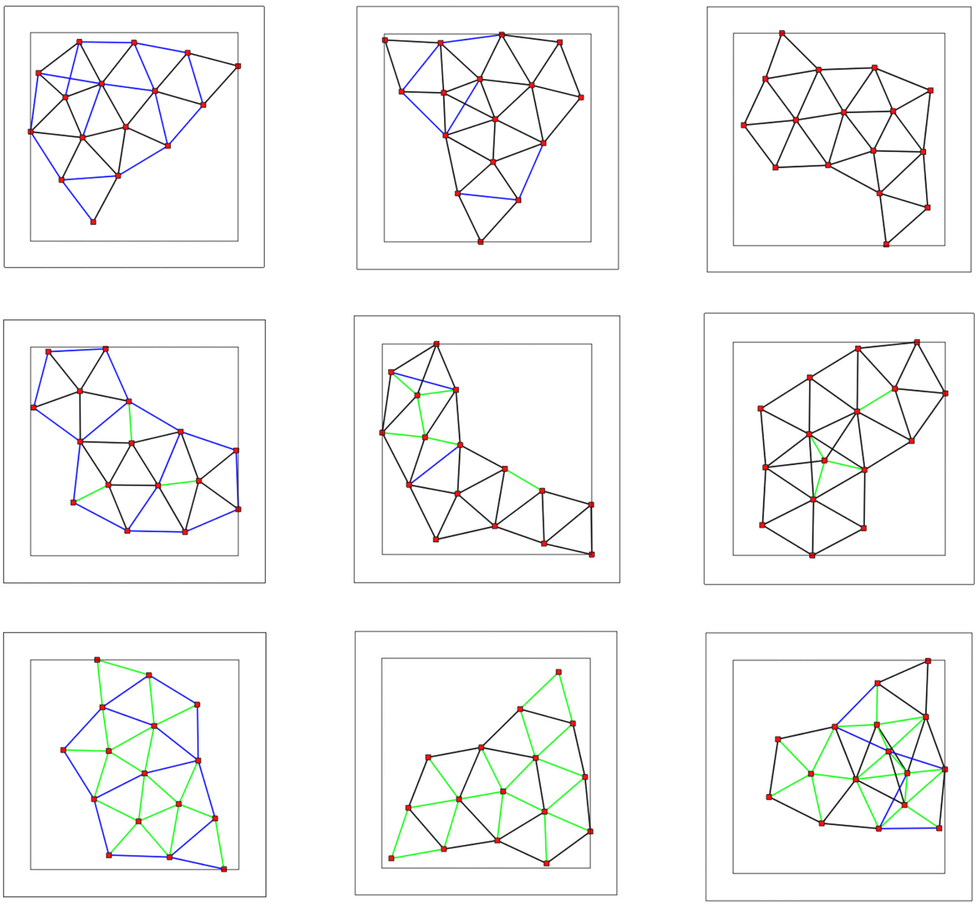
Towards aperiodic tesselation: A self-organising particle spring system approach



Marianthi Leon

This dissertation is submitted in partial
fulfilment of the requirements for the degree of
Master of Science in Adaptive Architecture and Computation

Bartlett School of Graduate Studies University College London

September 2008

I, Marianthi Leon, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

The derivation of novel programming methods for the generation of aperiodic tiling patterns, predominantly in 2d space, has attracted considerable attention from both researchers and practicing architects. So far L-Systems and quasicrystals are the only tools which can be used for the creation of aperiodic tiling patterns. This project attempts to create a self organizing particle spring system for aperiodic tiling formation on a 2d surface. The proposed method simulates natural dynamic procedures and applies a generative particle spring system for tiling formation. The initial inspiration of the thesis is the realization of tiling patterns for non-planar geometries, by using the previously stated method. The architectural reasoning behind that would be to use a minimal number of types of prefabricated units (e.g. Penrose rhombuses) to create an irregular and complex pattern or geometry.

Word count: 9.160

Contents

Acknowledgements

I would like to thank: my supervisor Sean Hanna for all his guidance, advices and encouragement and for never forgetting the initial target; Alasdair Turner for the necessary processing and programming advices and his optimistic point of view; Marios Theofanous for his engineering viewpoint and for proof reading my thesis.

Most of all I would like to thank my family for their continuous and unconditional support.

List of figures

1. Introduction

Tessellation of non-planar curved surfaces with aperiodic tiling patterns is a challenging but formidable task, which necessitates the derivation and use of elaborated and efficient algorithms and is still an open research topic. The derivation of novel programming methods for the generation of aperiodic tiling patterns, predominantly in 2d space, has attracted considerable attention from both researchers and practicing architects (Hwang et al, 2006) over the past three decades.

In 1961 Wang (Wang, 1961) asserted that aperiodic tiling patterns cannot be created (and hence cannot exist) by aggregating tiles of specific geometry. This would suggest that local-rule based systems are inappropriate for aperiodic tiling generation. Several researchers succeeded in to contradicting this assertion from a mathematical point of view (Penrose, 1974; Senechal, 1990; Robinson, 1971). It was not until recently that algorithms employing the logic of L-Systems and of quasicrystals, which succeeded in producing aperiodic tilings, were developed. So far L-Systems and quasicrystals are the only tools which can be used for the creation of aperiodic tiling patterns.

This project attempts to create a self organizing particle spring system for aperiodic tiling formation on a 2d surface. The proposed method simulates natural dynamic procedures and applies a generative particle spring system for tiling formation. The initial inspiration of the thesis is the realization of tiling patterns for non-planar geometries, a not yet achieved approach, by using the previously stated method. The architectural reasoning behind that would be to use a minimal number of types of prefabricated units (e.g. penrose tiles) to create an irregular and complex pattern or geometry.

1.1 Overview of the natural inspirations to architecture

Nowadays, the natural processes are structured and simulated by virtual multiplicities and head for a never achieved actual state. As Deleuze and Guattari mention in "a Thousand Plateaus" (Deleuze, Guattari, 2004) , it is all about the process; systems tend to move either towards reality or towards virtuality, consequently, processes can either be more or less stable with diverse possibilities. Neither the structures of such processes, nor their completed products "merit the same ontological status as processes themselves" (Deleuze, Guattari, 2004). Subsequently, these systems are capable of including the concepts of coding, which is the process of ordering matter as it is drawn into a body, stratification, that is the process of creating hierarchal bodies, and territorialisation or, in other words, the ordering of those bodies in "assemblages," an emergent unity joining together heterogeneous bodies in a "consistency".

This theory is applied within the framework of the current project, in the sense that natural theories and equations are infused within coding so that a virtual dynamic simulation decides about the tiling formation; different parts, nodes, springs, forces and mathematical theories, are assembled for generating a desired formation. Thus, the project's constrains lie in the dynamic and generative notion, the absence of stability results into the incapability of having a certain final result.

The real and digital fusion is signified by the way structures, buildings, and cities evolve, transform and dissolve and it is becoming feasible through the generation, imagining, evolution and control means (Bentley, 2003). On the one hand, nature's programming approach, which combines natural and artificial methods, produces new architectural organisms, new environments and new natures through dynamic processes. On the other hand, technology animates space and users and programs animate matter; "architectural concepts are expressed as generative rules so that their evolution may be accelerated and tested" (Fraser, 1995) and the rules are described in a programming language which produces a code script of instructions for form-generation. In other words, the perfection and variety of natural forms is the result of the relentless evolutionary experimentations and of systems' elaborations in all the possible matter's scaling; hence, programmable nature is a source of inspiration for architectural concept in coded form.

Architecture has frequently drawn inspiration from nature, either from forms and structures, by using analogies and metaphors, or more recently from the inner logic of its morphological processes by using some form of algorithmic procedures, analogous to nature's genetic scripting. Examples of the first kind include Frei Otto's fibre experiments that led to branched constructions (Otto, 2006, figure 01) and Gaudi's particle systems applied to Colonia Guell Church, while examples of the second kind can be found in include the Watercube (the National Swimming Center in Beijing (figure 02), with the randomized appearance of the structure but with a strict, rational geometry that is found in natural systems such as crystals, cells and bubbles (Hwang et al, 2007)), in Anthony Gormley's sculpture and in the algorithmic idea underlying the application of different types of diagrams (e.g. Voronoi diagrams) applied in order to describe the form of the 'Body Space Frame' project.

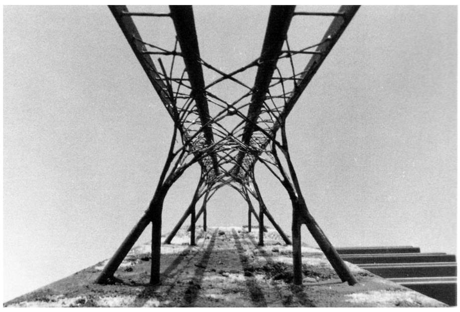


Figure 01: Frei Otto's branched construction (Otto, 2006)

Figure 02: Watercube, Beijing's national Aquatic Centre, (Hwang et al, 2007)

1.2 Tilings

Tilings or tessellations of a plane are firstly an art, initiated early in the history of civilization. Tilings are repeated patterns or motifs in a more or less systematic manner, and various cultures seem to have emphasized on different aspects of art production (figure 03), either as a human portraits' or landscapes representation tool or as complex and colorful geometric designs (Grunbaum, Shephard, 1989). The tilings created by ancient civilizations are made of stone, ceramic or similar materials which fit together without appreciable gaps to cover the plane of some other surface. Famous examples are to be seen in the Alhambra at Granada in Spain, which were an inspiration for a famous tilings' artist, M. Escher (Senechal, 1990, figure 04). Tessellation techniques are used nowadays in computer graphics, in order to manage sets of polygons by dividing them to create complex surfaces.

Figure 03: Detail of Alhambra's periodic tilings, a source of inspiration for Escher (Grunbaum, Shephard, 1989

Figure 04: M. Escher, Circle Limit III, 1959, a tessellation of a hyperbolic plane (Senechal, 1990)

A certain tiles' set, in the Euclidean space and especially plane, 'admits' a tiling if nonoverlaping copies of the tiles in the set can be fitted together to cover the entire space. Periodic tilings are those that remain invariant after being shifted by a translation (Wieting, 1982). On the other hand, "a set of tiles (closed topological disks) is called aperiodic if the plane can be tiled with tiles congruent to those in the set, but without any translational geometry" (Ammann et al, 1992). These aperiodic tilings have applications starting from recreational mathematics up to solid-state physics.

To explain the term "aperiodic tiling" it is necessary to review the definitions "tiling" and "aperiodic". Tiling is a family T= {T_1, T_2,...} of sets (the tiles) which cover the plane without overlaps or gaps. If all the tiles are congruent (oppositely or directly) to a minimal set P= {P1, $P_{2,...,}$ Pn} of tiles, then these are called the prototiles of the tiling T, and they 'admit' the tiling T (Ammann et al, 1992). The symmetries of T are the isometries of the plane that map T onto itself, and T is nonperiodic if it does not have any translation as symmetry. A set of P of prototiles is called aperiodic if every tiling that P admits is nonperiodic (Ammann et al, 1992, figure 06). The first known aperiodic set was developed by Berger R. (Berger, 1966), and consisted of 20426 tiles' shapes which were afterwards reduced to 104 tiles. In the early 70's R. Robinson (Robinson, 1971) devised a relatively simple aperiodic set of six square-shaped tiles with various notches and extensions on their edges to prevent periodic arrangements (figure 05), and R. Penrose (Penrose, 1974) found even simpler sets, with the most popular one consisted of two shapes, named kites and darts (figure 07).

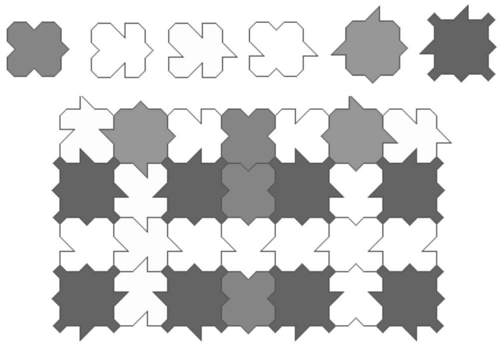


Figure 05: Robinson Tilings, the six simple tiles are based on Wang's work, who achieved tiling a plane with different-coloured edges, called Wang dominoes (Wang, 1961).

