Traversing digital matter states to fill irregular volumes

by

Harri Lewis

Supervised by

Prof. Paul Richens

Dr. Paul Shepherd

A thesis submitted for the degree of Master of Philosophy
University of Bath
Department of Architecture and Civil Engineering
October 2011

Acknowledgments

I would like to thank the following for their inspiration, guidance and support:

My supervisors, Paul Richens and Paul Shepherd

Adrian Bowyer and Chris Williams of the University of Bath

Tom Foley, Duncan Horswill and John Harding of Ramboll Computational Design

Jakob Lange of BIG architects

My parents, Rollo and Patricia

Abstract

This thesis explores the challenges faced by engineers when presented with a highly complex and fixed volume to support and fill with structure. This situation may arise in the design of free-form architecture, around existing buildings or within large-scale sculptures.

An interactive software application based on a particle simulation moving between matter states from gas to liquid to solid was developed. Combinations of simulated Brownian motion, inter-particle and environmental forces were used to imitate each state. Various algorithms, including a real-time adapted Delaunay tetrahedralisation and a recursive bounce routine, were implemented and used to create stable space frames that fit within any imported volume.

A novel technique called freeze-thaw optimisation is introduced and initial tests that produced a reduction in deflection for a range of space frame support conditions are presented.

The technology was successfully implemented as a case study to design a roof structure for the upcoming Tallinn Town Hall project by BIG architects and Ramboll UK.

This research has significance beyond structural space frame design, including finite-element meshing, optimised cellular or foam metals, bespoke furniture design and computer animation.

Keywords

generative engineering, verlet integration, adapted Delaunay tetrahedralisation, volume-filling, freeze-thaw optimisation, particle simulation.

Table of Contents

1. Introduction and background

The concept of an allowable volume in which to place structure will be familiar to architects and engineers. This paper explores a software-based solution to the situations where this volume is a highly complex form requiring support throughout. This introduction places this work in the context of current digital design techniques, form finding and engineers' responses to free-form architecture.

Digital tools in design

Software is now used throughout the design process in the vast majority of architectural and engineering projects. These digital tools range from computer-aided design (CAD) through finite-element analysis and digital fabrication packages, to bespoke applets created for specific project challenges. Comparable to tools used in any other creative process, these digital tools have an influence on the output. This can be a positive influence if it inspires the designer or allows the realisation of a complex form. However, if a designer becomes reliant on a certain tool they can become restricted by it, or start to use it in inappropriate ways. When using a CAD package a skilled user should be able to create any form they desire, however, some forms will be *easier* and *faster* to create than others. For example, using Non-Uniform Rational Basis Splines (also known as NURBS) it is easy to create an arbitrary freeform doubly curved surface with no understanding of how the software is generating the surface and, more importantly, no consideration of structural performance or fabrication. Although this may be liberating for the architect (and could be argued to have a value during early stages of experimentation) the lack of validation of its form can have a detrimental effect on the project when an engineer attempts to recreate the curves at the building scale. A better option might have been to use a tool that creates a developable surface (zero Gaussian curvature) or form finding a surface that minimises bending stresses when converted to a solid. It is important that designers understand the tools they are using, the limitations, where they are influencing the design and ideally a designer should be capable of creating their own tools.

There is a symbiotic relationship between digital tools and design ideas. To quote John Lassetter, Pixar founder and developer of Renderman software, "*The art challenges the technology, and the technology inspires the art*" (Orwall 1998). For example, the design concept of using paper as a form finding tool (used by Frank Gehry) required the development of Digital Project - modelling software capable of handling these complex forms. Digital Project was then attributed as a factor in the development of the next generation of freeform architecture. The collaborative nature of the design, analysis and construction process has also pushed the development of software design – with every advance that allows an architect to produce complex and irregular geometries, new tools to rationalise, optimise and analyse these forms have been and are continuously being developed.

Digital tools can be put into two categories – those that recreate a previous workflow within the computer (with the advantages this can bring) and those that allow new workflows that are impossible without the use of a computer. In this thesis the development of software that falls into this second category is explored.

Form finding and physical simulations

Form finding experiments can provide useful information and embed a logic and elegance into structural design. The basic concept is to create a scale model and allow a material to interact with physical forces (such as gravity or surface tension) and study the often non-linear deformations. Some successful examples where designers have utilised these methods include:

- Antoni Gaudi's use of intricate hanging chain models to guide the design for masonry vaults for the crypts of the Church of Colònia Güell and the Sagrada Familia in Barcelona.
- Frei Otto's use of soap films to find minimal surfaces to aid the development of tensile membrane structures for the German Pavilion in the Montreal Expo '67.

- Heinz Isler's experiments with hanging fabrics for the design of thin reinforced concrete shells for Norwich Sports Village.

These projects were successful as the structure is considered at the same time as the form; it is perhaps no coincidence that these designers are often considered both architects and engineers.

There have been several attempts to digitally recreate these kinds of experiments. There are many advantages to this – creating a physical form finding model can be expensive, time consuming and retrieving the three dimensional information about the deformed shape and forces within the material can be an inaccurate and tedious task. A digital model can easily be exported and integrated into the CAD, structural analysis or CAM workflow. This ultimately means, once the digital simulation has been programmed, a designer can explore far more experiments at minimal cost.

There is currently some interesting research in taking form-finding experiments that fit into the second category of digital tools – those that are only possible using digital simulations, see below. This may be because the forces to be modelled cannot be easily produced or maintained accurately (magnetism, inter-particle physics), the scale is unworkable (too big or too small) or the time frame is too fast or too slow or unwieldy for another reason (chemical reactions). If a physical process is understood it can be modelled (or approximated) digitally, tightly controlled and quantifiable simulations can be run, forces can be altered and time can even be reversed.

Daniel Piker, an independent researcher and the developer of Kangaroo (a physics simulation plug-in for Rhino (McNeel 1995)) has explored *"pseudo-physical materials – virtual materials with custom rules for how they respond to deformations, which do not correspond to the behaviour of any real material."*(Piker 2011). In an early test he created a triangulated material with internal forces, which allow the triangles to increase in size but remain equilateral. Another powerful aspect of digital simulations is that unlike physical models, the topology can be easily updated during the simulation; as a basic example, this could be used in particle simulations where particles within a certain range are connected.

Engineers' response - pre/post rationalisation

These form-finding techniques (both physical and digital) are a way of pre-rationalising a complex structure before detailed structural analysis, and as shown by Gaudi, Otto, Isler et al, can help produce structures that are more efficient, aesthetically pleasing, or at least less intrusive. However, these may guide a design towards an aesthetic not desired by the architect - as ultimately some control is lost in the process. When this pre-rationalisation step is not taken or if the results are rejected and an irregular and unconventional form has been chosen, there are two options for the structural engineer.

One response is to exactly match the form required by the architect by fitting traditional engineering arrangements "by eye". By fitting heavy standard members around the arrangement, with often complex and unique connections, a solution can usually be found. However, this can be a painstaking and expensive process during design, fabrication and erection and, because of the complexities and time required to tweak the design, efficient solutions are not the norm. Structures designed in this way often have an appearance of awkwardness and in the worst examples heavy structure is used simply to support a façade or sculptural form or even resist awkward loading that the complex arrangement exerts upon itself, see Figure 1.

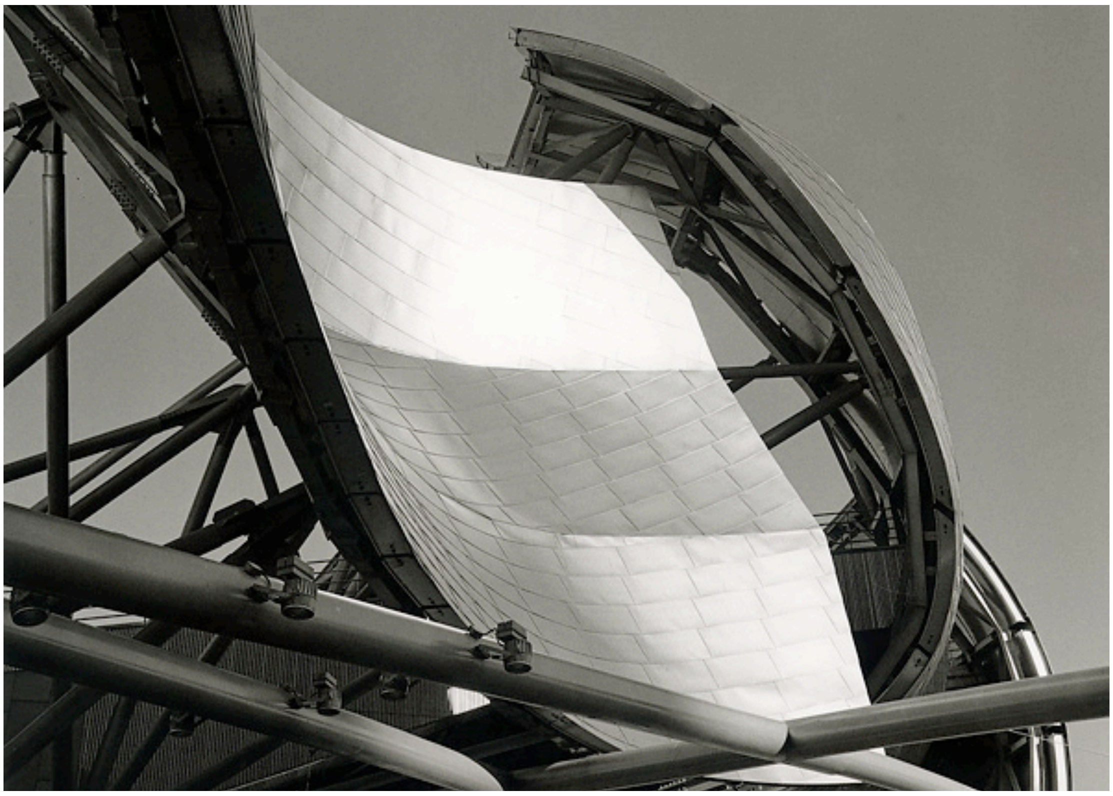


Figure 1: Frank Gehry's Pritzker Pavillion, Millenium Park, Chicago, 2005 © David A Goldfarb.

Clearly this should be avoided if possible. Engineer and designer Cecil Balmond suggests - *"with free form, what you have to realise is that every point of curvature is a structural opportunity, and that's why it is so important to work together now. The danger is that if we stick with the assumptions of structure as they are, and architects do not try to understand new structural possibilities, you end up with post-and-beam logic underpinning free form, and you are not getting anywhere"* (Leach, Turnbull et al. 2004). Perhaps just as objectionable as the material and structural inefficiency is the spatial inefficiency – post and beam construction will not match the surface closely and will therefore create a volume of wasted space between the surface and the structure. Furthermore, this method loses much, if not all, of the interesting form created by the free form surface for the interior perspective.

The second response has been to take the desired form and post-rationalise it. This differs from the form-finding techniques described earlier – starting with a given form an arrangement of structure or cladding is devised that fits around it with minimal effect on the form. Using a variety of techniques the form is tweaked to improve desired performance or produce optimal conditions for fabrication such as zero Gaussian curvature. This technique has proved successful for a range of goals – member lengths, connection similarity, cladding rationalisation and structural performance (Eigensatz, Deuss et al. 2010). However, this post-rationalisation often occurs too far down the design process for the maximum benefits to be achieved and a resistance to alter the form at this stage is to be expected.

This paper explores a third response – where the form is fixed throughout but the aim is to still create elegant and optimised structure to support it.

Differences between architects' and engineers' software

It can be argued that, when working with complex and irregular forms, the architectural judgement of whether the result is satisfactory (from an aesthetic point of view) takes no longer than with a traditional building. However, whilst an experienced engineer can look at a traditional structural arrangement and be able to quickly judge whether a workable solution is possible, this process will take disproportionately longer for an irregular form and therefore cannot be done in parallel – Prof Kristina Shea suggests "*engineers need tools to keep up with the creative processes of architects*" (Leach, Turnbull et al. 2004).

To be able to respond at the same speed (and therefore guarantee more of a focus on engineering logic in scheme design) an intuition and understanding of workable structural logic that can be applied to complex and irregular forms must be developed. It has been argued that one way of improving this tacit knowledge is with the instant feedback gained from interactive real-time simulations (Zarzycki 2009). If these simulations include a generative engineering component, this understanding can be further improved. For this reason it was decided by the author that any software developed must be a real-time generative design solution that based on sound engineering and manufacturing principles.

It has been suggested that when the form or volume in which a structure exists can be changed there are various methods to improve the structural performance. However, when the structural engineer is restricted to a fixed but irregular and complex volume (that the structure must support *and* be contained within) the current solution, of fitting structure by eye and implementing different arrangements for every irregular area, is not satisfactory from an engineering or aesthetic perspective. There are a number of situations where this could be the case – existing surrounding buildings, protected sight lines, architectural or artistic vision. For example, when working on a large sculpture (that requires internal structural support) the outer surface material must be supported at regular intervals and any alteration of form is unlikely to be acceptable. Or perhaps a solution with a volume of a particular shape is so optimal with respect to a number of other design parameters that if a satisfactory structural solution can be found then it is to the advantage of the project.

There has been various explorations of these challenges using digital tools however all attempts have various limitations. These are explored in2. Literature review & previous work.

Aims and case study

This thesis aims to explore the challenge of creating an interactive framework to fill complex and irregular but fixed volumes with a stable and functional structure.

The objectives that the tool should achieve are:

1. A digital output of a 3-dimensional structure that can be exported for integration into a CAD model, structural analysis or fabrication software.
2. The software must be able to work with any concave shape that is input by the user including internal voids.
3. The structure must support the entire surface of the volume.
4. The software should allow the user to define fixed support points, both within and on the surface of the volume.

5. The software should run in real-time and provide visual feedback for the user.
6. The structure should be optimised to minimise material or deflection.
7. It should be faster than current engineering solutions.

In order to assess the effectiveness of the software it will be used to create a structural solution for a complex roof volume on a live project - Tallinn Town Hall currently being designed by BIG architects and Ramboll UK structural engineers. The output of the software will be compared with an existing, conventional structural design proposal.

Structure of the thesis

Chapter 2 explores in depth relevant previous work and investigates existing algorithms and concepts to benefit the development of the software. In chapter 3, the software development process is described and the various solutions are explained in detail. In chapter 4 initial testing and calibration of the software is presented, starting with basic volumes such as spheres and tetrahedrons and moving onto more complex concave volumes. In chapter 5 an optimisation routine is introduced and results from initial tests are presented. Chapter 6 outlines in detail the use of the software in respect to the Tallinn Town Hall case study and the software is quantifiably analysed and assessed. In chapter 7 conclusions are drawn and recommendations for further work are made.

2. Literature review & previous work

Volume filling

There have been many experiments in creating generative forms. Selected examples were explored to assess their suitability for volume filling and generation of structural systems.

L-systems

A Lindenmayer system (L-system) "*...a technique for defining complex objects by successively replacing parts of a simple initial object using a set of rewriting rules or productions*" (Prusinkiewicz and Lindenmayer 1990). The system is made up of a seed and replacement rules. At each iteration each character in the current string of text is replaced by another set of characters (depending on the replacement rules). A replacement rule could, for example, be to replace an "A" with "AB" and a "B" with an "A". These replacement rules can also have probabilities to inject a random element into the structure. Once all the iterations have been carried out, each character in the resulting text represents a particular drawing function. These actions could correspond to "draw line 1 unit long in forward direction", "rotate yaw +90 degrees", "store current location and direction", "go back to last stored location and direction" etc.

An interesting experiment in using L-systems in generative building design (Hansmeyer 2003), includes a path memory which is implemented to avoid self intersections, and clash detection checks which ensure the structure does not cross an undulating floor, see Figure 2.

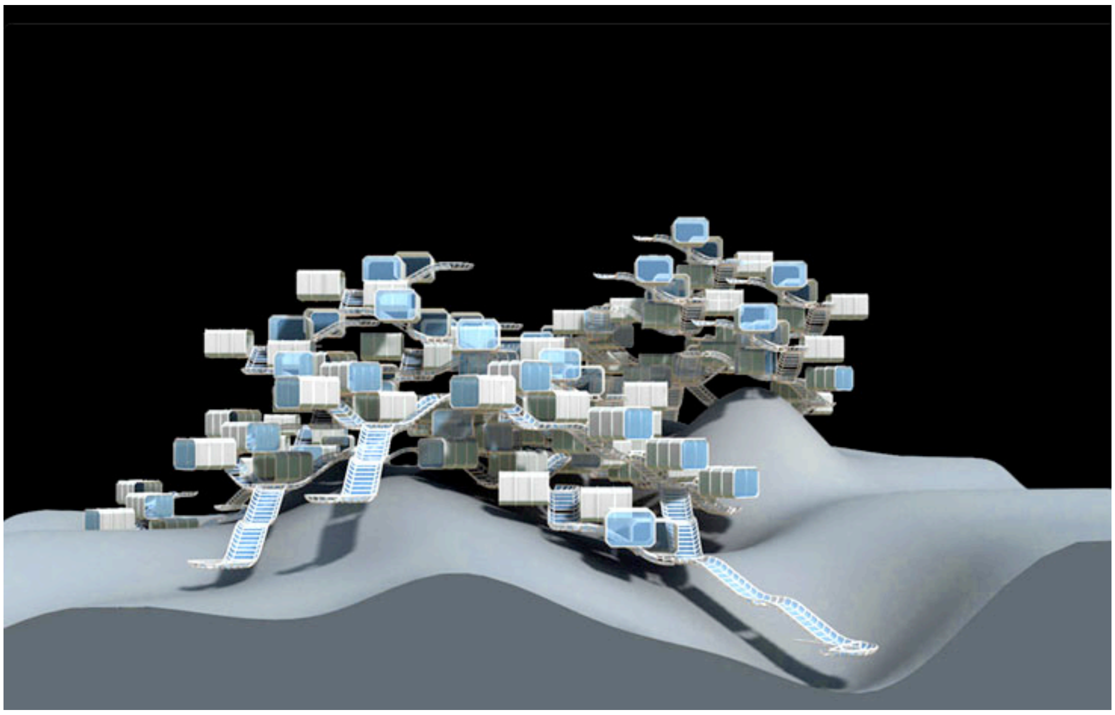


Figure 2: Generative model using L-systems boundary obeying behaviour © Michael Hansmeyer.

Although this technique could be used to contain a structure within an irregular and concave boundary, the advantages of the initial simplicity and beauty are outweighed as the data for the structure grows exponentially and creates a linear data structure. This makes them unwieldy and unable to interact efficiently with itself, other l-systems or its environment. All of these aspects are of priority when analysing structural and spatial performance.

For further research the author suggests ideas used in L-systems are developed to a more object-orientated structure, where parallel data structures can be created and which should be more effective for generating efficient and logical structures.

Diffusion Limited Aggregation

Diffusion Limited Aggregation (DLA) is "*the process whereby particles undergoing a random walk due to Brownian motion cluster together to form aggregates of such particles*" (Wikipedia 2011). This can be used to create 3-dimensional structures that can be used to fill complex volumes. The growth can be guided using pre-defined curves or physical simulations and they have been used to grow structures using simultaneous structural analysis to reject unstable additions (Zarzycki 2009) see Figure 3. In this example each particle is represented as a truncated octahedron.

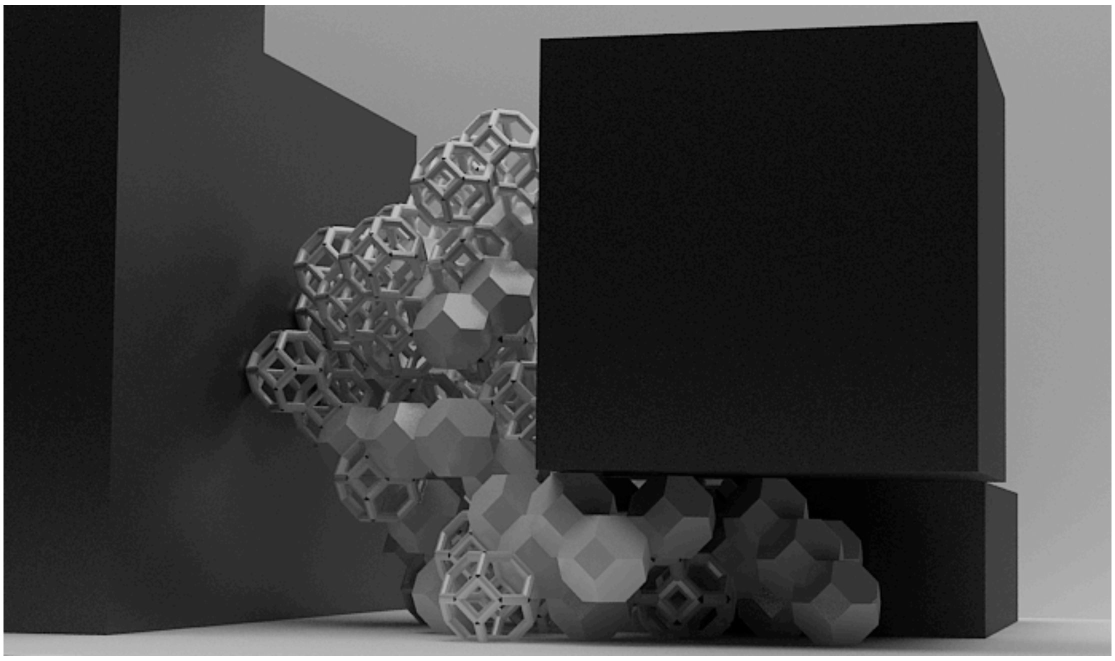


Figure 3: Diffusion limited aggregation constrained by its surroundings © Zeta Kachri - Parasitic Ecologies.

Equilateral tetrahedrons could be used for each particle - this would develop into a space frame topology with an irregular overall form constructed from inherently stable units. Areas of exploration for optimisation could include guidance of structure to zones of high strain and/or adapting the size and shape of individual "tetrahedron particles". The author concluded that this would be an interesting area for future research although it was not further explored for this project.

Bubble meshing

When analysing a 3D shape using the finite element method it is necessary to divide the geometry into a discrete mesh of (often tetrahedral or hexahedral) elements. These elements should fill and fit the volume exactly and therefore any automated methods that are used to achieve this are also of interest to this research. One such technique is bubble meshing. Shimada and Gossard (1995) observed that *"a pattern of tightly packed spheres mimics a Voronoi diagram, from* *which a set of well-shaped Delaunay triangles and tetrahedra can be created by connecting the centres of spheres"*. To fill a volume with spheres they take an "*initial guess*" for the centres using hierarchical spatial subdivision and then perform a "*dynamic simulation for a force balancing configuration*" to relax the mesh which essentially reduces the intersections between spheres. This force-based distribution is an effective way of ensuring that bad elements (such as nodes close together or flat tetrahedrons) are avoided – these situations are unstable and therefore the mesh will not settle in these positions. "*Adaptive population control*" was implemented which calculates how many particles are adjacent to a particular particle and adds or removes particles if necessary.

Bubble meshing is an effective method for producing meshes (see Figure 4) and, as tetrahedrons are inherently stable structures the mesh also has favourable structural properties. A number of concepts presented by Shimada and Gossard (1995) are further explored and adapted for this project, see Software development.

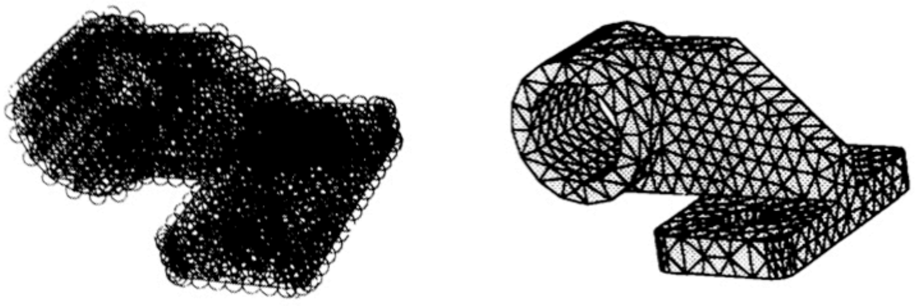


Figure 4: Loose packing of bubbles (left) and the resulting 3D mesh made up of tetrahedrons (right) (Shimada and Gossard 1995).

Another example of the use of a bubble analogy for filling space is the Weaire Phelan structure (Weaire and Phelan 1994) - currently proven to be the most efficient way of dividing an infinite volume into cells of equal volume with minimal surface area between them. This was used successfully for the Watercube Aquatics Centre in Beijing (PTW Architect and ARUP) in a design that "*achieves every goal with one idea: mechanical, acoustics, light, structure, architecture*" (Weinstock 2006). Other interesting features to note are –

- The volume filling structural philosophy works throughout the structure - walls, roof and primary façade support.
- Only four different types of nodes and ten lengths of bar members were used.

With the geometry specified by the packing process the engineers sought to adjust the size of the member cross-sections from three different thicknesses. To achieve this an automated optimisation scheme was implemented. The complex load paths throughout the structure mean this is not a straightforward task and if the members were too big the structure could not support its self-weight. An iterative cycle was run which checked every member (in each of the 200 load cases) and increased any overstressed members by one size and reduced any members that were less than half stressed. According to Arup engineer Tristram Carfrae - "*After only fifteen cycles, the number of overstressed elements rapidly comes down to virtually zero [and it] converges to what turned out to be a very efficient structure*" (Weinstock 2006). Although this is a relatively simple optimisation strategy it seems to have been effective. For a structure of this type the process highlights a reliance on automated and efficient computational calculations - as an estimate, analysing 22,000 members for 200 load cases 15 times equates to 66,000,000 member checks. To put this in perspective - if an engineer were to complete a beam hand check every minute, twenty-four hours a day, the process would have taken over one hundred and fifteen years.

