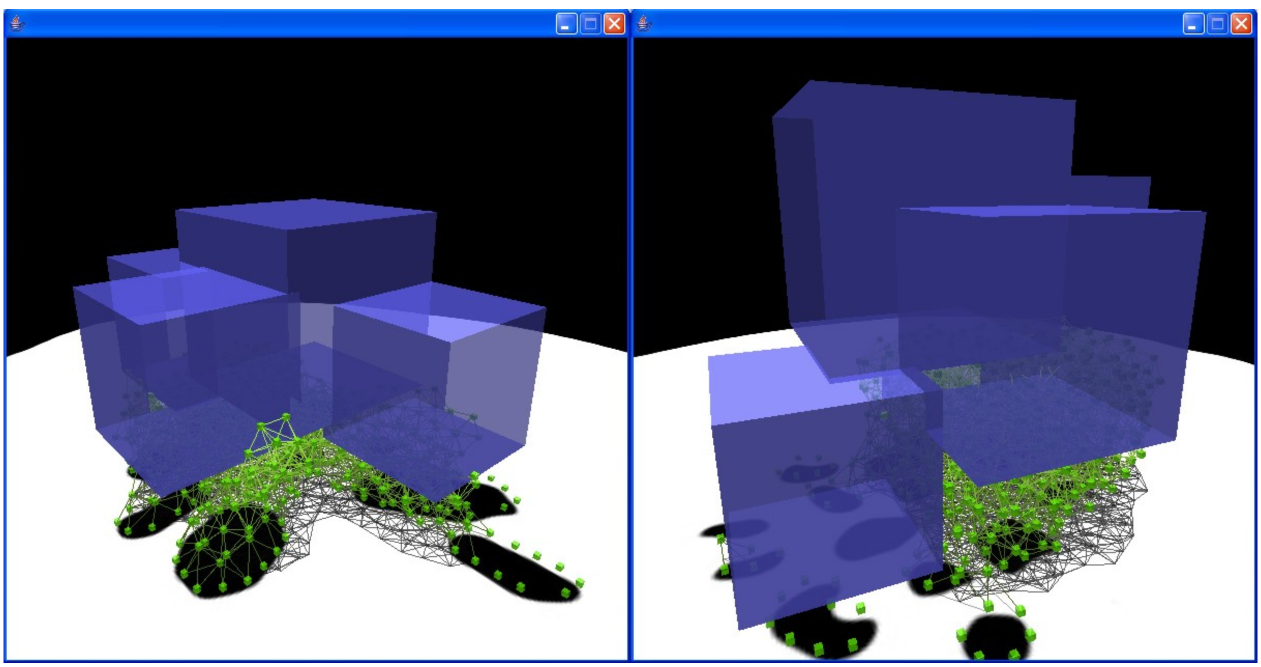
The features introduced in this version caused changes, which made the construction look very different from when just pure growth algorithm was used. Most striking one is that the construction doesn't embrace boxes any more. In fact, embracing doesn't contribute to ability to bear the load , so 'arms' that grew before around boxes were removed by algorithm automatically as they didn't experience any stresses. In this form, program moves elements from idle positions to where they can help by transferring most of the stress.

Using built-in console (commands steering the program), every simulation was run in stages. First, agents producing 'driving force' are settling down from chaos (random vectors) to directional movements that embrace the blocks. After few seconds, when they're ready, growth procedure is started and elements start appearing. During many attempts, it was made clear that its better to run growth procedure for a while, then wait a moment for nodes to settle down, then run growth again and so on. This way, simply makes the model more coherent, whereas when growth runs rapidly and adds elements too quickly for too long, structure expands too much and forms many unnecessary branches.

After structure finishes growing (because it reaches maximum node number for example), its good to leave simulation for few minutes, turning off driving force (agents), and wait. During that time, program picks idle elements, those that don't perform any structural task, don't transfer any force. After removing each of those, structure distributes other nodes evenly, compensating the change. Removed node is added in another location, that needs support, and thus optimisation takes place step by step, and even though nothing is growing any more, structure keeps reorganising itself.