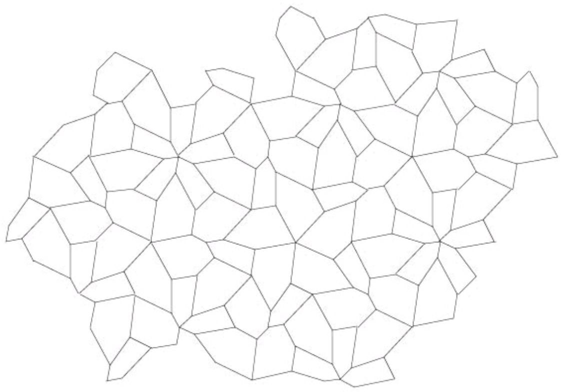


Figure 06: Ammann R. also discovered a number of aperiodic tilings

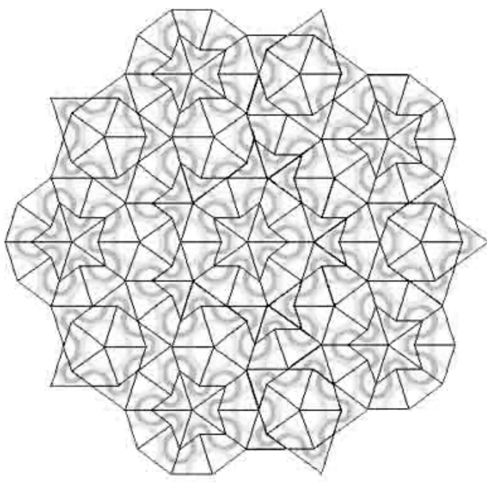


Figure 07: In 1974, Roger Penrose discovered an aperiodic tiling that uses only two shapes, nicknamed kites and darts.

Penrose tilings are named after Roger Penrose, who investigated these sets of tiles in the 1970s. Penrose found a way of tiling a plane with pentagons and avoiding any gaps by attaching three other shapes, called a star, a boat and a diamond. Afterwards, certain rules were formed in order to match the different pieces in an effective way, without leaving any gaps while covering a plane. Penrose found later two more sets of aperiodic tilings, the previously mentioned, kites and darts, and a third one consisted of two different rhombuses, a thin one and a thick one. Most of the tiling sets obtained with Penrose tiles are aperiodic, with an exception of two that involve a fivefold symmetry and a mirror symmetry. Penrose tilings' properties are non-periodic and do not include translational geometry. Furthermore, any finite region in a tiling appears infinitely many times in that tiling despite the non- periodicity property. Last but not least, Penrose tilings can simulate quasicrystals, since they include fivefold symmetry and long range order.

Penrose tilings of the third kind can be described by rhombus tiling, with equal sides but different shapes. There are two different rhombuses; the thin one has four corners with angles of 36, 144, 36, and 144 degrees and it can be bisected along its short diagonal to form a pair of acute Robinson triangles. The thick rhombus has angles of 72, 108, 72, and 108 degrees and it can be bisected along its long diagonal to form a pair of obtuse Robinson triangles. A Penrose tiling cannot be constructed simply beginning with a single tile and adding tiles, one at a time, on a fixed scale, based purely on local translation methods, making random choices when there is freedom of choice (Wang, 1961). Experimentations proved that this method produces a configuration with imperfections, clashes, and inconsistencies. Nevertheless, it appears that some physical substances actually do form into non-periodic quasi-crystals with rotational but not translational symmetry, in a pattern which resembles this aperiodic tiling. Such quasi-crystals are typically quite small, their size probably being limited by how far they can proceed by simple local growth before engendering a clash with the pattern.

1.3 Quasicrystals

The ways in which human mind understands nature and is inspired from its processes, depend on the scales of both distance and time used for the phenomena being studied, i.e. micro scale up to mega scale, and on the possible connections between the different scales (figure 08). Starting from micro scale, the "autopoietic"[1] systems of chemical transformations "pull themselves up by their own bootstraps and become distinct from their environment through their own dynamics, in such a way that both things are inseparable", (Maturana and Varela, 1998). Similarly to that, Penrose mentions (Penrose, 2000) that there is an abstract but inherent connection between different scales, from the relativity theory to quantum mechanics[2], a connection which depends on parameters' modifications (figure 09).

[1] "An autopoietic machine is a machine organized (defined as a unity) as a network of processes of production (transformation and destruction) of components that produces the components which (i) through their interactions and transformations continuously regenerate and realize the network of processes (relations) that produced them; and (ii) constitute it (the machine) as a concrete unity in the space in which they (the components) exist by specifying the topological domain of its realization as such network". (Maturana, Varela, 1980)

[2] Penrose regards consciousness as "a feature of human brains, and probably of many animals as well, which must be accommodated in our overall picture of reality. Human beings can produce arguments that cannot be run on a computer. There must be some physical correlate to this ability to beat even an ideal computer (Turing machine). Quantum processes are taking place in the brain, involving entangled states. Consciousness consists in becoming aware of something definite, so it must involve the OR process (Objective Reduction is an instantaneous event- the climax of a self-organizing process in fundamental space-time (www.Quantumconsciousness.org/penrose-hameroff/consciousevents.html)). Hence the OR process must be non-computational." (Penrose, 2000)

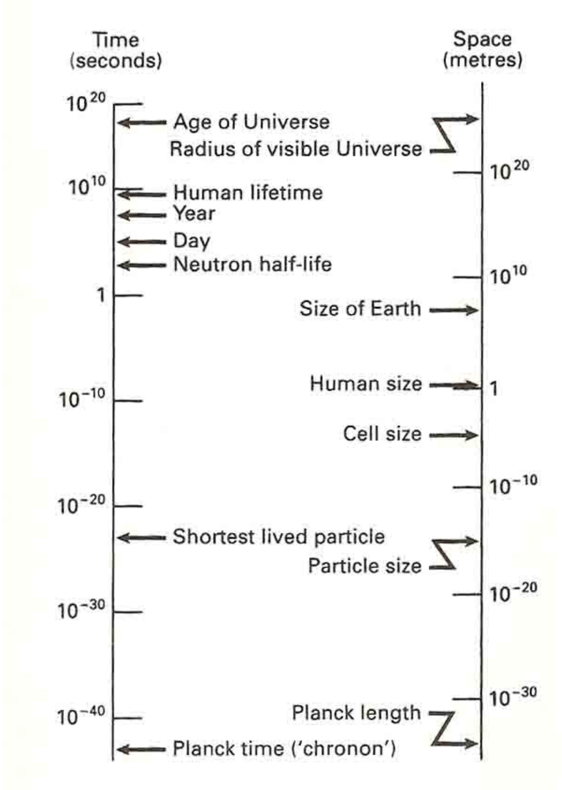


Figure 08: on left-hand side of the diagram, time scales are shown and on the right hand side are the corresponding distance scales, initiating from the very shortest time-scale which is physically meaningful and continuing with the days the years and on the top of the diagram, the present age of the universe. On the right hand side, distances corresponded to time-scales are depicted. The length corresponding to the Plank time is the fundamental unit of length, called the Plank length and is derived from the combination of Einstein's General Relativity with quantum mechanics. "The translation from the left to the right-hand axis of the diagram is via the speed of light, so that times can be translated into distances by asking how far a light signal can travel in that time" (Penrose, 2000).

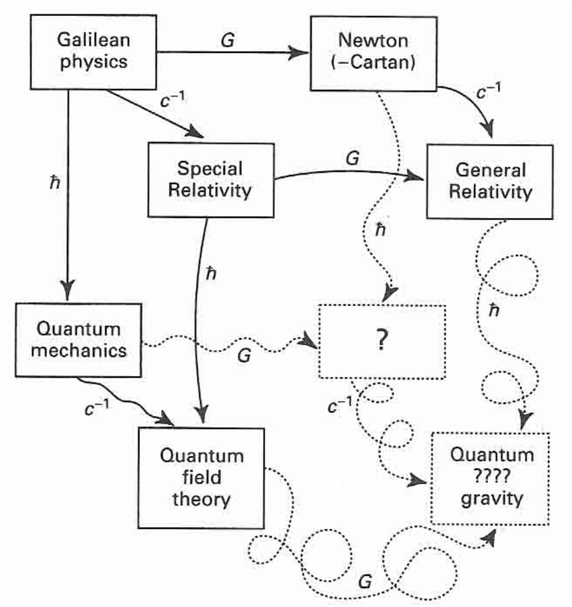


Figure 09: illustration of Penrose's theory including the necessary distortions

Furthermore, the different ways of understanding nature affect the possible inspirations; finding global patterns in locally determined physical processes, i.e. parallelisms in the research scale, is an example of such an approach. Tilings can be found in that way in micro scale natural formations. Aperiodic tilings, especially Penrose tilings, make feasible the atomic quasicrystals' visualization and study, which is a recent discovery about the crystalline atomic structure formations, and they provide explanations for the complex matter's structure.

Quasicrystals are formed in nature by certain alloys under extreme conditions of pressure and temperature, e.g. annealing, or quenching. Most of the known chemical substances, of all the compounds and condensed matter in general, have an energy state which is represented by a periodic crystal structure. The discovery of quasicrystals fundamentally changed the understanding of structural order on atomic scale because of their aperiodic structure. Moreover, tiling patterns and especially Penrose ones provide a means of representation for quasi-crystalls, since the sliced quasicrystal surfaces exhibit an aperiodic tiling pattern.

Traditionally, the atomic structures of pure solids have been divided into two classes, the crystal structures and the glassy structures, with the first ones being highly ordered, having a long-range translational order characterized by a periodic spacing of unit cells, long-range orientational order and a rotational point symmetry. On the other hand, a glassy structure has none of the long range correlations of the crystals but is modelled by spheres that are densely and randomly packed together (Levine et al, 1985). Quasicrystal, a quite recent notion of a new kind of ordered atomic structure and an in-between condition of the previous two, is introduced the last twenty years and can represent a new phase of solid matter which is found in nature.

Quasicrystals are structured forms, which are non-periodic but also ordered. Recent studies in that topic entail that a structure of perfect quasicrystals can be interpreted in terms of a single quasi-unit and simple energetics as shown on experiments on Al72Ni20Co8 as shown in figure 10 (Steinhardt, 2000). Specifying or determining the atomic structure is made simple since the only degrees of freedom are the atom types and the atom positions within the quasi-unit cell. The new paradigm implies a close physical relationship between quasicrystals and crystals. Both can be described in terms of the close-packing of a single low-energy cluster. The key difference is that the atomic arrangement in the case of quasicrystals is constrained to allow atom sharing among neighbouring clusters. Since atom sharing works for only special arrangements, the quasi-unit cell picture naturally explains why quasicrystals are less common than crystals.

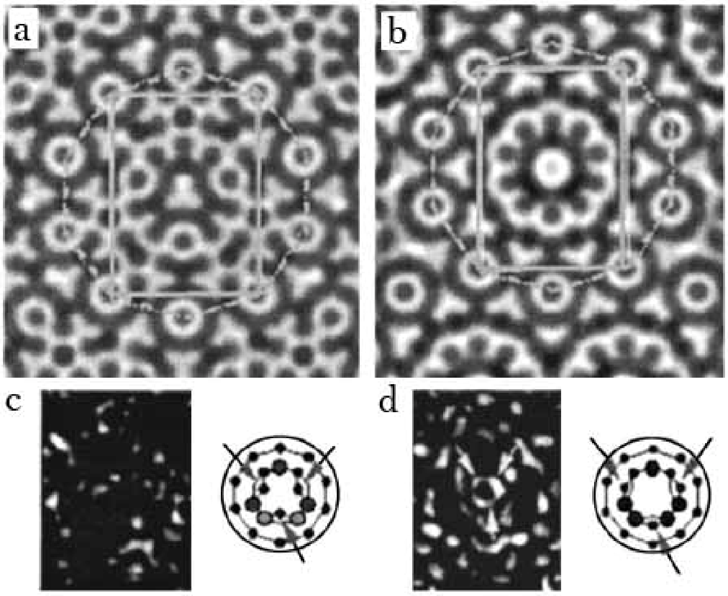


Figure 10: Simulated images of the atomic models in Fig. 2 are shown in (a) and (b), respectively. Differences between the observed and simulated image contrasts of the rectangle regions in (c) and (d) show that the model in Fig. 2(a) with a central triangle of sites in the center (broken symmetry) fits better. A contribution to the difference is the central triangle of sites in (a) vs. a 10-fold symmetric ring of sites in (b) (see arrows) (Steinhardt, 2000).

