Universidade de Aveiro Departamento de Física,
2016

Diamantino Castanheira da Silva

Evolução de variedades topológicas da perspectiva das redes complexas

Complex Network View of Evolving Manifolds

Universidade de Aveiro Departamento de Física,
2016

Diamantino Castanheira da Silva

Evolução de variedades topológicas da perspectiva das redes complexas

Complex Network View of Evolving Manifolds

Masters thesis to Aveiro University for the acquirement of the Masters degree in Physics, supervised by Dr. Sergey Dorogovtsev and Dr. Rui Costa, Researcher Coordinator at the Department of Physics of Aveiro University.

the jury

Prof. Dr. Manuel António dos Santos Barroso
Professor Auxiliar do Deparamento de Física da Universidade de Aveiro

Prof. Dr. Nuno Miguel Azevedo Machado de Araújo,
Professor Auxiliar da Faculdade de Ciências da Universidade de Lisboa
(examiner)

Dr. Sergey Dorogovtsev
Equiparado a Investigador Coordenador do Departamento de Física da Universidade de Aveiro
(advisor)

acknowledgements

Esta tese não teria sido possível sem o apoio do meu orientador, o professor Sergey Dorogovtsev e do meu co-orientador Rui Américo. Quero agradecer a todos os colegas do Departamento de Física e em especial aos meus pais, por me apoiarem nesta caminhada.

Abstract

Neste estudo investigamos redes complexas formadas por triangulações de variedades topológicas em evolução, localmente homeomórficas a um plano. O conjunto de transformações dessas redes é restringida pela condição de que a cada passo todas as faces se mantenham triangulares. Neste trabalho adotamos duas abordagens principais. Na primeira abordagem crescemos variedades usando várias regras simples, que progressivamente adicionam novos triângulos. Na outra abordagem relaxamos a estrutura de variedades grandes, mantendo o número de triângulos constante.

As redes resultantes da evolução destas triangulações demontram várias características interessantes e inesperadas em redes planares, tais como diâmetros ”small-world” e distribuições de grau tipo lei de potência.

Finalmente manipulámos a topologia das variedades pela introdução de ”wormholes”. A presença de ”wormholes” pode mudar a estrutura da rede significamente, dependendo da taxa a que são introduzidos. Se introduzirmos ”wormholes” a uma taxa constante, o diâmetro da rede apresenta um crescimento sub-logarítmico com o número de nodos do sistema.

We study complex networks formed by triangulations of evolving manifolds, locally homeomorphic to a plane. The set of possible transformations of these networks is restricted by the condition that at each step all the faces must be triangles. We employed two main approaches. In the first approach we grow the manifolds using various simples rules, which progressively had new triangles. In the other approach we relax the structure of large manifolds while keeping the number of triangles constant.

The networks resulting from these evolving triangulations demonstrate several interesting features, unexpected in planar networks, such as small-world diameters and power-law degree distributions.

Finally, we manipulate the topology of the manifolds by introducing wormholes. The presence of wormholes can change significantly the network structure, depending on the rate at which they are introduced. Remarkably, if we make wormholes at constant rate, the network’s diameter shows a sub-logarithmic growth with the number of nodes in the system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and structure

Network theory has been successful in describing a series of phenomena in various systems such as social, biological or technological systems. Important models have been developed in the last decades, such as the Erdős-Rényi model [1] in the late 1950s, the Watts-Strogatz model [2] and the Barabasi-Albert (BA) model [3] [4] in the late nineties, which were fundamental to understand various aspects of complex networks.

However, in recent years there has been a new interest about the geometrical characterization of networks, which these important models did not address. This is an important aspect for various real networks such as transportation networks, communication networks or even the brain, which structure depends on the space in which it is embedded.[5]

This new perspective about networks has been used to study routing problems in the Internet [6, 7, 8], data mining and community detection [9, 10, 11]. The study of properties such as curvature and hiperbolicity can be used to determine metrics of networks embedded in a physical space [12, 5], with technological applications such as wireless networks [13].

Even in the field of quantum gravity, there are studies using pre-geometric models, were space is an emergent property of a network of a simplicial complex [14, 15, 16, 17] , which are structures formed by gluing together simplices such as triangles, tetrahedra, etc.

The objective of this work is the study of 2D evolving manifolds, using methods from network theory. The evolution of these manifolds will be made by a set of rules that obey various topological constraints, in order to observe several characteristics, such as degree distribution and average distance of the embedded network.

This document starts by giving a basic introduction to complex network theory and topology (chapter 2). In chapter 3, we present our approach to this problem and introduce the rules used to evolve the manifolds. In chapter 4, we present the results of the simulated networks, discussed them and derive some analytical results. In chapter 5, we extend the model in order to observe the evolution of manifolds by changing their topological properties, consisting on the introduction of wormholes. The last chapter, chapter 6, is a reference to all the methods and characteristics used in this work.

1.2 Related work

The theme of space and networks has been approached from various perspectives by many authors. They explored several aspects, ranging from the discovery of embedded metrics in communication networks [8], to the routing problem in several types of network [7] or even the emergence of geometric and statistical characteristics using networks randomly built from merging simplexes [1] [18] [19].
The following works [18] [19].
done by Bianconi et al. were a starting point to our present work.

In one of these works [18], the network model was constructed by adding simplexes of dimension d to the network, building a simplicial complex [2]. δ-faces ($\delta < d$) were defined as the simplexes of dimension δ belonging to each d-simplex, and the generalized degree of a δ-face was defined as the number of d-simplexes incident to it.
Each (d-1)-face of a d-simplex can be shared by 2 d-simplexes at most, so each new d-simplex can only connect to another simplex if it has some (d-1) face not belonging to 2 d-simplexes. Every vertex of each d-simplex is generated with a random energy value. When a new simplex is introduced, it will connect to a random (n-1)-face, with probability proportional to the energy [3] of that face (see preferencial attachment at section 2.3).
The simulation results showed that the average of the generalized degree of the δ-faces forming the simplicial complex can either follow Fermi-Dirac, Boltzmann or Bose-Einstein distributions, depending on the dimension of the δ-faces and of the d-simplexes.

A more simple model [19], was also developed by Bianconi et al., focused in the emergence of curvature in the construction of surfaces based in triangles. Here curvature (R_i) is related to the number of adjacent triangles (t_i) to each vertex i, as $R_i = 1 - t_i/6$. Although curvature is not a priority in our work, the method developed in this work to evolve a surface is quite interesting.
This model has two parameters m, which is the maximum number of triangles connected to an edge, and p, which is a continuous control parameter.

To build such surfaces, they started with just a triangle and then at successive steps used a combination of two processes. The main process is executed in every step, adding a triangle to an random edge which has less than m triangles attached to it. An additional process is executed with probability p, consisting in the connection of vertices at distance 2 from each other. This process will only choose two vertices such that each of the two edges between them have less then m adjacent triangles.
By using different combinations of the two parameters, it was possible to generate ”complex network geometries with non-trivial distribution of curvatures, combining exponential growth and small-world properties”, such as a very slow growth of distance between vertices relative to the number of vertices (see section 2.3).