7.5. Testing the solutions – strong and weak points.

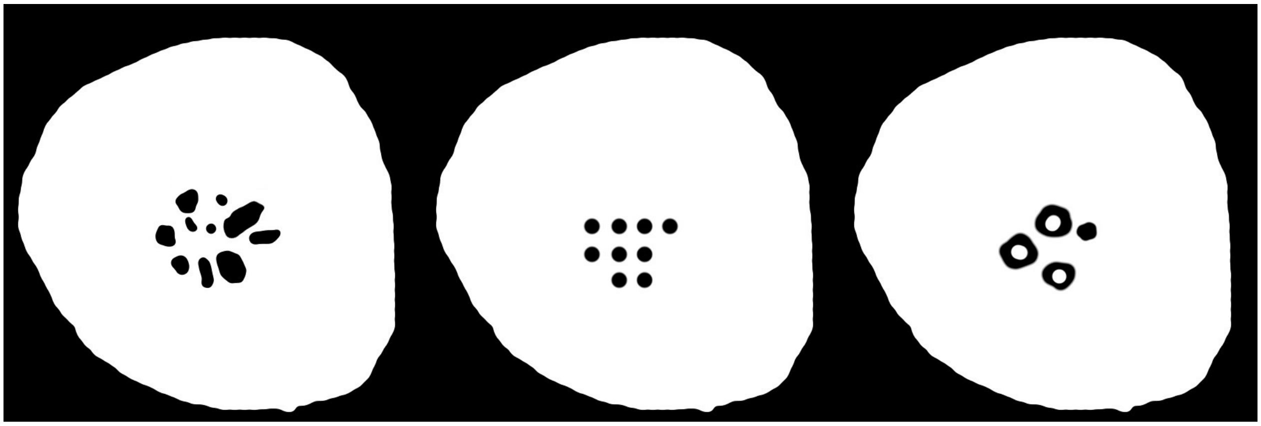
Sometimes, seed wasn't well fit to host construction able to hold boxes. When parts of it were too far from the blocks, nodes were separated from it, dragged away and moved to become part of bigger bulk in the middle. It was caused mainly by agents dragging them too fiercely or by lack of sufficient number of surrounding nodes, that could keep it together. That can be observed on image below. Model on left picture lacks nodes connected to right side of the seed. When it starts happening, other, single nodes break connections as well, as they don't have any others to connect to. Therefore, some parts of the seed become abandoned and thus idle. Most of the times this is a result of efficient elimination of the elements that don't take part in distribution of forces, because they are too far from supported box (left image). Sometimes, very rarely, this happens as a result of incorrectly interpreted seed (image on the right). Biggest field beneath the lowest box wasn't populated with nodes at the beginning, thus box lacked enough support and fell on the ground.



Parts of the seed become idle (left). Lack of support causes box to fall down(right).

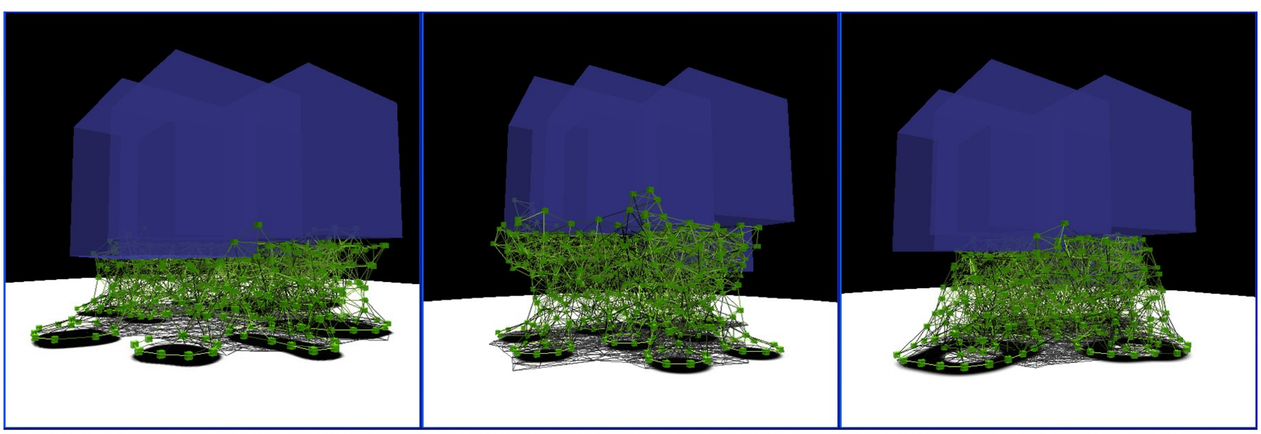
To understand better strong and weak points of presented algorithms, I decided to test it in three different environments. Three different seeds were chosen, and program was enriched with data-gathering functions for further analysis. Number of properties will be analysed :

–amount of overstressed joints
–number of nodes comprising a seed (firmly attached to black fields)
–efficiency of the seed (%), showing how many seed nodes are used by the structure
–approximate height of the blocks, as they're lifted differently depending on the seed
–number of unique connections (linear construction elements) between nodes



seed A seed B seed C

For each seed there were three separate sessions, from which results were approximated. Number of nodes was constant during all of them, to provide better comparison in all situations. Growth stopped automatically after this number reached 600, and all measurements were taken around three minutes after beginning of each simulation.



seed A seed B seed C

Following table shows results of those measurements.