The quasiperiodic structures by coverings are described as quasi-unit cell approach (Steinhardt, 1996) and regard a process where a single structure motif is sufficient to construct a quasiperiodic structure, while for a tiling at least two different unit cells are needed. Quasiperiodic structures are created in metallurgy when annealing alloys, i.e. warming them at high temperatures (up to a significant fracture of the alloys' melting point) and then letting them slowly cool by air. Alloys get a crystal structure, which depends on the temperature, the alloying elements and their mixture and the annealing process (Grushko, Holland-Moritz, 1997). An example of that can be found in alloys containing Aluminium, Nickel or Cobalt. "The high quality of the basic Ni-rich phase was reflected in a SAED pattern with almost no diffuse background and in HRTEM images showing a highly perfect pentagon Penrose tiling decorated with _20 _A clusters with decagonal symmetry. In a narrow temperature range around 900 _C, the Ni-richest d-phase was found to be stable for a composition d-Al70.2Co5.4Ni24.4. Quasilattice parameters were reported to increase with decreasing Ni-concentration while the lattice parameter along the tenfold axis increases less then 1%" (Steurer, 2004).

The certain quasi-unit cell derives from overlapping rules as it is shown from the decagonal $Al_{70}Ni_{15}Co_{15}$ example (Steurer et al, 1993). A decagon with overlap rules forcing a covering fully equivalent to a Penrose tiling, that contains the maximum possible number of these decagons, was discovered by Gummelt (Gummelt, 1996). There are many theoretical research papers on tilings, especially on the ones serving as quasilattices of decagonal phases; the most famous one is on the Penrose tiling (Ingalls, 1992&1993). The associated models are mainly based on diffraction data and/or on bulk and surface microscopic images (figure 11, 12 & 13).

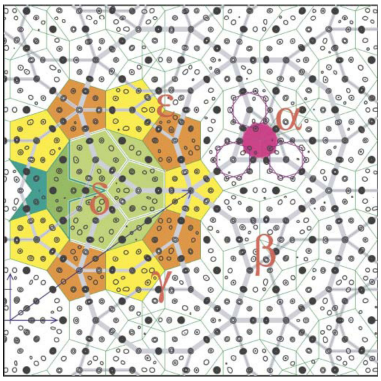


Figure 11: The projected electron density is shown in this contour plot. A pentagonal Penrose tiling is shown with green lines. A group of five Gummelt decagonal clusters is drawn in, labeled by Greek letters. The centre of the clusters α, γ, β, appear clearly asymmetric, while clusters β, δ, appear approximately decagonal. (Steurer, 2004)

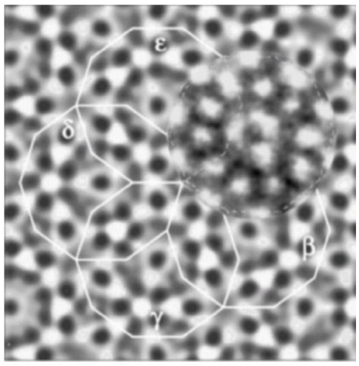


Figure 12: The projected electron density on a resolution comparable to that of the HAADF image (High Angle Annular Darkfield imaging) from a sample with composition Al72Co8Ni20 (Abe, Tsai, 2001) copied upon the place of Gummelt decagon α. Black corresponds to zero density and white to maximum density.

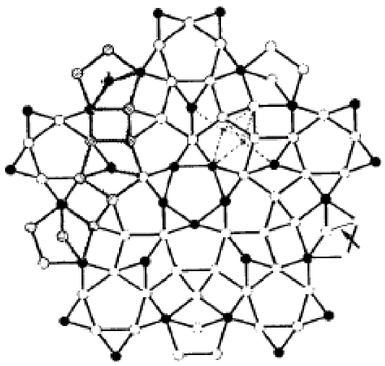


Figure 13: Schematic drawing of the planar basic structural unit of a columnar cluster (monopteros). A crucial role is played by the triangle inside the decagons which can adopt three different equivalent positions as indicated. Open circles correspond to Al and closed circles to Ni/Co atoms.

1.4 Problem's Definition and Thesis Aim

The representation of quasicrystal or aperiodic crystalline structures on 2d surfaces is based on aperiodic tilings, which were discovered during the 1960s; the programming simulation techniques utilize particularly the Penrose tilings. Even though the process which takes place in nature is not completely knownto date, the existing computer simulations apply either L-system methods or cut-and project quasicrystals' methods in order to visualise the tiling' s formations. The basic conceptual difference between these two methods, lies in the closed, rule-based nature of L-systems, which therefore fall in the broader class of bottom- up approaches, contrary to the cut and project method, which is essentially a top- down generation approach.

An example of a top- down, quasicrystal cut and project method on a 2d surface was utilized for the design of the Watercube Aquatic Centre in Beijing. The approach essentially consists of breaking down a system to gain insight into the subsystems, which comprise it. An overview of the system is first formulated, specifying but not detailing any first-level subsystems. A Weaire- Phelan Foam "base cluster" subsystem is composed of 8 irregular tetrahedral, two with hexagonal and pentagonal faces and six with only pentagonal faces, and can be repeated infinitely in the form of nested polyhedra to compose a system. Each subsystem is then refined in yet greater detail, sometimes in many subsequent subsystem levels, until the entire specification is reduced to base elements. When this structure is rotated by a particular angle and then sliced, an irregular 2d surface pattern emerges. These tiled surfaces are used to cover a cube, which is the shape of the Centre (figures 14 & 15). "Though the foam is regular and only composed of two different tetrahedral polygons, the slicing of infinite geometry produces a block of incredible tiling variation" (Hwang et al, 2007). This method is called "cut and project method", since the structure is cut and then projected on a 2d plane, which in this particular case consists of the cube's sides. For structural purposes, the number of the resulting tilings has been limited to seven variants in the roof and sixteen in the walls.

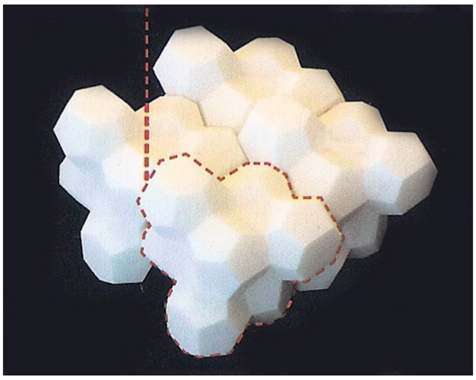


Figure 14: Entire geometry , the idea behind the Watercube((Hwang et al, 2007)

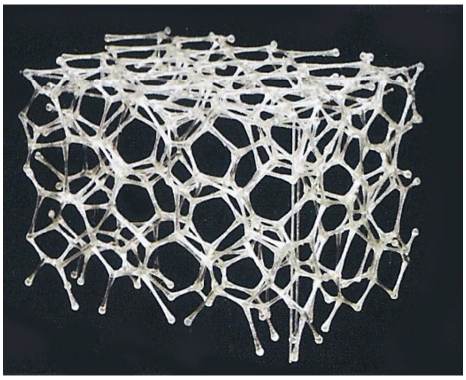


Figure 15: Sliced geometry, the actual Watercube (Hwang et al, 2007)

L-systems are a theoretical bottom- up framework for studying the development of simple multicellular organisms which develop into more complicated ones (Lindenmayer, Prusinkiewicz, 2004); in other words it is a parallel rewriting system composed of a set of rules and symbols that allow for the morphological description of a variety of organisms and complicated structures. L-systems are introduced so as to simulate development, with all their subunits or symbols being transformed simultaneously into a subsystem. Adding together systems gives rise to greater systems, which are in turn linked, sometimes at many levels, therefore making the original systems sub-systems of the emergent and complete L-system. (Lindenmayer, Jurgensen, 1992). Since Penrose tiling can be obtained from L-systems either with the method of inflation or deflation (figure 16), this means that aperiodic tiling and Penrose tiling are characterized by self-affinity, therefore that the tiling is fractal in nature (Gardner, 1977). The recursiveness simulated with L-systems starts developing from the initial state and continues with as many rules as possible are applied simultaneously per iteration; the application to the Penrose tiling is initiated with an amount of certain tilings. The L-systems resemble the algorithms used for generation of fractal objects such as Peano curve, Koch's recursion curve, etc. and allow for considering the tilings as cluster growth and as natural fractal phenomenon.

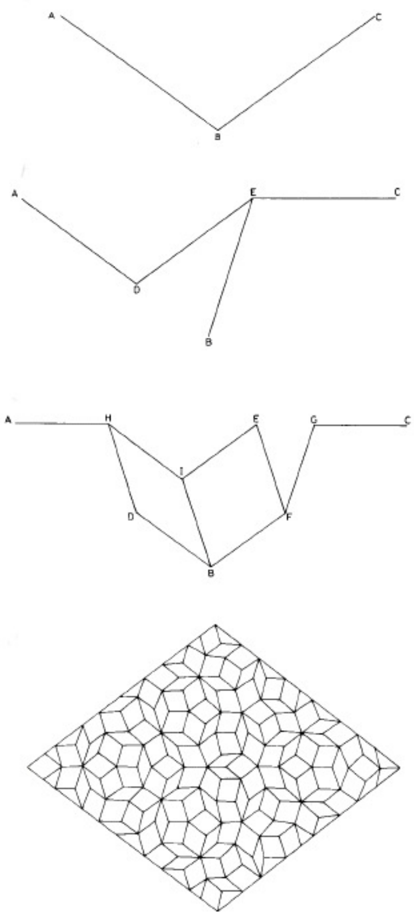


Figure 16: Algorithmic generation of Penrose tilings by using L-systems

A question that emerges, which later defines the topic of this study, is whether these processes, the natural ones or the programming ones can be replaced by an easier geometric and programming one; in other words, whether a Penrose tiling pattern can emerge in a bottom- up way by allowing a particle spring system to rearrange nodes in 2-d space. Therefore, the project aims in developing a self-organizing system, starting from an initially given number of nodes which comprise a spring system that attracts, repels and re-arranges these nodes to create Penrose tiling patterns. The system can be set to be either free to expand in the 2d space or constrained by linear boundaries, thereby defining a quadrilateral which has to be tiled.

The idea of this thesis is inspired by the ways Anthony Gormley's sculptural problems are faced; Gormley's sculpture "Body/ Space/ Frame" is composed by a 25 meter high steel lattice, which resembles the shape of human body in a crouching position and is located on the coast Zuiderzee. The problem in Gromley's case was to fill the idealised shape of the human figure, with a structurally stable space frame network (figure 17). "Within this framework, algorithms have been devised specifically for the project to assist in placing the structural members and their connections in a manner that both describes the form of the body in space and provides structural integrity" (Hanna, http://www.sean.hanna.net/bodyspaceframe.htm). Top- down geometry and bottom- up self organisation are examined in this project. The 'rules' examined at the beginning are aesthetic demands on final form, whereas the spring system actually produces these requirements naturally without explicitly stating them. The aforementioned problem can be rephrased as a search for a panelling structure, either 2d or 3d which is either a flat surface or a complex curved surface that is comprised of aperiodic tiling patterns and in particular, Penrose Tiling.