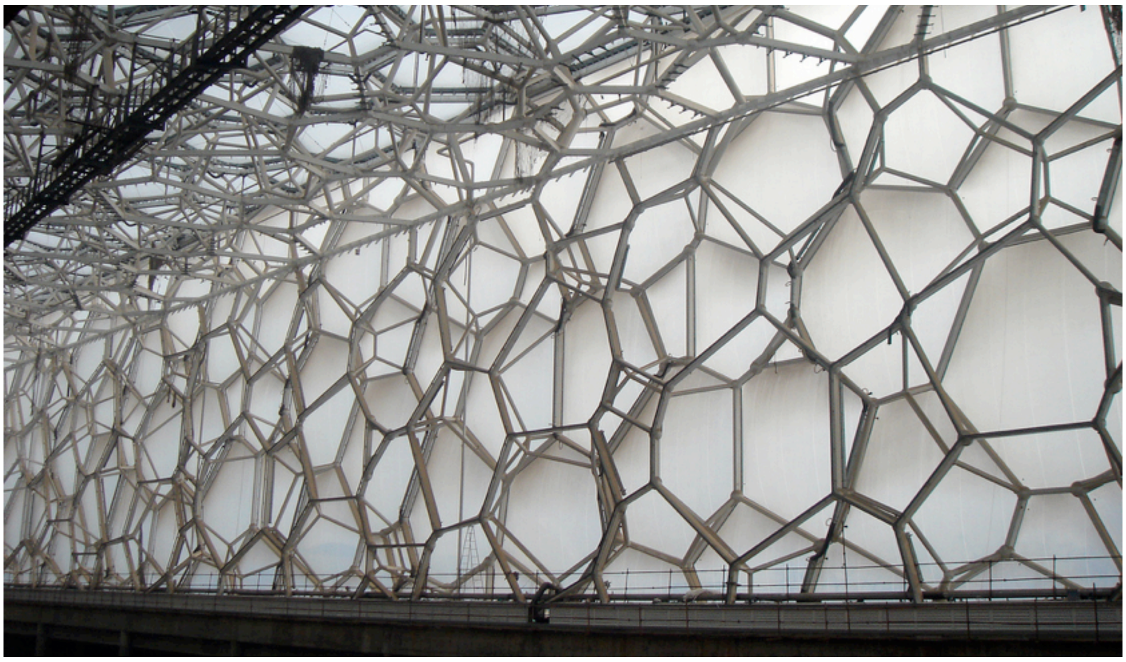


Figure 5: Internal view of the Watercube during construction.

Foam metal is a "*cellular structure consisting of a solid metal, frequently aluminium, containing a large volume fraction of gas-filled pores*" (Wikipedia 2011). It is manufactured by injecting gas or a foaming agent into molten metal to produce extremely lightweight and porous materials that are generally used for energy absorption rather than structural purposes. The metal forms a structure similar to 3D Voronoi cells around the bubbles.

A Voronoi structure made up of the edges of the faces between cells relies on a combination of axial forces through the members and bending stiffness at the connections to resist load. A better arrangement of the internal structure for structural use would be the dual of the Voronoi or the Delaunay (see Design and optimisation section below), which will produce tetrahedrons. The main advantage would be that all the forces would be transferred axially, which is more efficient than in bending. Although this arrangement cannot be recreated using bubbles, 3D printing technology could be used to create these arrangements at a range of scales. An optimisation strategy that could be used to further improve the structural performance of these materials is discussed later.

In the paper "Using simulations and artificial life algorithms to grow elements of construction" (Jaworski 2006) structures are presented that use a system of agents simulating the principle of hot air movement where they "*move vertically upwards, avoiding obstacles, and this forms paths of possible growth directions, which 'embrace' [the] avoided object*" (see Figure 6). These agents guide the movement and development of a separate bubble mesh to create a stable structure to support load.

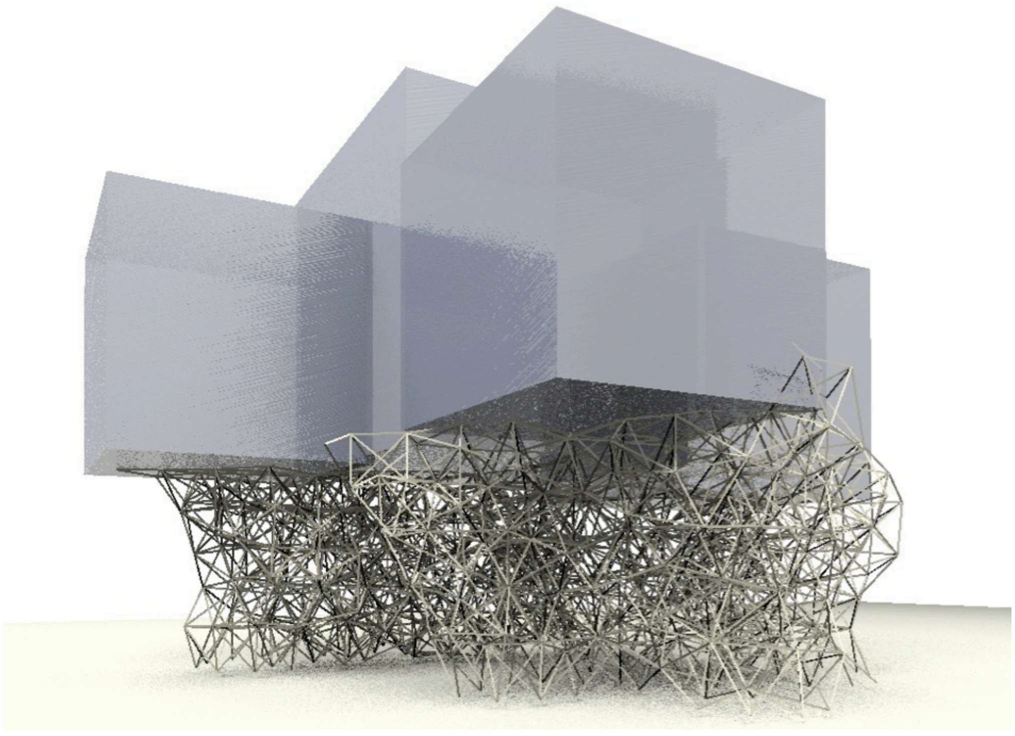


Figure 6: The result of "hot air" agent based guidance of a bubble mesh to support a volume (Jaworski 2006).

Approaches for using particle-spring simulations to fill an "airtight" volume with structure are explored by Kanellos (2007). Extensive testing of the performance of structures filling cubes against more traditional arrangement was documented, that indicated the method is capable of generating space frame topologies.

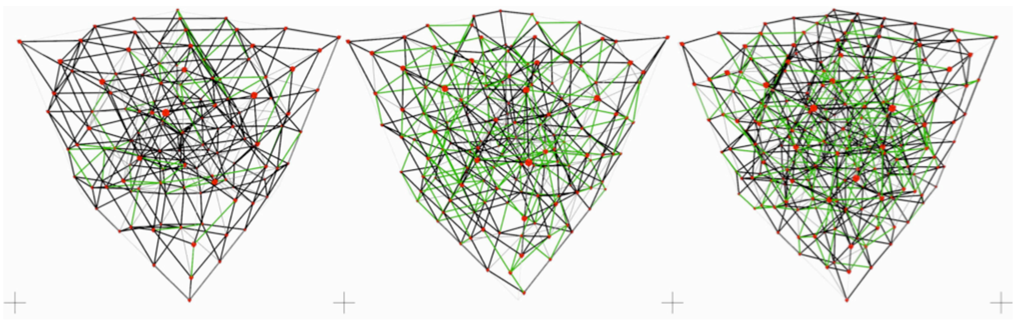


Figure 7: Examples of volume filling of a cube (Kanellos 2007).

However, the distribution of points appears to be uneven. Issues relating to non-convex boundaries were mentioned (it seems intersection checks were carried out on the plane the faces were on, rather than the actual faces) and the connectivity routine seems to be sub-optimal. In both this and the previously described project, springs were created between any two particles within a certain distance of each other. In the middle of a set of particles (that are already well distributed) this can lead to a good connectivity of the particles with its neighbours. However, two particles further than this distance apart would never be connected – even if there were no particles in between and connectivity would be advantageous for stability. In other words, complete tetrahedrons are not guaranteed as can be seen in Figure 7.

Two further experiments of interest are presented by Wallner (2010). They target many of the aspects this thesis is looking to solve – "*irregular space frames inside arbitrary (including non-convex) boundary volumes with predefined support areas*". In the first algorithm the edges of Voronoi structure (built from a set of randomly placed points) are used to produce a "*traffic system*" within the volume and then shortest routes between the specified support zones and load bearing zones are found and straightened, see Figure 8.

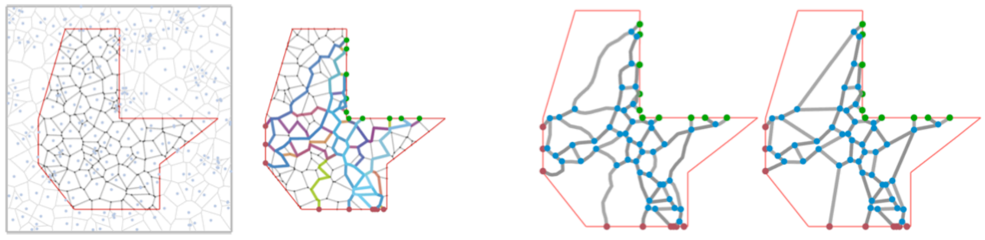


Figure 8: A 2D representation of the Voronoi paths structure generation (Wallner 2010).

The second approach uses a "*repulsive force field for the calculation of curve-skeletons for three-dimensional objects*". Figure 9 shows structures designed using these algorithms.

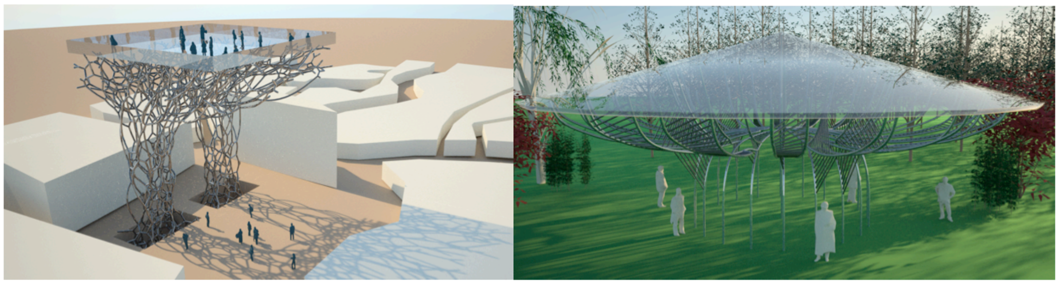


Figure 9: Structures designed using Voronoi traffic system (left) and skeletonisation technique (right) (Wallner 2010).

Whilst both of these examples are interesting experiments in applying geometrical algorithms to architectural design, the writers state that it is not their "*concern to generate structures in regard to statics in the first instance*" and the structures are "*intended as input for a genetic algorithm which optimizes the static stability*". This is a conflicting approach to the aim to of this thesis and highlights the difference between generative design and generative engineering. It could be argued that generative design such as this creates the same problems for engineers that are created by using freeform NURB based design as the static structural stability is completely disconnected from the initial design.

Sculptural solutions

As mentioned previously, volume filling may be of use in the rationalisation of sculptural works. In fact, the history of internal structure and support for sculptures is the driving force behind much of the developments in the field and has relevance to the structural design for buildings. At least as far back as the Statue of Liberty internal armature has been used to support doubly curved façades – essentially the same system was used in Frank Gehry's Guggenheim Museum in Bilbao (Winslow 2010). Perhaps the pinnacle of this method is Anish Kapoor's "Cloud Gate" with its flawless mirrored surface and "*aircraft wing dimensional tolerance*" (Chadwick, Thomas et al. 2009).

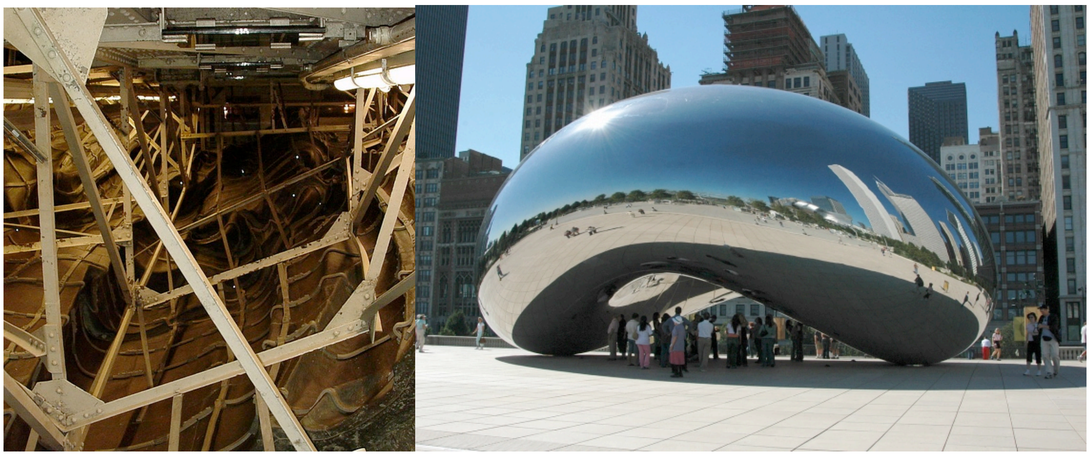


Figure 10: Internal structure for the Statue of Liberty (left), Anish Kapoor's Cloud Gate (Kapoor 2011) (right).

Antony Gormley's recently completed "Exposure" is an explicit example of volume filling structure to match a given surface; in fact, the structure ***is*** the sculpture and vice versa, see Figure 11. In a similar fashion to the bubble meshing algorithm, the surface is discretized separately from the internal volume, in this case a "*Voronoi pattern is found on the surface (found by a cellular automata method on an original hi-res scanned mesh* [of the artist's body]*) and the interior found by a kind of spring system*" (Hanna 2011).

Another example of an interesting method for volume filling in sculptural design can be seen in "The Tall Tree and the Eye", again by Anish Kapoor, see Figure 11. Irregular sphere packing is a stable configuration of spheres created by randomly pouring spheres into a container and compressing the pack. It creates a stable structure against compression as long as the container can withstand the lateral forces (Song, Wang et al. 2008). It would be challenging to create a digital model of irregular packing without using a physical experiment or digital simulation as a basis. To achieve the arrangement the Advanced Geometry Unit at Arup used a physics simulator within an animation package to quickly create digital models with an inherent stability for design exploration and reflection studies (Tuffanelli 2010).

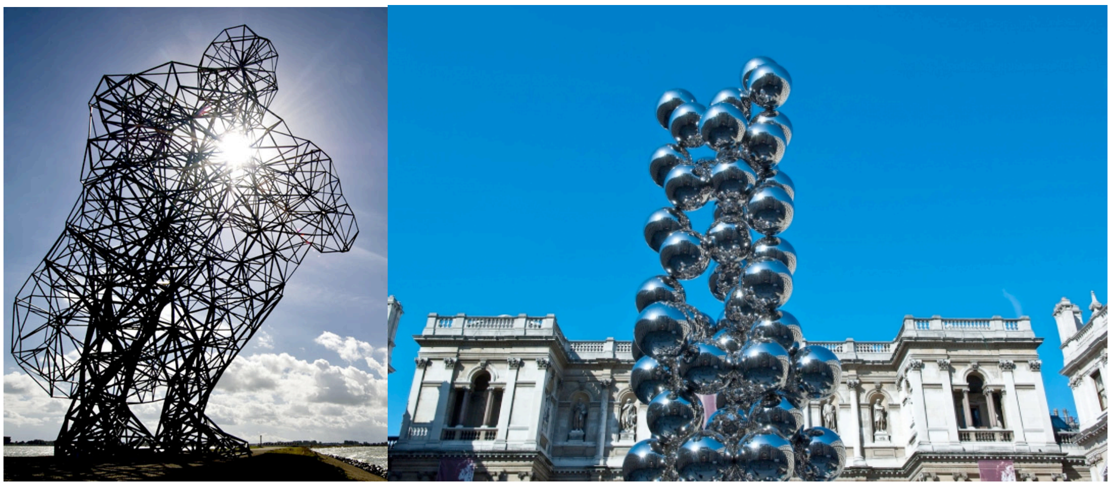


Figure 11: Antony Gormley's Exposure (Gormley 2011) (centre), Anish Kapoor's Tall Tree and the Eye (Kapoor 2011) (right).

Design and optimisation of space frames and trusses

Many of the examples presented above are based on the space frame. As post and beam construction has been rejected, unless the volume is to be filled with solid material or the surface used as a monocoque structure then the natural choice for this project is also a form of space frame construction.

In terms of architectural use, Buckminster Fuller popularised the use of space frames and used them to realise his famous geodesic domes. Interestingly, their origins can be linked to volume filling and the close packing of spheres (Marks and Fuller 1973). The system distributes forces throughout the structure efficiently, is robust and can be easily pre-fabricated and erected.

When designing spatial structures the chosen form should be related to the boundary conditions and/or supports. This can be seen in the orthogonal arrangement seen in typical spaces frame used for rectangular boundary conditions and the concentric circles seen in spatial dome arrangements. Spheres can be approximated if subdivided into equilateral triangles but for more irregular boundary conditions a regular subdivision pattern may not exist. A possible solution for this is to project a grid and trim to fit the boundary, however, this will lead to inconsistent support spacing. It is concluded that the only way to ensure regular boundary supports when working with an irregular shape is with an *irregular* configuration. When moving beyond a single layer-grid into a multi-layered space frame this issue of boundaries having to match the form of the individual unit is exacerbated. Therefore, regular modules must be abandoned in order to fill irregular volumes. Whilst some of the advantages of regular configurations may be lost, there are opportunities within this shift – anisotropic frames may produce more interesting forms and have the potential to become more efficient, see Optimisation. In terms of manufacture these irregular configurations add an element of complexity as nodes and members become less repeatable and interchangeable. However, whilst this is not an ideal situation, there are various tools available to engineers to ease this process. For example, a parametric modelling system could automatically create structural models for all required nodes and digital fabrication techniques such as 3D printing could manufacture and automatically label them (to ease the assembly process).

Delaunay

The use of Delaunay Tetrahedralisation (DT) has been previously noted and different methods to compute the algorithm are explored elsewhere (Cignoni, Montani et al. 1998). Perhaps the most fundamental advantage is that the result is always built up of only tetrahedrons and is therefore not a mechanism, which is beneficial for a generative engineering process. There are other advantages, for example each triangular face of the tetrahedrons is "*as nearly equilateral as possible*" (Shimada and Gossard 1995) for a given point set. This may be desired from an aesthetic viewpoint but more importantly this means the spacing of the edges entering the same node is maximised which is advantageous for manufacture and erection.

Interestingly, if the nodal arrangement for a Warren truss is used as an input then the Delaunay edges will exactly match the arrangement of structural members. This also works for a Pratt truss (although the diagonals may not all face the same direction and may produce a Howe/Pratt truss hybrid) and in three dimensions the Octet truss (albeit with the top square triangulated – which can be advantageous if significant horizontal loads are present). This property of DT and an adapted algorithm is explored and tested in3. Software development.

Optimisation

An example of a structural optimisation routine was shown for the Watercube and if the software is to produce efficient results a suitable optimisation routine should be found. There are many factors in design that one can optimise, however adjusting one can often have an adverse effect on another. A good example of this situation would be a preference for a minimum volume of steel use, which could have a negative influence on the stiffness of a structure. When multiple load cases, aesthetics, serviceability criteria and manufacturing considerations are included these problems can quickly become extremely complex. To find a good balance between these interrelated factors genetic algorithms (Mitchell 1996) can be used.

By encoding positional information for a set number of nodes and analysing the resulting triangulation it has been shown to be possible to optimise 2D trusses for a reduced total stress (in comparison with a regular truss, see Figure 12) and produced similar arrangements to previously proven topologically optimised arrangements (Harding 2010).

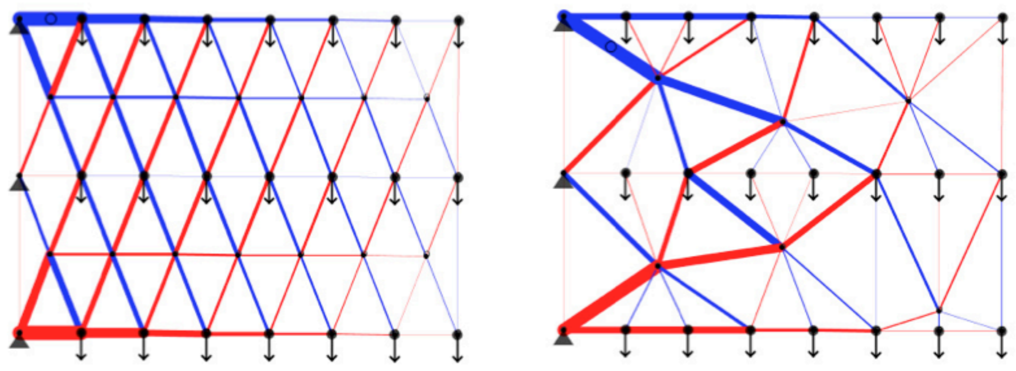


Figure 12: Comparison between regular arrangement of truss node positions (left) and positions found using a genetic algorithm (right).

A method introduced by Schein and Tessmann (2008) is to use a set of fixed node positions and optimise connections between random pairs between a top and bottom layer of a space frame. Three fitness functions were used to find an optimal solution; these were structural performance (deflection under dead load), number of elements and intersections with predefined volumes that should contain no structure - "*the design process is driven by architectural and structural parameters simultaneously*" (Schein and Tessmann 2008).

In his paper Winslow (2010) describes an interesting approach, where a number of selected regions on a surface are optimised in response to multiple load cases. Using these points as a basis the rest of the structure is built by interpolating parameters between the points. Although the example is a single layer grid, it should be possible to expand this to 3D.

An example where a 3D space frame was optimised with nodes that were free to roam is presented by Papapavlou (2008). Whilst under certain load cases it was found to out perform an "engineered" regular space frame (in terms of a better distribution of strain), see Figure 13, the results were not consistently better and appear so random that there seems to be no learning process for the architect or the engineer.

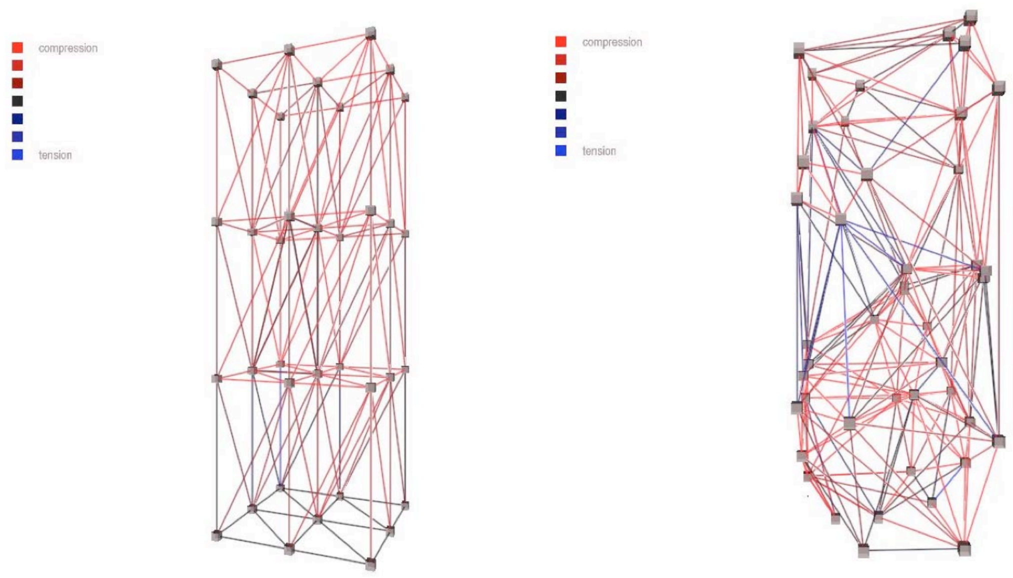


Figure 13: A regular node arrangement (left) and a generated space frame (left) (Papapavlou 2008)

This is unexpected in comparison to Figure 12 - where the result can be understood structurally. This may be as a result of the extra dimension, which leads to 3D load sharing behaviour that cannot be easily comprehended or errors in the algorithm.

Whilst GA’s have been shown to be a flexible way of solving complex problems, the author decided to find a solution that optimises in a more direct manner, in real-time and by adapting a single solution (rather than producing many solutions and choosing between them) in the hope that it can lead to an improvement in the understanding of complex space frames. This doesn’t necessarily mean the number of parameters we optimise for have to limited - if a direct force-based optimisation routine is used, conflicting parameters should find a balance between each other. This balance could also be interactively adjusted by the user and rather than having to change probabilities and re-run the GA.

Further to the Watercube example a space frame could be optimised by reducing the length of highly stressed members. This will not only increase the buckling resistance of the member but should also lead to a denser mesh in the surrounding area which should result in more load sharing and essentially more material to support the load. An advantage of this over cross-section optimisation is that all the members could be the same shape - which may be advantageous aesthetically, financially and for connection design.

John Harding of Ramboll Computational Design experimented with this process in 2D (see Figure 14) and in 3D (with nodes constrained to a surface) (Harding 2011). In both cases the deflection of the system was reduced. Implementing this method on a 3D space frame, while maintaining support for the outer volume, is an interesting avenue for further study and would also be particularly relevant to the DT 3D printed foam metal concept described earlier. One can imagine a cross section of a tube with a dense mesh on the perimeter and a more sparse mesh in the centre in response the relative stresses. Although this is essentially creating a tube the idea becomes far more significant when considering complex cross sections with complex loading. In a way this can be seen as an extension of the pseudo physical materials – a material that pushes density towards areas of high stress.