	Seed A			Seed B			Seed C		
Session	1	2	3	1	2	3	1	2	3
Average box height	31	30,5	32,89	43,5	42,03	41,44	37	38,7	40,43
% of overstressed members	7,45	7,68	7,6	6,45	6,58	6,84	7,8	7,24	7,45
Nodes attached to seed	104	105	104	72	72	72	88	85	82
Seed efficiency (%)	91	90,47	87,5	95,8	97,7	100	100	98,82	97,5
Number of unique connections	3256	3233	3191	3261	3217	3353	3424	3389	3362

Which after approximation looks as follows:

	Seed A	Seed B	Seed C
Average box height	31,46	42,32	38,71
% of overstressed members	7,58	6,62	7,5
Nodes attached to seed	104,33	72	85
Seed efficiency (%)	89,66	97,83	98,77
Number of unique connections	3226,67	3277	3391,67

7.6. Conclusions

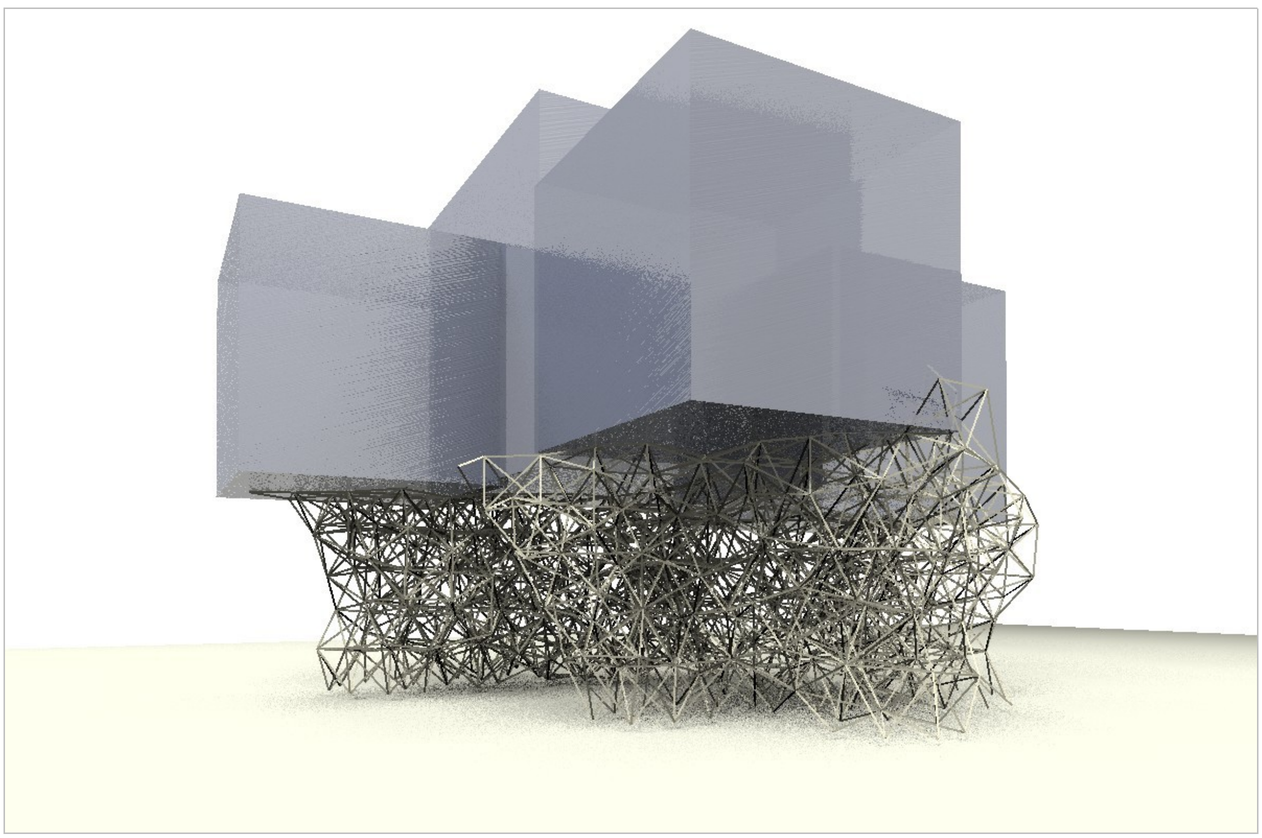
A number of interesting things could be observed. First, most visible one is that height of boxes is directly linked to number of nodes attached to the seed (104,72 and 85). The more nodes pinned to the ground as part of the seed, the lower the structure is. Answer to that dependency lies in fact that volume of this structure is constant, and that number of nodes reflects the area of the seed. (the bigger the seed, the more nodes at ground level). Area multiplied by the height gives volume, so if we want tall structure, we have to use tight seed, if we want a lower one, and thus more stable, we'll choose a larger seed.
The lowest model was built on seed A (average height 31,46) – number of seed nodes is 104, the tallest one (42,32, seed B) has only 72 .
Another visible thing, connected to the latter, is seed efficiency. If the seed is spread too far, nodes detach from it, and the value goes down. That is why model with seed A has efficiency of only 89,66 whereas B and C have 97,83 and 98,77 respectively. This is a very good indication that seed is not appropriate. This also means, that manipulation of height with size of the seed has its limits.
Number of unique connections (rigid profiles forming small beams) stays at the same level, most of the time, probably because it depends only on number of nodes, and this number is constant (600).
Percentage of overstressed members is a good information, however hard to use in practice. According to the table, it has some relation to the height of the structure (in the tallest one, seed B it is the lowest among all – 6,62%), but it may be hard to prove, because other models keep it on