[1] Simplexes of n dimensions are full connected networks of $n + 1$ vertices.

[2] A simplicial complex is a set of simplexes in which any face of these simplexes also belongs to the simplicial complex. A face of a simplex is a simplex of lower dimensionality, belonging to it.

[3] Here energy of a δ-face is defined as the sum of the energies of all it’s vertices.

Chapter 2

Basic notions

2.1 Network basics

A network, G, also known as a graph, can be mathematically described as a collection of vertices $G(N)$ and edges $G(E)$. Each edge is simply the connection between two vertices of the network.

A network can be fully represented by its adjacency matrix a_{ij}, in which a_{ij} will be equal to the number of edges from i to j. Normally a_{ij} will have values 0 (disconnected) or 1 (connected by a single edge). A graph can be directed or undirected. A directed graph can have edges that only permit transitions from a vertex v_i to v_j but not from v_j to v_i ($a_{ij} \neq a_{ji}$). When both directions are permitted for all edges, a graph is called undirected ($a_{ij} = a_{ji}$).

In this work, we will only use undirected networks, therefore, the degree of a vertex will be equal to the number of links incident to it. Using the adjacency matrix, the node's degree calculation is simply:

$$k_i = \sum_{j=1}^{N} a_{ij} \tag{2.1}$$

in which N is the number of vertices of the network.

2.2 Ensemble approach

The approach we are using in this work will focus on *random networks* which are not single networks but a statistical ensemble [20]. The *statistical ensemble* is defined as a set of particular networks in which each one of them has a given probability of realization. In this work, each ensemble will be formed by networks grown by a set a rules, that begin with the same initial conditions and have the same number of evolution steps. A clearer definition of this process will be discussed in chapter 3.

In order to obtain statistics of a certain rule, we will not use the properties of single realization, but the expected characteristics of the possible realizations of this type network, taking into account the realization probability. To illustrate this approach, let us use a small random network composed of exactly 3 nodes. Each possible link can exist independently with probability p, so each member of this ensemble will have an existence probability as depicted in figure 2.1.

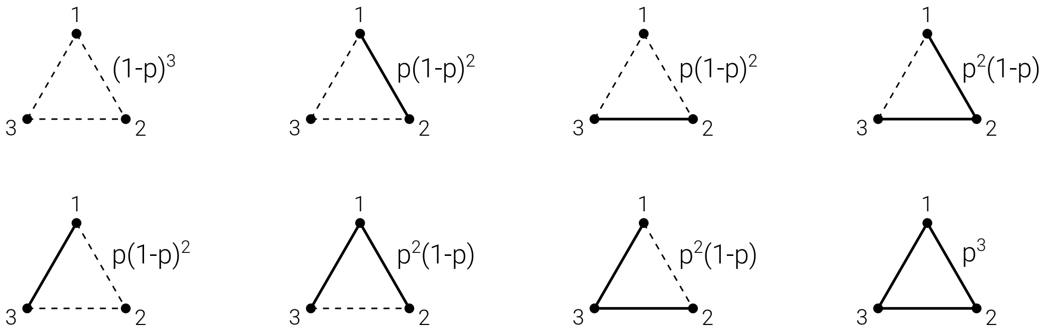


Figure 2.1: Ensemble example.
The probability of each member of the ensemble is shown near it.

The degree distribution $P(k)$ in a network will be equal to the expected fraction of vertices with degree equal to k. It is defined as:

$$P(k) = \frac{\langle N(k) \rangle}{N} \tag{2.2}$$

in which $\langle N(k) \rangle$ is the average number of vertices with degree k, averaged over all the members of the ensemble and N is the number of vertexes. Therefore, the nth moment of the degree distribution, can be calculated by:

$$\langle k^n \rangle = \sum_k P(k) k^n \tag{2.3}$$

A path between vertex i and vertex j is a sequence of vertices beginning in i and ending in j, such that each vertex is connected by an edge to the next vertex. The length of a path is equal to the number of edges in it.
The distance between two nodes i and j, l_{ij}, is the minimum length path from vertex i to vertex j. This path is called a geodesic. The average distance in a specific network s is the mean distance between all possible pairs of distinct vertices:

$$\langle l \rangle_s = \frac{2}{N(N-1)} \sum_i^N \sum_{j>i}^N l_{ij} \tag{2.4}$$

Therefore, the average distance in a ensemble, in which a network s has as weight $P(s)$ is:

$$\langle l \rangle = \sum_s P(s) \langle l \rangle_s \tag{2.5}$$

2.3 Reference network models

To give a better idea of the meaning of the measured quantities, here we present an overview of some basic models and that will serve as a basis of comparison when characterizing our networks.

The classical random network is built from N initial nodes. For each distinct pair of nodes, there is a probability p of creating a link between them.

The small world network is a kind of network that has some specific characteristics related to average distance. To be small-world, the average distance between vertices should grow logarithmically with the system size.
One of the most simple models that manifest this kind of characteristics is the Watts-Strogatz model [2], which starts with a lattice structure forming a ring and with the addition of *shortcuts*, the network mean distance falls dramatically. In fact, this modification makes the mean distance growth with system size to go from linear to logarithimic.

A very important mechanism for growing networks is the preferential attachment mechanism. One of the first models to demonstrate this mechanism was the Barabasi-Albert model [3, 4], in which a new vertex introduced in the network, connecting to m random vertices, given a weight function proportional to its degree. Basically, that mechanism tends to increase the number of connections of the high degree nodes. In fact, the BA-model will converge to a heavy-tail degree distribution $P(k) \sim k^{-3}$.

In terms of diameter growth, we can select two networks with very distintic behaviour, lattices and classical random networs. The diameter of an lattice grows with $\sim N^{1/d}$, in which N is the number of vertexes and d is the number of dimensions. In contrast to this growth, the classical random graph grows with $\sim \log(N)$, which is a slower growth than any lattice.

2.4 Simplexes and Manifolds

A basic building block of topology is the n-simplex. A 0-simplex is a point, a 1-simplex is a line, a 2-simplex is a triangle and a 3-simplex is a tetrahedron.
In order to construct more complicated spaces, we can merge faces of various simplexes, obtaining a *simplicial complex*. As an example, the surface of a cube can be considered as twelve triangles glued together.
Manifolds are a class of spaces of topology, that near each of its points resembles a n-dimensional euclidian space. These manifolds can be resultant, for example, of polynomial or differential equations.
In order to study its global structure, it is helpful to triangulate it, that is, to construct a homeomorphism to a simplicial complex. For example, the surface of a sphere is a 2D manifold, homeomorphic [1] to a cube. [21]

2.5 Object of work

The manifolds that interest us resemble a 2D euclidian space near each of its points. Some examples of this type of manifolds are a spheres or a torus.
Using triangles, which are 2-simplexes, we can build a simplicial complex that is homeomorphic

[1] Homeomorphism is an isomorphism between two topological spaces, in which every point of one topological space can be continuously mapped in the other topological space, in either direction.

to those manifolds. Each edge will be shared between 2 triangles and each triangle will have 3 distinct neighbouring triangles. The resulting simplicial complex will be a *surface*, from which the nodes and links of the network will be naturally its vertices and edges. We will consider that all edges are undirected.