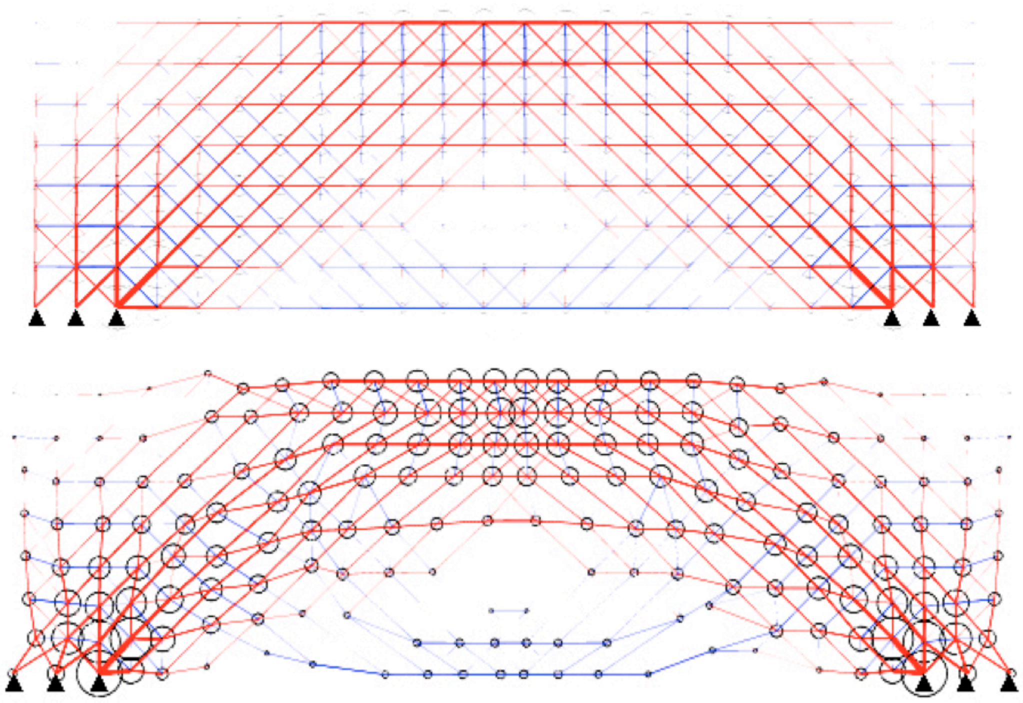


Figure 14: Comparison of initial ordered layout of nodes and members (top) with optimised layout where nodes have been moved towards areas of high stress (bottom) (Harding 2011).

The similar process was used by structural engineers AKT in the optimisation of the Land Securities bridge where a hexagonal mesh constrained to a surface was moved towards area of high stress (Kaijima and Michalatos 2007).

3. Software development

As a result of the literature review it was decided that a volume-filling tool would be developed in the form of a particle simulation reflecting physical behaviour where possible. Delaunay tetrahedralisation will form the basis of the topology between particles to create a stable space frame structure and an optimisation routine using force-based relaxation will be implemented.

Particle state analogy

In the physical world, a good example of volume filling behaviour is the diffusion of gas particles as they completely fill a container of any shape in which they are placed, whereas a liquid or foam will only fill the volume if the correct numbers of particles are present. Shimada and Gossard (1995) highlight this, where careful consideration and adaptation of the number of particles or bubbles was necessary. Therefore it was proposed to use the kinetic theory of gas and simulated Brownian motion to create particles that can fill a volume with little or no input from the user. However, once the volume is filled, the gas particle analogy becomes less relevant. In fact, a "cooling" of the gas into a liquid or solid state becomes more appropriate and the forces and topology algorithms should change to reflect that.

Development platform

The software was developed with the Java programming language initially on the Processing platform (Fry and Reas 2004) which facilitated early exploration of both the particle simulation and l-systems. However, in order to manage the multiple windows, user interaction and an increasing program complexity, development was moved onto the Eclipse development environment (http://eclipse.org/). See Appendix A for interface layout.

Development of the physics simulation sandbox

A winged edge data structure was constructed. This allowed the creation multiple 3D meshes made up of vertices, edges and faces with a record of the topology within them. The collection of each group of vertices, edges and faces were stored in a Mesh class along with various methods for displaying and adapting the mesh. Although the vertex position could be changed they were simply points in space. To simulate physics the Mesh class was extended to become the Particle class.

In situations where the force (and therefore acceleration) varies over time (depending on the velocity or position of an object) the velocity can be calculated by integrating the acceleration function with respect to time. However "*differential equations in which the quantity we're solving for is part of the equation are not always easily — if at all — solvable by analytical means*" (Verth and Bishop 2008), therefore a numerical method is sought. There are various numerical integration methods available for calculating positions of particles that obey Newton's second law. Popular methods are Runge-Kutta, Verlet, Implicit (or backwards Euler) and Semi-Implicit. They are all slightly different ways of solving initial value problems (or more accurately a *series* of initial value problems). The methods vary in terms of simplicity, speed of calculation and stability, and some are more suited to certain types of simulation than others. For example, Verlet is popular for use in the simulation of molecular dynamics as it is stable and fast (Verth and Bishop 2008) and so was chosen for this application.

The specific details of the method are explained by Verth and Bishop (2008). For this thesis all that is important are the inputs and outputs of the process. The main extension of the Particle class is that each vertex knows not only its current position (inherited from the Mesh class) but also its position in both the previous and next frame. From now on these will be identified as P0 = previous position, P1 = current position and P2 = next position.

To run the simulation, a function within the Particle class "runVerletStep" is called which in turn calls any functions that impart a force (discussed in Forces on particles) and all of the forces on each particle are summed. Using this force the Verlet step is carried out and P2 is calculated. This process can be looped as many times as required per frame, which has the effect of speeding up the simulation.

Boundary collisions

Once the particles are moving around the sandbox, a system must be implemented to contain them. In the bubble meshing technique (Shimada and Gossard 1995), particles are constrained to every vertex, edge and face of the enclosing volume. For the purposes of finite element modelling all of the edge and vertex information should be maintained and this method is suitable, however for this research as it is too closely related to the mesh topology of the boundary volume. If the bubble meshing strategy were implemented, the face size of the boundary volume would also be the maximum face size for the generated structure. Ideally the generated structures should not be limited in this way - it should be possible able to make a low-resolution structure bounded by a high-resolution boundary volume and vice-versa. It would be possible to adapt the algorithm to allow particles that are fixed on an edge (or face) to move onto the connecting edges (or faces) and thus spread out beyond this limit. However, a solution that does not force particles to be defined as vertex, edge, face or volume particles is preferable.

It should be noted that the surface vertices in the bubble mesh method are used to repel the free volume particles and prevent their escape. Therefore, if there are no particles fixed to the surface we must instead rely on the boundary to hold the particles inside. To achieve this, an algorithm to check each particle's position against the volume (and calculate a corrected bounced position if necessary) was implemented. Particles must never escape the mesh, but they should be free to slide around and move off it.

From the Verlet integration function the P1 and P2 for every particle in the simulation is known. The first step is to check whether the line between these two points intersects any faces, if so, a reference to the face that is intersected first and the location of intersection is required.

If a face is defined as a triangle enclosed by three vertices, each face check can be made using a series of tests (of increasing complexity). As soon as any test returns false it can be concluded that no intersection takes place and the process can quickly move onto the next face. The tests are as follows:

1. Are P1 and P2 on opposite sides of a plane defined by the face?
2. Is the intersection point with the line P1-P2 and the plane within the face?
3. Is the distance from P1 to the intersection point shorter for this face than for any others?

If all of these return true then the particle intersects the face in question and it is the first face it hits as it travels from P1 to P2 and so the positions P1 and P2 must be adjusted accordingly.

An initial solution (of how to adjust a particle after an intersection) was to place the new P2 on the intersection point and reflect P1 in the plane of the face so it was outside the volume and would provide the "bounce" momentum for the next frame. Whilst this solution worked and seemed to re-create the bouncing of particles accurately, a problem arose if a particle was pushed up against a face with a force greater than the bounce momentum (forces between particles will be discussed later but this situation happens frequently). This leads to an intersection every frame and any in-plane velocity (which would represent the particle sliding on surface of the plane) would be lost. This sliding is desirable if particles are to be well spaced across the boundary. This can be relieved slightly by offsetting the particle away from the face a small distance in the direction of the normal, allowing some in-plane movement to occur, but the particles remained "sticky" and this offset would sometimes push a particle through an adjacent face and outside of the volume.

To avoid these problems, a recursive bounce algorithm has been implemented. P2 is reflected with respect to the plane of the face and the bounce vector (the line from the intersection point to P2) is then checked to see if it intersects any other faces and, if so, the step is repeated until the bounce vector does not intersect any face, see Figure 15. This process reduces the length of the bounce vector by the distance the particle travels within the boundary each step, which, unless the particle intersects exactly onto the boundary between two faces, stops the recursion entering an infinite loop. If the intersection point is touching two or more faces the bounce vector will never reduce and an infinite loop would occur. In these instances the intersection point is moved by a small fraction along a vector pointing towards the centre of one of the faces and the recursion can continue. Once each particle has been checked and the P1 and P2 positions have been updated the scene can be drawn and the next Verlet step with P0 and P1 now equal the last frame's P1 and P2 respectively can begin, see final frame in Figure 15.

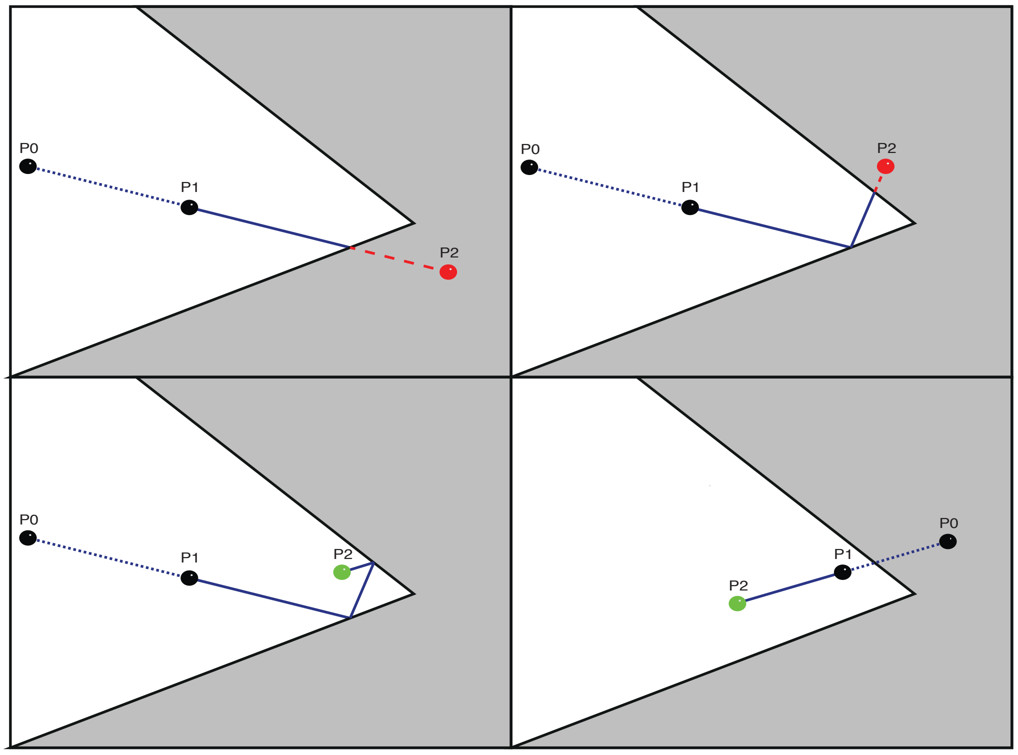


Figure 15: Positions calculated from Verlet integration (top left) corrected P2 position for intersection (top right) corrected P2 position after another boundary intersection (bottom left) and updated positions for next Verlet step with adjusted P0 (bottom right)

Import of volumes

An .obj file importer was implemented which allowed the author to import any 3D mesh rather than having to hardcode vertex positions and connectivity. As previously described, the intersection test works between a ray and a triangle but rather than limit the meshes that can be used, the intersection test is able to divide a polygonal face made up of any number of vertices into triangles. Any mesh can be imported but for the volume filling to work correctly the mesh should be manifold with welded vertices.

In order to make the intersection tests more reliable (in terms of rounding errors) each face has one direction in which the particles bounce off - which is defined by the order in which the face vertices are listed in the .obj file. When exporting a mesh from Rhino for example (other modelling software could be used - .obj is widely recognised) it is a straightforward process to "flip" the normals and therefore make sure that the particles are contained within the required regions. The bounce directions are displayed in the software so the user can check the model is set up correctly.

Fixed particles

One of the initial aims was to be able to include fixed points, as this will provide the freedom to create many different arrangements whilst ensuring the structure always fits in with the surrounding supporting or supported structure.

Three options to achieve this were implemented. Firstly, all of the face vertices from the volume mesh can be used to create fixed particles at each position. This is similar to the bubble meshing technique with the option to interactively remove some if required. The second option is to import an additional .obj file with support positions and then import the volume separately, which allows the freedom to create and edit support points (and test a number of options) without having to adjust the boundary volume. Finally, the user can create and interactively place a fixed particle during the simulation.

Ultimately, as the space frame is not limited to a regular arrangement this functionality can be used to create regular supports on irregular boundaries or vice versa. An engineer will be able to use this to focus or guide the loads imposed upon the structure to particular points.

Fixed particles behave in the same way as non-fixed particles in terms of the forces they exert but their positions are not updated by the Verlet integration step.

Connectivity between particles

Once the application had particles obeying Newtonian physics and contained within volumes the next step was to decide how they should interact with each other and any other forces we want to simulate in the environment. To transfer these forces between particles and to produce the final stable structure a connectivity network was required.

As discussed the members for the stable structure would be calculated using DT. An incremental construction algorithm (Cignoni, Laforenza et al. 1995) was implemented which builds up tetrahedrons from an initial starting vertex. The Delaunay topology is stored as an object that extends the Mesh class with an additional data structure containing all the triangular face and the tetrahedron objects. The objects are automatically self-referencing so it is possible to choose any face and traverse to the tetrahedrons on both sides and also the vertices and newly created edges from the original mesh that the Delaunay was derived from. Hull faces (those with a tetrahedron on one side only) are distinguished from internal faces and stored in a separate array. The option to re-calculate the DT every frame or fix the topology was implemented – both were used, depending on the current state of the particles, see Solid state section below.

To test both this implementation of the DT and prove its viability for re-creating truss arrangements, a series of tests were carried out. As expected the square faces of the octet truss were triangulated but apart from that it gives a satisfactory result, see Figure 16.

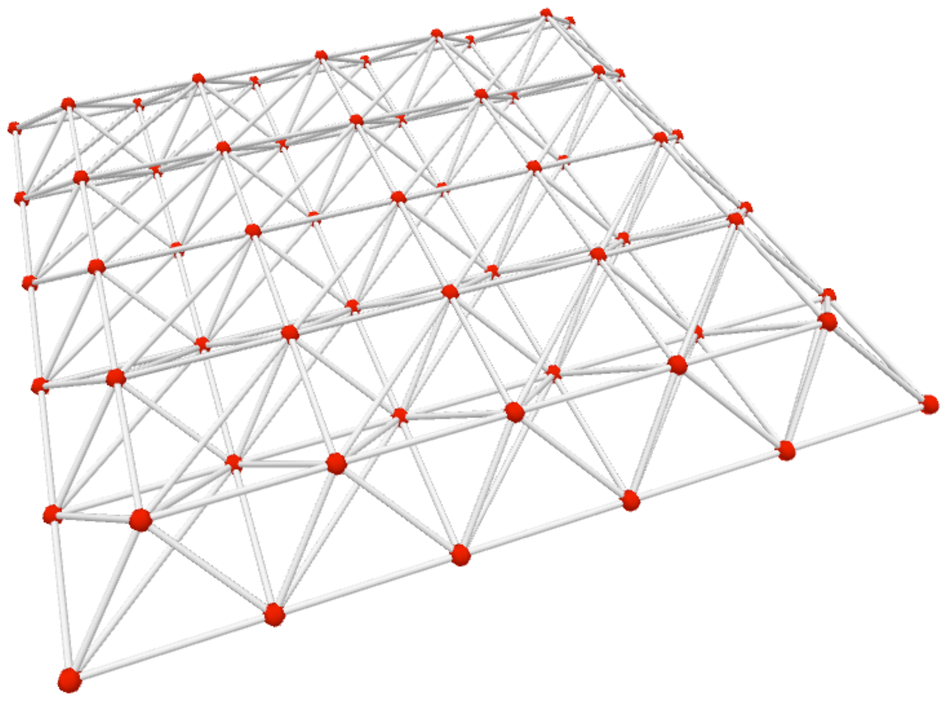


Figure 16: Test of Delaunay tetrahedralisation on a standard octet truss arrangement of nodes.

There are issues with Delaunay that must be resolved. For example it will currently create a convex hull around each of the points, which will lead to edges travelling through a concave boundary (even if particles are constrained within). To fix this we must adapt the DT in a number of ways.

Adapted Delaunay tetrahedralisation

Boundary recovery in Delaunay algorithms is a way of adapting the input so that the boundaries become part of the triangulation and are therefore maintained. Vertices are recursively added along the boundary edges and/or faces until this is achieved (Shewchuk 2002). However this is unsuitable for a number of reasons – particles should not be fixed to the surface of the volume, exactly matching every face and edge of the volume is not desired (just to stay inside them) and particles should not be automatically added during the simulation (as this will have a negative effect on reaching a stable equilibrium). The solution was to allow all of these Delaunay edges to be created and then hide all that go through a boundary (using a similar ray-triangle intersection test as for the bounce algorithm). The reason these edges were hidden rather than deleted is explained in the Forces on particles section.

If two adjacent particles are either side of a concave edge, despite the edge travelling slightly through the boundary it is quite possible the user will want to keep the edge as part of the DT. To let the user decide, a slider was implemented which allows edges below a certain adjustable length to remain, see Figure 17 showing this situation and the effect of the slider.

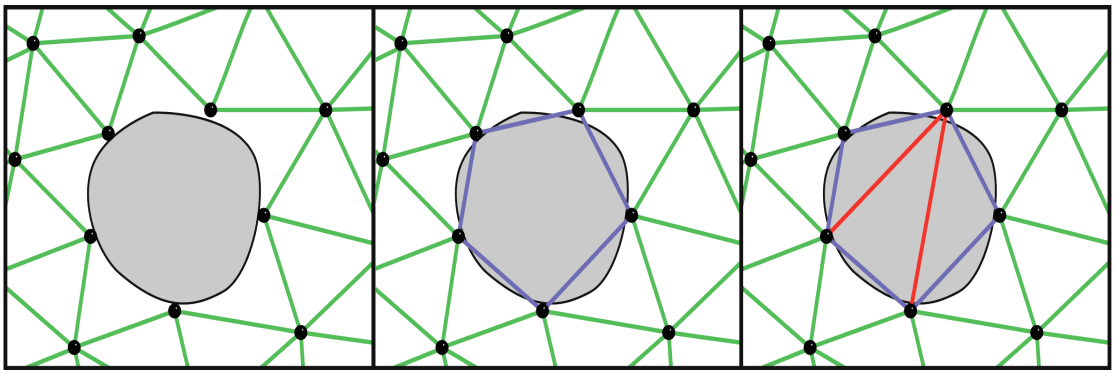


Figure 17: Effect of increasing the allowable intersecting edge length through a boundary. Zero allowable edge length (left) increased allowable edge length providing support around the void (centre) further increased, questionably too far as edges cross the void (right).

A related problem also arose with fixed points that were on the boundary. There are three conditions for when two fixed boundary particles are connected:

1. On different faces and the edge travels totally within the volume (this edge should be kept).
2. On the same face so the edge travels directly on the boundary (this edge should be kept).
3. On different faces on a concave mesh and the edge travels entirely outside the volume (this edge should be hidden).

If a particle that is exactly on a face is said to have not yet intersected it then all the edges will fail an intersection test and will be kept. Conversely, if as a particle touches a face an intersection is said to have taken place then all the edges will be hidden. There are a number of ways this problem could be solved – the direction of the edge could be compared to the normal of the face and any with an angle equal to or greater than 90° (normals point away from the side of the particles) would be kept. Or the midpoint of the edge could be checked to see if is outside our volume and hiding if so. Both of these require extra computation every frame and should be avoided. The solution was to offset each fixed particle a small distance into the volume. This has to be executed just once (when the fixed particles are created) so is computationally more efficient and even though the offset is small, the original exact position is stored and reverted to when the final structure is exported from the application.

A direction for the offset had to be calculated that always points inside the mesh. This process was complicated by the concave and irregular topology of the mesh and the functionality that allows the user to import fixed points anywhere on the surface and inside the volume. An algorithm was implemented to identify and deal with each of the possible different cases; these are where the fixed particle is:

1. not touching any faces (no action required).
2. in contact with a single face.
3. on the edge between two faces.
4. on a vertex between three or more faces.

For case 2 the direction is simply taken as the inverse of the face normal. For case 3 the inverse of the average of the two face normal is taken. Case 4 was more complicated, initially the same method from case 3 was implemented (with more faces) however it quickly became apparent that this did not always produce the expected result and in some cases would push the particle outside of the volume. The problem was that each face had an equal input in the final direction and for meshes such as those shown in Figure 18 the denser mesh on one side will overcome the single face and point outside the volume. As one of the aims was to be as flexible as possible with the imported meshes to be filled this had to be solved. To achieve the desired result a weighted vertex normal algorithm (Jin, Lewis et al. 2005) was implemented. This takes each face that the fixed particle is touching in turn and multiplies the normal by the angle between the two adjacent edges (see **θ** in Figure 18). The result for each face is added and we take the normalised inverse of the result.

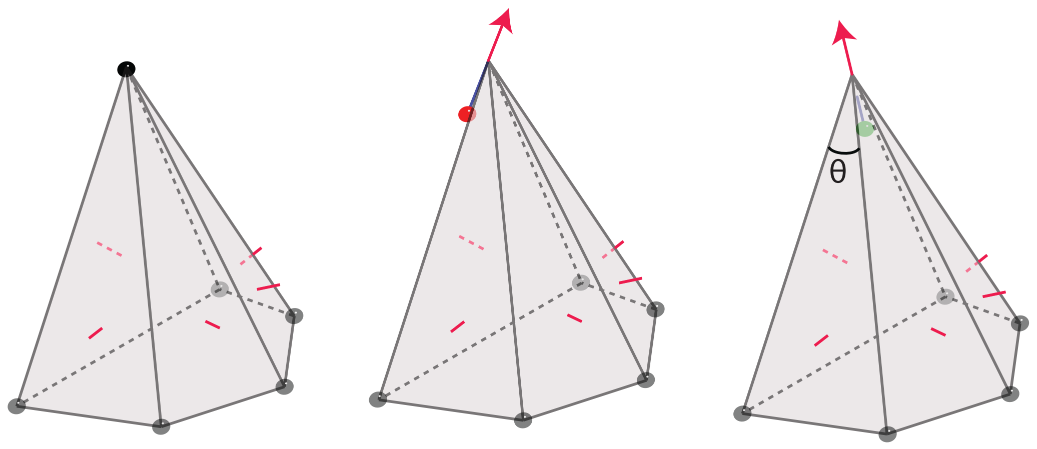


Figure 18: Fixed vertex to be offset into the volume shown in black (left) vertex normal offset calculated by averaging face normals (centre) vertex normal offset calculated by averaging face normals adjusted by angle between adjacent edges (right).

The DT is the linking of four vertices whose circumsphere has the smallest radius, producing the greatest possible angles between edges and thus avoiding flat tetrahedrons. However, there are instances where extremely flat and undesired tetrahedrons are part of the valid solution and unfortunately the process of pushing particles up against planar faces is a certain way to produce them. These tetrahedrons, although correct for DT, will not contribute structurally as the edges are disproportionately long (therefore prone to buckling) and will complicate erection and aesthetic and therefore should not be included in our mesh.

The situation occurs as four particles are pushed up against a boundary (but all slightly different distances from it). As there are no other particles beyond the boundary to form a more equilateral tetrahedron they link together forming an almost flat tetrahedron with a very large circumsphere. To avoid this behaviour any such tetrahedrons should be rejected and the face (that would have been the base of the tetrahedron) should be set as a hull face during the DT construction sequence. If these tetrahedrons were to be identified by circumsphere radius alone this could (for the same radius threshold) reject a large equilateral tetrahedron and accept a small but flat tetrahedron. A better measure is to take the ratio of the circumscribed sphere and the inscribed sphere for the tetrahedron. This function approaches zero for very flat tetrahedrons with a maximum of 1/3 for an equilateral. This was implemented with a slider to control the cut-off threshold during the simulation.

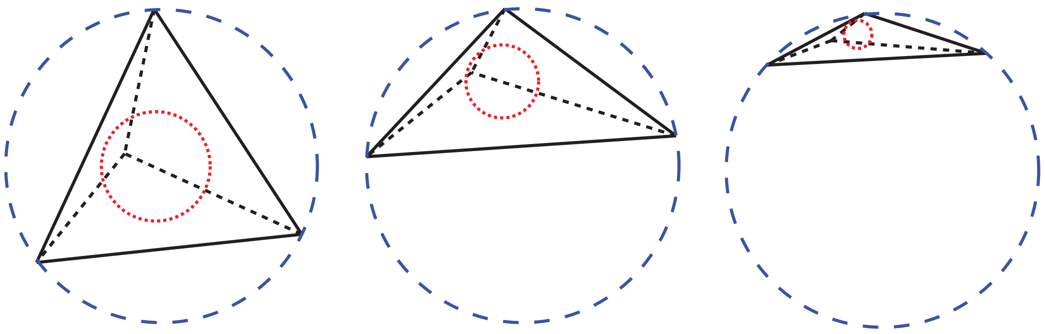


Figure 19: Comparison of three tetrahedrons with the same circumsphere radius (shown in blue) but decreasing inscribed sphere radius (shown in red) as a result of increasing flatness.