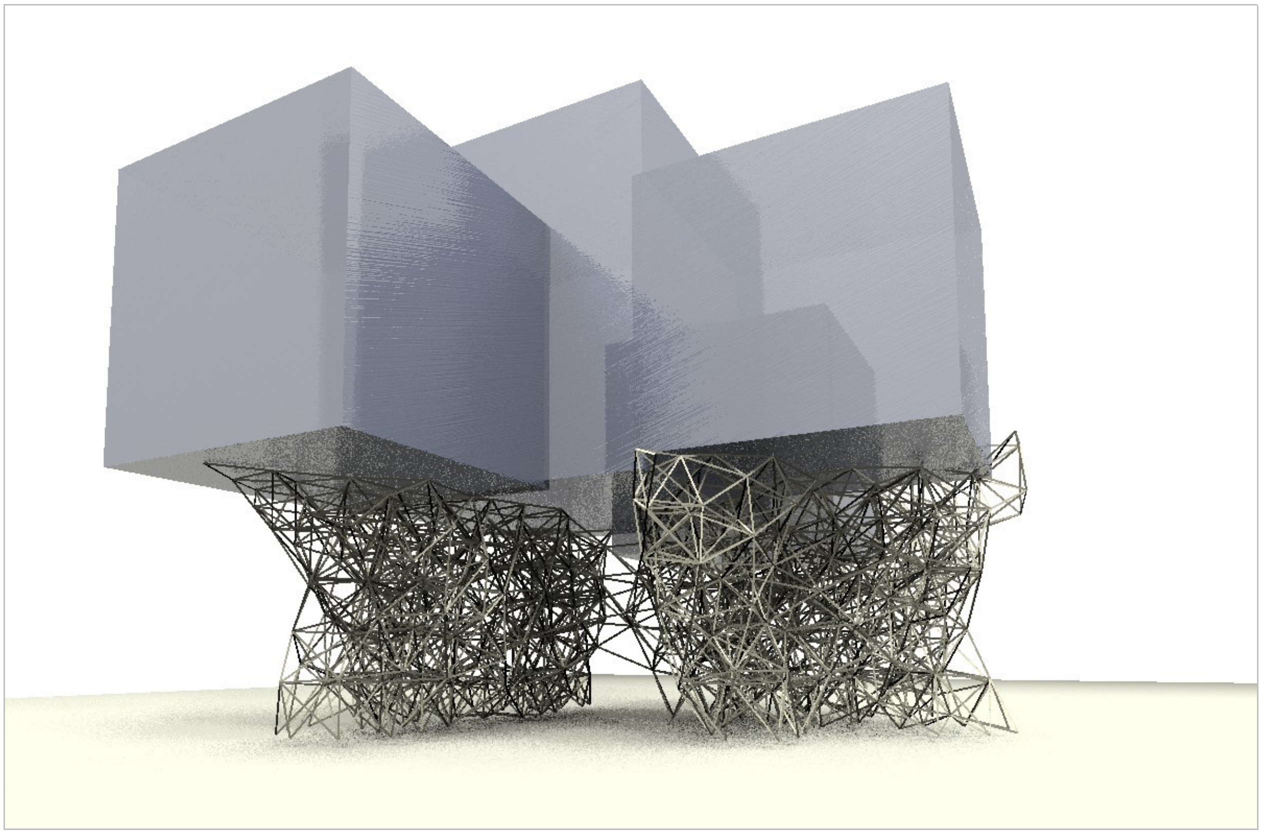
the same level despite of varying height. It is supposed to show how many of elements are stressed of more than 10%, so it will be useful in further structural analysis.