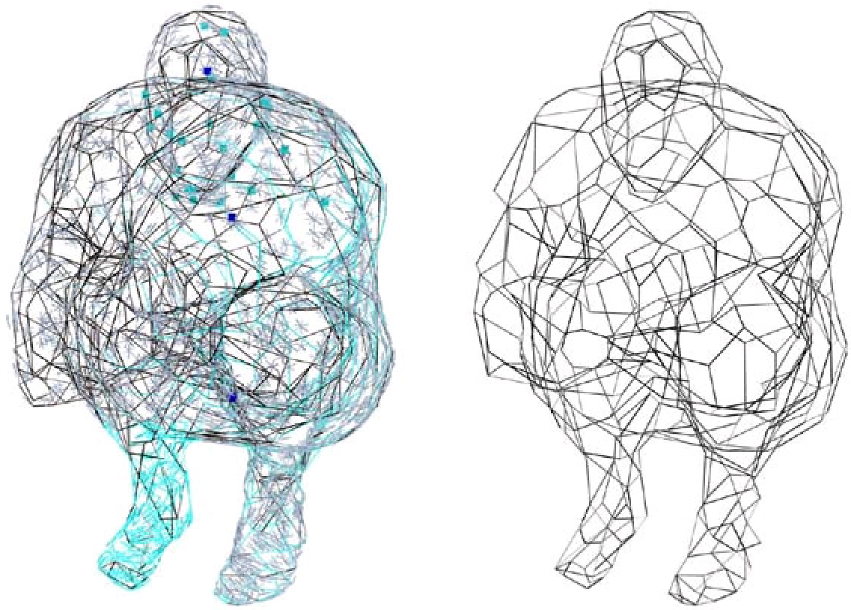


Figure17: Anthony Gormley's "Body/Space/Frame", "The geometric concept that evolved in design was that of an external skin of open polygons, a sort of tensile net resembling bubbles or sea foam, surrounding a triangulated interior of 'starbursts' of structural members. In addition to the differing structural implictions of rigid and pin jointed nodes, the coupling of the two systems required several approaches to member placement" (Hanna, http://www.sean.hanna.net/bodyspaceframe.htm).

The project's main limitations are those that set the boundaries for the research; first of all, the integration of all the characteristics of an aperiodic tiling pattern is the most important part. In other words, the creation of aperiodic tiles in such a way that they agree with the chosen structural process, that is the self-organized system. This system consists of a fixed number of nodes within certain boundaries within the 2d space; it tries to connect these nodes with strings and at the same time evenly distribute the nodes within certain distances. The tiles' topology is intended to approximate the form of the given boundaries in the best possible way.

1.5 Thesis Outline

The purpose of this project is to generate 2d surfaces comprised of aperiodic and especially Penrose tiling patterns using a bottom-up approach within the framework of a self-organising particle spring system. Within these borders, certain parameters are added afterwards to optimize the result. In Section 2 the basic algorithmic methods used are described, while in Section 3 the proposed method is further analyzed and the basic components of the algorithm are explained. Section 4 focuses on the experimentations with the algorithm and its variations and some interesting results, while Section 5 summarises the conclusions and makes some suggestions for future research and analysis. The basic components of the code are presented in the appendices.

2 Review of Related Work

In this section, particle- spring systems, dynamic relaxation and self- organization, which constitute different parts of the experimented code, are reviewed. The proposed method for dealing with the problem of tiling generation utilizes a numerical method of an algorithm that dynamically simulates a particle-spring system (Kanellos, 2007). These particles are positioned on a surface and are connected with linear elastic springs and are assigned an initial velocity and damping coefficient; their response (oscillation about their initial configuration) is traced by means of a dynamic relaxation algorithm. The particle spring system continuously tries to organize itself into more complex structures under the influence of external forces.

2.1 Particle Systems

A Particle System is a collection of points in 2d space or point masses in 3d space which are connected with springs, are affected by external forces and obey the laws of physics. They are particularly used in computer graphics as rendering techniques for simulating and visualising chaotic natural and artificial phenomena like fire, water, explosions, etc.

The algorithms usually applied to particle systems are called 'Force directed algorithms'; space is discretised into a number of nodes, which represent bodies of the system, with forces acting on or between them. These forces are often based on physical laws, and therefore have a natural analogy, such as magnetic or electrical attraction or repulsion or gravitational attraction. It is also possible for the system to simulate forces acting on the bodies with no direct physical analogy, e.g. the usage of logarithmic distance measure rather than the Euclidean one.

The 'Force directed algorithms' are essentially an iterative procedure to derive equilibrium configurations for the nodes, on which the forces act, regardless of the nature of the applied forces. The algorithms are applied in discrete time steps, with the respective nodal forces being adapted at the end of each step according to the new nodal configuration (Kaufmann, 2001). They can also be viewed as an iterative procedure for minimising the potential energy associated with the forces acting within the field of the nodes, since stable equilibrium configurations correspond to local minima of the potential energy function.

The particle spring systems can be also utilized to find structural forms on which only axial forces act and to specify the equilibrium position corresponding to a specific set of actions while allowing user's interaction within the simulation process. These solutions are helpful when searching for an equilibrium position of a structural system, which can be a membrane structure, a roof system or a lightweight structure. Kilian and Ochsendorf (2005) applied this concept by considering a chain of weightless, axially stiff springs connecting certain masses inside a simulated 3d space; for specific values of the springs' length, stiffness and specific masses the equilibrium configuration changes. Gaudi's physical hanging chain models for Colonia Guell Church (figure 18 & 19), which are based on the theory 'reversion of the catenary'[3], work in a similar manner to the certain simulation. This resulted in vaulted forms that require less material and therefore smaller mass" (Otto, Rasch, 2006).

[3] "Catenary is the theoretical shape of a hanging flexible chain or cable when supported at its ends and acted upon by a uniform gravitational force (its own weight) and in equilibrium. The chain is steepest near the points of suspension because this part of the chain has the most weight pulling down on it. Toward the bottom, the slope of the chain decreases because the chain is supporting less weight." (http://en.wikipedia.org/wiki/Catenary)

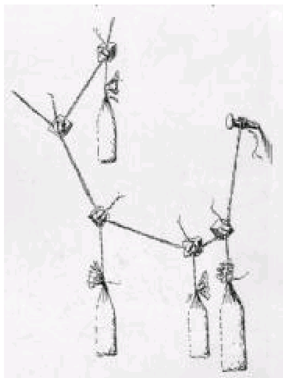



Figure 18 & 19: Gaudi's hanging model, a system of threads represents columns, arches, walls and vaults. Sachets with lead shot resemble the weight of small building parts (http://me-wserver.mecheng.strath.ac.uk/group2003 /groupl/Inspirational%20people/Antoni%20Gaudi_files/Antoni%20Gaudi.htm)

2.2 Dynamic Relaxation

Dynamic Relaxation is an algorithmic method for finding the deformed configuration of a cable or fabric structure subjected to a set of forces. It is based on discretising the continuum into nodes on which the total mass of the structure is lumped and tracing the pseydo-dynamic response (oscillation) of the structure to the applied forces (Topping, Khan, 1994). This method is used in structural analysis to derive the solution of a static problem by using a fictitious damped dynamic analysis. In this way a static problem may be solved with explicit vibration analysis, thereby eliminating the need for forming and inverting the structure's stiffness matrix which can result in reduced computational times for finely discretised structures (Hibbitt, Karlsson & Sorensen 2006). It is essentially a direct application of Newton's Second Law, where "A force 'F' acting on a body gives it an acceleration 'a' which is in the direction of the force and has magnitude inversely proportional to the mass *m* of the body":

$$F = m \times \alpha$$

The Dynamic Relaxation system is comprised by particles which are positioned on a surface and are connected with linear elastic springs, with an initial length and a damping coefficient. It has been used extensively for analysis and design of cable and membrane structures, or tensile structures, which can carry only tension. Tensile structures are membranes with high tensile but practically zero compressive stiffness. Therefore, no compressive (and hence no bending and shear) stresses can be developed and they resist loads solely by transforming their geometry, in a way that their tensile stiffness is activated. The origins of the design of contemporary tensile roof-structure are based on physical modelling techniques, which are inspired by natural forms and self-formation techniques like the aforementioned Gaudi's example; the research in the sphere of this subject lies in the operative forces that can bring about synthesis, change and transformation (Otto, Rasch, 2006). These modelling processes have a life on their own; they are set in motion and the results are not always predictable.

The dynamic relaxation method has been applied to the British Museum Great Court Roof, where a "combination of analytic and numerical methods were developed to satisfy architectural, structural and glazing constrains" (Williams, 2001). The roof covers a 70 by 100 meters area containing the 44 meter Reading Room (figure 20). The roof is comprised of triangular grid of steel members welded to node pieces with one flat panel of double glazing for each triangle of the structural grid. These nodes of the mathematical model are allowed to float freely on top of the surface of the shape, to slide on it without friction, so as to control the maximum size of the triangles to some structurally difficult areas, like towards the centre of the construction or towards the borders of it.

Another example of an application of dynamic relaxation on the analysis of a roof structure is the Montreal baseball stadium which is a lightweight structure of the cable-stayed system type, with a double curvature membrane covering an ellipse-shaped opening of 200 by 140 meters. Membrane shape is obtained by a uniform elliptic pretension with seventeen anchorage points all around the perimeter and twenty six internal suspension points. Due to the flexibility of the roof of the Montreal Olympic Stadium (figure 21), "fluid acts on the directly exposed outer surface of roof as well as on its internal surface depending on the internal volume changes. The roof encloses a fluid volume that can be classified as an appendage of external fluid where the connection is due to exchange surfaces, such as stand access openings, etc. In case of large openings, the fluid field produces point-to-point variable internal pressures, acting on the internal side, which are rather small but not negligible when measured against external pressure level. Conversely, such pressures are not generated when the volume is closed" (Lazzari et al, 2008).

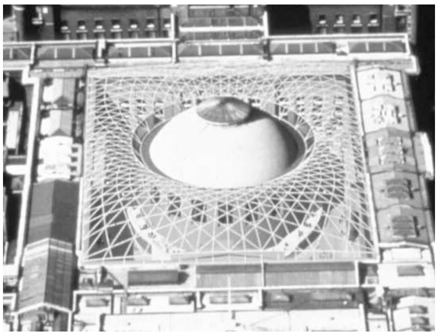


Figure 20: British Museum Great Court Roof (William, 2001)

Figure 21: Montreal Olympic Stadium roof (Lazzari et al,2008)

2.3 Self- Organizing Systems

Self organizing systems are used as an inspiration reflected in various complex phenomena and paralleled with global patterns; they are regarded as an environmentally driven evolution process understood on the basis of the same variation and natural selection and are normally triggered by internal variation processes, which are usually called "fluctuations" or "noise".

"Self organization is a spontaneously formed higher-level pattern of structure or function[4] that is emergent[5] through the interactions of lower-level objects" (Flake, 1998). The emergent behavior or self-organization is about simple processes leading to complex results; it's about the whole being more than the sum of its parts. Self-organizing systems usually exhibit what appears to be spontaneous order (Beer, 2004); it results from the system's continuing efforts to organize itself into more complex structures under the influence of external forces. Additionally, the system is considered to exhibit emergent properties, and the different parts that compose the system tend to organize themselves according to local information, processes that work near the 'edge of chaos' (Kennedy, 2001). In other words, self-organization is a process where the organization of a system increases in complexity without being controlled by the environment or an encompassing or external system.

Examples of such self- organized systems can be found in germs, bugs, varmints and widgets societies. "Termite builders, for instance, are one kind of self-organizing system. There is no central control, the intention of the population is distributed throughout its membership—and the members themselves are unaware of the 'plan' they are carrying out. Actors in the system follow simple rules and improbable structures emerge from lower-level activities, analogous to the way gliders emerge from simple rules in a cellular automaton" (Kennedy, 2001). The same applies to insects; different hormones are responsible for different types of messages as an output of gland secretion in response to a stimulus.

[4] Function is a mapping from one space to another. This is usually understood to be a relationship between numbers.