Forces on particles

As described in the Development of the physics simulation section, for each step of the simulation a force to be imposed for each individual vertex is calculated and then used in the Verlet integration step. This force can be a combination of a number of different influences and any function can add its own force vector to the running total. It is reset each step (any momentum is calculated from the Verlet) and new updated forces are found depending on the new positions of the vertices. To simulate matter moving between different states we can use different combinations of forces and connectivity routines. See Figure 20 for a chart highlighting the active forces and differences in the states.

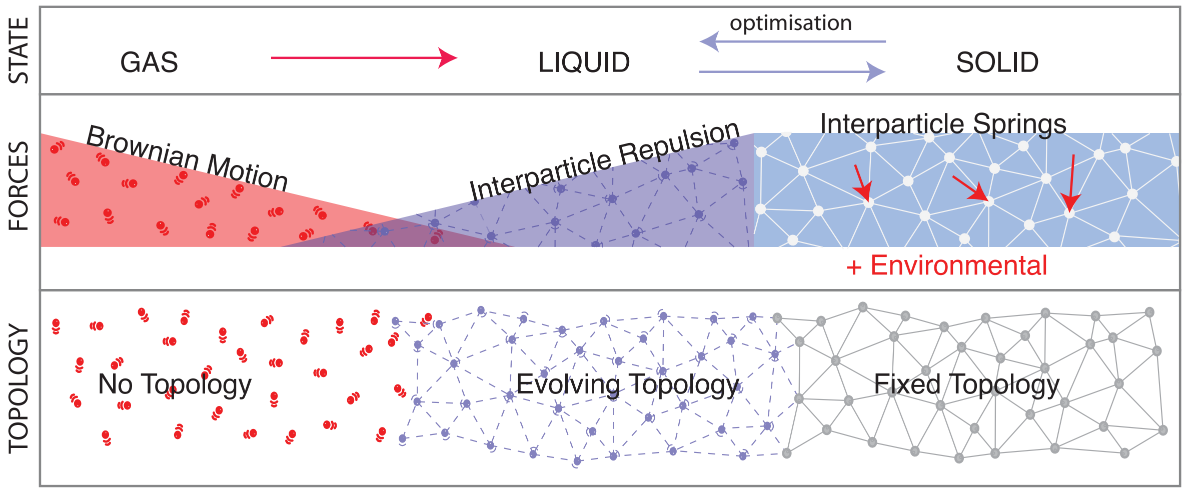


Figure 20: Chart showing the transition between the different states of the simulation and the different forces and topology.

As can be seen in Figure 20 there are four forces to simulate – Brownian motion, interparticle repulsion, interparticle springs and environmental.

Brownian motion

Brownian motion is the motion of a suspended particle as it is bombarded by millions of gas particles. Since far fewer particles are to be modelled similar behaviour can be invoked by directly placing a random force vector on each particle. This will simulate the motion of gas particles filling a volume. The magnitude of the random force can be thought of as an analogy to the temperature of the environment as hotter particles will move faster and therefore further each step. Pseudo-code for the implementation of this force is as follows;

```
void brownianMotion() {
        for(Particle p:particles) {
                if (p.fixed==false) {
                        Vector force= forceVectors.get(p.index);
                        Vector randomForce = randomVectorMagnitudeLessThan1();
                        force.x += randomForce.x*temperatureFactor;
                        force.y += randomForce.y*temperatureFactor;
                        force.z += randomForce.z*temperatureFactor;
                        }
                }
        }
```

This simple step recreates the behaviour effectively. An example of a torus behind filled in this manner is shown in Figure 21, a fading trace showing the previous 100 positions are shown in the image to highlight the motion.

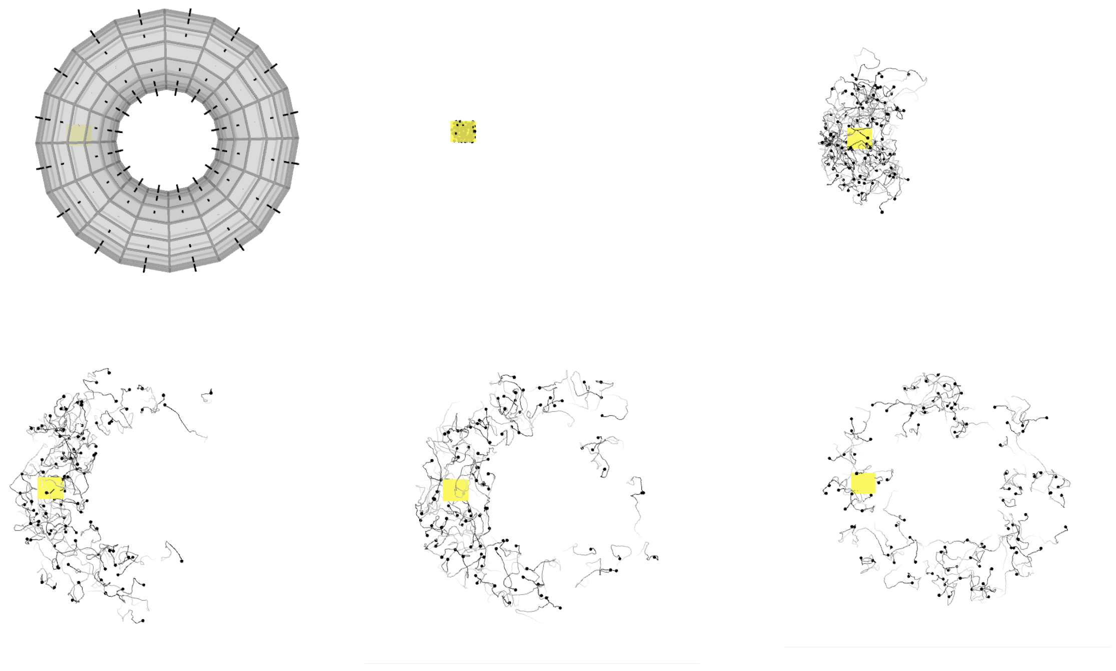


Figure 21: Sequence showing a torus being filled with 100 particles under the Brownian motion force only.

By using Brownian motion with no inter-particle forces, complex volumes can be filled effectively. An initial investigation found that having inter-particle forces during the first stages of volume filling created bottlenecks or blockages in complex volumes.

Inter particle connectivity

The first step to modelling inter-particle force is to decide which particles exert a force on which others – the force topology. Every particle can interact with every other particle, however this is an n^2 operation and (assuming that distance will have some effect on the magnitude of the force) the vast majority of particles that are picked up by this will exert an insignificant or zero force. Every particle within a certain range would be connected but this still requires each particle to check every other one to find the distance (to speed up this process a kd-tree could be implemented, see Moore (1991)). It was decided that a far more efficient result could be achieved by using the existing Delaunay structure to find the nearest neighbour. There were challenges in maintaining flexibility while using this method, for example how to connect to more than just the directly Delaunay connected particles but ultimately a number of advantages to combining in this way were found see Boundary force reduction.

When an edge is created it adds a reference to itself to the two particles it joins. This was adapted so that each particle also has a reference to the particles at the end of all of its connecting edges. This means that once the DT is complete a list of all the particles that are directly connected to each particle is produced. In order to extend the range of the forces, the number of "steps" through the DT can be increased. If the number of steps is set to one then the force affects only the directly connected particles. If this number is increased to two then the force will affect all the directly connected particles and all of *their* directly connected particles, see top right image in Figure 22. A Boolean array (the same size as the number of particles in the simulation) is used to check whether a particle is already in the list of affected particles and ignored if so. The range can be taken to any number of steps however the exponential nature of the process means within 3 or 4 steps every particle is often connected to every other particle. If this behaviour were desired it would be more straightforward to ignore the DT and add them all directly.

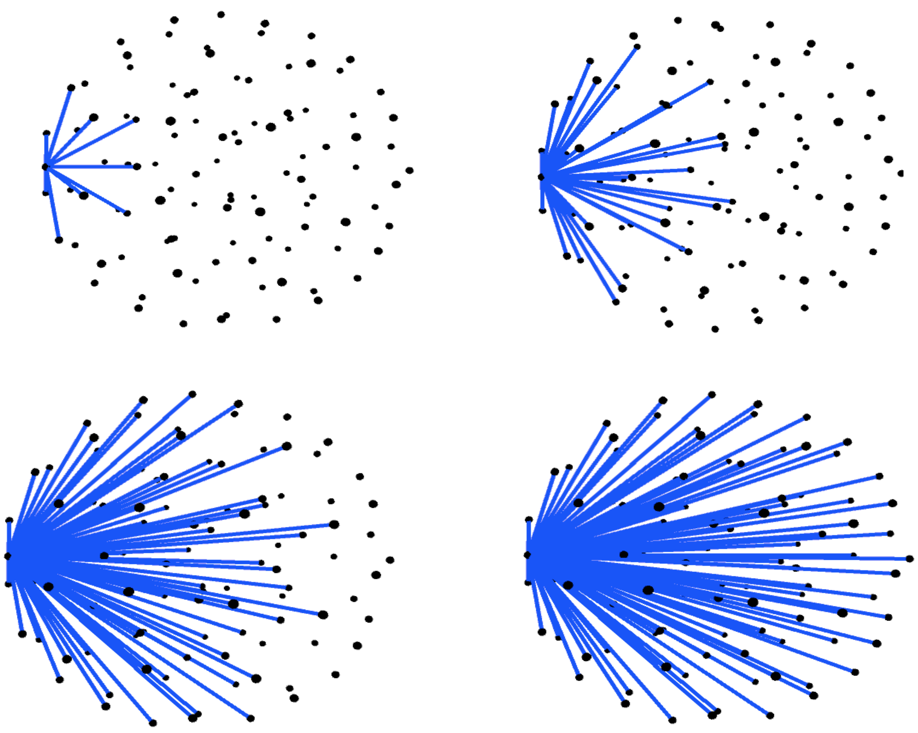


Figure 22: Effect of increasing the number of steps of influence for the inter-particle forces (blue line identifies two particles exerting a force on one another).

During the simulation an interactive slider allows the user control the range of the force and move through multiple steps of the structure. See Figure 22 for a sequence of images showing the range of influence as the number of steps is increased from 1 to 4. The appropriate range will depend on the type of force being used (see 4. Initial testing and calibration of software) and which state the simulation is in. Having the forces transmitted in this way should be considered a computational efficiency step rather than a reference to any particular physical particle process.

The exact method for calculating the force is established in the next section (after various functions between distance and force had been tested). In order to maintain volume-filling behaviour during the liquid phase and avoid the need to calculate rest lengths to suit the volume and particle density, only repulsive forces are used in the liquid state. Once the solid stable state is reached a more rigid structure modelled on stiff springs is used.

Boundary force reduction

For concave volumes it was initially assumed that if the connection between two particles intersects a boundary then that force is ignored. However, if the aim is for the particles to settle and achieve a stable solution during the liquid phase (where there is no fixed topology) this method can cause instabilities. A situation can arise where, as two particles move around a concave corner (that is cancelling the repulsive force), they will be able to reach a proximity that leads to a very large repulsive force when they finally move past the influence of the boundary, see Figure 23. This can causes the particles to shoot away de-stabilising the rest of the simulation and creating a vacuum for other particles to fill and possibly repeat the problem.

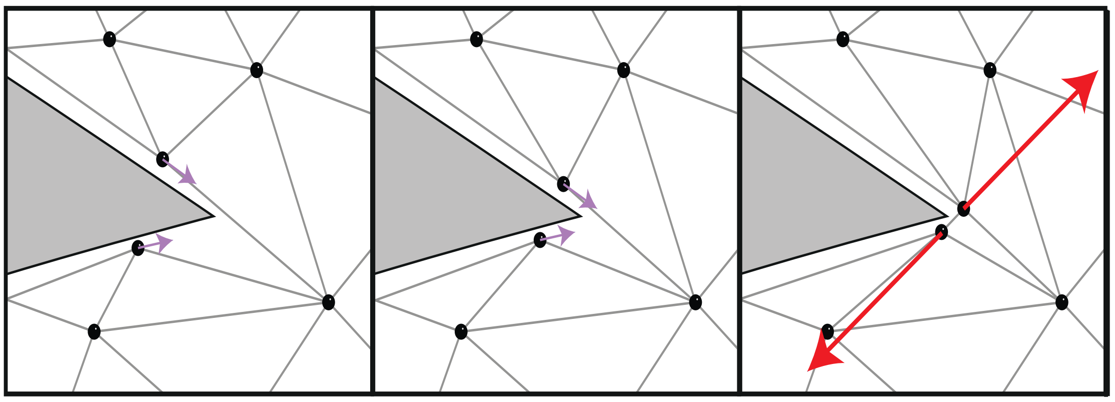


Figure 23: Sequence showing the behaviour of particles as they move around a concave edge and exert a large repulsive force, which can lead to instabilities.

If forces through the boundaries are allowed this problem will not arise, however particles either side of an internal void or concave corner could repel each other enough to stop the particles reaching the boundary and therefore not support the whole surface as required. In order to strike this balance a reduction factor was implemented. By dividing the connection between particles into inside and outside sections, and multiplying the outside section by a reduction factor, an increased distance is found which is then use to calculate the force, see Figure 24.

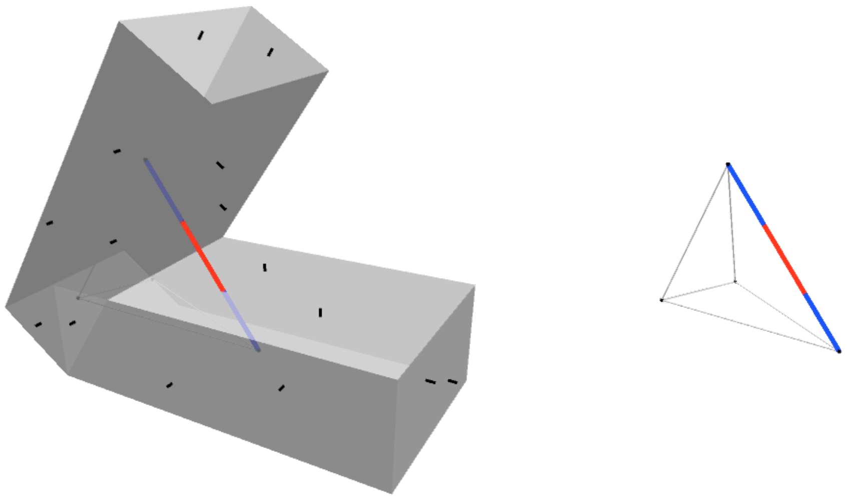


Figure 24: Intersecting force topology divided into inside (blue) and outside (red) sections

This will always give a gradual increase of the force as they move around a concave corner rather than a sudden large force. It also creates a relative vacuum or low pressure area inside internal volumes which attracts the particles (or rather repels them less) so the void is well supported by structure

As both the adapted DT and the force factoring functions needs information about edge-boundary intersections this information was shared whenever possible. For example, during the liquid stage the DT is carried out, then the force factoring checks are made, which marks the edges that intersect and adjusts the forces and finally any edges that were marked are hidden for the adapted DT.

Solid state

Fixing the topology and replacing each DT connection with a spring, of natural length equal to the distance between particles, achieves the solid state. These springs provide the forces for the Verlet integration each frame. The strain in each spring is calculated and multiplied by the stiffness (set by the user) to give a force. Then for each particle (which is now essentially a node in a rigid space frame) each connecting edge's normalised direction is multiplied by the force and summed. This force is then added to the particles force vector in the force vector array for this step in the simulation.

Visual feedback of the stresses in the structure is provided during the simulation via a node size and edge thickness increase with the stress in the member and coloured red to highlight members in tensions and blue for members in compression, see Figure 25.

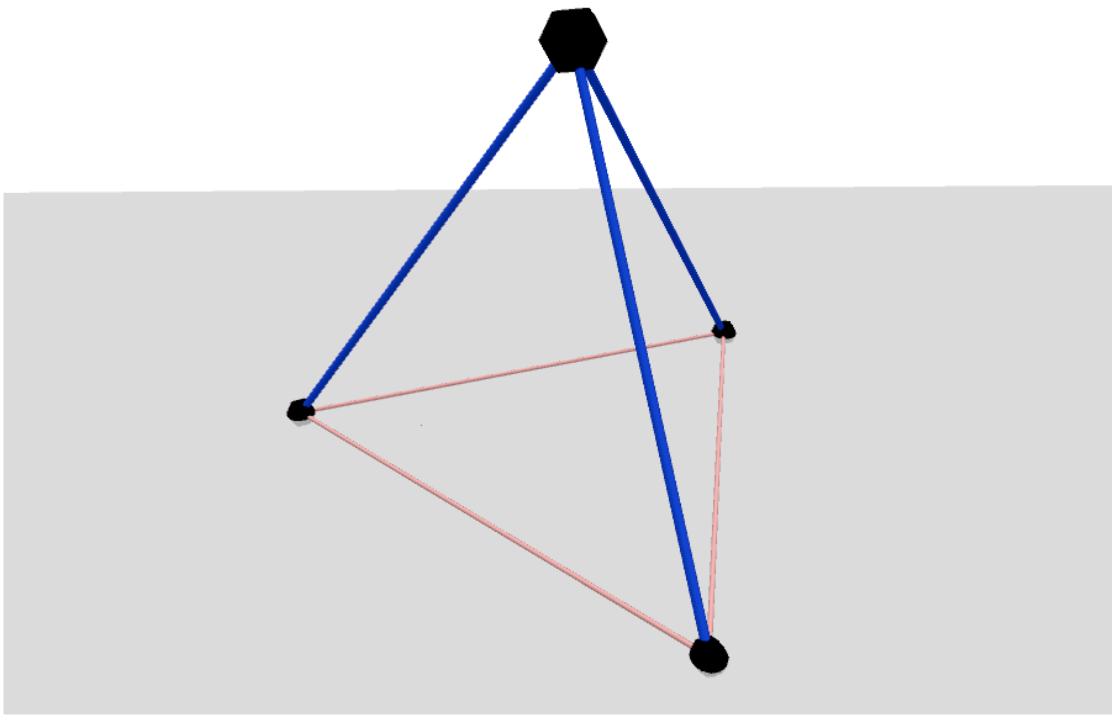


Figure 25: Equilateral tetrahedron in the solid state with self-weight under gravity, compressive members shown in blue, tension in red. Node size represents the relative strain in the node's connecting edges.

In the solid state the mass of the connecting springs must be taken into account (as this is the self weight of the structure). The mass of the node is taken as half the total natural length of all the connecting springs multiplied by a fixed weight per unit length for each spring.

The springs simply resist forces imposed upon them – therefore if no external forces are applied the structure stays in equilibrium with no stress in the members. Any additional forces were previously described as environmental forces and the most fundamental of these is gravity - which can be implemented by simply adding a vertical acceleration to each node during the Verlet integration. In fact any force in any direction can be applied to the structure in this way. For this research, only gravity and self-weight are applied but the framework could easily be adapted to apply area loads (such as a wind or floor load) by calculating the amount of load to impose on each node from the area of the triangles it touches.

With the topology fixed and springs holding the structure in place the boundary can be removed and the structure can start to deform. The framework that was previously used to contain particles within a boundary can now be used as a surface to interact with the structure. For example, as seen in Figure 25, a simple flat ground plane has been imported and under gravity a tetrahedron has been “dropped” onto it.

During the solid state, fixed particles act as supports for the structure. In most cases this is the required behaviour as the fixed particles are often used to allign the structure with pre-defined support points. However, fixed particles can also be used to align the structure for aesthetic purposes or with connecting but not supporting features such as glazing mullions. To allow this each fixed particle can be set as fixed in both the liquid and solid phase or only in the liquid phase.

Particle management

A particle emitter was implemented which allows the user to interactively create new particles in the simulation. It is shown as a yellow cube (see Figure 21) which can be dragged and placed anywhere within the simulation. The particles are randomly seeded within this cube (the range can be adjusted) to ensure they are not placed in the exact same position. This was done because the DT containing two particles in the same position does not create an edge between them - therefore no inter-particle force is created to move them apart. With the flexibility to move the emitter between volumes multiple separate space frames can be create simultaneously.

Particles can be selected individually for deletion or randomly from the simulation to decrease the density once they have already been spread throughout the volume. This random deletion ignores any fixed particles.

4. Initial testing and calibration of software

This section outlines the testing and calibration that was carried out to prove the concept and determine how the inter-particle forces should be calculated.

Brownian motion

It was previously shown how the Brownian motion could fill a torus; to confirm that the particles will fill a shape with a bottle neck a further series of tests were carried out in which a function was added so when a particle hits a face, the face is marked red, see Figure 26.

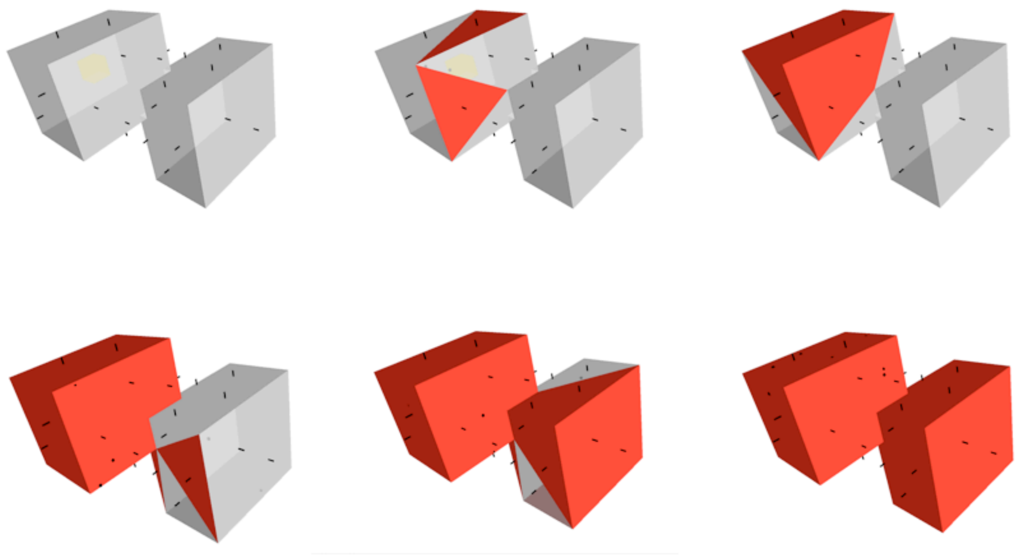


Figure 26: Sequence showing particles under Brownian motion passing through a bottleneck and hitting every face in a boundary.

The test was carried out on two overlapping cuboid volumes with a central bottleneck and the Stanford bunny (see Appendix B). For both tests all faces were reached by the particles. The Brownian factor which effects the "temperature" of the system can be manually adjusted depending on the size and shape of the volume and the number of particles.

Inter-particle forces - liquid

Previous studies (see 2. Literature review & previous work) had varying and contradictory results for various inter-particle forces so independent tests were required. In order to calibrate the system the aims must be explicitly defined. At this stage (i.e. before any optimisation routine) the aims for the final arrangement were defined as:

1. As equal edge lengths as possible (throughout the mesh).

2. Each tetrahedron to be as close to equilateral as possible.

An estimation of the degree of success of aim 1 can be measured by the standard deviation of the lengths and 2 can be measured using an equilateral fitness, ε_g:

$$\varepsilon_g = \frac{1}{m} \sum_{i=0}^{m} \left(A - \frac{r_i}{R_i}\right)$$

Equation 1

where m is the total number of tetrahedrons, if A is a constant (set as 1/3 if we are measuring tetrahedrons), r_i is the radius of the inscribed sphere and R_i is the radius of the circumscribed sphere. Equation 1 from Shimada and Gossard (1995), it should be noted that A is incorrectly stated in the paper as $\sqrt{}(2/11)$ rather than 1/3 for tetrahedrons. Using this equation, a mesh made up of perfect equilaterals would result in $\varepsilon_g = 0$. Any variation from an equilateral will reduce the value of ri/Ri and therefore the greater the value of ε_g. Although these two results are measuring similar aspects of the mesh (an equilateral tetrahedron will have equal edge lengths), check 2 does not take into account the size of the tetrahedron (which at this stage we want as consistent as possible) so a combination of the two is a good guide for mesh quality.

In order to test and calibrate the inter particles forces a series of tests were carried out. The chosen volume to fill for the tests was a high-resolution sphere for the following reasons:

1. The shape has a minimal effect on the shape of the tetrahedrons – in contrast, a cube will introduce 90° angles in the corners and edges which are not conducive to equilateral tetrahedrons.

2. As the particles move to reduce inter-particle forces they will move into "valleys" in the boundary mesh, the higher resolution and the more consistent the curvature the lesser this effect will be.

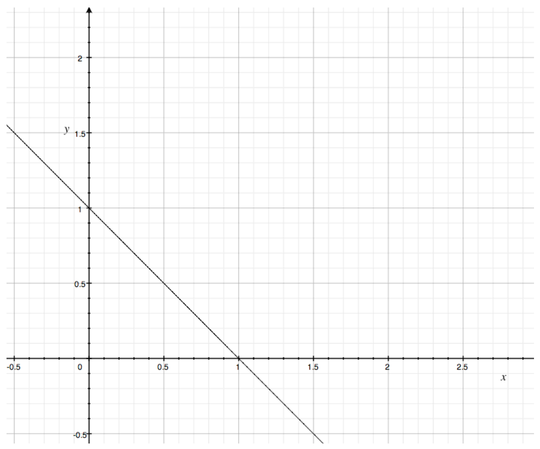
It should be noted that the behaviour of a cube affecting the shape of the boundary tetrahedrons (as highlighted in reason 1) is not a negative aspect for the form filling but simply negative for the test. For the final solution boundaries should certainly affect the form of the tetrahedrons.