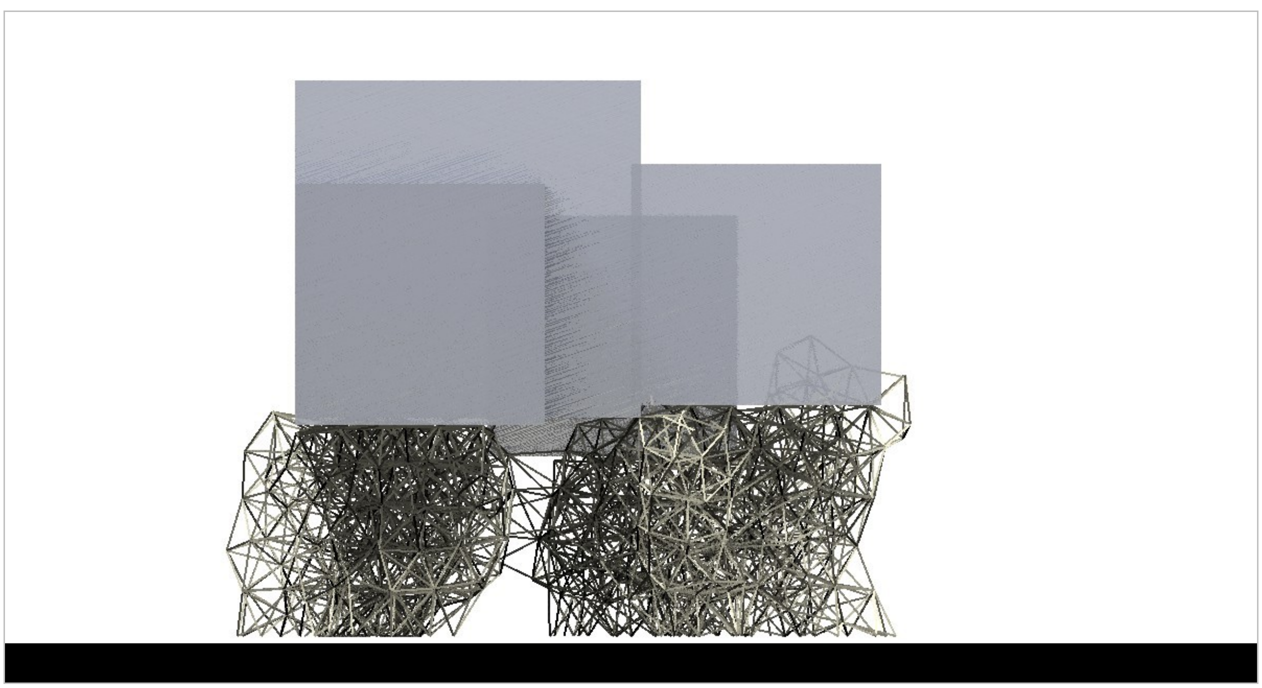
All above properties have been analysed during the simulation, and thus are describing a living model. However, when it comes to physical forces, it is important to analyse them 'on the fly', and use data to form the model, creating feedback loop. Model will be changed, analysed again, and changed according to numerical statistics of forces within it. This process may be repeated endlessly, allowing structure to be as efficient as possible.

7.7. 3D model.

Geometry was exported from Processing application to Generative Components, part of Bentley's Microstation, where it was modelled and rendered. Now it can be digitally fabricated and assembled in real world.