This object must obey some constraints. Each edge must have only 2 neighbouring triangles and the Euler characteristic (see section 2.6) must be constant if the *topology* [2] does not change.

As we will see later (chapter 3) the construction of this object will depend on the chosen rule. It could start from a minimal surface and then be grown or starting with a large surface and then be relaxed.

In contrast with Bianconi's model [19], in which only edges that were not saturated could evolve, that would lead to the creation of a frozen nucleus and only at the border would be evolution. In our model, every edge, vertex or face is available to evolve.

2.6 Relations between simplexes

Given the properties of our object of work, we can get some basic quantitative relations between the simplexes of the surface. One of the most fundamental equations, that all networks grown in this work will follow, is Euler's formula:

$$\chi = F + N - E \tag{2.6}$$

In which F is the number of faces, N is the number of vertices, E is the number of edges and χ is the Euler characteristic. χ gives us the number of wormholes [3] , H, this surface has, using the relation $\chi = 2(1-H)$. A sphere has $H = 0$, a torus has $H = 1$ and so on. In closed surfaces, the topology is directly connect to χ, therefore, if the topology of a surface does not change, χ must be constant. Also, when a rule is not changing any number of simplexes of the surface, obviously χ will not change.

Let us obtain further relations between the simplexes, to see what limitations we have in the simplexe's evolution. Knowing that each edge has 2 neighbouring triangles and that each triangle has 3 neighbouring edges, we get the relation $3F = 2E$. Therefore, the relation between the number of edges and vertices is given by:

$$N = \frac{1}{3}E + \chi \Leftrightarrow E = 3(N - \chi) \Leftrightarrow F = 2(N - \chi) \tag{2.7}$$

Given the obtained relations between the simplexes in equation 2.7, if we add a single vertex, we need to add 3 edges and 2 faces (triangles). But if we try to add only 1 edge or 1 face, that will create a fraction of a vertex, what is impossible. So, removing or adding a vertex at each step is the minimal evolution we can do.

[2] We will consider a strict meaning for topology, in which two topological surfaces are equal if they have the same number of wormholes.

[3] See the operative explanation of wormhole in chapter 5.

Chapter 3

Rules

The evolution of the networks that we will study are clearly dependent on the evolution of the surfaces, which will be made in two different approaches, by growing or relaxing these surfaces using very simple rules.

The growth of surfaces starts from a small initial surface, with enough nodes for the rule to operate, such a tetrahedron or a octahedron. At each step, the rule introduces 1 new node, 2 new triangles and 3 new edges to the surface. The quantity of introduced simplexes will be the same for any rule, but the way in which these elements are connected to the remaining surface will be what distinguishes each rule.

On the other hand, the rules for relaxation will not change the number of nodes, triangles or edges. The initial surface that they start with must have a considerable number of vertexes, and these rules will just change the structure of this network.

To name these rules, it will be used the letters G (growing) and R (relaxing) followed by a reference number. The list of all used rules can be consulted at the end of this chapter (section 3.7).

3.1 Common definitions and procedures

Before exploring the rules used in this work, some definitions and procedures related to the evolution of a surface must be explained. Many of them will use random choice of elements, and unless otherwise specified, we will assume that this choice is made uniformly.

Neighbours order around a vertex

The set of triangles adjacent to a vertex has a border forming a ring, connecting all the neighbouring vertices of that vertex. Therefore, if we define a orientation around that ring, we can define a range of neighbours from a neighbour X to neighbour Y.

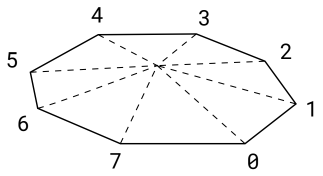


Figure 3.1: Order of neighbours around a vertex.

Expansion of a vertex into an edge:
One of the basic operations made to the surface consists in the ”expansion” of a vertex into an edge.
To expand a vertex A, and the two edges that connect to its neighbours B and C, we proceed as follows:
1) Create two new vertices, A1 and A2, and connect them.
2) Connect A1 to the range of neighbours of A from B to C, included.
3) Connect A2 to the range of neighbours of A from C to B, included.
4) Eliminate A.

Contraction of an edge into a vertex:
Another operation consists in the elimination (contraction) of an edge into a vertex, eliminating one vertex.
Let us call vertex A1 and A2 to the ends of a chosen edge, and vertex B and C to the two common neighbors of A1 and A2. The contraction procedure of an edge into a vertex follows the following steps:
1) Eliminate edges A1-A2, B-A2 and C-A2.
2) Connect all remaining neighbors of vertex A2 to vertex A1.
3) Eliminate vertex A2.

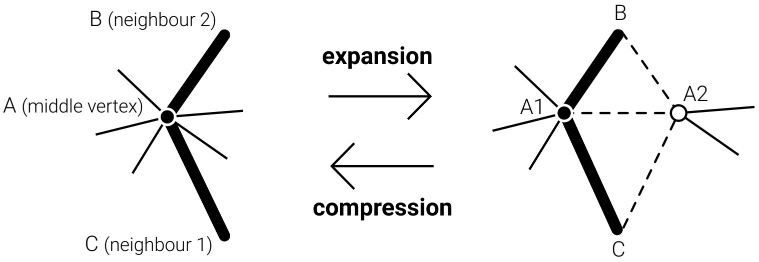


Figure 3.2: Expansion and contraction operations.
The thick lines and full vertices represent the selected elements.
The dashed lines and empty vertices represent the new elements.

Flip
Given a chosen edge, we will have its two end vertices, A and B and two common neighbours to A and B, let us call them C and D. To make the flip operation, we simply eliminate the edge between A and B and create a new one between C and D.

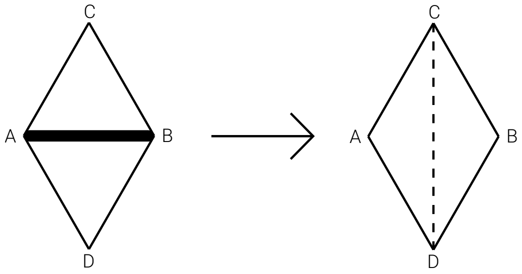


Figure 3.3: Flip operation diagram.