For each test, 64 particles were randomly placed in the centre of the sphere and both Brownian motion and the tested force was applied to the particle. Then the Brownian motion was removed and the structure was allowed to reach equilibrium, at which point the values for the edges lengths, population standard deviation and the equilateral fitness were recorded and an image the final form saved. Three tests were taken for each force. To test the algorithms and implementation of the tests they were run on an equilateral tetrahedron - as expected the standard deviation was approaching zero ($<10^{-5}$) and the equilateral fitness was exactly zero.

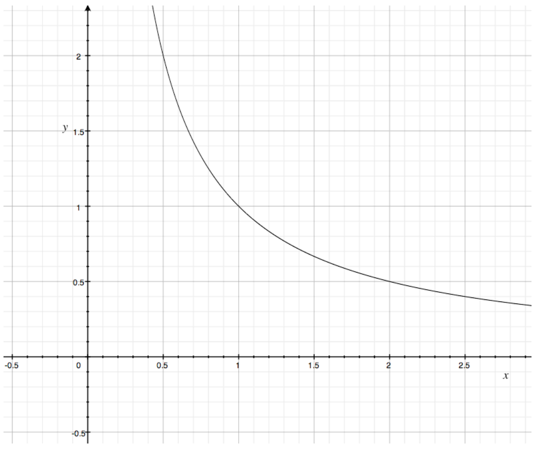
For the liquid stage, only forces with no attractive force component were implemented. This was done for two reasons. Firstly, if there are attractive forces and the "natural length" of the force is too small the particles form a stable ball in the centre of the volume and will not push against the boundaries. This problem can be overcome with an interactive or automated variable spring lengths or by adding more particles to fill the volume. However, as previously discussed we do not want particles to be added or removed automatically and by having repulsive forces based on inverse power laws the volume should always be filled whatever the strength of the force (although the rate may be very slow for weak forces).

Secondly, as the constantly updating DT can suddenly connect two distant particles, if there are attractive forces these are pulled towards each other (very strongly if the force is based on distance), this is problematic for concave shapes since, as the two particles pull towards each other, their previous positions are filled by other particles and the cycle starts again. These situations do not lead to a stable solution.

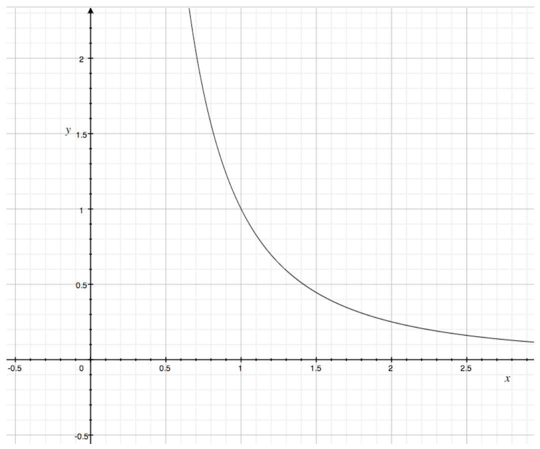
Six different repulsive forces with varying force-distance relationships were tested; see Figure 27 for force-distance graphs for each. For each test an interactive force factor could be adjusted to scale up the force to suit the scale of the test – the important factor being tested is the fall-off of the forces. See Appendix C for images of the final mesh for each the tests.



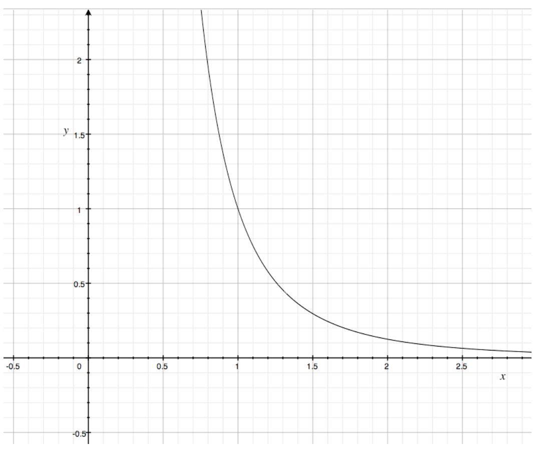
Force 1: y = 1 – x



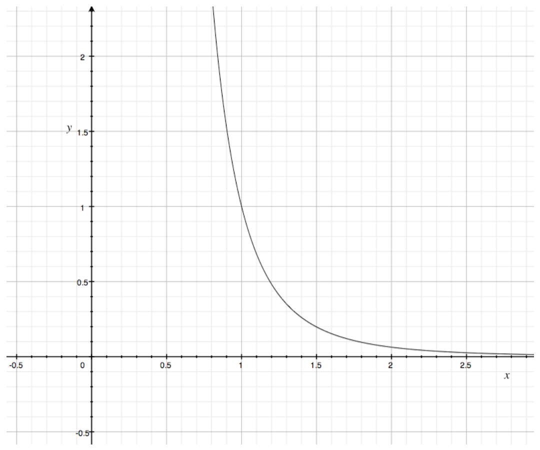
Force 2: y = 1 / x



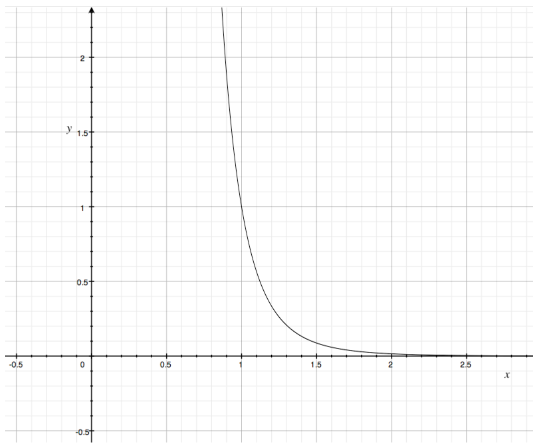
Force 3: y = 1 / x^2



Force 4: y = 1 / x^3



Force 5: y = 1 / x^4



Force 6: y = 1 / x^6

Figure 27: Force-distance graphs for the six functions tested (force on y-axis, distance on x-axis).

During testing it was immediately clear that force 6 was not suitable - the fall off of the forces was far too rapid which this meant the particles were not at all well spaced and the mesh was particularity unstable when new particles are added, therefore the tests were not concluded. As can be seen from Figure 28 the best performing were forces 1 and 5. It is proposed forces 2 and 3 did not provide a quick enough falloff of force strength whilst still having a very large maximum.

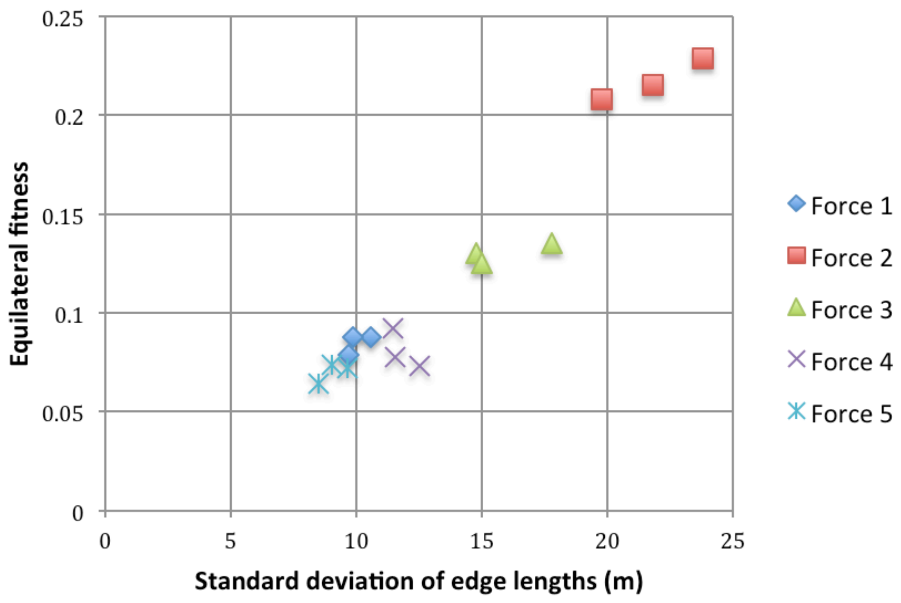


Figure 28: Comparison of equilateral fitness and standard deviation of edge lengths for each force.

In order to further differentiate between forces 1 and 5, more tests were performed, this time filling a tetrahedron rather than a sphere (again with 64 particles). In these tests the power law performed considerably better; this can be attributed to the increase in force as particles get too close – with power laws this is strongly resisted where as with a linear force particles are able to settle much closer. This behaviour is most noticeable between fixed points. See Figure 29 which shows particles along the edges with visibly better spacing. This observation is supported by the equilateral fitness's of 0.12 for force 1 and 0.59 for force 5.

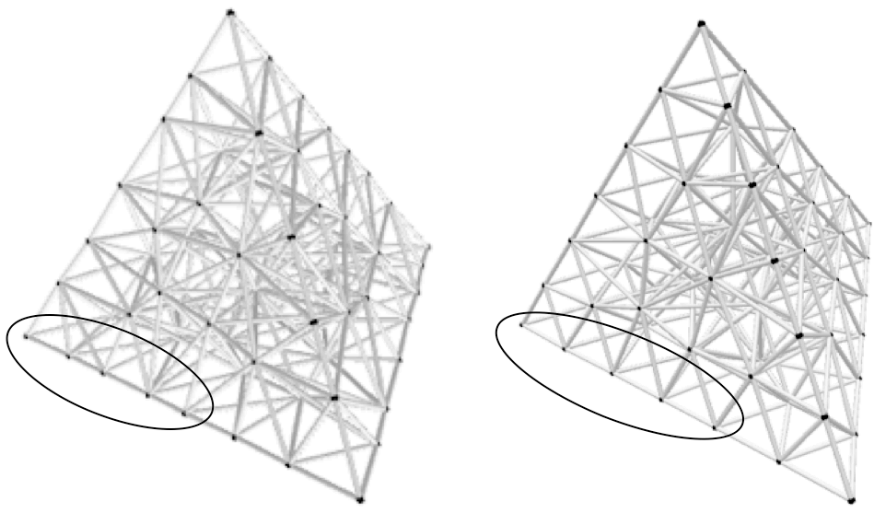


Figure 29: Comparison between linear and power law forces filling a tetrahedron, force 1 (left) and force 5 (right).

It was concluded that while there is potential for further study in analysing different force-distance functions, force 5 provided a very good functionality to test the framework on a case study and attempt to optimise the forms. A cut off value for very small distances was included to avoid the extremely high forces that would result is two particles get too close (this usually only happens when particles are added).

A number of further examples of volume filling using force 4 can be seen in Figure 30 and Figure 31.

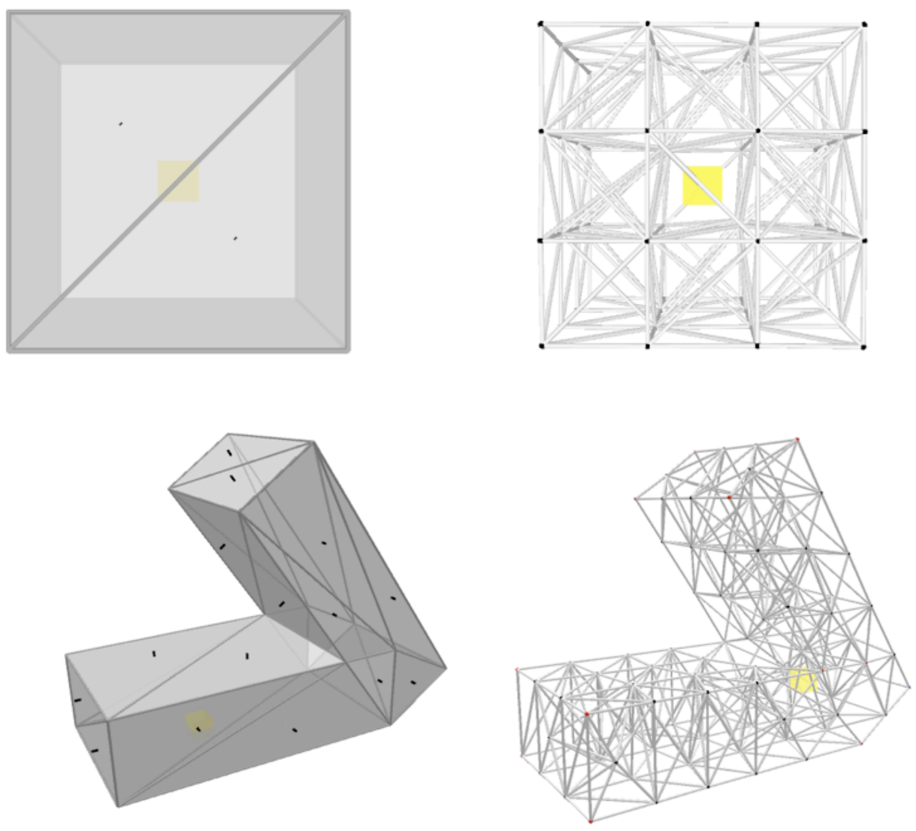


Figure 30: Volume filling examples. Original imported volumes (left) and resulting space frames (right).

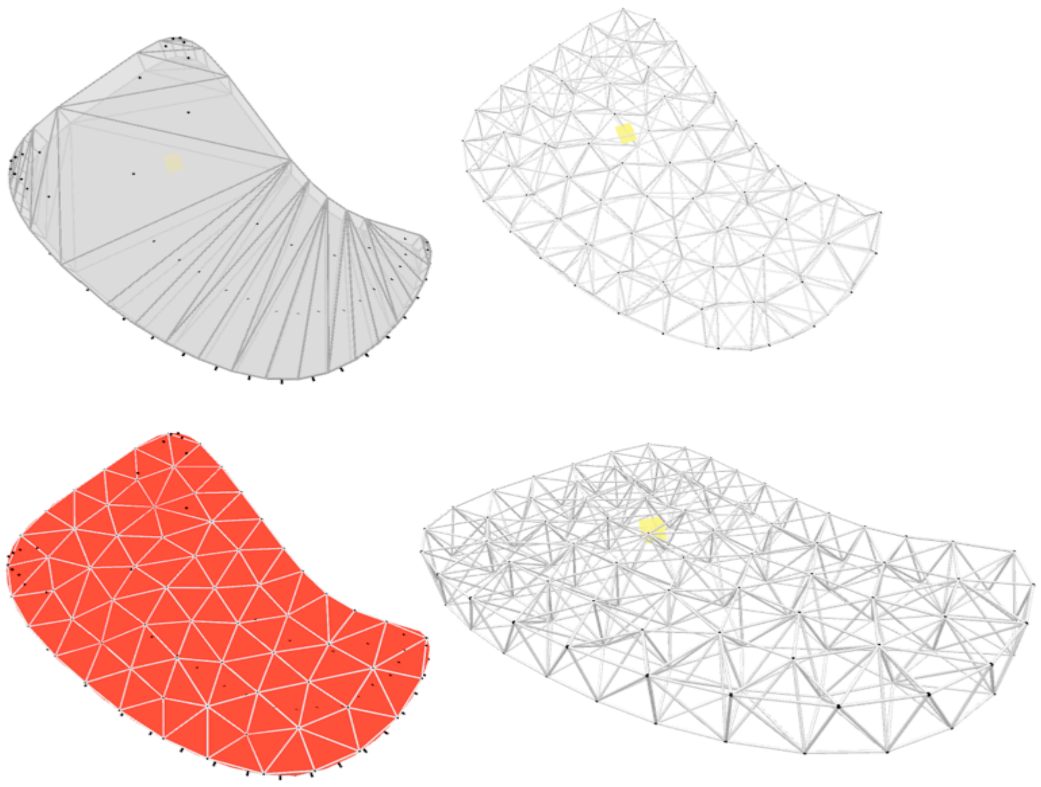


Figure 31: Irregular flat boundary volume (top left) used to create a single layer space frame (top right). Top layer highlighted (bottom left) and isometric view (bottom right).

These results show a good distribution of particles and compare favourably with previous volume filling space frame generation systems, see Figure 7.

Inter particle force – solid and environmental

To test the solid phase, a stable mesh was produced from the gas and liquid phase, the DT fixed and all edges changed to springs with their current length set as the natural length. The fixed supports were chosen (or imported), the boundary volume was removed and gravity was applied.

The structure deflected as expected with the highlights for tension and compression working correctly. The stiffness of the members and the gravitational force and weight of the structure (and therefore the stresses and deflections) are relative and not currently based on real values. This is reasonable for the purposes of this research as the stability of the structure and the areas of high and low stress are still valid. The testing of the solid phase confirmed that the process creates stable structures.

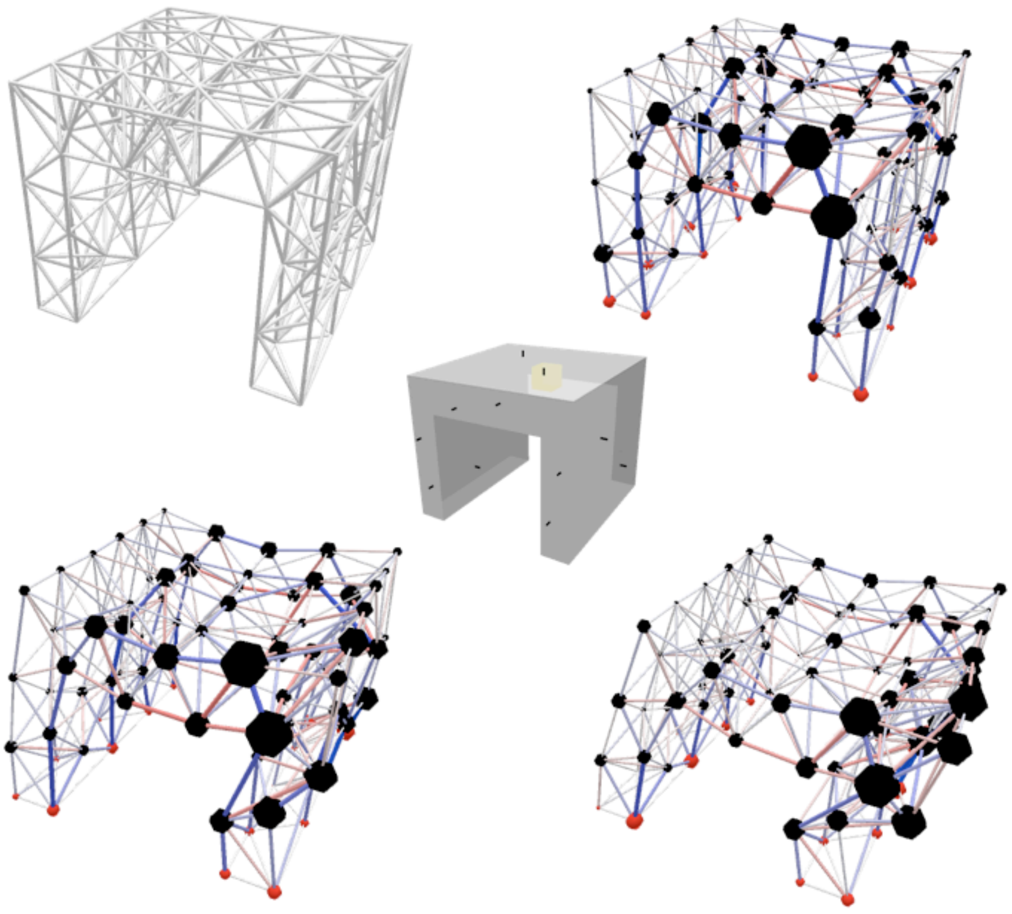


Figure 32: Solid model created from central volume, unloaded, then with increasing gravitational force until collapse.

Figure 32 highlights an interesting element of the process – asymmetry. For certain forms, there may not be a symmetrical stable arrangement of particles – for example an uneven number of particles filling a cube. It was also noted that running the volume filling process twice with identical settings did not always produce the same structure. This is as a result of the random seeding and random element of the Brownian motion leading to different solutions. This highlights that each output from the software is *a* solution to the problem rather than *the* solution.

5. Freeze-thaw optimisation

Development

The solid state gives a clear picture of which areas of the structure are experiencing the greatest stress and deflection and conversely those that are not working to their full capacity. If this disparity can be evened out a more efficient structure should be produced. A process involving traversing back and forth between solid and liquid states named "freeze-thaw optimisation" was developed which is described along with results from a series of tests in this section.

As described in 2. Literature review & previous work optimisation routines where structural material is moved towards areas of high stress have shown some success in 2D and surface optimisation. If the length of highly stressed members is reduced this could pull structure away from the boundaries and therefore lose the support. Instead, if the relative repulsive forces are adjust in different areas of the simulation, the internal structure should still maintain the volume filling behaviour whilst becoming more or less dense in different areas.

Each particle object has a relativeForceFactor field, which remained equal to 1 up until the point of the first solidification. During the solid phase and application of the environmental forces the absolute strains are summed at each particle and now used to alter this factor. In areas of high stress this factor will be reduced and vice versa. To manage ranges of stresses the values for particle stresses were scaled into a range between 0 and 2, see Equation 2.

$$relativeForceFactor_i = 2 * \left(1 - \frac{particleStrain_i - particleStrain_{\min}}{particleStrain_{\max} - particleStrain_{\min}}\right)$$

Equation 2

A method for adjusting this factor by taking the average of each factor with the connecting particle was implemented. This was done to create areas, rather than individual particles, of high stress. See Equation 3.

$$averageForceFactor_i = \frac{1}{n+1}\left(relativeForceFactor_i + \sum_{j=0}^{n} relativeForceFactor_{connected} \right)$$

Equation 3

Where n is equal to the number of connected particles. Whilst the simulation is in the solid state these values are calculated continuously and stored as it thaws and changes back to a liquid. As the structure is not contained within the volume during the solid state (to allow it to deflect) when it thaws it must go back inside the volume. To achieve this the position of each particle is stored as the simulation freezes and reverted to as the simulation thaws.

As a result of this step each particle in the liquid stage has a relativeForceFactor (and averagedForceFactor) ranging between 0 and 2. For each force calculation between particles their respective force factors are averaged and the force (calculated by their distance) is multiplied by this value. Figure 33 illustrates the theoretical result of this process.

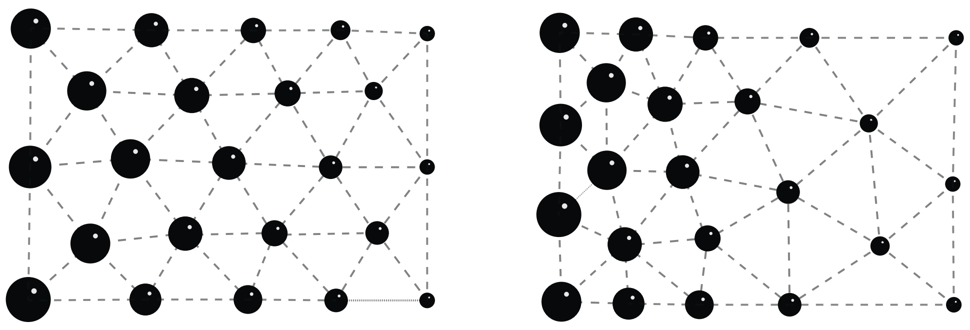


Figure 33: Structure in the solid state with each particle's relative total strain indicated by size of particle (left). Structure after thawing with reduced force factors in areas of high strain leading to a denser mesh (right).

To allow further control and fine-tuning of this process a further factor was implemented, the forceFactorPower. The relativeForceFactor is raised to the power of this forceFactorPower, which can be interactively controlled by the user. As the relativeForceFactor is between 0 and 2 this has the effect of increasing the relative difference in forces factors and therefore increasing the effect of the optimisation. If the force was simply multiplied by a factor this will have no effect of the relative forces and the mesh will stay the same but, if anything, become less stable.

Initial testing of optimisation

To quantify the effectiveness of the optimisation, tests were carried out on a cuboid under various support conditions. A single fixed support (cantilever), two fixed supports (see Figure 34) and simply supported. The maximum deflection for any particle and the total sum of deflections was recorded for six iterations of the freeze-thaw loop using a range of forceFactorPowers. Fixed particles were updated every iteration – for example, in the cantilever test, any particle touching the support face was fixed before loading applied (and released during the thawing process). The maximum deflection results for these initial tests are presented in Figure 35.