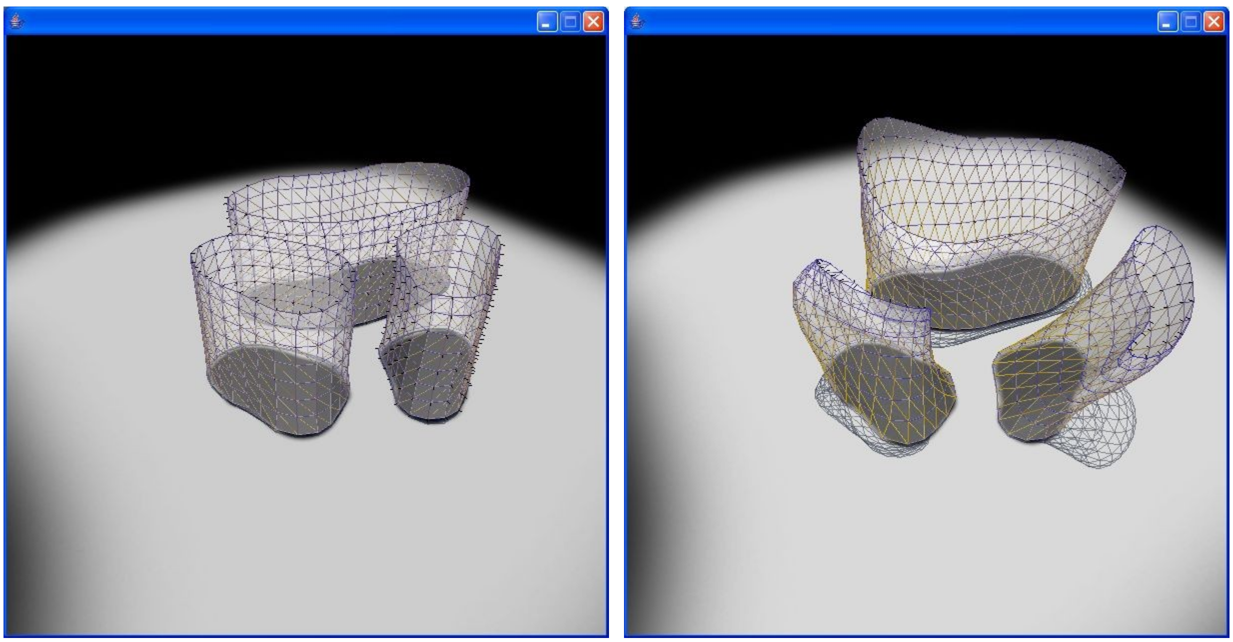






8. Quick look at non-developmental models

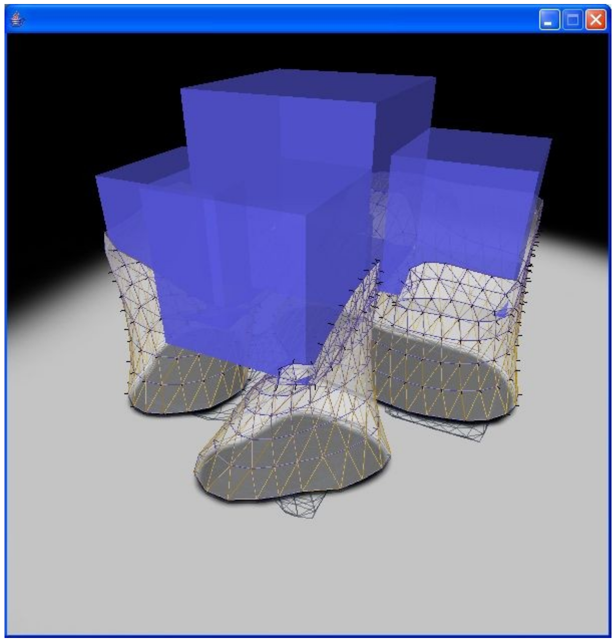
Even though principles of generative approach to growth subject have been explained, it is still worth presenting what's happening on the other side of the barricade, which is a non-generative approach. The main difference is that model doesn't grow, but is inserted with predefined topology (connections between the nodes). We still can have a seed, but instead of generating few initial elements and then adding others during simulation, we will generate everything before simulation runs. That defines all the connections at the beginning, and only thing that changes afterwards is the geometry.



Pre-calculated input stretched by virtual forces

8.1. Pre-defined construction

This time, seed was used to generate loops enclosing black areas, which then were lifted up and repeated few times, forming 'chimneys'. Produced mesh seems to be far simpler than previous model, but it has few advantages. Most importantly it is well ordered, structure of connections is the same for each node, and this fact helps with determining and using some important properties. Also a good thing here is that actually, they are surfaces, meaning that they have a continuity, and they have normals (vectors perpendicular to the surface).

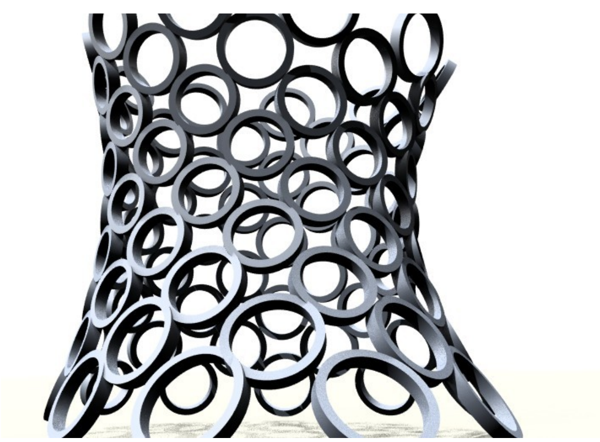


Surfaces wrap around boxes, adaptation driven by agents

Using previously defined 'driving force' we may make the structure wrap around the cubes. Construction shown above wasn't then grown, but rather formed, or 'sculpted' by hundreds of little agents moving upwards in resemblance of hot air movement, and thus driving force. The closest thing, this 'driving force' could be compared to is probably phototropism, where each tube would act similarly to a branch of a tree looking for resources for a photosynthesis. In nature however, we don't have trees that voluntarily look forward to supporting heavy boxes, so it's clear that what we're dealing with here is adaptation of natural mechanism to non-natural human needs. But that is what these simulations are for.