3.2 Rules for growth

In order to grow a network, we just need to insert a new vertex. A simple way to do that consists in splitting simplexes of the surface. So we started by designing rule G1, that will divide triangles and rule G2 that will divide edges.

Rule G1
This rule will simply introduce a new vertex, by subdividing a triangle in 3 triangles at each step. Schematically, we can represent the evolution steps like this:

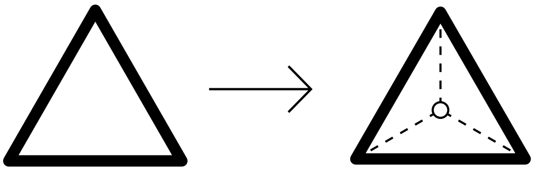


Figure 3.4: Rule G1 diagram.

Rule G2
This rule will introduce a new vertex by dividing an edge into two. Given the constraints of our model, all the faces must be triangles, then the two adjacent triangles to the original edge are also divided.

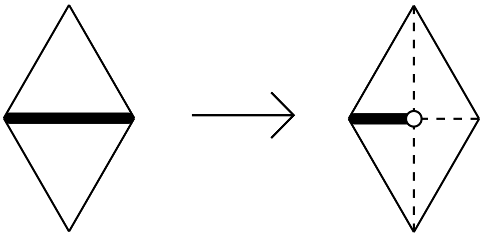


Figure 3.5: Rule G2 diagram.

Rule G3

The main idea of this rule is to choose one sequence of three vertices such that they form a connected chain, and then expand the middle vertex, using the expansion mechanism (see the initial part of this chapter and figure 3.2). To do that, a random vertex is chosen an then two neighbours are also chosen at random. This rule make its choices in the most simple and uniform way as possible. No preferential functions are used, just the simple selection of simplexes.

Rule G4
This rule is almost equal to the last one (G3), but it will make a flip after the expansion of the vertex, adding some structural destruction. But not without a price.
There are some limitations in the implementation of this algorithm. When choosing a second neighbour, this cannot be next to the first one. This is so, because they share an edge, and the flipping would create a new edge between them, something that is forbidden in our model. Therefore, using nodes of degree 3 as the middle vertex is forbidden. Because of this limitations, the most simple and uniform initial surface had to be an octahedron, instead of a tetrahedron.

Rule G5
This rule also uses the expansion mechanism, like in rule G3, but makes its choices based on edges, not on vertices. We start by choosing an edge at random, the next edge will be chosen from the set of edges connect to that first edge.

Rule G6
This rule is similar to rule G5, but it simplifies the choice procedure of the new edge. Instead of building a set of all edges connected to the first chosen edge, it just choose one of it's ends, and then chooses an edge connected to that end.

3.3 Variations of rule G3

Rule G3 follows a very simple procedure, having no functions of preferential choice, simply choosing simplexes uniformly at random.
During the implementation of the algorithm that simulated rule G3, we asked ourselves ”what if we change the probabilities of choosing the second neighbour next to the first neighbour?”. So as a first experiment, we reduced it to half of any other neighbour choice. As a consequence of this, the Hausdorff dimension decreased. So, we learned a way of manipulating the dimensionality of this network, by manipulating the way that the neighbours are chosen. Our observations of section 4.4, suggest that the closer the two chosen neighbours are, the higher will be the dimension. With this insight, we then created rules based in rule G3, but with different choice mechanisms. These new rules will give us information on the impact of these new choices in the characteristics of the network, such as degree distribution, average distance and dimensionality.

Rule G3.1
Inspired by the previous experience, we created a new rule in which we forced the choice of the two neighbours, the most far apart from each other.

With this in mind, we start by choosing the first neighbour at random. Using the ordering of neighbours around the middle vertex (see section 3.1), and assuming that the first neighbour is in position i, then the choice of the second neighbour will be in position $i \pm k/2$, if k is even, or we choose between $i \pm (k+1)/2$ or $i \pm (k-1)/2$ if k is odd.

Rule G3.2
In the reverse approach we can try to get the highest dimension possible by choosing the nearest two neighbours.
This rule clearly relates to rule G1, where the two neighbours are chosen in the same way. The difference lies in the middle vertex choice. In the case of algorithm G3.2, the vertex is chosen uniformly, but in the case of algorithm G1, the choice of vertex is proportional to the vertex degree.

Rule G3.3
To test if choosing high degree nodes for the middle vertex, leads to a lower dimensionality, we created this algorithm. The first choice (global) will be an triangle at random. Then the highest degree vertex is chosen as the vertex to expand. The two neighbours are chosen at random.

Rule G3.4
This rule is a variation of the previous rule (G3.3), but instead of a triangle as the first element to be chosen, it will be an edge. The choice for middle vertex, by highest degree rule will be maintained.

Rule G3.5
To simplify the process of choice, this rule simply chooses an edge at random and then chooses one of it's ends at random. The two neighbours of the middle vertex are also chosen at random.
This rule will create a preferential choice for the middle vertex.

Rule G3.6
This rule is almost the same as rule G3.5, but the vertex that will be expanded is the highest degree end of the chosen edge.

3.4 Rules for relaxation

The other set of rules, the rules for relaxation, begin with an already grown network. The main idea of these rules is to modify the network's structure, keeping the number of simplexes constant, until it reaches an equilibrium state, where the network's characteristics, such as the average degree distribution, stop evolving. The initial network can be generated by a growing algorithm, or it can use a predefined initial surface (see section 6.8).
In fact, the kind of initial surface is almost irrelevant [20] , assuming that every simplex configuration for the same number of vertices is reachable from any other configuration, after many iterations the resulting surface should be independent of the initial one. This holds true for all the rules for relaxation, except for rule R1.

Rule R1
This rule simply moves an vertex of degree 3 from one triangle to another triangle. To do this,

this rule combines rule G1 and another rule that will do the reverse of G1, by removing a node of degree 3. It is obvious that this rule has several limitations. It only works if nodes of degree 3 exist, so the initial surface must be generated by an algorithm that guarantees those kind of vertices, such as a surface evolved by rule G1. There is also the risk of having 4-cores [1] that will not relax. There is also the risk of having so little degree 3 nodes, that they only be relocated from one triangle to another, not changing the structure very much, or taking a huge number of steps to reach equilibrium.

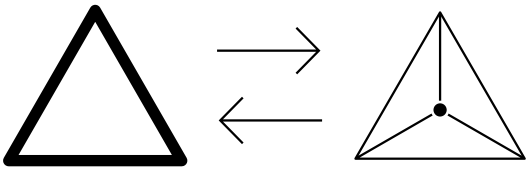


Figure 3.6: Rule R1 diagram.

Rule R2
This simple rule is based in the flipping mechanism presented in section 3.1. At each step, it will flip a random edge from the surface (fig 3.3). It's main purpose is to shuffle the possible correlations between neighbouring vertices.