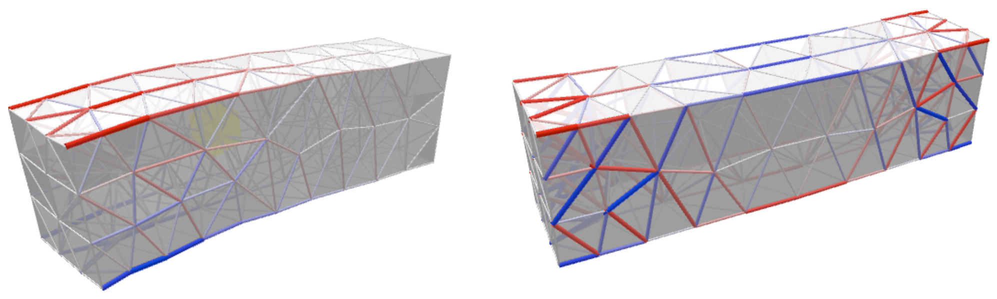
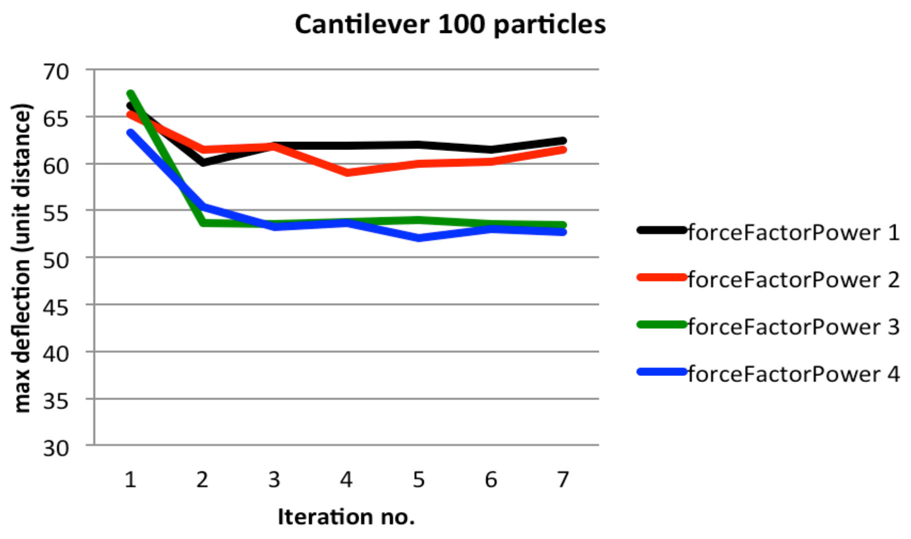
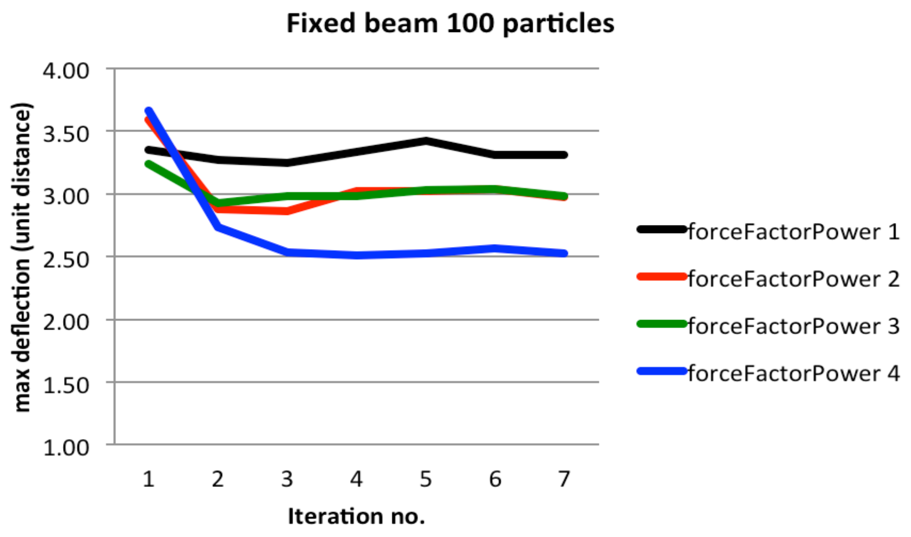


Figure 34: a cantilever (left) and fixed beam (right) in the solid state under gravity after three iterations of freeze-thaw optimisation. In both examples an increased density of particles can be seen near the supports.





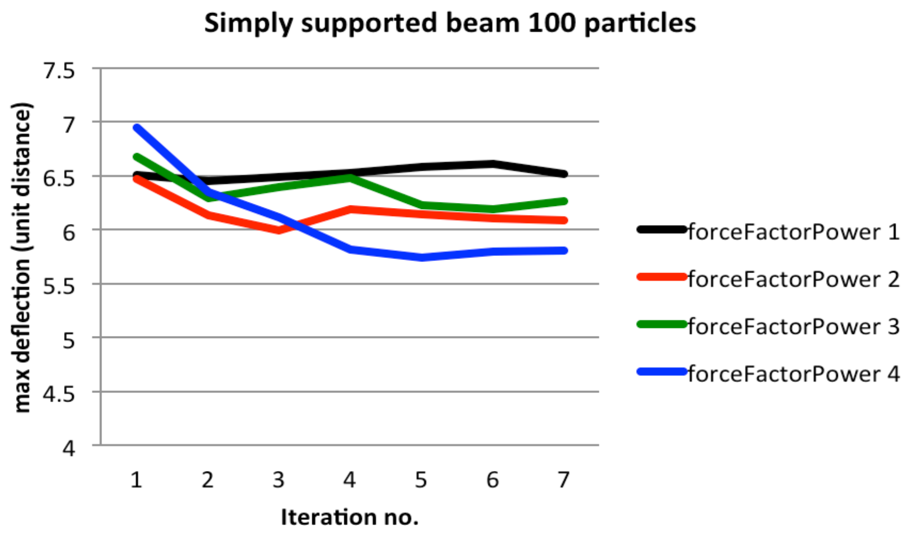


Figure 35: Results from initial testing of freeze-thaw optimisation routine under various support conditions.

The results indicated that the technique was capable of optimising the arrangement of particles to reduce deflection under self-weight with a forceFactorPower equal to 4 produced the best results for all support conditions. The sum each particle's deflection was also reduced, see Appendix D.

The evidence suggesting the optimisation technique works for a range of support conditions is promising – it implies that for complex volumes where a range of support conditions are used the structure can still be optimised.

It was noted that the volume filling behaviour was maintained for all the tests – i.e. the structure did not shrink away from the boundaries. It was also noted in every test and iteration a statically determinate structure was created.

The results were encouraging and the process is used on the case study, however, further testing and development of freeze-thaw optimisation is required, for details see Further work.

6. Case study

Tallinn Town Hall

To test the functionality of the process a case study was completed focusing on Tallinn Town Hall (Jordana 2009), a large multi-purpose building currently in development by BIG Architects and Ramboll UK, see Figure 36.

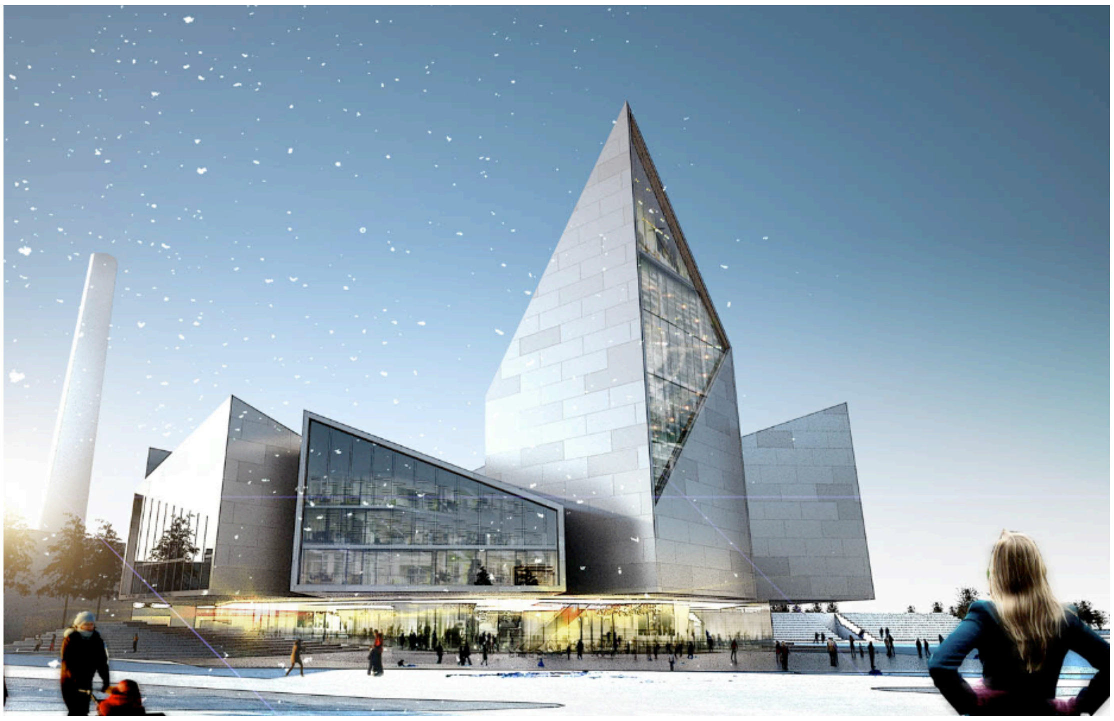


Figure 36: Visualisation of Tallinn Town Hall (BIG 2011).

The focal point of the building is the main spire that is conceived as a "*democratic periscope*" - see Figure 38. Suspended from the angled roof is a huge mirror that provides a view from the surrounding city into the debating chamber to involve the public with the democratic process and vice versa. Between the mirror and the roof is a volume in which a staircase leads the public up to a viewing area at the top of the tower.

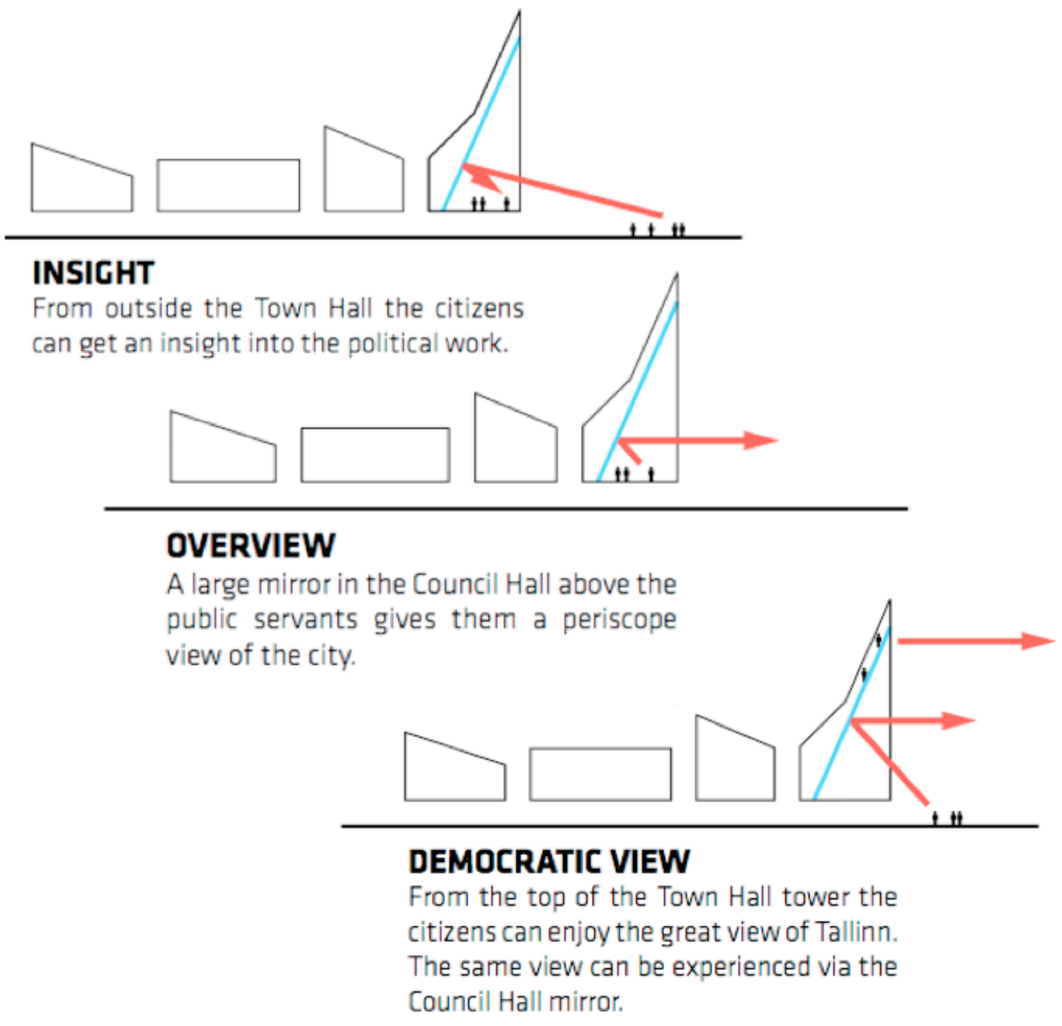

INSIGHT
From outside the Town Hall the citizens can get an insight into the political work.
OVERVIEW
A large mirror in the Council Hall above the public servants gives them a periscope view of the city.
DEMOCRATIC VIEW
From the top of the Town Hall tower the citizens can enjoy the great view of Tallinn. The same view can be experienced via the Council Hall mirror.

Figure 37: Diagram showing the various functions of the mirror and space within the main spire roof structure (BIG 2011).

Despite the volume being approximately square in plan, the angle of the roof creates a space that is highly irregular with a continuously changing cross section, differently spaced support conditions on all edges and an off-centre structural core. Originally, an irregular staircase route was to be imported with the volume as a structural void for the particles to avoid, however after seeing preliminary testing of the application the architects chose instead to allow the structure to form, optimise and then weave an interesting staircase route through the result. The main structural core of the building was imported as a void for the particles to avoid, see Figure 39. The volume is designed to be backlit at night, which changes the reflective properties of the mirror and make the supporting structure visible from the outside so aesthetics should be examined both at close range (from the viewpoint of someone on the stairs) and at the scale of the whole building.

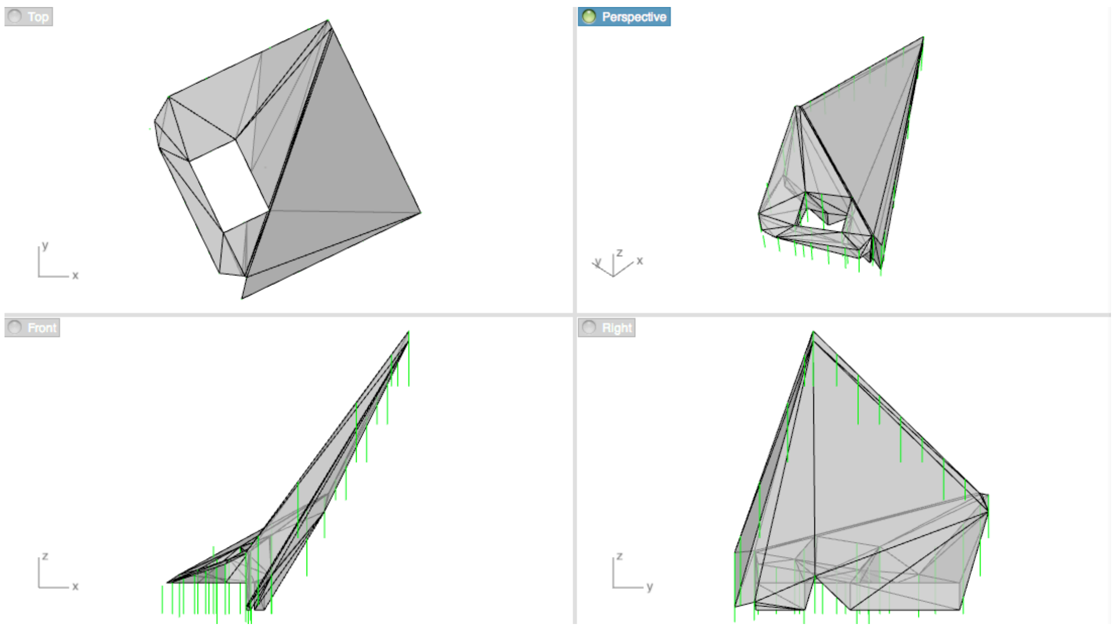


Figure 39: Boundary volume to be filling for the case study.

The structure must fit within this volume, provide support for the roof, stair and mirror (either directly or with secondary support). Using the particle simulation the volume is filled with structure therefore no secondary support is required – an important advantage. The surrounding structural support lines (shown in green) were intersected with the volume and the intersection points imported as fixed particles. The Ramboll UK structural design team produced a design for an orthogonal truss arrangement, which is used for comparison.

The volume filling application was run and the resulting stick model was exported as a .dxf file to Rhino. To maintain continuity across the kink in the roof a small allowable intersecting edge length was allowed. A quick post-processing stage was carried out to delete any undesired intersecting edges and group elements with respect to the loading they would resist and the structural frame was ready for analysis, see Figure 40.

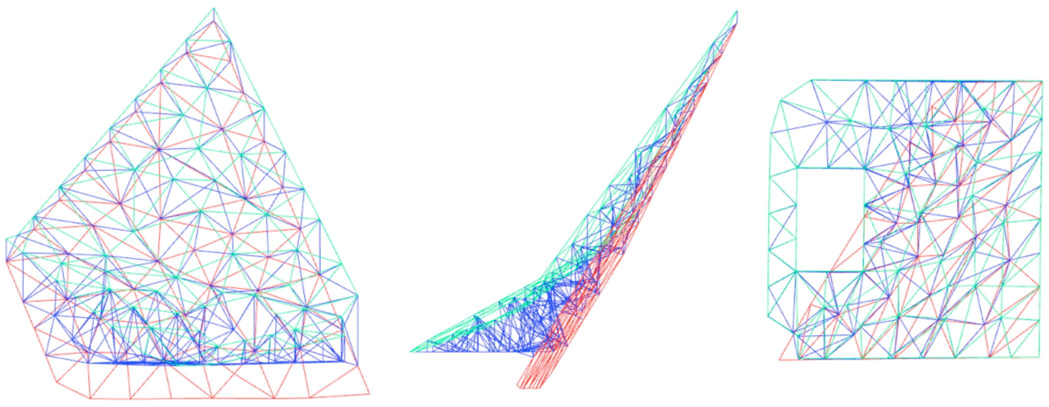


Figure 40: Stick model ready for detailed structural analysis. Different colours highlight the loading areas, for example, red elements are those supporting the dead load of the mirror.

The process was fast – a stable structural model that filled the volume and fitted in to existing supports was created and post-processed in approximately twenty minutes – far quicker than manually fitting the orthogonal truss into the awkward volume.

The process was then repeated to include freeze-thaw optimisation. The fixed points were adapted so the connections at the top of the glazed face were fixed only during the liquid phase – i.e. they do not act as structural support. Six iterations of the optimisation were carried out and the structure exported and post-processed as before, see Figure 41.

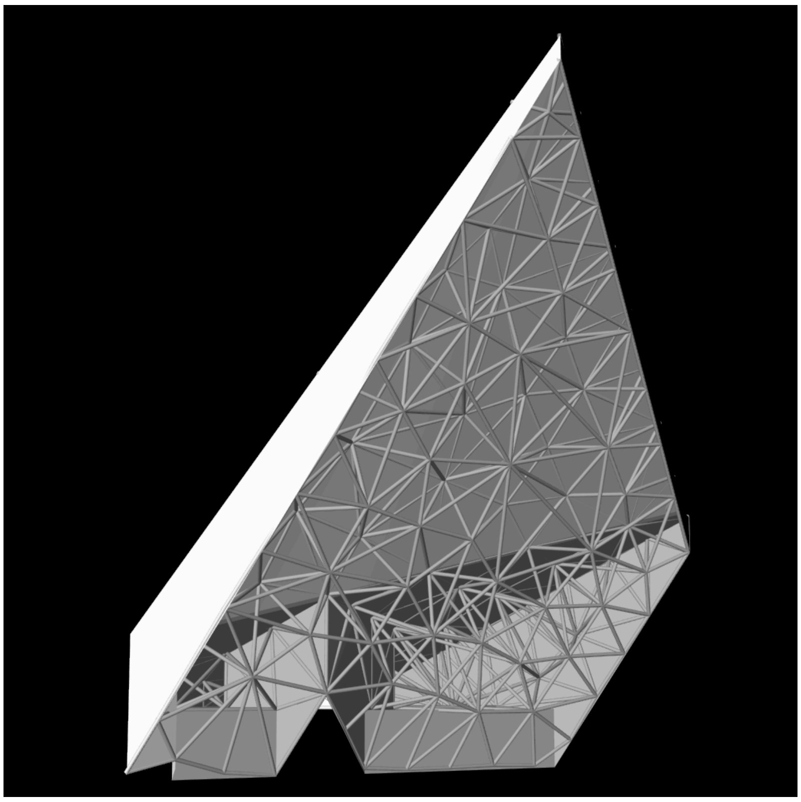


Figure 41: Render of the freeze-thaw optimised space frame within the volume.

Both models were then imported into structural analysis software, each edge was defined as a circular hollow steel section and wind, dead, super-dead and live loads were applied.

Analysis results

The analysis confirmed that the software creates stable structure capable of supporting all the surfaces and loads required. See Table 1 for comparison self weight and deflection.

Table 1: Headline results from structural analysis

	Orthogonal	Volume filled	Volume filled + freeze-thaw
Self weight of structure	154,385 kg	131,003 kg	133,705 kg
Maximum deflection	54.3mm	66.9mm	66.7mm

The data showed that the structures that were been automatically generated performed well in comparison to the orthogonal truss. As orthogonal trusses are one of the most efficient structural forms for spanning large distances this data highlighted the effectiveness of the software and was an exciting result.

Unfortunately, the freeze-thaw optimisation did not have a positive effect on the structural performance. A minimal improvement in deflection was recorded however this was more than counteracted by increase in self-weight. See 7. Conclusions for discussion of this result.

7. Conclusions

Overview

The aim of this research was to create a software solution to fill highly irregular and concave volumes with stable space structure. This was accomplished using an interactive particle simulation designed to increase engineers' tacit understanding of how space frames can fit within such volumes whilst providing a 3D structural output.

A literature review and investigation into previous work suggested that this was an area in which recent progress has been made but was yet to be satisfactorily solved.

The process emulated the behaviour of particles as they traverse matter states from high-energy gas to a liquid to solid structure. Different connectivity routines and combinations of various inter-particle and environmental forces defined the different matter states. The software development included an adapted Delaunay tetrahedralisation routine, a recursive bounce algorithm, boundary force reduction, allowable intersecting edge length control and a mesh quality measurement system. Quantitative testing and calibration was carried out which proved the effectiveness of each stage of the simulation and of the process as a whole.

An iterative algorithm called freeze-thaw optimisation was introduced to improve the structural efficiency of the output. It worked by collating member strains produced within the solid state and using these values to alter the relative repulsive force for the connecting particles in the liquid state. Initial testing on a cuboid volume produced positive results with deflection reduced for a variety of support conditions.

The software was used to design a roof structure for a live project – Tallinn Town Hall in Estonia. Within minutes two viable structural options (with and without freeze thaw optimisation) were produced. These performed approximately as well as an orthogonal truss with roughly 15% less steel albeit with 18% greater maximum deflection. The use of imported fixed points proved to be an effective way of fitting the structure into surrounding more conventionally design arrangements. The case study proves the software can successfully tackle and automate the design problem of how to fill irregular volumes with stable structure.

The results were undoubtedly more aesthetically interesting than the orthogonal truss whether they are *superior* is a matter of opinion - feedback from BIG architects has been very positive. The case study demonstrated how flexible the process is to design changes, rather than freezing the volume any changes by the design team could very quickly be integrated and imported into the software.

It was noted that the freeze-thaw optimisation did not improve the structural efficiency of the case study model. It is suggested that this was due to the difference between the loading for the detailed structural analysis and that used during the optimisation, which will be the subject of further work.

Uses

The software was developed in response to freeform architecture and sculpture that required internal structural support, however the functionality of filling any given volume with a stable and even mesh of tetrahedrons is a powerful tool with a wide range of potential uses.

There is an argument that the output contains a utilitarian logic and efficiency making it suitable for civil engineering structures (if manufacturing challenges can be overcome). For example, a designer of an industrial process may require structural support at a particular point in space whilst avoiding a particularly complex existing environment. These situations could be quickly solved and optimised using the software.

It is also suggested that the software may be used to provide inspiration or guidance for structural arrangements rather than a direct output. For example, in Figure 34 it could be noted how the 3D structure handles a change in density towards areas of high stress and this could influence an engineer designing a structure manually. As the structure was comparable in weight to the orthogonal truss it may also be used as a fast way to estimate steel weight required in a scheme design – even if the required final structure is to be a more traditional truss arrangement.

The software may also be used at a smaller scale. For example, highly bespoke furniture could be designed around volumes based on 3D scans of constrained spaces or even people. Another promising area of exploration would be at the scale of 3D printing technology. The development of optimised foam metal-like titanium components with high strength to weight ratios will be of interest in the manufacture of prostheses and high performance mechanical engineering.

The development was partly inspired by the bubble mesh technique (Shimada and Gossard 1995) for creating meshes for finite element purposes and this software is also a valid method for their creation. Interestingly, the freeze-thaw optimisation step will create a dense mesh in areas of high stress, which is desirable for finite element analysis. High-resolution meshes could be created by increasing the number of particles or by sub-dividing each tetrahedron.

It was noted by the author that as the structures deformed during the solid state they acted as if the original imported volume was filled with a jelly-like material. As the framework already includes intersection tests and a bounce algorithm it was possible to bounce objects against a floor and produce very convincing behaviour. This is of interest in the fields of computer games and 3D animation. In addition to this, if forces pass a certain limit the structure could even break up into the individual tetrahedrons (or groups) and reproduce the shattering of an object.

Further work

A robust and practical framework was created, nevertheless there is potential for further development and research.

The simulation ran in real-time and was usually capable of doing multiple Verlet steps per frame. However when working with a large number of particles and/or a complex volume it became noticeably slower. It is suggested that the implementation of a k-d tree data structure would considerably speed up both the construction of the DT and the boundary collision algorithm.

The allowable intersecting edge length routine (as shown in Figure 17) provided a simple way of adjusting the mesh if desired. However, with complex volumes different areas may be suited to different allowable edge lengths. This meant a manual post processing stage was sometimes required to delete any unwanted edges. By analysing the angle of the edge with respect to the orientation of the faces being intersected it may be possible to create a more robust algorithm.