[5] Emergence refers to a property of a collection of simple subunits that comes about through the interactions of the subunits and is not a property of any single subunit. For example, the organization of an ant colony is said to "emerge" from the interactions of the lower level behaviors of the ants, and not from any single ant. Usually the emergent behavior is unanticipated and cannot be directly deduced from the lower-level behaviors.

Society is another more complex example of a self-organizing system with global properties that cannot be predicted from the properties of its individual members. A computer simulation called 'the Warsaw Simulation System' contributed significantly to the determination of the extent to which group-level phenomena result from individual-level processes. "The program simulates the changes of attitudes in a population resulting from the interactive, reciprocal, and recursive operation of a theory regarding social impact, which specifies principles underlying how individuals are affected by their social environment. Surprisingly, several macrolevel phenomena emerge from the simple operation of this microlevel theory, including an incomplete polarization of opinions reaching a stable equilibrium, with coherent minority subgroups managing to exist near the margins of the whole population" (Nowak, et al, 1990).

Latané calls the current incarnation *dynamic social impact theory,* and his findings have developed beyond simple polarization. Dynamic social impact theory results, whether in simulations or studies with human subjects, are seen to possess four characteristics, as described by Latané in numerous publications (Nowak, et al, 1990). These are consolidation, clustering, correlation and continuing diversity.

3 Generic methodology of the code

3.1 Algorithm's overview

The methodology followed in the current project is comprised of three basic algorithmic components: the particle- spring system, dynamic relaxation and self- organization; these have been discussed in detail in Section 2 and are combined herein within the framework of the current project, in order to deal with the geometric problem of generating an aperiodic Penrose tiling pattern. Their main characteristic is the bottom- up approach for the creation of the self- organized tiled surface, where dynamic global and local rules of particle- node interaction and correlation are applied.

The main concept is to start from a node-spring system and subsequently generate a tiling-like surface with the aid of application of suitable nodal forces. Despite the conceptual simplicity of the followed methodology, some significant problems arise, which prevent the successful generation of Penrose tiling patterns. These problems and some potential solutions are discussed in Section 4.

The programming language Processing (Reas, 2007) has been extensively utilized throughout the project for the application of the algorithms. The surface to be tiled is bound in a way that the nodes are obliged to settle and to achieve an equilibrium state within well-defined boundaries as explained in §3.2.6.

3.2 Description of the algorithm

In the remainder of this Section, a breakdown of the algorithm in its basic components is attempted. Each component as well as their interaction is discussed in detail.

3.2.1 Nodes

A cluster of points or nodes is generated within 2d space. These include three main characteristics, which are updated in each time-step: velocity, position and direction, as can be seen in §A1. Later on, an identity (ID) will be labelled to each one of them, in order to confront the problem of 'line crossing' as discussed in Section 4. The nodes are assumed initially to have zero mass for simplicity; it should be noted that the original inspiration for the particle spring system, namely Gaudi's analogical experiments applied to 3d space, which was discussed in §2.1, did not include this assumption.

Initially (first iteration) the nodes have a random position, within a priori specified boundaries; they are also assigned a small and random initial velocity and a damping coefficient. Subsequent iterations lead to changes in the nodal position in an attempt to achieve (global) equilibrium throughout the whole system. The position of each node is derived from the sum of its current position and its equivalent velocity (as specified at the end of the previous increment) for each time step. The equivalent velocity is determined by the interaction of the node with its neighbouring ones. Hence, the updated position is derived from the previous position and the particles interactions. Since each node affects its neighbouring ones, the position of the nodes is continuously updated, upon each iteration, until all nodes are in equilibrium under the nodal forces they are subjected to. In some cases, this process results in a never ending loop, since the boundaries, spring forces, initial number of nodes and specified drive (as defined in the subsequent subsection), are not necessarily compatible with each other and hence the determination of an equilibrium nodal configuration may not always be feasible.

Therefore, an artificial damping coefficient is introduced (see §3.2.4) and it is in this way it is ensured that a static (yet not necessarily in equilibrium) nodal configuration will be obtained within a finite number of iterations.

3.2.2 Creation of springs

This subsection focuses on the establishment of spring connections between adjacent nodes, which depends on the nodes' proximity. In every increment and for the current (in that increment) instantaneous position of the nodes, the program checks the possibility of establishing a spring connection (and thereby defining a nodal pair), based on the proximity of the nodes comprising the potential nodal pair; i.e. it checks whether the distance d between two nodes is smaller than a specified threshold, which is a multiple of a user-defined parameter called '*drive',* and if it is less than that, then a connection between these two nodes is established and a line connecting these two is depicted on screen. The magnitude of the drive can be increased or decreased by pressing the "q" and "a" buttons respectively. More on this subject is discussed in §3.2.5.

After all appropriate spring connections have been created, the algorithm subtracts a user-defined length l_0, smaller than the drive ($l_0 < drive$), from the distance d. If $d - l_0 < 0$, a repelling force is applied between the two nodes, whereas if $d - l_0 > 0$, an attractive force is applied on the node pair (figure 22). The repulsion's and attraction's direction is set parallel to spring's direction. The sum of all nodal forces, to which each node is subjected, is derived from the spring interactions and is stored as a vector, which is subsequently added to the velocity vector at the beginning of the next increment. The algorithm is repeatedly applied for each time step and it recalculates the new nodes positions along with the spring links that have been drawn, until global equilibrium is obtained, or until the damping (§3.2.4) applied to the velocity vector leads to a premature termination of the iteration procedure.

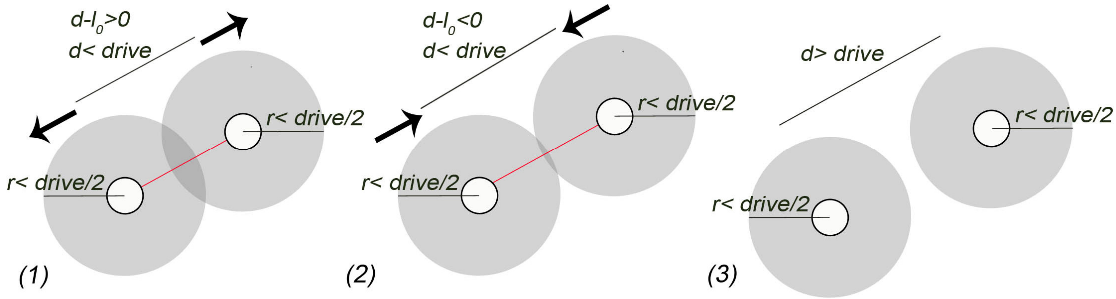


Figure 22: possible nodes relations

(4) *the pair of nodes is close enough to establish a spring connection, but their distance d is less that the l_0, as a result, the two nodes are repelled*

(5) *the pair of nodes is close enough to establish a spring connection but their distance d is greater that the l_0, as a result, the two nodes are attracted*

(6) *the pair of nodes are too far away from each other, as a result, there is no spring connection is established*

3.2.3 Spring forces

The spring forces are vectors parallel to the line connecting each pair of nodes i.e. the subtraction of their position vectors. Experimentations proved that the most important forces for the fastest and computationally most efficient determination of the system's equilibrium configuration are the repulsion ones. When having both repulsion and attraction forces, other important features of the code, like self- organization and tiling creation, are downgraded and a chaotic system emerges after a few iterations. As a result, the initial choice of having both repulsion and attraction forces is abandoned and only the repulsion forces are kept, in order to obtain the desired length. Hence the l_0 parameter referred to in §3.2.2 is also discharged.

3.2.4 Damping of nodal velocity

Nodal velocity is damped from each time- step (each iteration) to the next one, so that a static nodal configuration can be assumed in a finite number of increments. The damping coefficient is set to a default value of 0.95, i.e. the nodes preserve 95% of their initial velocity magnitude which is later added to the new applied forces in the subsequent step. It should be noted that in this way the desired nodal distances (§3.2.5) may not always be obtained. By setting the damping coefficient to unity, the damping effect can be removed and preservation of the nodal velocity holds.

3.2.5 Formation of tiling patterns

Since the main target of this project is to create Penrose tiling patterns, some geometric relations between the drive parameter and the tiles' dimensions have to be specified. Three basic lengths, namely d_0, d_1 and d_2, are defined in figure 23.

Two different types of rhombuses are used as Penrose tiles in the current project as depicted in Figure 24. Relations yielding rhombuses' small diagonal lengths d_1 and d_2 as a function of their side d_0 and angles are defined by equations (1) and (2).

$$d_1 = 2 \times d_0 \times \sin\left(36^o/2\right) \quad (1)$$

$$d_2 = 2 \times d_0 \times \sin\left(72^o/2\right) \quad (2)$$

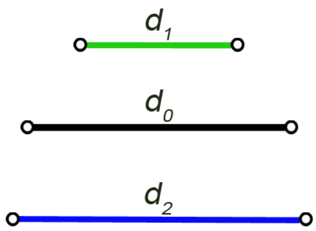


Figure 23: The three lengths involved in the desired Penrose Tiling pattern.

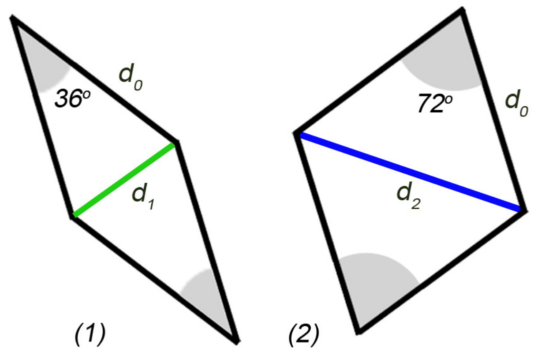


Figure 24: The two Penrose rhombuse:, the first one (1) with angles of 36^0 and 144^0 and the second one (2) with angles of 72^0 and 108^0.

Since each rhombus is made up of two symmetric triangles, each of which has two sides equal to d_0, the spring system attempts to create those triangles rather than the whole rhombus. It is however acknowledged that a random combination of the sides d_0, d_1 and d_2 can generate ten different triangles out of which only two have the desired geometry as shown in figure 25. Furthermore, it can not be guaranteed that the resulting triangle pattern will necessarily include the rhombuses. However, within the framework of particle- spring system only a triangulated tiling pattern can emerge. Otherwise, a fundamentally different approach which would consist of placing building blocks (rhombuses) within a bounded rectangle and ensuring that no overlapping or gaps occur, should be pursued. As stated by Wang (1961) such an approach does not lead to penrose tiling formation.

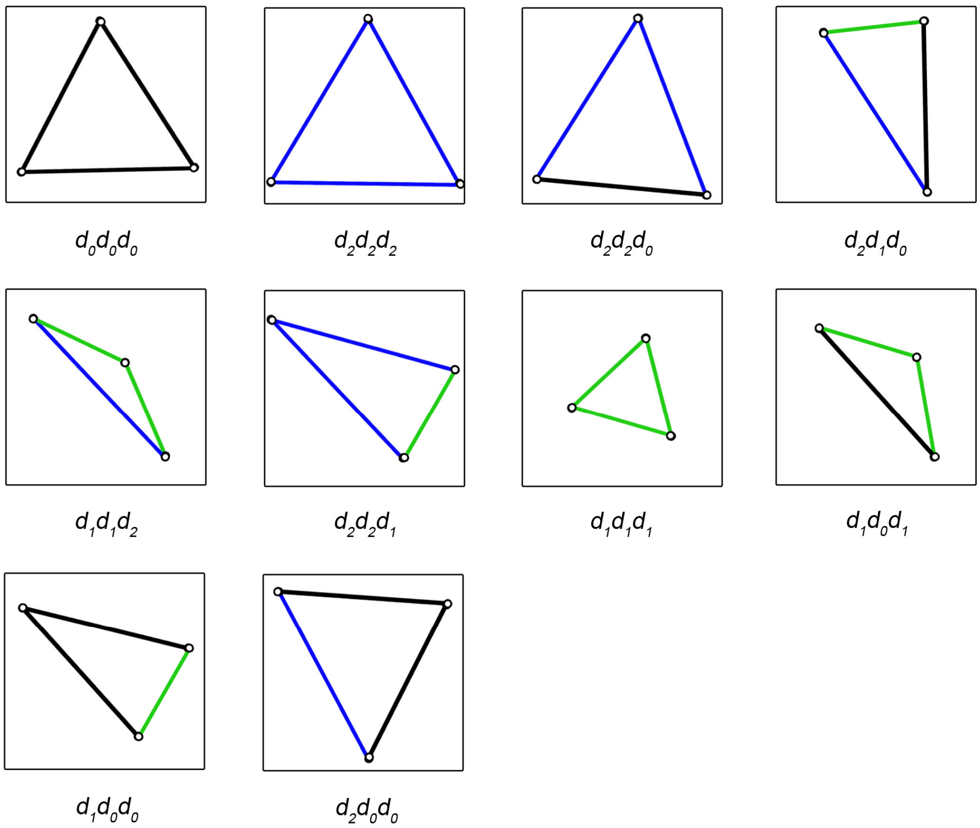


Figure 25: All the possible combinations of the three different sides with the last two ones being those that regard the Penrose tiling.