Rule R3
This rule combines the contraction mechanism and an expansion rule. The expansion rule introduces a vertex and the contraction rule removes one vertex. Therefore, the number of simplexes is maintained. In this case, rule G3 was chosen as the expansion rule.

Rule R4
This last rule is just an variation of R3, where rule G4 was used in the expansion, instead of rule G3.

3.5 Reduction to fundamental rules

All the rules that were presented, can be expressed in terms of more fundamental rules. These fundamental rules are based in the ”expansion” and ”contraction” operations. By using them, we can emulate any other rule, being only a matter of finding the right sequence of operations. Some rules will be recreated, to demonstrate this reduction.

Reduction of rule G1
In order to get the same evolution as rule G1, the choice of the three vertices must be made such that they all belong to the same triangle. To do that, we can starting by choosing a triangle at random, like the original rule, but then we choose one of the vertices at random. The two neighbours obviously will be the two remaining vertices of the chosen triangle.

[1] k-core is a subgraph of a network in which every member has at least k neighbours also belonging in the k-core

This procedure will increase the degree of the middle vertex and the two neighbours, and also create a new vertex of degree 3 connected to the all the vertices of the triangle.

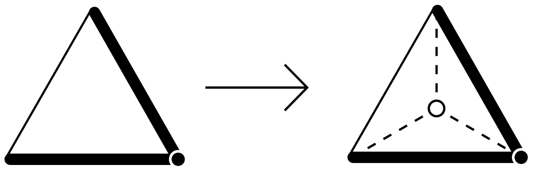


Figure 3.7: Algorithm G1 emulated by the expansion operation

Reduction of rule G2

To recreate this rule, we start by choosing an edge uniformly at random, just like in the original procedure (section 3.2). Then, one of its ends is chosen at random, as the middle vertex. The 2 neighbours become the 2 common vertices of the chosen edge. With this choice, we just expand the middle vertex.

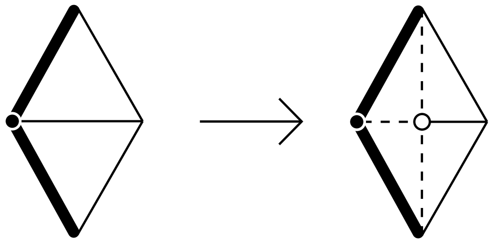


Figure 3.8: Algorithm G2 emulated by the expansion operation

Reduction of rule R1

The rules of relaxation do not change the number of simplexes in the surface. Therefore, we will need to combine the contraction mechanism with the expansion mechanism. In section (section 3.4) we saw that R1 consists of two phases, one in which a degree 3 vertex is eliminated and a second phase where a degree 3 vertex is inserted, like in rule G1. The reduction of the first phase starts with a choice of a vertex of degree 3. Then we apply the contraction mechanism using as middle vertex, the chosen vertex, and the two neighbours, any of the vertex's neighbours.

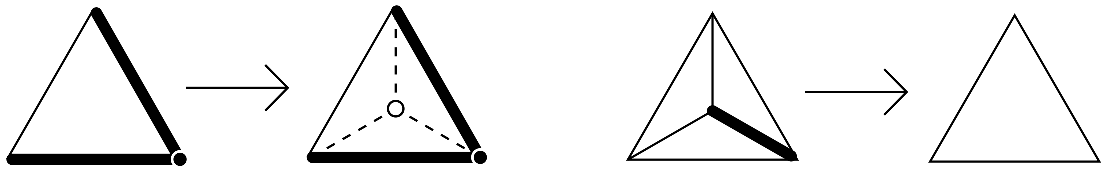


Figure 3.9: Algorithm R1 emulated by the expansion and contraction operations. The final stage of the expansion and the initial stage of the contraction could not correspond to the same elements.

Reduction of rule R2
This rule can be also reduced into two stages, a ”expansion” stage and a ”contraction” stage. The first stage is equal to rule G2, where we end up with an extra vertex. In the second stage, the idea is to merge this new vertex with one of two vertices that were initially disconnected, and the flipping is complete.

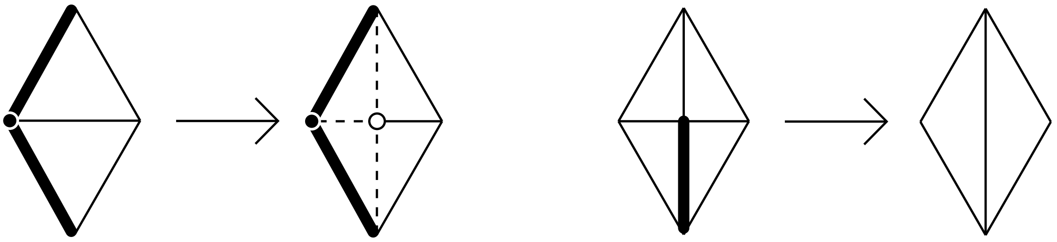


Figure 3.10: Algorithm R2 emulated by the expansion and contraction operations. The final stage of the expansion and the initial stage of the contraction must correspond to the same elements.

3.6 n-dimensional rules

Rules G1 and G2 can be generalized to higher dimensions. Let us suppose that we build a manifold made of n-simplexes. Each n-simplex has $n+1$ faces ($n-1$-simplexes), that can be at most shared by two n-simplexes. The initial geometry is a $(n+1)$-simplex, in which each n-simplex has $n+1$ neighbouring n-simplexes.

In this setting, G1 can be generalized to an operation that starts by choosing one of the n-simplexes and then connecting a new vertex to all the $n+1$ vertexes of this chosen n-simplex.

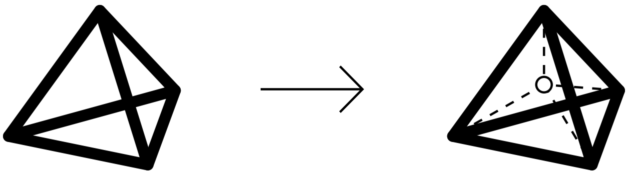


Figure 3.11: Rule G1 ($n = 3$) diagram.

The generalized version of G2 could be thought as an operation that chooses a random face ($(n-1)$-simplex) and divides each of the two n-simplexes that share this face into n n-simplexes. In general this consists in connecting a new vertex to all the vertices of the two n-simplexes, but for 2-simplexes, we must first eliminate the chosen face (a link).

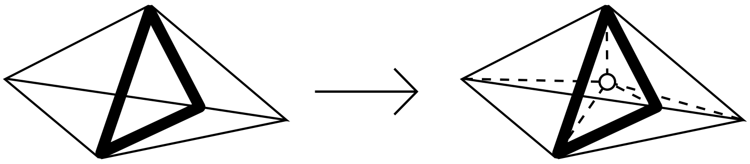


Figure 3.12: Rule G2 ($n = 3$) diagram.