During the solid state the simulation does not take into account buckling of the individual members. This is an important aspect of structural analysis and should be taken into account - especially as members' lengths are increased as a result of the optimisation. For tubular sections (commonly used in space frames) this is a straightforward process. A related development would be to update the simulation to produce accurate stresses and deflections rather than the current settings - which are used and set primarily to ensure a stable simulation.

It is expected that the space frames produced by the simulation will often be required to support secondary structure, such as glazing or lightweight roof panels. These materials will have a maximum span and it would be advantageous if the spans between the structure on the surface of the volume were restricted to this distance. Identifying any edges on the surface and imposing a spring force during the liquid state may be able to achieve this.

The completed DT has a list of hull faces, which represent the convex hull of all the particles (which can be seen in Figure 34), however this does not currently respect a concave boundary volume. If an algorithm were implemented that set all faces pushing up against the surface of the volume as hull faces it would allow area loads to be calculated and applied to the structure. Using these accurate area loads further testing of the freeze-thaw optimisation can be carried out. In addition, comparative tests with traditional arrangements will provide further validation. However it should be noted that the process is designed to work within volumes where there are no traditional arrangements.

Structures also must withstand a variety of load cases throughout their lifespan, to respond to this in the optimisation it would be feasible to run a series of load cases during the solid phase and average out the particle strains (or possibly take the worst case) for each particle. It is also suggested that a normal distribution between -1 and 1 is used to adjust the relativeForceFactors rather than the forceFactorPower.

Rather than increasing the density of the structural arrangement in areas of high stress it may be interesting to do the opposite and create less dense areas in combination with increased member size. This mirrors the structural behaviour of a tree, where the most highly stressed area of the structure (in both compression and bending) is the base of the trunk – where there is a single heavy "structural member". In contrast - in the areas of low stress, such as the ends of the branches there is a high density of lightweight "structural members". A structural analogy for this would be large columns that subdivide and decrease in size multiple times to support a large surface area of façade.

Finally, research into how to simplify the manufacture and construction of the resulting structure should significantly increase the viability of the process within the building industry. Connection details in particular should be repeated or based on a standard detail wherever possible.

This research reveals what can be achieved when form-finding philosophies are used within a digital environment and designers are no longer limited to learning from only inert or physically measurable behaviour. It is part of the next generation of form-finding methods and another demonstrable justification of the use of computers in design.

References

BIG (2011). Tallinn Town Hall - Preliminary Design Booklet.

Chadwick, A., N. Thomas, et al. (2009). Liquid threshold. London, Atelier One.

Cignoni, P., D. Laforenza, et al. (1995). "Evaluation of parallelization strategies for an incremental Delaunay triangulator in e3." Concurrency: Practice and Experience **7**(1): 61-80.

Cignoni, P., C. Montani, et al. (1998). "DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed." Computer-Aided Design **30**(5): 333-341.

Eigensatz, M., M. Deuss, et al. (2010). Case Studies in Cost-Optimized Paneling of Architectural Freeform Surfaces

Advances in Architectural Geometry 2010. C. Ceccato, L. Hesselgren, M. Pauly, H.Pottmann and J. Wallner, Springer Vienna**:** 49-72.

Fry, B. and C. Reas (2004). Processing, Massachusetts Institute of Technology.

Gormley, A. (2011). from http://www.antonygormley.com/.

Hanna, S. (2011). Personal communication. Lewis, H.

Hansmeyer, M. (2003). L-Systems in Architecture. http://www.michael-hansmeyer.com.

Harding, J. (2010). An Applied Genetic Algorithm, University of Bath. (unpublished report).

Harding, J. (2011). Personal communication. Lewis, H.

Jaworski, P. L. (2006). Using simulations and artificial life algorithms to grow elements of construction. Bartlett School of Graduate Studies. London, University College London.

Jin, S., R. R. Lewis, et al. (2005). "A comparison of algorithms for vertex normal computation." The Visual Computer **21**(1): 71-82.

Jordana, S. (2009) BIG wins International Competition to design Tallinn's new City Hall. http://www.archdaily.com/26391

Kaijima, S. and P. Michalatos (2007). "Discretization of Continuous Surfaces as a Design Concern." Lecture.

Kanellos, A. (2007). Topological Self Organisation: Using particle-spring system simulation to generate structural space-filling lattices. Bartlett School of Graduate Studies. London, University College London.

Kapoor, A. (2011). from http://www.anishkapoor.com.

Leach, N., D. Turnbull, et al. (2004). Digital tectonics. Chichester, Wiley-Academy.

Marks, R. W. and R. B. Fuller (1973). The Dymaxion world of Buckminster Fuller. Garden City, N.Y., Anchor Books.

McNeel, R. (1995). Rhinoceros 3D, computer program.

Mitchell, M. (1996). "An introduction to genetic algorithms." from http://www.netlibrary.com/urlapi.asp?action=summary&v=1&bookid=1337.

Moore, A. (1991). An introductory tutorial on kd-trees Technical Report No. 209, Computer Laboratory, University of Cambridge.

Orwall, B. (1998). What's the Buzz? Wall Street Journal. US, Dow Jones & Company, Inc.

Papapavlou, A. (2008). Structural Evolution: a genetic algorithm to generate structurally optimal Delaunay triangulated space frames for dynamic loads. Bartlett School of Graduate Studies, University College of London.

Piker, D. (2011). Pseudo-Physical Materials.

http://spacesymmetrystructure.wordpress.com/2011/05/18/pseudo-physical-materials/

Prusinkiewicz, P. and A. Lindenmayer (1990). The algorithmic beauty of plants. New York, Springer-Verlag.

Schein, M. and O. Tessmann (2008). "Structural analysis as driver in surface-based design approaches." International Journal of Architectural Computing **6**.

Shewchuk, J. R. (2002). Constrained Delaunay Tetrahedralization and Provably Good Boundary Recovery. 11th International Meshing Roundtable, Springer-Verlag.

Shimada, K. and D. C. Gossard (1995). Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. Proceedings of the third ACM symposium on Solid modeling and applications. Salt Lake City, Utah, United States, ACM**:** 409-419.

Song, C., P. Wang, et al. (2008). "A phase diagram for jammed matter." Nature **453**(7195): 629-632.

Tuffanelli, C. (2010). Mirrors and spheres: The geometry within the "Tall Tree and the Eye", Advanced Geometry Unit.

Verth, J. M. V. and L. M. Bishop (2008). Essential Mathematics for Games and Interactive Applications: A Programmer's Guide, Morgan Kaufmann.

Wallner, F. G. a. G. (2010). Algorithms for generation of irregular space frame structures. Vienna, University of Applied Arts.

Weaire, D. and R. Phelan (1994). "A counter-example to Kelvin's conjecture on minimal surfaces." Philosophical Magazine Letters **69**(2): 107-110.

Weinstock, M. (2006). Natures Verb. Barcelona, Actar.

Wikipedia (2011, 11 June 2011). "Diffusion-limited aggregation." Retrieved 15 June, 2011, from http://en.wikipedia.org/w/index.php?title=Diffusion-limited_aggregation&oldid=433739641.

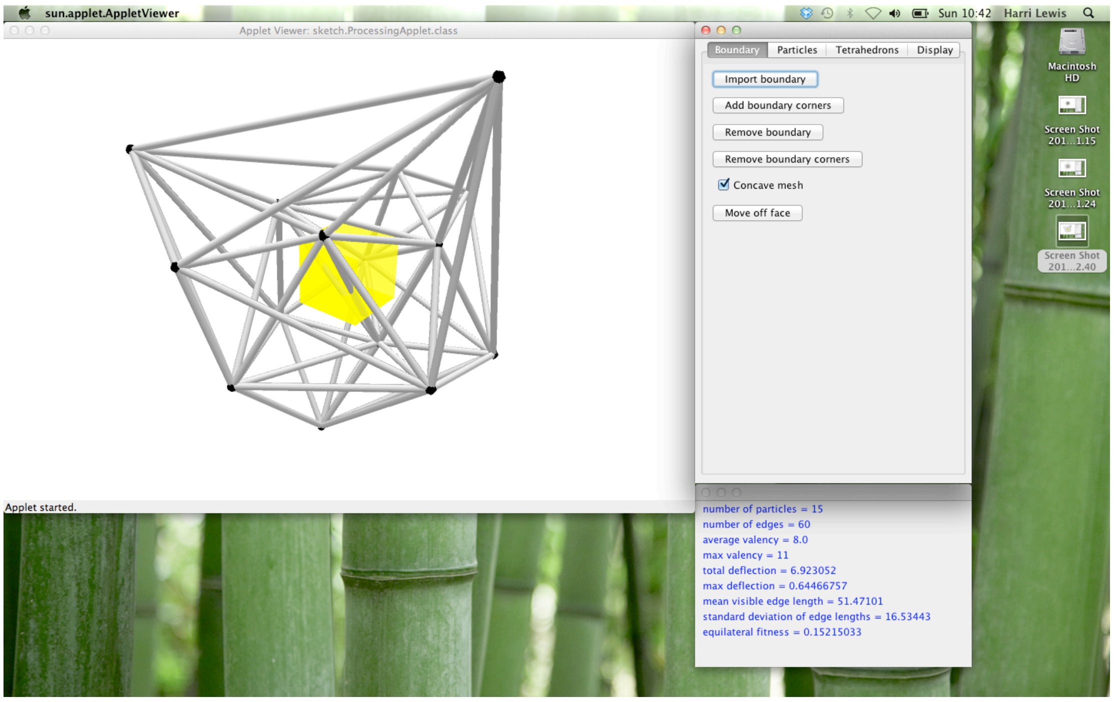
Wikipedia (2011). "Metal foam". Retrieved 21 July, 2011, from http://en.wikipedia.org/wiki/Metal_foam

Winslow, P. (2010). "Multi-objective optimization of free-form grid structures." Structural and multidisciplinary optimization **40**(1): 257-269.

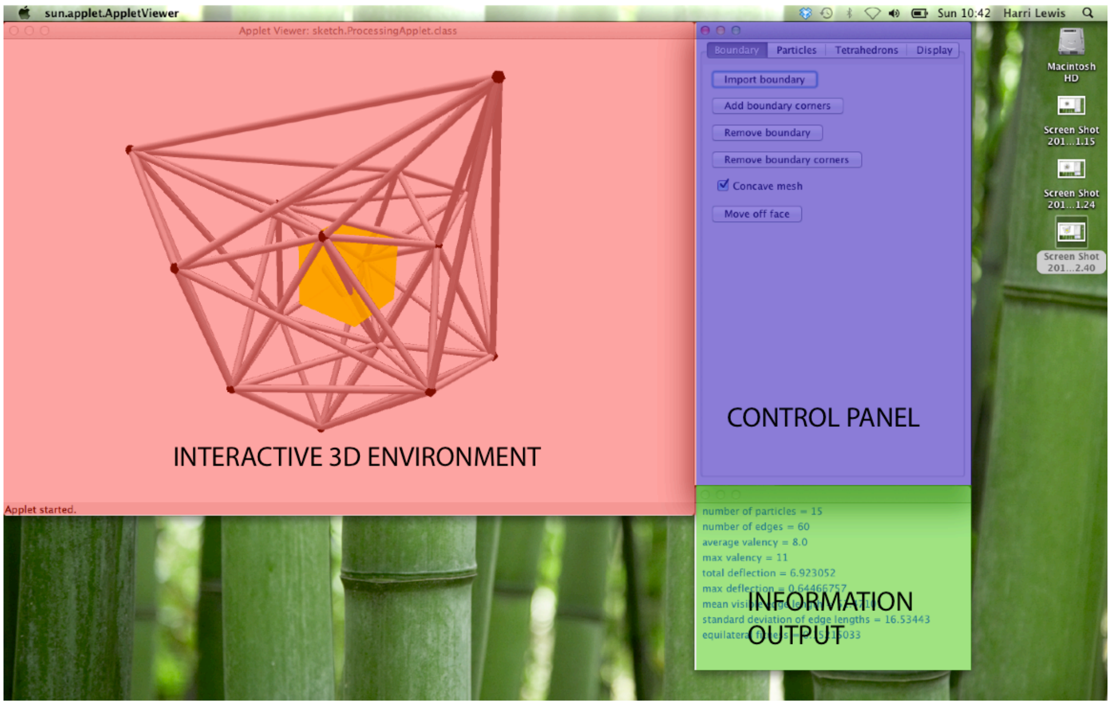
Zarzycki, A. (2009). Form-making in architecture: performance and simulation based design approach. ACM SIGGRAPH ASIA 2009 Posters. Yokohama, Japan, ACM**:** 1-1.

Appendices

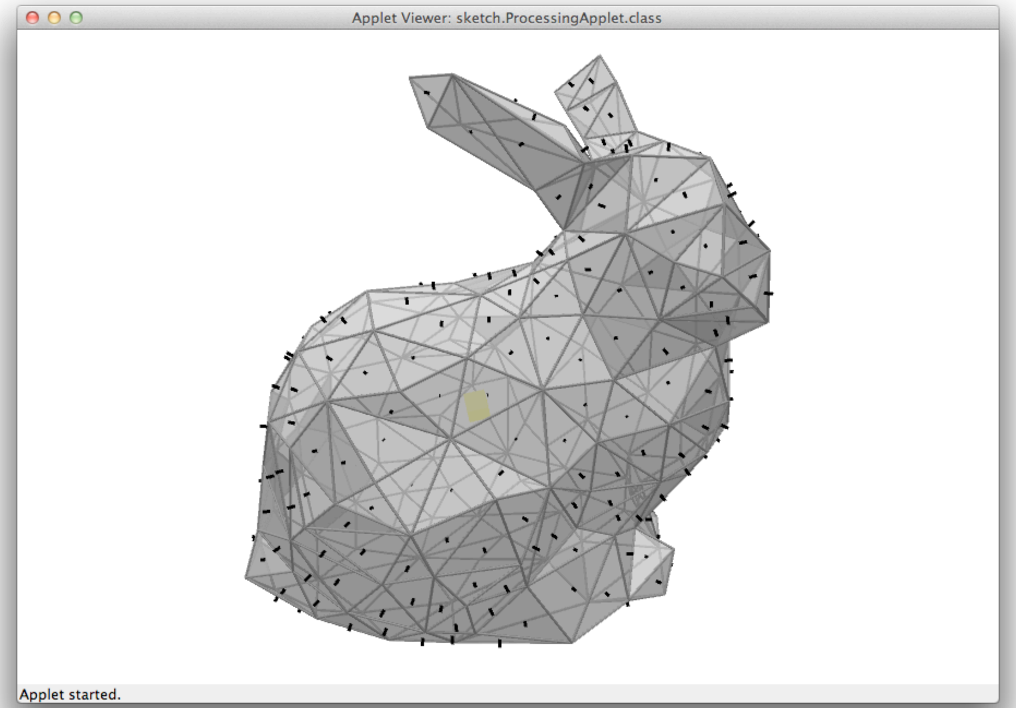
Appendix A

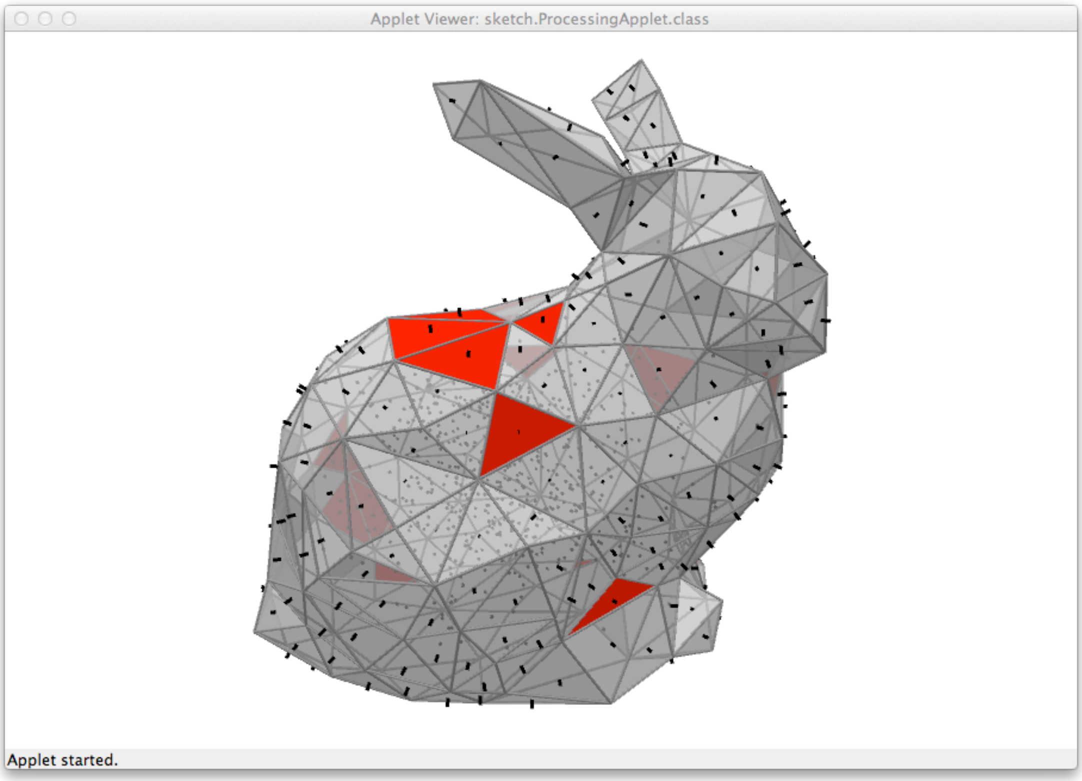


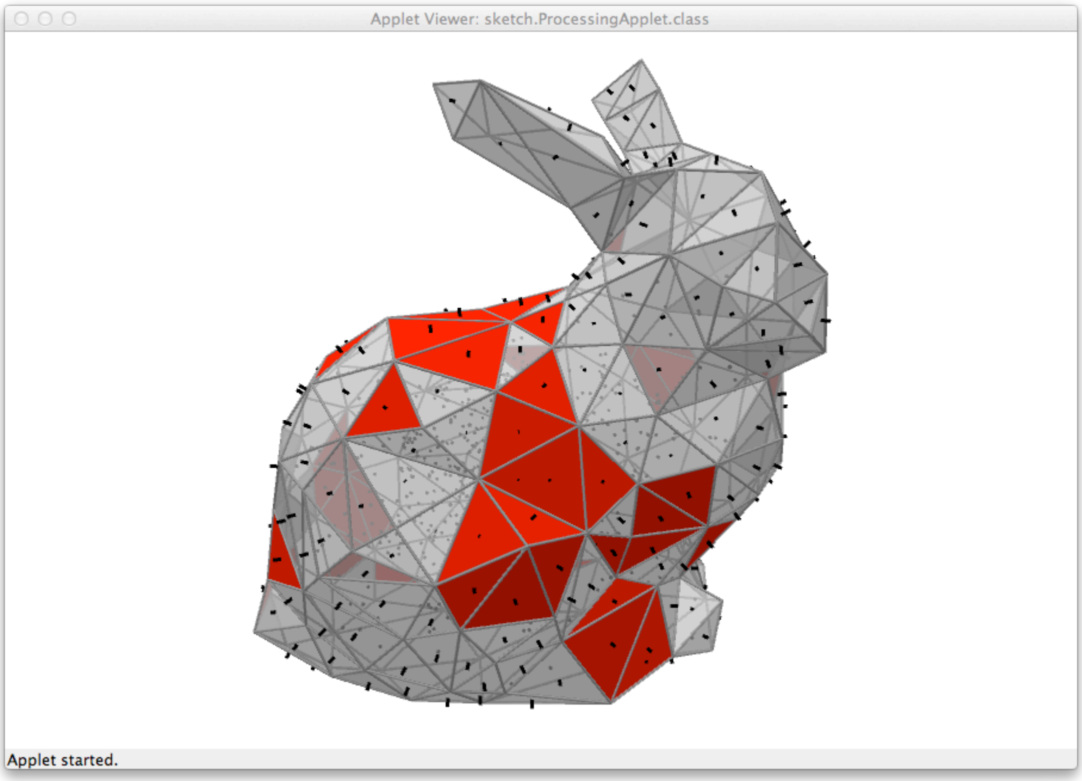
```
number of particles = 15
number of edges = 60
average valency = 8.0
max valency = 11
total deflection = 6.923052
max deflection = 0.64466757
mean visible edge length = 51.47101
standard deviation of edge lengths = 16.53443
equilateral fitness = 0.15215033
```



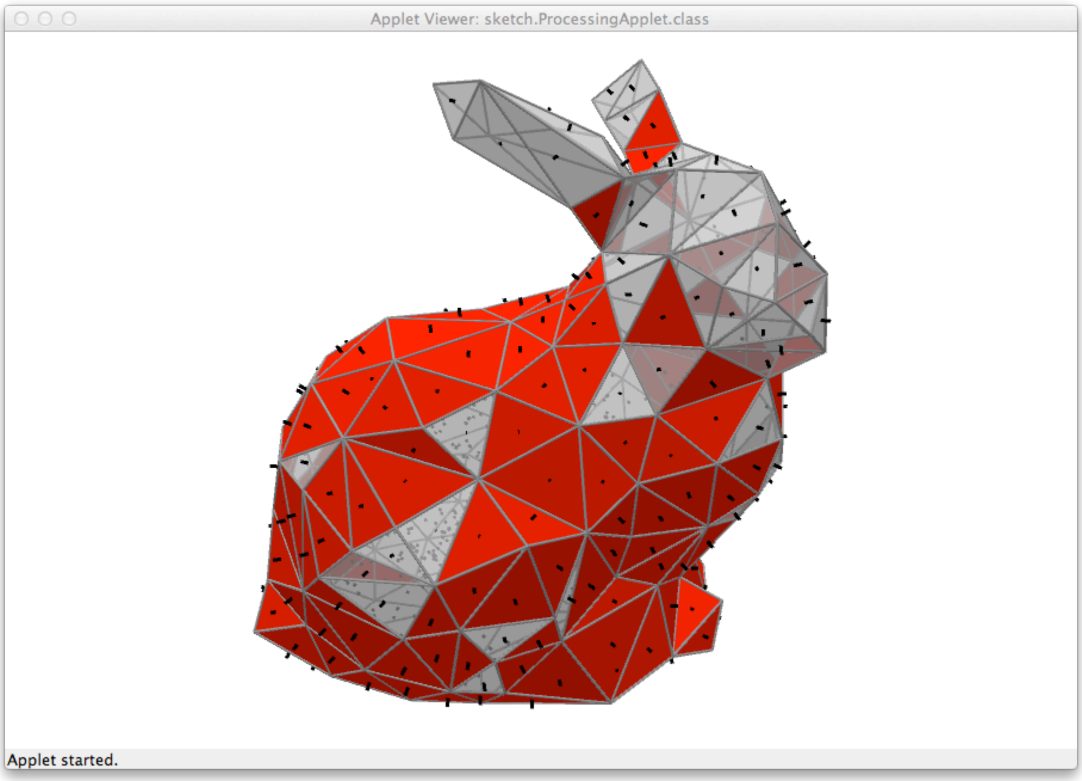
Appendix B


Applet Viewer: sketch.ProcessingApplet.class
Applet started.

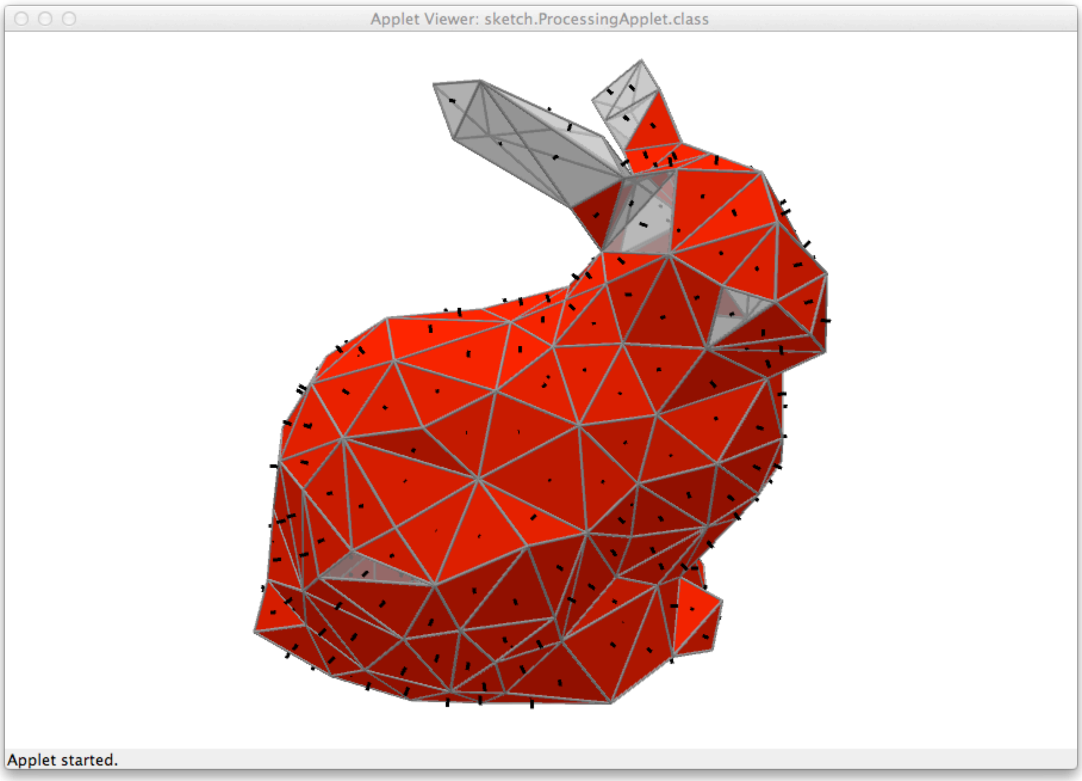

Applet Viewer: sketch.ProcessingApplet.class
Applet started.