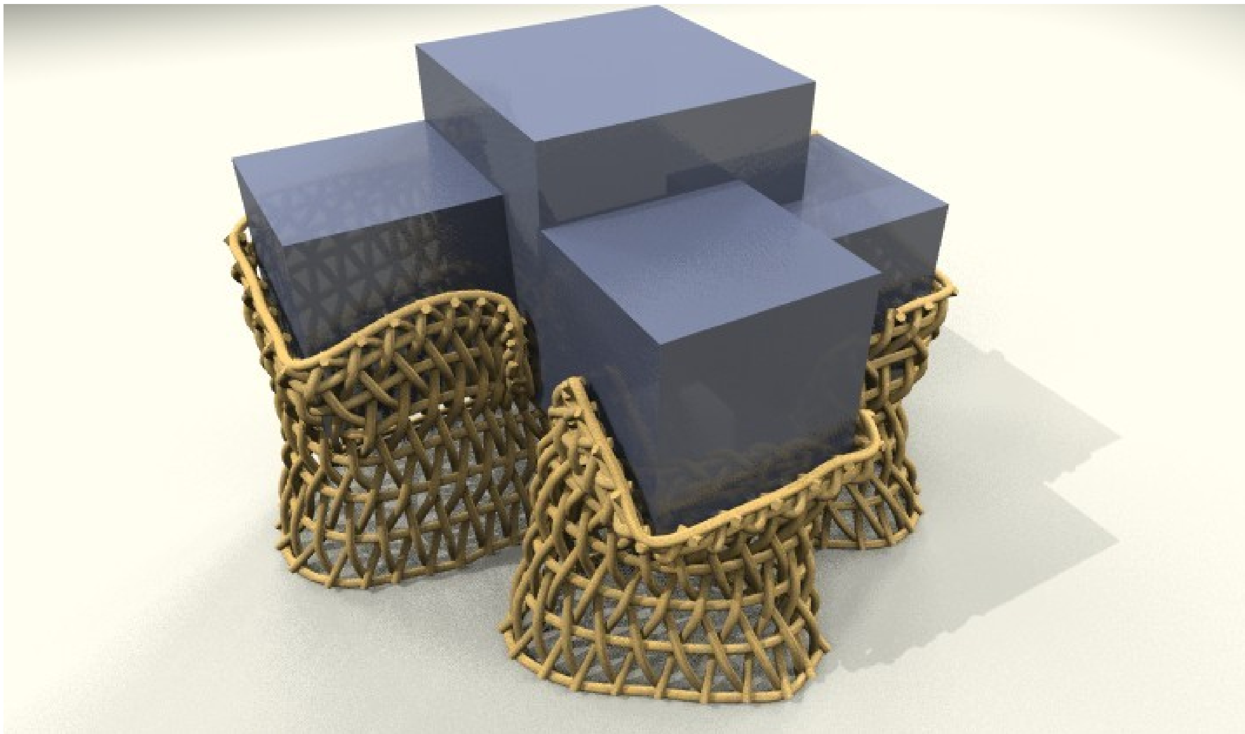
8.2. Creating 3D model.

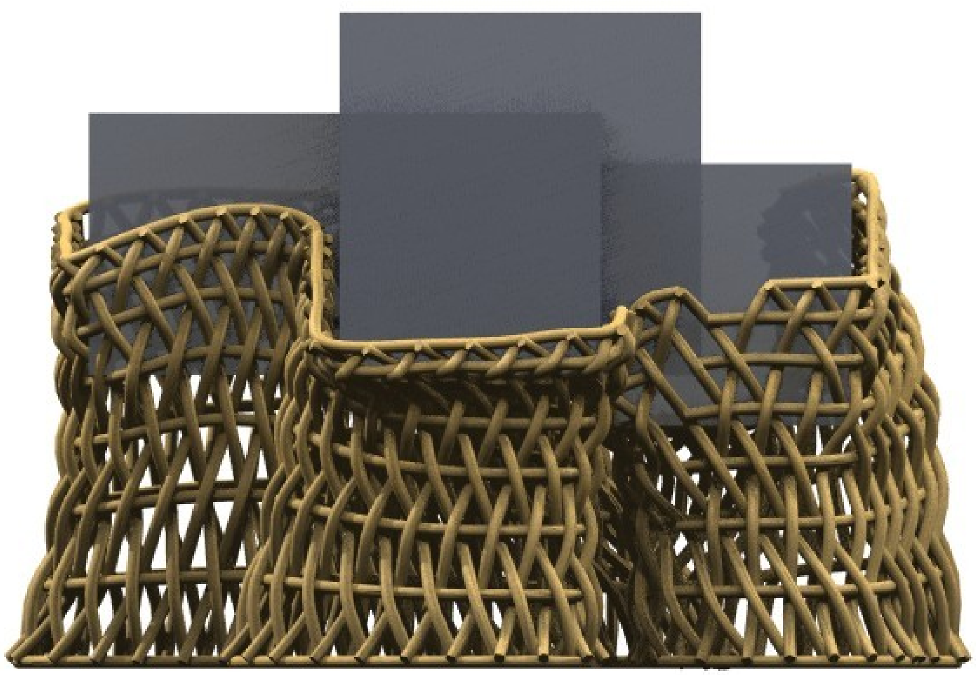
This is how topology and geometry of construction was generated. Now data can be exported to 3d modelling applications, where it can be visualised or fabricated. Most of CAD modelling was done with use of Generative Components and Microstation software. GC offers good scripting capabilities, which are usually sufficient to model advanced geometry, but it is not able to provide a 'playground' for iterative and thus generative processes. That is why combination of *Processing* software with Generative Components may give huge flexibility and versatility for introduced ideas.



Testing ideas in CAD software (circular cell-like structure and a basket), GC scripts.