3.7 Rules table

The following tables give a precise description of each rule. The diagrams show selected elements as thick lines and full points, and new geometry as dashed lines and empty points. If a diagram has numbers, they represent the order of selection.

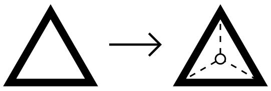
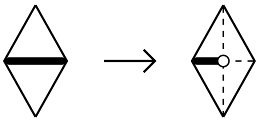
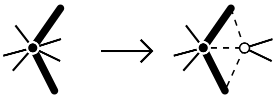
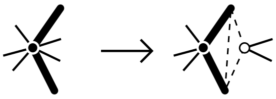
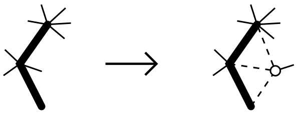
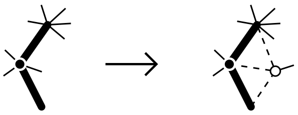
Code	Steps	Diagram
G1	1) Choose one of the triangles uniformly at random; 2) A new vertex is introduced and connected to each vertex of the chosen triangle.	
G2	1) Choose a random edge uniformly at random; (This choice will implicity choose the edge's two adjacent triangles.) 2) Remove the chosen edge; 3) Add a new vertex and connect it to the four vertexes of the two chosen triangles.	
G3	1) Choose a vertex at random; 2) Choose one of the vertex's neighbouring edges at random; 3) Choose another neighboring edge at random, excluding the edge chosen in 2); 4) Expand the middle vertex into an edge.	
G4	1) Proceed as in rule G3; 2) Flip the new edge.	
G5	1) Choose a random edge of the network; 2) From the set of edges that are connected to this chosen edge, choose another edge at random; 3) Expand the common vertex between the two chosen edges into an edge.	
G6	1) Choose a random edge of the network; 2) Choose one of it's ends at random; 3) Choose uniformly at random one of end's edges, excluding the edge chosen in 1); 4) Expand the vertex chosen in 2) into an edge.	

Table 3.1: Rules for growth description.

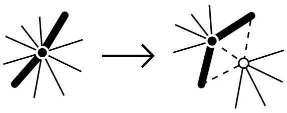
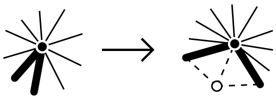
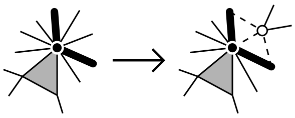
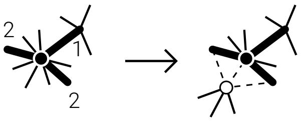
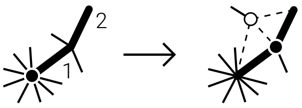
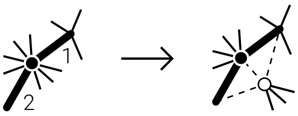
Code	Steps	Diagram
G3.1	1) Choose a random vertex; 2) Choose a random edge that is connected to that vertex; 3) The 2nd neighbouring edge will be the opposite edge from the 1st. If the degree of the node is odd, then with equal probability, one of the two ”oposite” edges will selected.	
G3.2	1) Choose a random vertex; 2) Choose a random edge that is connected to that vertex; 3) The 2nd neighbouring edge will be chosen of one of the two nearest edges to the edge chosen in 2), using the order around the vertex chosen in 1).	
G3.3	1) Choose a random triangle of the network uniformly; 2) Choose the highest degree node from the triangle. If more than one node has the maximum degree, choose one of those vertices at random; 3) From the set of edges neighboring the vertex, choose two edges uniformly at random; 4) Expand the chosen vertex in step 2) into an edge.	
G3.4	1) Choose a random edge of the network uniformly; 2) Choose the edge’s highest degree end. If they have the same degree, choose one of them at random; 3) From the set of edges neighboring the chosen vertex, choose two edges uniformly at random; 4) Expand the chosen vertex in step 2) into an edge.	
G3.5	1) Choose a random edge of the network uniformly. 2) Choose on of the edge’s ends uniformly at random. 3) Given the order of the edges around the vertex, if the node has a even degree, choose the opposite edge relative to the edge chosen in step 1). If the node has an odd degree, choose at random between the two farther edges relative to the first chosen edge. 4) Expand the chosen vertex in step 2) into an edge.	
G3.6	1) Choose a random edge of the network uniformly; 2) Choose the edge’s highest degree end. If they have the same degree, choose one of them at random; 3) Given the order of the edges around the vertex, if the node has a even degree, choose the opposite edge relative to the edge chosen in step 1). If the node has an odd degree, choose at random between the two farther edges relative to the first chosen edge; 4) Expand the chosen vertex in step 2) into an edge.	

Table 3.2: Rule G3 variations description.

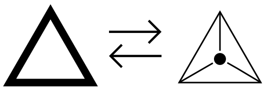
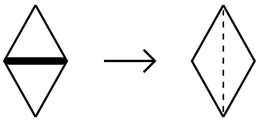
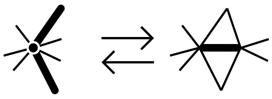
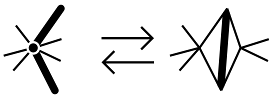
Code	Steps	Diagram
R1	1) Choose a vertex with degree 3, uniformly at random; 2) Remove it; 3) Choose a triangle at random; 4) Add a new vertex and connect it to the 3 vertices of the chosen triangle.	
R2	1) Choose an edge uniformly at random. Let's call this edge end vertices, A and B, and the common neighbours of A and B, C and D; 2) Remove the chosen edge; 3) Add an edge between C and D.	
R3	1) Choose a random edge; 2) Compress this edge into a vertex; 3) Use rule G3 to expand a vertex into an edge.	
R4	1) Choose a random edge; 2) Compress this edge into a vertex; 3) Use rule G4 to expand a vertex into an edge.	

Table 3.3: Rules for relaxation description.

Chapter 4

Results

4.1 Initial results

Using the relations obtained in 2.6, we can calculate some network characteristics. To calculate the average degree, we can use $\langle k\rangle = 2E/N$ [22]:

$$\langle k\rangle = 6\frac{N-\chi}{N} \tag{4.1}$$

Given the constraints of our surface, only in the case were the topology is the same as a torus ($\chi = 0$), the mean degree is exactly $\langle k\rangle = 6$. In all other cases, this value is only achieved asymptotically for very large networks.

The clustering coefficient measures the probability that two neighbours of a vertex are also neighbours between themselves. There are two different clustering coefficients. We have the network clustering coefficient [20] given by:

$$C = 3\,\frac{\text{number of loops of length 3 in a network}}{\text{number of connected triples of nodes}} \tag{4.2}$$

In the case of our networks, we will have:

$$C = 3\,\frac{F}{\sum_i k_i(k_i-1)/2} = 12\,\frac{N-\chi}{N(\langle k^2\rangle - \langle k\rangle)} = 2\,\frac{\langle k\rangle}{\langle k^2\rangle - \langle k\rangle} \tag{4.3}$$

We see that for networks with diverging second moment, $C = 0$.