The rhombuses' side d_0 (and hence d_1 and d_2 as well) is defined with respect to the user-defined drive parameter and hence its magnitude can be controlled by the user. The application of Equations (1) and (2) imposes geometric constraints on the spring formation process and forces the nodal connections of the desired length proportions to be formed, resulting in tiling patterns. It is noted that it is the ratio of lengths and not the lengths themselves that are of interest, since all actual lengths can be scaled up and down according to the drive parameter. However, controlling the frequency of occurrence of each of the three lengths necessitates the introduction of further user-defined parameters.

The drive parameter is set equal to d_0 which is the largest of the three lengths. Hence a spring connection between adjacent nodes is established when their respective distance d is smaller than $d_2=1.1755\times drive$. Four cases can arise according to the magnitude of the distance d with respect to the drive parameter:

- if $d> 1.1755\times drive$

  no spring connection is established
- if $d< (0.618+0.389\times f) \times drive$

  the minimum spring connection d_1 is established
- if $d< (1+0.1755\times h) \times drive$ and $d> (0.618+0.389\times f) \times drive$

  the medium length spring connection d_0 is established
- if $d> (1+0.1755\times h) \times drive$

  the largest spring connection d_2 is established.

The "f" and "h" are additional parameters, which implicitly affect the relative frequency of occurrence of each of the three spring connection as will be demonstrated in Section 4.

3.2.6 Surface's boundaries

The node-particle system is developed within certain boundaries, in order to achieve a tiling formation within a plane of specific dimensions. The initial nodal position is set within rectangular boundaries and the algorithm calculates for every increment the nodal position and checks whether they are positioned outside the boundaries. The boundaries are rigid (i.e. non-deformable) and the particles are not allowed to cross them. When in contact with the boundaries, an additional velocity, perpendicular to the boundaries and facing inwards, is applied on the nodes. In this way, it is ensured that the movement of the nodes remains within the specified boundaries. The vector operations for node-boundary contact detection and response are depicted in figure 26.If the node collides to more than one border (i.e. corner of the bounded rectangle), then the new position vector is the sum of the previous nodal position vector and the two additional due to contact with both boundaries. However, it is possible to disregard the influence of the boundaries on the spring system, by pressing the "f" button during the execution of the program; this activates a Boolean function which renders the boundaries ineffective. By pressing the "b" button, the boundaries can be reapplied.

.

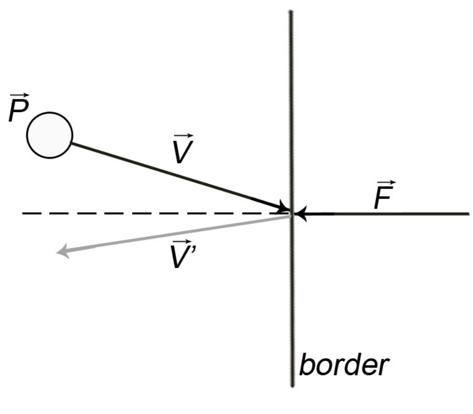


Figure 26: Vector calculations to detect contact with the boundaries and determine the nodal response once the contact has occurred.

4 Experimentation and results

This section summarises the various experimentations with the basic code features and their variations and depicts some illustrative results. It focuses on the influence of the most important user-defined parameters on the generated results. These are the f and h parameters, which affect the frequency of occurrence of the three basic spring lengths, the damping coefficient and the activation/deactivation of the boundaries. Furthermore, the need for adjusting the main code in order to minimise overlapping of the spring connections is explained.

It should be noted that the number of nodes, the dimensions and geometry of the bounded region and the drive parameter all affect essentially same thing, namely the dimensions of the resulting tiling patterns with respect to the bounded region, which is an inherent part of the tiling problem itself rather than the current code. Therefore all of these parameters are set to specific values and hence their individual influence on the results has not been considered, in order to obtain comparable results and assess the effect of the inherent parameters of the code (f and h parameters, damping coefficient, etc). For all figures depicted in this Section, the bounded region is square with all sides equal to 250, the number of the nodes is 16 and the drive parameter is set equal to 45.

4.1 Controlling the frequency of occurrence of d_0, d_1 and d_2

As stated in §3.2.5 two additional parameters, “f” and “h” have been introduced in the code so that the relative frequency (i.e. number of occurrence of each length of spring connection normalised by the total number of spring connections established) within the particle system of the three basic lengths d_0, d_1 and d_2 can be controlled. The code forces spring connections to be established between adjacent nodes ($d<d_2=1.1755\times drive$). Once a connections is established, equal and opposite forces are applied to each node of the node pair, which force the spring connection length to assume one of the three basic lengths, namely d_0, d_1 and d_2, depending on the distance on the nodes of the node pair (as calculated for the given iteration):

- if $d> 1.1755\times drive$

  no spring connection is established

- if $d< (0.618+0.389\times f) \times drive$

  the minimum spring connection d_1 is established

- if $d< (1+0.1755\times h) \times drive$ and $d> (0.618+0.389\times f) \times drive$

  the medium length spring connection d_0 is established

- if $d> (1+0.1755\times h) \times drive$

By adjusting the values on f and h parameters, the generation of generation of some connection lengths increased or decreased, or even excluded. For f=h=0.5, the spring connection length tends to the closest of the three basic lengths, whereas for f=1 and h=0 the generation of d_0 lengths is excluded. On the contrary, most d_0 lengths can be generated for f=0 and h=1. Figure 27 depicts the effect of these parameters on the generated results.

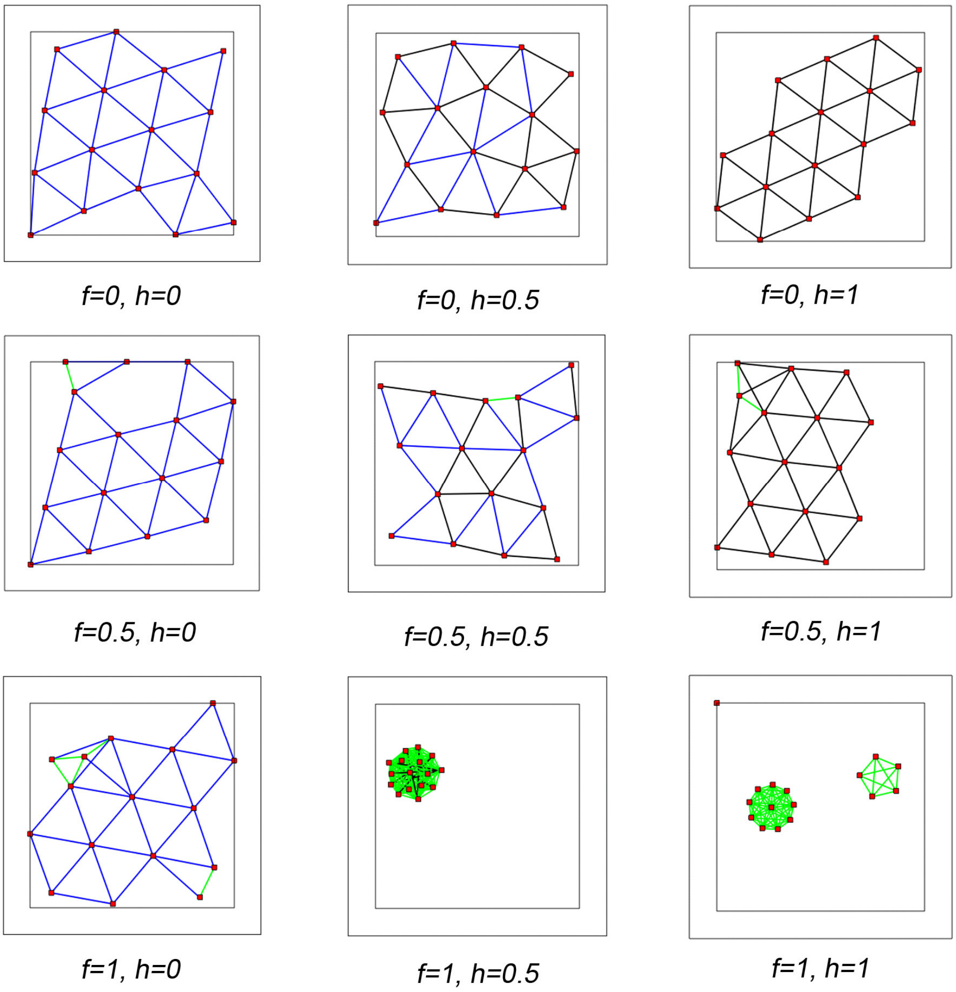


Figure 27: effect of the "f" and "h" parameters on the generated results (the damping coefficient is set to unity)

From the previous figure it can be seen that some overlapping of nodal connections (line crossings) takes place in certain cases, which can lead even to the collapse of the whole algorithm as illustrated for the cases (f=a, h=0.5) and (f=1, h=1). This problem is dealt with an adaptation of graph theory equations where mathematical structures are used to model pairwise relations between objects from a certain cluster. The additional pieces of code test whether the generation of a new spring connection should be avoided in case of overlapping springs. The code sorts the nodes, gives them an identity and afterwards compares the two nodes of each pair so as to judge if a new connection could be established. Having sorted the overlapping issue, the effect of f and h parameters is revisited in figure 28.