Below results produced by application written in Processing, and exported to Generative Components.





9. Summary

Presented simulations were dealing with 'inter-molecular' forces between elements, thus defining morphology of entire construction upon defined input and embodied creativity exhibited by algorithm. This way, however, is only one of many ways to perform such calculations. I believe, as mentioned before, that these are tools, that may be used in many different situations and environments, with different inputs and adjusted values. There are many other ways of creating this kind of emergence, that could use additional techniques, not presented here, like GA's (genetic algorithms), neural networks (Kohonen Self Organizing Maps – fascinating subject of investigation), cellular automatons and others.

Simulating forces is not that complex any more, as reality shows. Many software companies sell programs for structural calculations which is now very popular among constructors, as great simplification to, used-to-be difficult tasks. Due to technological evolution we have huge amount of resources and knowledge that allows us to solve complex problems almost instantly. Various software packages accessible over the net, thousands of free libraries with mathematical functions ready to use and serve us in whatever we need.

And as even most complicated computer procedures still consist of very simple components, going down to simple multiplications and divisions in processor's registry, all of those mechanisms built upon those simple operations should be easy to understand, when broken into smaller parts and 'digested' without hurry.
Therefore, it all becomes understandable for anyone, it's just a matter of proper attitude and commitment. However, as Paul Graham ('Hackers and Painters') said 'painter doesn't have to know how to make paint, he only has to know how to use it', and thus the approach, from which it comes out vividly, that all new techniques in computation are just ordinary tools.
These tools are no different in principles to pencils, calculators or rulers, because they have scope of application, input, and output. Knowing that, we can combine them together, creating new quality each time we run the program, still thinking about them as a help in design process, not being afraid of some spontaneous creativity exhibited by the algorithm itself.
It's not going to limit us as a designers, it's not going to take our freedom of decisions away. When used skilfully, it will bring a lot of inspirations and hints (not even hidden, but clearly shown in front of us), which will definitely improve overall quality of the design.

10. Conclusions

Constructions generated by simulations seem quite unreal or unnecessary to some people. It always was, and probably will be for some time, a challenging thing to convince those doubtful minds to why these are better. There always will be someone who will say 'why didn't you use simple grid of columns 9x9m, concrete floor slabs and straight envelope ?' or 'what are these blobs for'.

In fact, in the past, many times curved surfaces appeared in architectural projects only because someone invented software that allowed to model those. This resulted in formation of so-called 'blobby' architecture, which sometimes doesn't have too much in common with the term 'generative', and is only shaped in that way for decorative or purely aesthetic purposes.

What should appear in design strategy, is pure honesty. With use of those tools, all elements of the design will have defined purpose, not as detached things, but as a part of the whole, performing something meaningful in the chain of dependencies between elements of the project. When generative approach is assembled properly, and algorithms well crafted, any part of the outcome, whether geometry, partitioning of space or functionality of the solution, will always be easily explainable, or it may even explain itself.

I believe that in future algorithmic approach to design, inspired by, and harmonious with laws of nature will be one of the main aims in creation of good architecture. However, we must remember that architecture is considered to be an art, backed with technology and science. And in art, not everything can be explained, measured and quantified. Therefore, even though algorithms are a big calibre tool in creating good buildings and towns, they will never replace an architect.

11. Bibliography

[1] Regio Godfrey, 'Koyaanisqatsi' (Life out of balance), Miramax 1983

[2] Oshii Mamoru, 'Ghost in the shell' (Kôkaku kidôtai), 1995

[3] William Gibson, 'Neuromancer'

[4] Frei Otto, Bodo Rash, 'Finding Form: Towards an Architecture of the Minimal ', Axel Menges, 1996

[5] Thompson, D'Arcy 'On Growth and Form', Cambridge Univesity Press,

[6] an essay from: Lightness; the inevitable renaissance of minimum energy structures. A Beukers & E v Hinte, 010 Publishers, Rotterdam.

[7] Prusinkiewicz, P., Lindenmayer, A.: The algorithmic beauty of plants. Springer (1991) http://algorithmicbotany.org/

[8] Philip Ball, 'The self-made tapestry : pattern formation in nature', Oxford University Press 2004

[9] John C. Hart1, BrentBaker, Jeyprakash Michaelraj, 'Structural simulation of tree growth and response' , Springer-Verlag 2003, published online at http://graphics.cs.uiuc.edu/~jch/papers/structgrowth.pdf

[10] Integrating generative growth and evolutionary computaiton for form exploration
Paper submitted to Genetic Programming and Evolvable Machines special issue on Developmental System. Martin Hemberg, Una-May O'Reilly

[11] 'The generation of bone-like forms using analytic functions of a complex variable' Miss EAO Nsugbe and Dr. CJK Williams, School of Architecture & Civil Engineering, University of Bath'

[12] Hackers and Painters: Big Ideas from the Computer Age, Paul Graham, O'Reilly Media (May 2004)