There is another clustering coefficient, the average local clustering coefficient. To calculate it, we obtain the local clustering for each vertex, that is given by:

$$C_j = \frac{t_j}{k_j(k_j-1)/2} \tag{4.4}$$

where t_j is the number of connections between the neighbours of vertex j.

In our networks, the number of connections between the neighbours of a given vertex is equal to the vertex's degree. We can calculate the average local clustering coefficient:

$$\overline{C} = \frac{1}{N}\sum_j C_j = \sum_k P(k)\frac{k}{k(k-1)/2} = \sum_k P(k)\frac{2}{k-1} \tag{4.5}$$

For an perfect delta-function distribution with mean degree equal to 6, (constant degree), which is the case of a 2D triangular regular mesh or a torus, the average local clustering is $\overline{C} = 2/5$

4.2 Experimental method

In order to observe the evolving characteristics of the network grown or relaxed by each rule, we made a series of computer simulations. Given the ensemble approach we are using, (section 2.2), in order to measure some quantity we would need to average it using all the ensemble of possible realizations. But that is impossible. So, for each rule, we generate a sufficient number of networks of that ensemble, so that we can obtain average results with low noise.

These networks must be large enough, approximating the ”infinite” case, because there are some quantities that can be only measured for very large networks, as is the case of dimensionality or average distance growth with system size.
One of the limitations of finite size networks is a cut-off in the degree distribution. That will limit, for example, the observation of power law behaviour for high degree values. We can have rough estimation the cut-off, k_{cut}, using:

$$N \int_{k_{cut}}^{\infty} P(k) = 1 \tag{4.6}$$

For many rules, we obtained degree distributions that approached power-law $P(k) \sim k^{-\gamma}$ or exponential $P(k) \sim \exp(-\beta k)$ distributions. We can obtain $k_{cut} \approx k_0 N^{1/(\gamma-1)}$ for a power law and $k_{cut} \approx k_0 + \frac{1}{\beta} \ln N$ for exponential distribution, in which $P(k < k_0) = 0$.

To have a better sense of the computational constrains of this simulations, section 6.9, in the appendix, has further information.

4.3 Rules for growth

Rule G1
After a considerable number of steps (2^{26}), we obtained the following degree distribution:

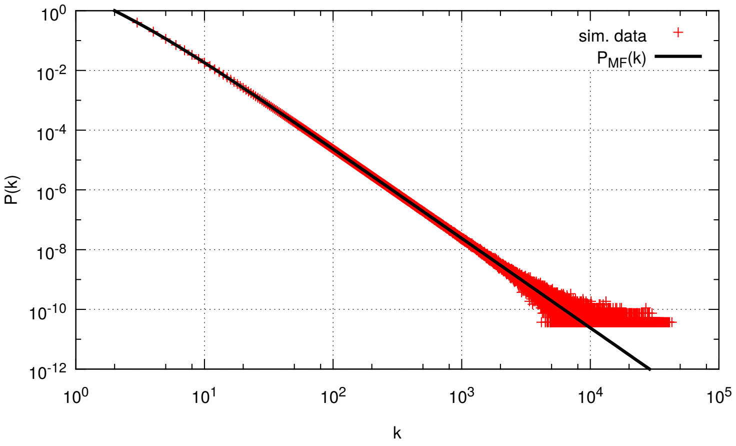


Figure 4.1: Rule G1. Degree distribution for system size $N = 2^{26}$.
Simulation data and analytical distribution $P_{MF}(k)$ of equation 4.10

Figure 4.1 shows that the degree distribution follows a power law, such that $P(k) \sim k^{\gamma}$, with $\gamma \approx 3$.

This is happening due to the preferential attachment mechanism that is operating in this system. In the BA model, we know that if the new vertex connects to old vertices proportionally to their degree, a power law distribution will emerge given enough steps.
In this rule's case, we see that a by choosing a triangle uniformly at random, we are in fact choosing vertices proportionally to their degrees, because the number of triangles adjacent to a vertex is equal to its degree.

Using a mean field theory, (see section 6.5), we can derive the exact degree for the infinite case. Assuming that there are no degree correlations, then we can build the following master equation:

$$N(k,t+1) = N(k,t) + 3P(k-1,t)\frac{k-1}{\langle k\rangle} - 3P(k,t)\frac{k}{\langle k\rangle} + \delta_{k,3} \tag{4.7}$$

in which t is the step number. This equation shows in the left hand side the number of vertices with degree k in time $t+1$ that will be equal to the current number of nodes plus 3 chances of obtaining vertices from degree $k-1$ using a probability proportional to their degree, minus loosing 3 vertices of degree k, with probability also proportional to their degree.
For $t \to \infty$, we can approximate $N(k) \approx tP(k)$, and $\langle k\rangle \approx 6$, then we will get

$$P(k) = \frac{1}{2}(k-1)P(k-1) - \frac{1}{2}kP(k) + \delta_{k,3} \tag{4.8}$$

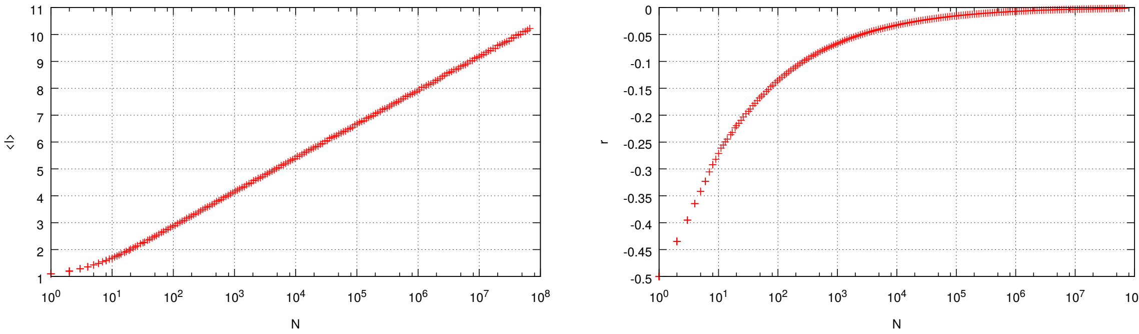
Reorganizing the terms, we will obtain the recursion relation:

$$P(k) = \frac{(k-1)}{(k+2)}P(k-1) + \frac{2}{(k+2)}\delta_{k,3} \tag{4.9}$$

Assuming that $P(k<3) = 0$, that will lead to:

$$P(k) = \frac{24}{k(k+1)(k+2)} \tag{4.10}$$

If we recall plot 4.1, we see that this analytical result fits perfectly the simulation data. Also, our assumption about non-existent correlations is supported by fig. 4.2b, were the Pearson Coefficient is approximating 0 for large networks sizes.