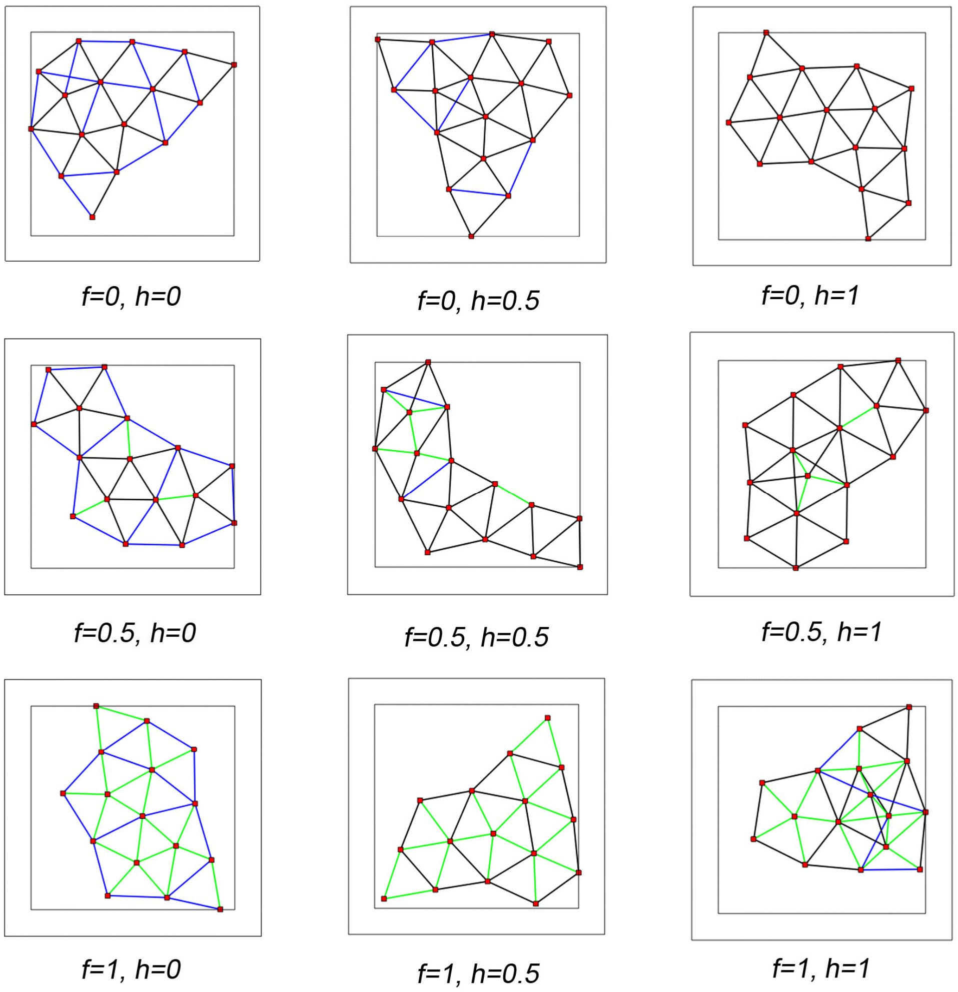


Figure 28: effect of the "f" and "h" parameters on the generated results including additional code to overlapping of spring connections (the damping coefficient is set to unity)

It can be seen that the additional code strings greatly minimize (but do not completely solve the problem) the occurrence of line crossings (overlapping of spring connections). In some cases aperiodic pentagonal tiling patters emerge. It is also demonstrated that the incorporation of the additional "f" and "h" parameters can affect the occurrence of the desired lengths, yet not in an explicit way (i.e. we cannot assign a specific probability of occurrence to a specific combination of f and h values). In the remainder of this Section both "f" and "h" are set equal to 0.5 and the effect of other parameters is examined.

4.2 The influence of damping

In the previous subsection the damping coefficient was set to unity, hence no damping was incorporated into the models. The lack of damping resulted in a never-ending oscillation of the particle spring system about a quasi-stable configuration. In some cases this oscillation was hardly noticeable indicating that a stable configuration was reached. All figures in §4.1 depict an instantaneous image of a moving system, rather than a stable equilibrium configuration. This subsection focuses on the effect of the damping coefficient and attempts to propose an optimal damping value (if any) which results in both the desired triangular tiling pattern and a stable equilibrium configuration.

The damping coefficient produces a stabilising effect, by decreasing the magnitude of the velocity vector upon each iteration. With decreasing damping coefficient global equilibrium can be rapidly reached. However the desired lengths of the nodal connections may not have been obtained, since the system may cease to oscillate due to the damping effect and not due to the attainment of a stable configuration. In such cases, the increase of the drive parameter is necessary so that oscillation may reinitiate. On the other hand, the absence of damping, may lead to a continuous oscillation of the particle system. It is a compromise between accuracy and runtime since any damping coefficient significantly smaller than unity alters the application of the main concept and introduces artificial 'stability' to the system. Figures 29, 30, 31 and 32 depict the generated results for varying damping coefficient and a drive parameter=30.

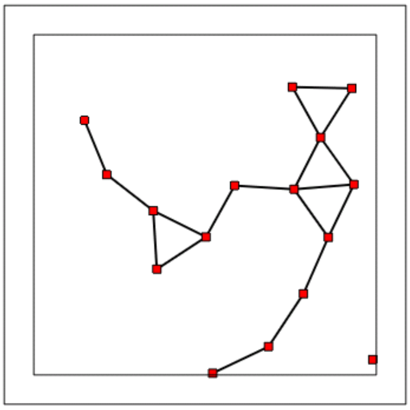


Figure 29: damping coefficient =0.5

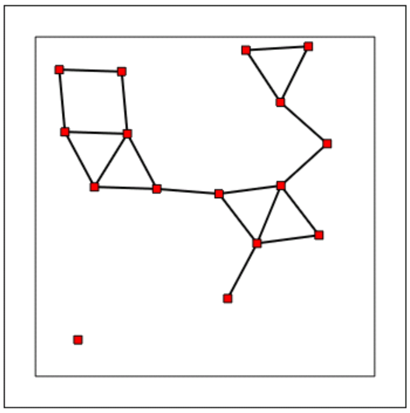


Figure 30: damping coefficient =0.9

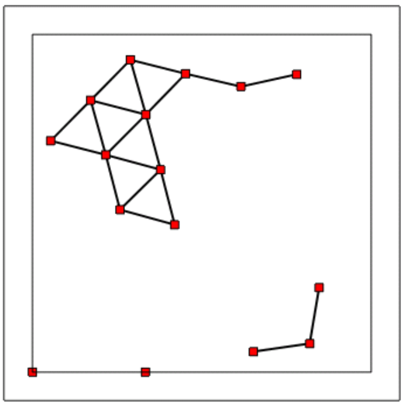


Figure 31: damping coefficient =0.99,

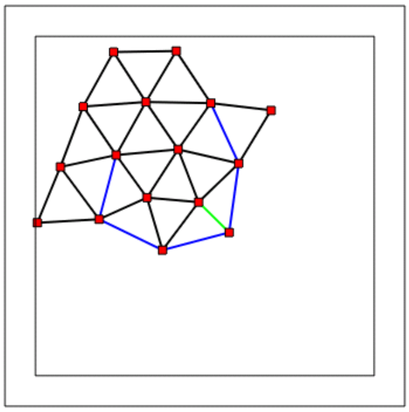


Figure32: damping coefficient =1 (no damping)

For low values of the damping coefficient (0.5 and 0.9), the particle system ceases to oscillate prior to the attainment of a tiling pattern, whereas for high values of the damping coefficient (0.99) more spring connections are established. For the case of no damping, a triangular tiling pattern is always reached, since the system's oscillation does not cease unless a stable configuration is reached. It should be noted that figures 29, 30, 31 depict a non-moving nodal configuration, whereas figure 32 is a snapshot of a moving one. Since it is the formation of tiling patterns rather than the minimization of the program's running time that this project is concerned with, it is proposed that no damping should be incorporated (damping coefficient set to unity) if a tiling pattern is to be reached in all cases (regardless of the drive parameter).

4.3 Activating/Deactivating the boundary constraint

As mentioned in §3.2.1, the boundaries within which the node particle system is placed can be deactivated by pressing the “f” key during the execution of the program. They can be reactivated by pressing the “b” key. Releasing the constraints imposed by the boundaries, results in the nodes escaping the screen, due to the repelling forces acting on the nodes, as can be seen in figures 33 and 34. Hence no tiling patterns can emerge.

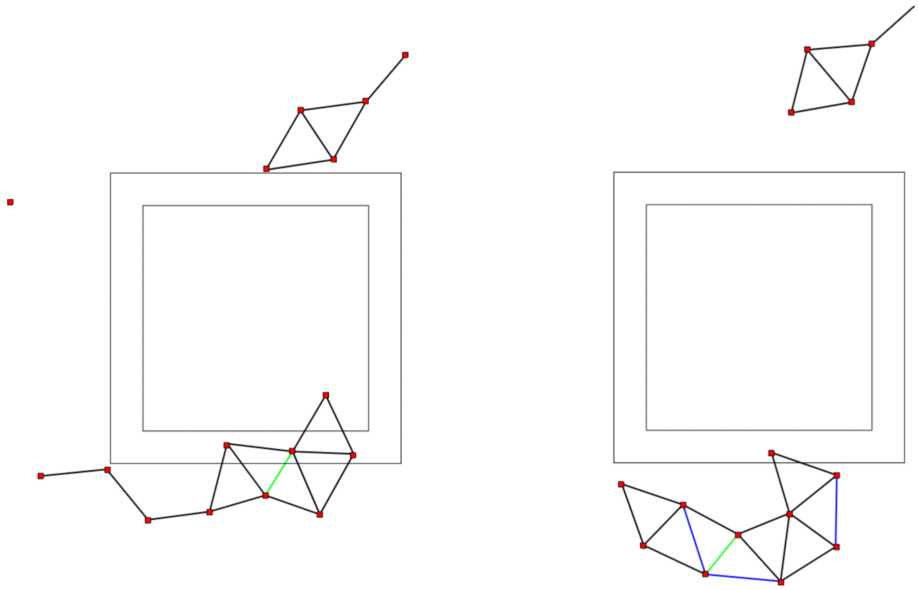


Figure 33 & 34: deactivation of boundary constraints and failure to generate tiling patterns

5 Conclusions and suggestions for future research

Having described the basic features of the code implemented within the current project, studied the affect of the various parameters and demonstrated what are believed to be interesting results, this Section summarises the conclusions and suggests potential improvements and/or extensions of the project.

The current project has utilised a particle spring system approach to derive Penrose tiling patterns. Specific lengths corresponding to the desired tiles' geometry have been specified and the code generates triangular tiling patterns with the desired lengths. However no Penrose tiling arises (except by chance) due to the fact that the combinations of the desired lengths constitute a total of 10 possible triangle out of which only 2 correspond to the rhombuses of the desired Penrose tiling. Furthermore, it is believed that no Penrose tiling pattern can emerge by enforcing a particle spring system approach, unless additional constraints are identified and enforced to the code, complementary to the geometric ones described in §3.2.5. The current code does produce triangular tiling patterns of specific proportions very efficiently, by using 10 different triangles as tiles. If two different lengths are specified, a total of 4 tiles can be generated.

The incorporation of the damping coefficient in the current problem greatly affects the feasibility of solution. The damping coefficient essentially reduces the effect of spring forces with increasing number of iterations; that should not pose a problem for a sufficiently large drive parameter with respect to the bounded region, but leads to an effectively premature termination of the algorithm when a small drive parameter is specified. Therefore it is concluded that preservation of the nodal velocity is a prerequisite if the desired nodal distances (tiling patterns) are to be obtained regardless of the number of nodes and/or the desired lengths within a given bounded region. On the other hand, if accuracy is not the main issue but a quick and effective way of connecting nodes is pursued (as was the case in the project of Kanellos, 2007), damping is a helpful but artificial stabilising technique to achieve that goal within a small number of iterations.

The definition of a bounded region is of vital significance for the successful application of the code. The absence of boundaries in conjunction with the application of repelling forces leads to failiure of the code; no spring connections connecting all nodes can be established. Moreover, the actual treatment of the boundaries by the code, i.e. whether the boundaries themselves can form a part of the tiling patterns, instead of just providing a framework for a tiling pattern is believed to depend on the depend on the geometry of the bounds and is a subject that could be investigated in the future. Furthermore, the tessellation within planar or non-planar curved boundaries or even non-orientable surfaces would also provide a challenging research project.

One further main future objective within the current algorithm is to determine the necessary and sufficient conditions under which a valid solution is obtained, in other words, to determine the sets of the main parameters (number of nodes, dimensions of the bounded rectangle, drive etc.) for which the problem is well defined and a solution is feasible. Furthermore, since there is some randomness associated with the initial location of the nodes a more probabilistic approach could be followed, according to which a probability for the feasibility of the solution could be assigned to any combination of initial parameter values, or, vice versa maximum permissible limiting values could be assigned to each parameter for a given probability of a feasible solution. Of course in this case the interaction between the various parameters has to be studied in more detail.