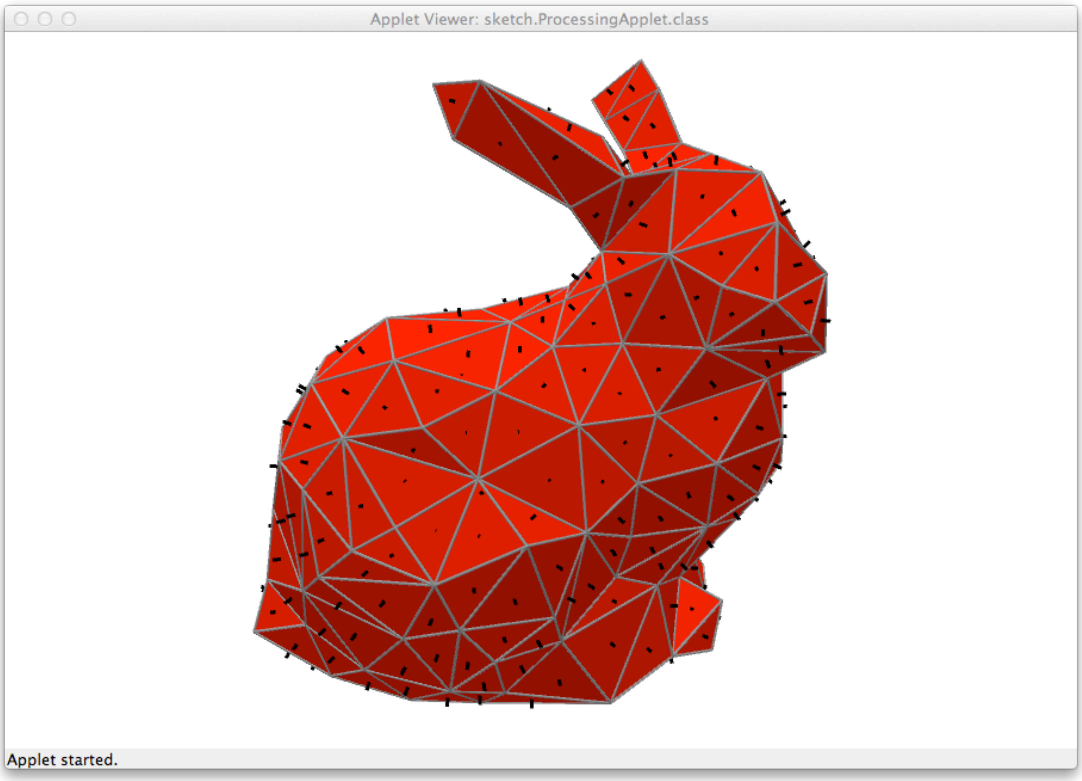

Applet Viewer: sketch.ProcessingApplet.class

Applet started.


Applet Viewer: sketch.ProcessingApplet.class

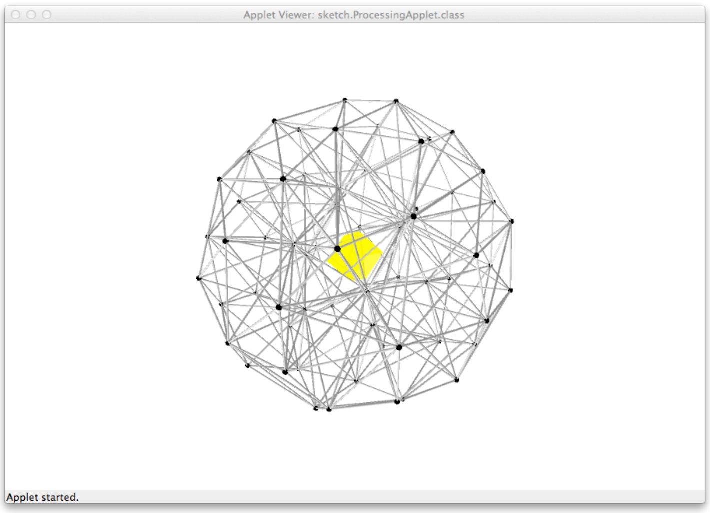
Applet started.


Applet Viewer: sketch.ProcessingApplet.class
Applet started.

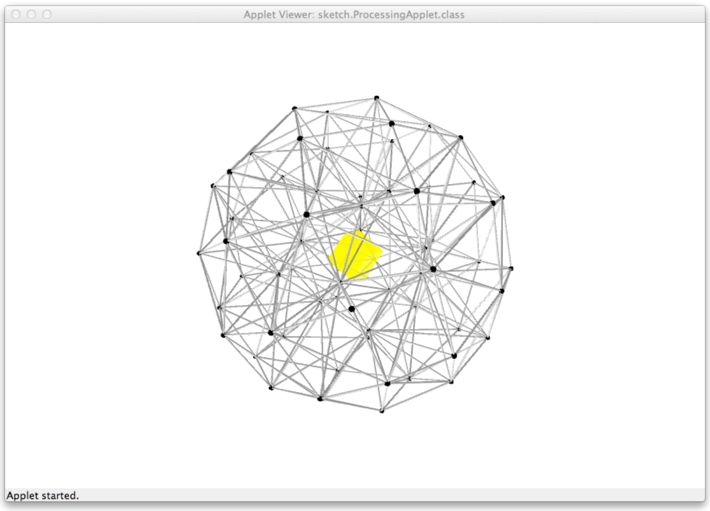

Applet Viewer: sketch.ProcessingApplet.class
Applet started.

Appendix C

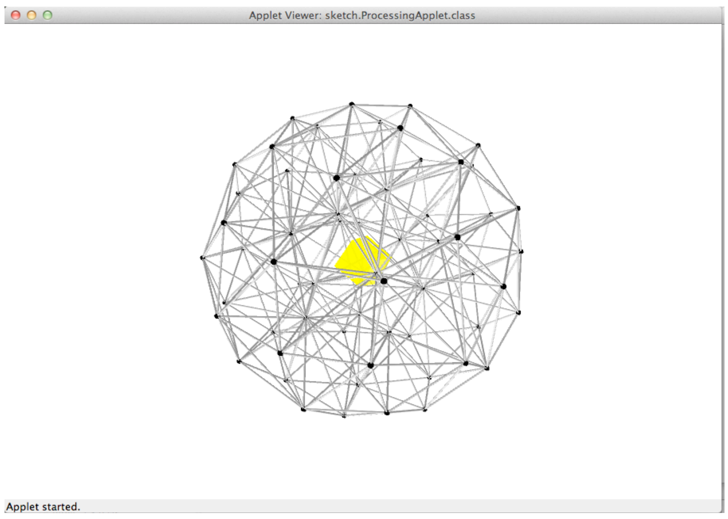
Force 1


Applet Viewer: sketch.ProcessingApplet.class

Applet started.

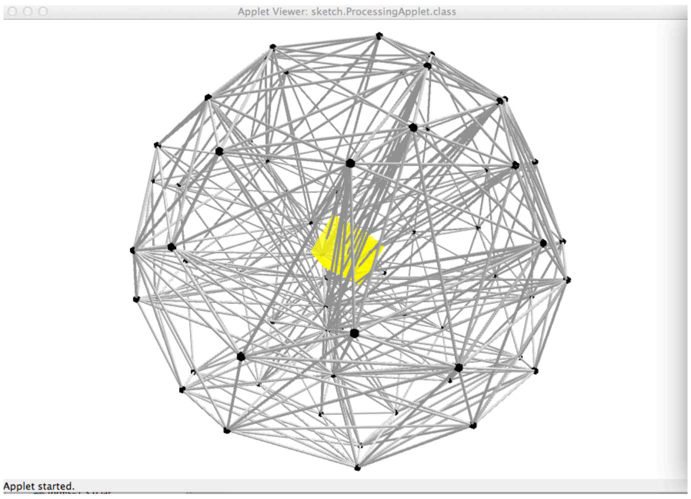

Applet Viewer: sketch.ProcessingApplet.class

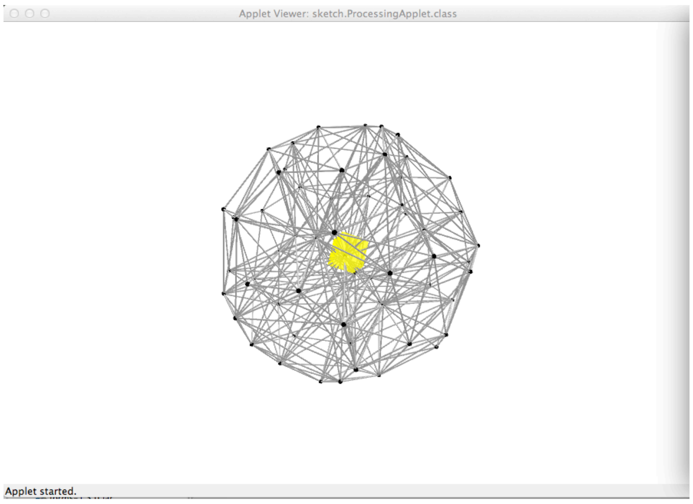
Applet started.

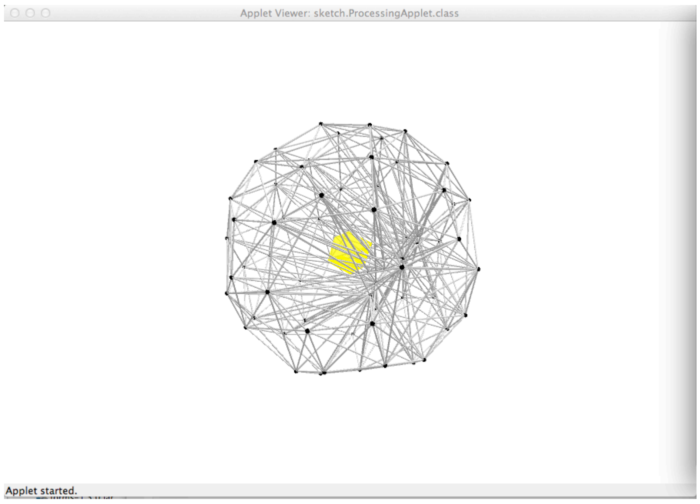

Applet Viewer: sketch.ProcessingApplet.class

Applet started.

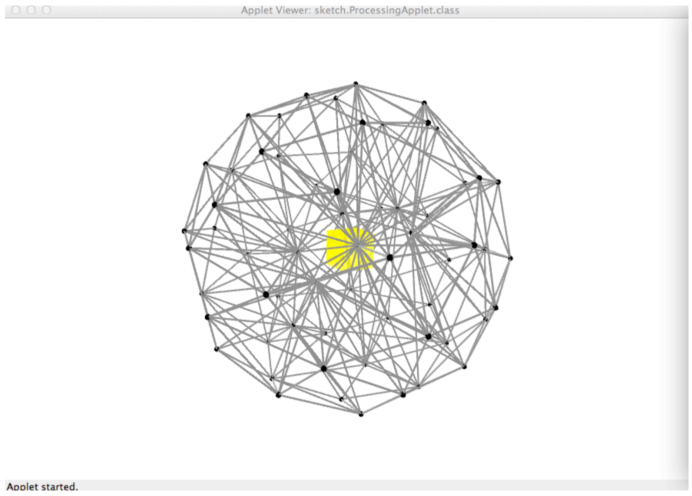
Force 2

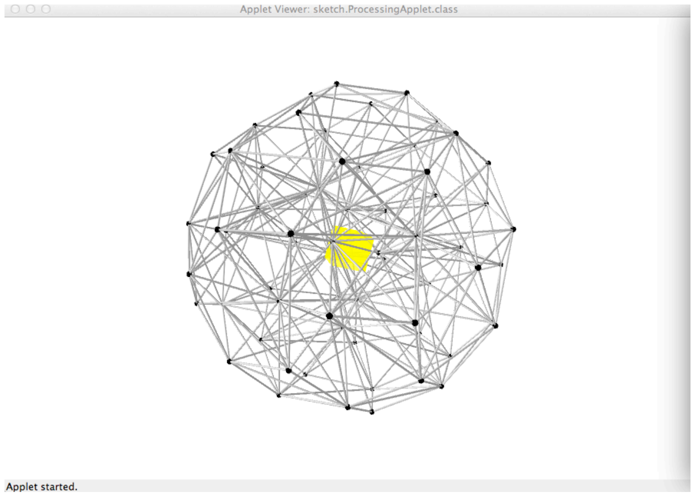

Applet Viewer: sketch.ProcessingApplet.class
Applet started.

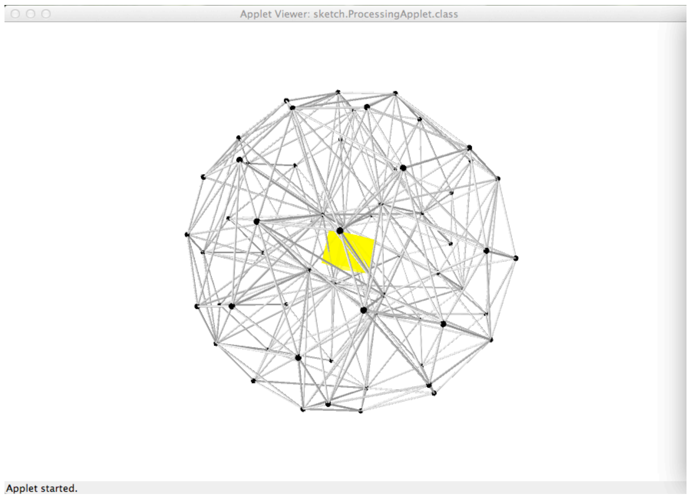

Applet Viewer: sketch.ProcessingApplet.class
Applet started.


Applet Viewer: sketch.ProcessingApplet.class
Applet started.

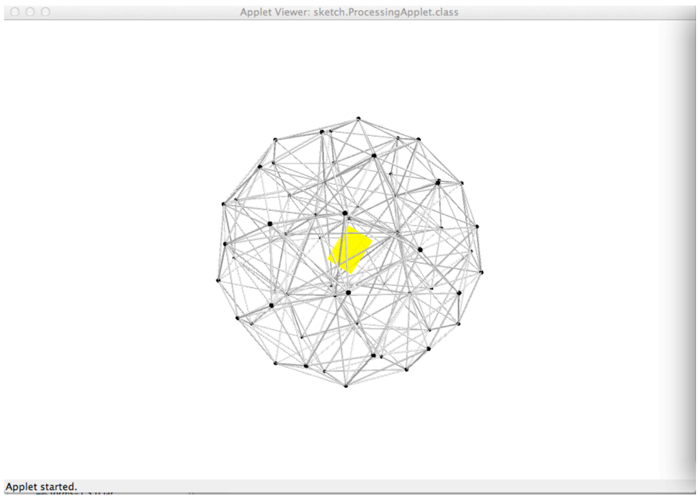
Force 3


Applet Viewer: sketch.ProcessingApplet.class
Applet started.

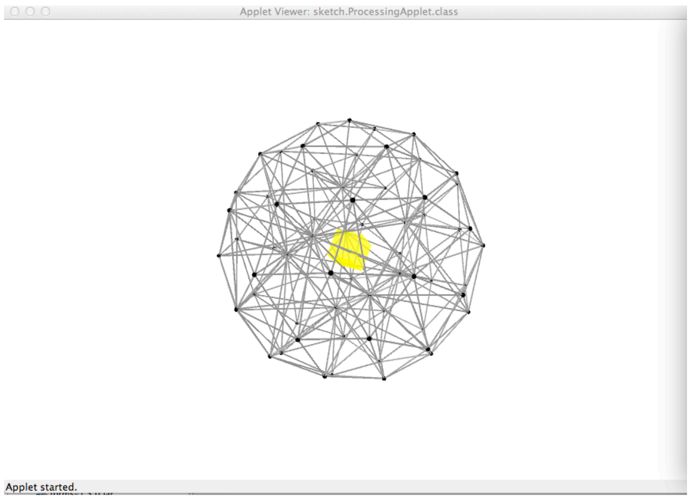

Applet Viewer: sketch.ProcessingApplet.class
Applet started.


Applet Viewer: sketch.ProcessingApplet.class
Applet started.

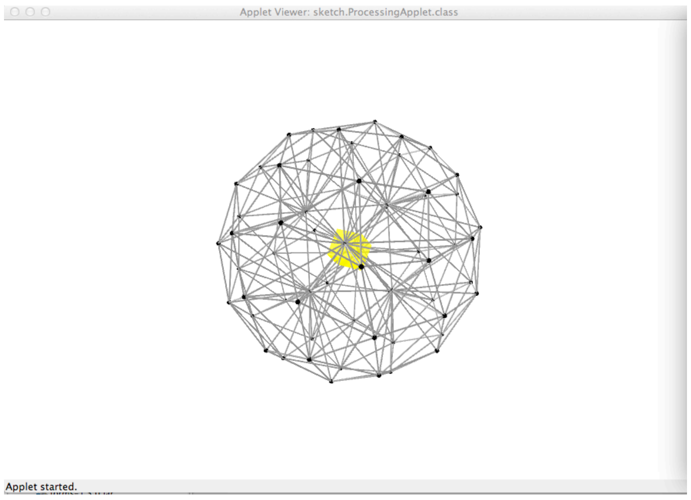
Force 4


Applet Viewer: sketch.ProcessingApplet.class

Applet started.

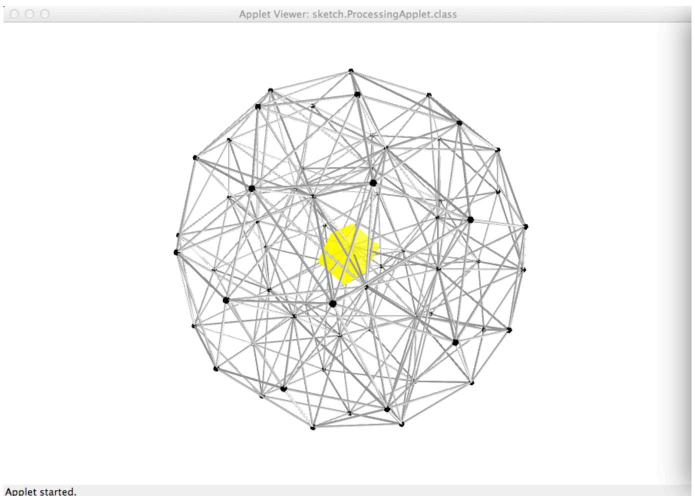

Applet Viewer: sketch.ProcessingApplet.class

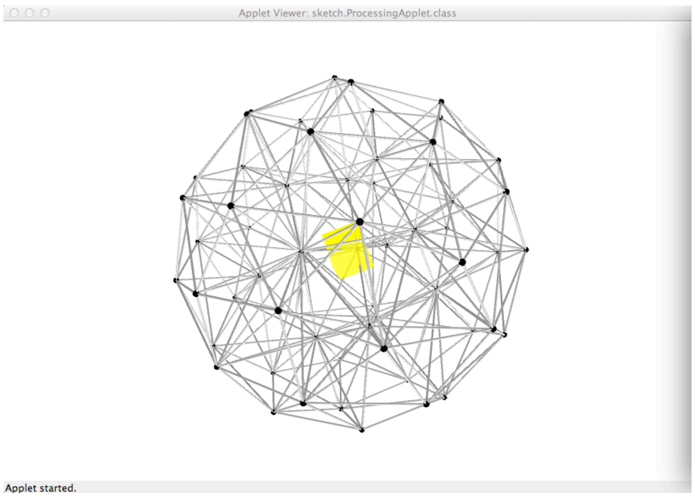
Applet started.

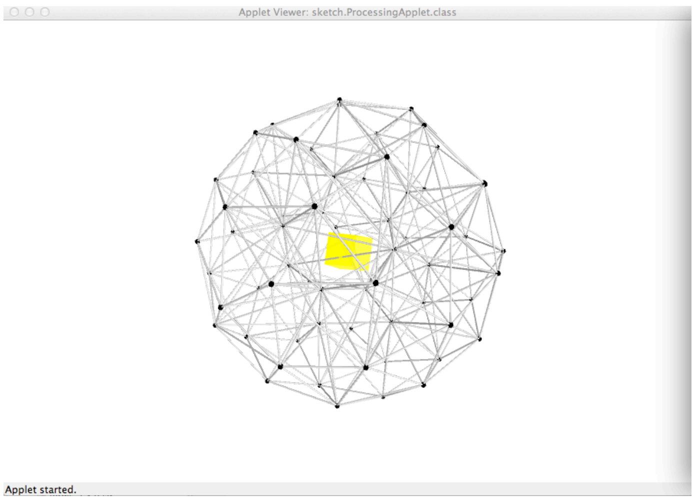

Applet Viewer: sketch.ProcessingApplet.class

Applet started.

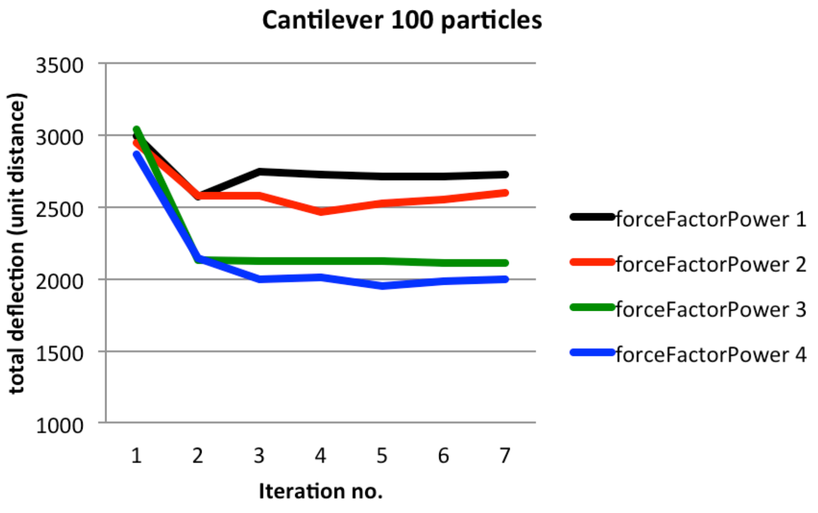
Force 5


Applet Viewer: sketch.ProcessingApplet.class
Applet started.

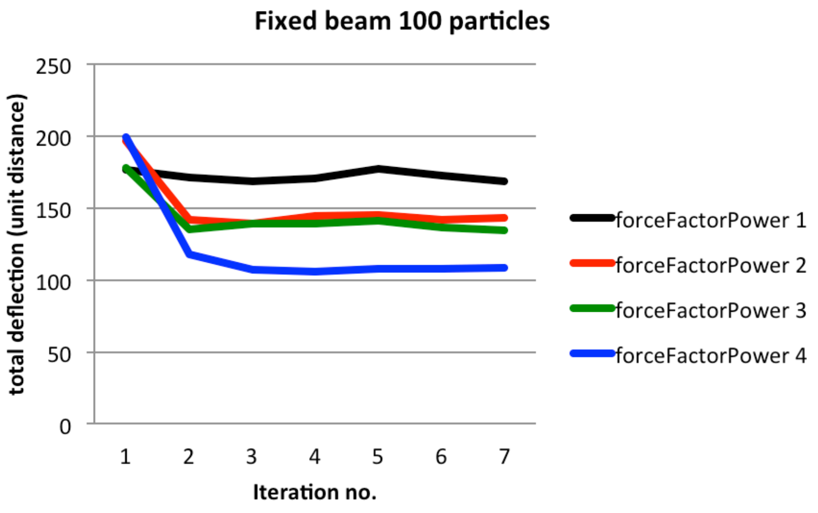

Applet Viewer: sketch.ProcessingApplet.class
Applet started.


Applet Viewer: sketch.ProcessingApplet.class
Applet started.

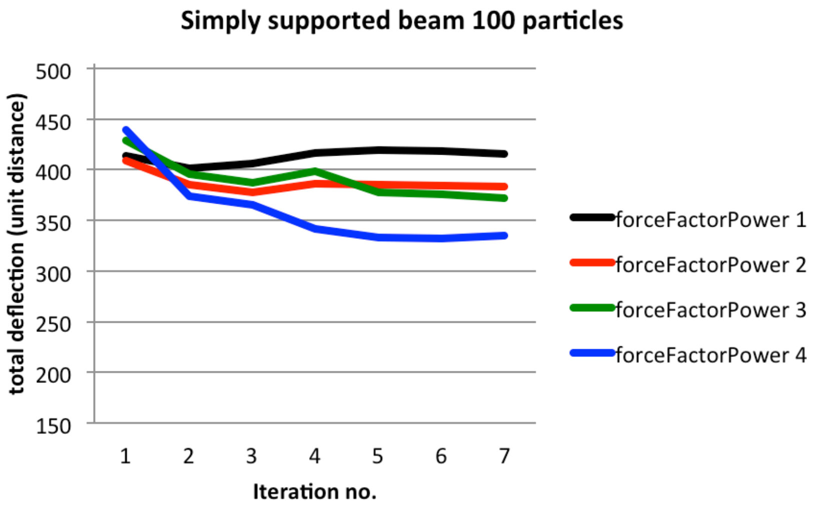
Appendix D



Cantilever 100 particles
total deflection (unit distance)
3500
3000
2500
2000
1500
1000
1
2
3
4
5
6
7
Iteration no.
forceFactorPower 1
forceFactorPower 2
forceFactorPower 3
forceFactorPower 4



Fixed beam 100 particles
total deflection (unit distance)
250
200
150
100
50
0
1
2
3
4
5
6
7
Iteration no.
forceFactorPower 1
forceFactorPower 2
forceFactorPower 3
forceFactorPower 4



Simply supported beam 100 particles
total deflection (unit distance)
500
450
400
350
300
250
200
150
1
2
3
4
5
6
7
Iteration no.
forceFactorPower 1
forceFactorPower 2
forceFactorPower 3
forceFactorPower 4