References

Abe, E., Tsai A. P., 2001, ‘Structure of quasicrystals studied by atomicresolution electron microscopy’, in JEOL News 36, pp. 18–21

Ammann R., Grunbaum B., Shephard G. C., 1992, ‘Aperiodic Tiles’, in Discrete & Computational Geometry, New York, Springer- Verlag

Bak, P., 1996, How nature works : the science of self-organized criticality, New York, Copernicus

Beer, R. D., 2004, ‘Autopoiesis and Cognition in the Game of Life’, in Artificial Life, Vol. 10, No. 3, pp. 309-326

Bentley P. J. & Kumar S. eds, 2003, On Growth, Form and Computers. London, Elsevier

Berger R., ‘The undecidability of the domino problem’, in Memoirs of the American Mathematical Society, no. 66 (1966), pp. 72

Deleuze, G., Guattari F. (eds) 2004, A thousand plateaus : capitalism and schizophrenia, London, Continuum

Flake, G. W., 1998, The computational beauty of nature : computer explorations of fractals, chaos, complex systems, and adaptation, Cambridge Mass., London, The MIT Press

Fraser J., 1995, An evolutionary Architecture, London, Architectural Association

Gardner M., 1977, ‘Mathematical games’, in Sci. Amer. 232, pp. 110–121.

Grünbaum, B., Shephard G. C., 1989, Tilings and patterns : an introduction, New York, Freeman

Grushko, B., Holland-Moritz, D., 1997, ‘Decagonal quasicrystals in Al—Co, Al—Ni and in their ternary alloys’, in Mater. Sci. Eng. A226, pp. 999–1003

Gummelt, P., 1996, Geometriae Dedicata, 62, 1-17Hwang I., Sakamoto T., Ferre A., Kubo M., Sadarangani N. & Ballesteros M., Prat R. eds, 2007, Verb Natures, Barcelona, Actar

Hibbitt, Karlsson & Sorensen, 2006, Inc. ABAQUS/Explicit User’s Manual, Volumes I-III and ABAQUS CAE Manual, Version 6.6, Pawtucket, USA

Ingalls, R., 1993, ‘Octagonal Quasi-Crystal Tilings’, in J. Non-Cryst. Solids 153, pp. 177–180

Ingalls, R., 1992, ‘Decagonal Quasi-Crystal Tilings’, in Acta Crystallogr. A48, pp. 533–541

Kanellos T., 2007, ‘Topological Self-Organisation: Using a particle-spring system simulation to generate structural space-filling lattices’, MSc thesis, Bartlett School of Graduate Studies, UCL, accessed September 2008, < http://eprints.ucl.ac.uk/4985/>

Kaufmann M., Wagner D., eds, 2001, Drawing graphs: methods and models, volume 2025 of Lecture Notes in Computer Science. Springer-Verlag

Kennedy J., Russel C. E., 2001, Swarm intelligence, San Francisco, London, Morgan Kaufman

Kilian A., Ochsendorf J., 2005, 'Particle-Spring Systems for structural form-finding', in Journal of the international association for shell and spatial structures: IASS, no. 148, pp. 77-84

Lazzari M., Majowiecki M., Vitaliani R. V., Saetta A. V., 2008, 'Nonlinear F.E. analysis of Montreal Olympic Stadium roof under natural loading conditions' in Engineering Structures, in press

Levine D., Steinhardt P. J., 1985, 'Quasiqrystals. I. Definition and Structure', in Physical Review B, vol 34, no.2, pp. 596- 616

Lindenmayer A., Jurgensen H., 1992, 'Grammars of Development: discrete-state models for growth, differentiation and gene expression in modular organisms', in Rozenberg G., Salomaa A.(eds) *Lindenmayer Systems*, Berlin Heidelberg, New York, Springer- Verlag

Maturana H., Varela J. F., 1980, Autopoiesis and Cognition, Dordrecht, Holland, Reidel

Maturana H., Varela J. F., 1992, The Tree of Knowledge: The Biological Roots of Human Understanding, Boston, Massachusetts, Shambhala Publications

Nowak, A., Szamrej, J., Latané, B., 1990, 'From private attitude to public opinion: A dynamic theory of social impact', in Psychological Review, 97, pp. 362–376

Otto F., Rasch B., Finding Form, Towards an Architecture of the Minimal, Axel Menges

Penrose R. 2000, The Large, the Small and the Human Mind, Cambridge, Cambridge University Press, Canto

Penrose R., 1974, 'The Role of Aesthetics in Pure and Applied Research' in Bull. Inst. Maths. Appl. 10 pp.266

Reas C., Fry B., 2007, Processing: A Programming Handbook for Visual Designers and Artists, Massachusetts, MIT Press

Robinson R. M., 1971, 'Undecidability and Nonperiodicity for Tilings of the Plane', in Inventiones math vol 12, pp. 177-209

Senechal M., 1990, Crystalline Symmetries: an informal mathematical introduction, Bristol, Philadelphia, New York, Adam Hilger

Steinhardt P. J., 2000, 'Penrose tilings, clusters models and the quasi-unit cell picture', in Materials Science and Engineering, pp. 205–210

Steurer W., Haibach T., Zhang B., Kek S., Luck R., 1993, The structure of decagonal $Al_{70}Ni_{15}Co_{15}$, in Acta Crystallogr. B 49, pp. 661-675

Steurer W., 2004, 'Twenty years of structure research on quasicrystals. Part I. Pentagonal, octagonal, decagonal and dodecagonal quasicrystals' in Z. Kristallogr. 219, pp. 391–446

Topping B. H., Khan A. I., 1994, ‘Parallel computation schemes for dynamic relaxation’, in Engineering Computations V. 11, I. 6, pp. 513-548

Wang H., 1961. ‘Proving theorems by pattern recognition- II’, in Bell System Technical Journal 40, pp. 1–42

Wieting, T. W., The mathematical theory of chromatic plane ornaments, New York, Dekker

Williams C., 2001, ‘The Analytical and Numerical Definition of the Geometry of the British Museum Great Court Roof’, <http://staff.bath.ac.uk/abscjkw/BritishMuseum/ChrisDeakin2001.pdf>

Web Documents

Hanna S. in <http://www.sean.hanna.net/bodyspaceframe.htm> (Sean Hanna on Gormley’s Body/Space/Frame), accessed August 2008

Williams C., 2001, ‘The Analytical and Numerical Definition of the Geometry of the British Museum Great Court Roof’, <http://staff.bath.ac.uk/abscjkw/BritishMuseum/ChrisDeakin2001.pdf>

Appendix

Pseudocode (Processing Programming Language)

A.1 Particle class

```
class Node
{
  int id;
  Vec pos;
  Vec dir;
  Vec tdir;
  Node(int i, float x, float y, float x_dir, float y_dir)
  {
    pos = new Vec(x,y);
    dir = new Vec(x_dir, y_dir);
    tdir = new Vec();
    id = i;
  }
  void move()
  {  pos = add(pos,dir);
  }
  void draw()
  {
    stroke(0);
    stroke(0,255);
    pushMatrix();
    strokeWeight(0.1);
    translate (pos.vec[0],pos.vec[1]);
    …
    popMatrix();
  }
}
```

A.2 Particle – spring system

```
int particles= 81;
float spring_thr = 1.3;// the edge of chaos or the beggining, too
tight clusters
(…)
Node [] nodes= new Node[particles];
TreeSet springs = new TreeSet();
Vector points = new Vector();
(…)
float d;  // distance between each pair of nodes
float side=250f; // square’s side
int connections=0; // connections between the particles when the
program starts
float drive; // the maximum distance; below that number new
connection is established

void setup()
{
  for (int i=0; i<nodes.length; i++)
  {
    nodes[i] = new Node (i,random(-side/2, side/2), random(-side/2,
side/2),random(-10f,10f),random(-10f,10f)); // nodes with random
initial position and random velocity
  }
}
void draw()
{
  for (int i=0; i<nodes.length; i++)
  {
    for (int j=i+1; j<nodes.length ; j++)
    {
      Spring a = new Spring (nodes[i],nodes[j]); // creating spring
connection
      if (!springs.contains(a)) {
        Vec nowi = add (nodes[i].pos,nodes[i].dir);
        Vec nowj = add (nodes[j].pos,nodes[j].dir);
```

```
        d = disto (nowi, nowj);
        if (d<=drive*spring_thr)
        {
          if  (!crossing(nodes[i],nodes[j]))  //  checking  if  the springs are crossing and if the answer is positive then avoid it
          {
            springs.add(a);
            connections++;
          }
        }
      }
    }
  }
  for (int i=0; i<nodes.length; i++)
  {
    nodes[i].dir = add (nodes[i].dir, nodes[i].tdir); // adding the existing applied forces with the new ones, coming from springs' application
    nodes[i].tdir = new Vec();
    nodes[i].move(); // adding velocity with each time- step and each iteration
    nodes[i].draw(); //drawing the nodes
    nodes[i].dir.scale(0.95);// dumping coefficient
  }
(…)
}
```

A.3 Spring class

```
class Spring implements Comparable
{
  Node node1; // node_1
  Node node1; // node_2, the pair of them is checked by the program in each iteration to create a spring
  float d; // the distance between the nodes
    Spring(Node nodex, Node nodey)
```

```
  {
    if (nodex.id < nodey.id) { // sorted nodes
      node1 = nodex;
      node2 = nodey; // every other node apart from the two
consisting the pair
    }
    else {
      node1 = nodey;
      node2 = nodex;
    }
  }
  int compareTo(Object o)
  (…)

  void go()
  {
    Vec now1 = add (node1.pos,node1.dir); // temporary node_1
position
    Vec now2 = add (node2.pos,node2.dir); // temporary node_2
position

    d = disto (now1,now2); // the temporary distance between node_1
and node_2
    Vec tdir_1 = sub(now1,now2); // new node_1's direction, parallel
to the spring and with a repulsing result
    Vec tdir_2 = sub(now2,now1); // new node_2's direction, parallel
to the spring and with a repulsing result

    (…)
    if (d<(0.61803 + 0.3892*f)*drive)
    {
      scalar = (drive-d)* 0.61803; //
    }
     if (d< (1 + 0.1755 * h)*drive && d >(0.61803 + 0.3892*f)*drive)
    {
     scalar =drive-d;
    }
      if(d>(1 + 0.1755 * h)*drive )
```

```
    {
      scalar = (drive-d)*1.1755;
    }
    scalar = scalar/10;
    tdir_1.scale(scalar); //
    tdir_2.scale(scalar);

    node1.tdir = add(node1.tdir,tdir_1);
    node2.tdir = add(node2.tdir,tdir_2);

    boolean compress=false;
(…)
    if (!visual )
    {
      line  (node1.pos.vec[0],  node1.pos.vec[1],  node2.pos.vec[0],
node2.pos.vec[1]); // drawing the spring line
    }
  }
}
```

A.4 Particles relation to boundaries

```
Node keep_in(Node nod)
{
  Vec new_pos = add (nod.pos,nod.dir);
  if (free) // by pressing "f" the nodes are without any borders
  {
    nod.dir=nod.dir;
  }
  if(rect)  //  by  pressing  "b"  the  nodes  are  going  back  within
borders
//the  following  code  regards  the  various  positions  outside  the
borders and the ways they lead the nodes back inside the rectangular
  {
boolean outside=false;
    if (nod.pos.vec[0]+nod.dir.vec[0] <= centre.vec[0]-side/2)
```

```
    {
      Vec                 bound                 =                 new
Vec(nod.pos.vec[0]+nod.dir.vec[0]+(side/2f),0);
      nod.dir = sub (nod.dir, bound);
    }
    if (nod.pos.vec[0]+nod.dir.vec[0]>=centre.vec[0]+side/2f)
    {
      Vec    bound    =    new    Vec(nod.pos.vec[0]+nod.dir.vec[0]-
(side/2f),0);
      nod.dir = sub (nod.dir, bound);
    }
    if (nod.pos.vec[1]+nod.dir.vec[1]<=centre.vec[1]-side/2f)
    {
      Vec bound = new Vec(0,nod.pos.vec[1]+nod.dir.vec[1]+(side/2));
      nod.dir = sub (nod.dir, bound);
    }
    if (nod.pos.vec[1]+nod.dir.vec[1]>=centre.vec[1]+side/2f)
    {
      Vec bound = new Vec(0,nod.pos.vec[1]+nod.dir.vec[1]-(side/2));
      nod.dir = sub (nod.dir,bound);
    }

  }
  return nod;
   }
```