(a) Average distance $\langle l \rangle$ evolution with system size N.

(b) Pearson coefficient r evolution with system size N.

Figure 4.2: Rule G1. Average distance and Pearson coefficient evolution.

Looking at the distance evolution in plot 4.2a, we see that the mean distance evolution is proportional to $\log t$.

We can explain this behaviour using the following arguments:
A path passing through the new vertex has to pass through two of the vertices of the chosen triangle. But we can shortcut that path, using the direct connection between these two old vertices. So, a path passing through the new vertex cannot be minimal. Therefore, the shortest distances between old vertices will not change. The only variation of the mean distance can only come from the distances between the other vertices and the new vertex.
Using $\langle l_i(t) \rangle$ as the mean distance to vertex i at time t, then, the mean distance at time $t+1$ is:

$$\langle l(t+1) \rangle = \frac{1}{N+1} \sum_{i=1}^{N+1} \langle l_i(t+1) \rangle = \frac{1}{N+1} \left(N \langle l(t) \rangle + \langle l_{N+1}(t+1) \rangle \right) \tag{4.11}$$

For $t \to \infty$, assuming $\langle l_{N+1}(t+1) \rangle \approx \langle l(t) \rangle + 1$ and $N + const. \approx t$, then:

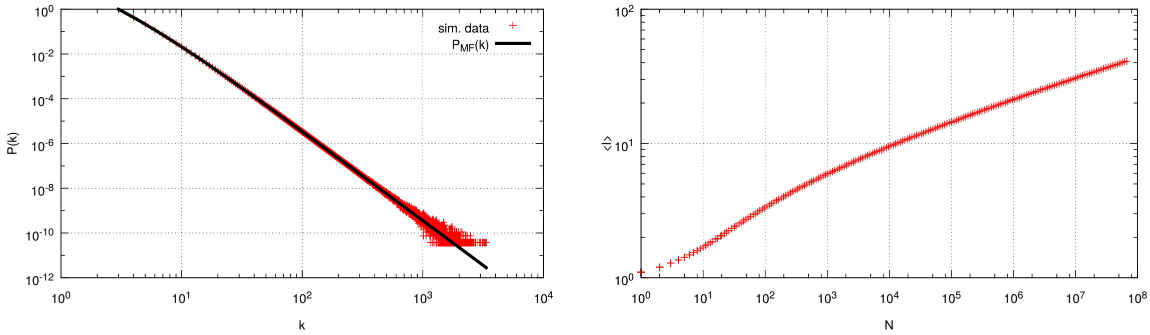
$$\frac{\partial}{\partial t} \langle l(t) \rangle \approx \frac{1}{t} \tag{4.12}$$

So, we see that the variation of average distance is proportional to the inverse of time, giving us a logarithmic growth.

If we calculate the average local clustering coefficient, using the degree distribution 4.10 in equation 4.5, we will obtain $\overline{C} = \frac{2}{3}$.

Given it's characteristics of growth, this network can be considered small-world.

Rule G2
Plot 4.3a shows that the degree distribution also follows a power law, $P(k) \sim k^{\gamma}$, with $\gamma \approx 4$.



(a) Rule G2. Degree distribution for system size $N = 2^{26}$. Simulation data and analytical distribution $P_{MF}(k)$ of equation 4.15.

(b) Rule G2. Mean distance $\langle l \rangle$ evolution with system size N.

Figure 4.3: Rule G2. Degree distribution and average distance evolution.

This can be explained by the similar behaviour to rule G1, where vertices with higher degrees have a better chance to increase their degree. Each vertex has a perimeter of edges that connect all its neighbours. The number of edges of this perimeter is equal to the vertex degree. If an edge of this perimeter is chosen, then this vertex will increase it's degree.

Using the method of mean field, used in the derivation of the degree distribution in rule G1, we will obtain the master equation. Again, no degree correlations are used:

$$N(k, t+1) = N(k,t) + 2P(k-1,t)\frac{k-1}{\langle k \rangle} - 2P(k,t)\frac{k}{\langle k \rangle} + \delta_{k,4} \tag{4.13}$$

leading to the recursion:

$$P(k) = \frac{k-1}{k+3}P(k+1) + \frac{3}{(k+3)}\delta_{k,4} \tag{4.14}$$

Assuming that for large t, $P(k < 4) = 0$, then the exact degree distribution is:

$$P(k) = 360\frac{(k-1)!}{(k+3)!} \tag{4.15}$$

that for large k, it will be $P(k) \sim k^{-4}$.

Plot 4.3b does not show a clear relation between the evolution of mean distance with system size. We have tried to grow the network as large as possible, but even then the relation with system size is not obvious. So, instead of trying to fit a function, we obtained the derivative in log-log space, giving us the exponent of a local fitting of a power law. For $N = 2^{26}$, we have obtained an exponent around ($\langle l \rangle \approx N^{0.17}$). The procedure is better explained in appendix 6.6

The analytical derivation of the mean distance of this rule could not be deduced as easily as in rule G1. The modification that G2 introduces, by dividing an edge, will only affect shortest paths that use that same edge, if they have no shortest alternative. To calculate the new average distance, we will need a strategy to quantify the shortest paths that are affected, a task that is much harder that in rule G1.

Rule G3

This rule is the most simple application of the expansion mechanism (recall section 3.1). Looking to it's degree distribution, we can see that it achieved one of the highest values of $\gamma \approx 6$ of all simulations (see table 4.1). This means that high degree nodes are very hard to create. We can explain that by the rule's mechanism itself, that splits the neighbours of the chosen middle vertex between itself and a new vertex. Clearly this mechanism reduces the chances for the survival of high degree vertices.

The mean distance variation with system size has an exponent ≈ 0.19, what is coherent with the dimensionality above 5.

Rule G4

Being a variation of rule G3, we wanted to see what is the impact of introducing the flip as the end step. It seems that the dimensionality has decreased and the degree distribution has a lower γ. Because a lower γ usually means smaller diameters, we were expecting that it would lead to a higher dimensionality. In fact, at each step, the extra flip is increasing 2 degrees for the two neighbours of the middle vertex, explaining the lower γ. The flip is also modifying the average distance in two ways: it is increasing the distance for some shortest paths that pass in the middle vertex, but is lowering the distance for shortest paths that pass in the two edges formed by the middle vertex and it's two neighbours.

Rule G5

We wanted to know how the choice of edges, instead of vertices (like in G3), would affect the network. This method of choosing elements of the surface, introduces a preferential choice because the higher the degree of a node is, the more chances has to be chosen.

The degree distribution shows a cut-off for k lower than G3, showing more difficulty in generating high degree nodes. We think this is happening because the preferential choice is leading to the destruction of high degree nodes.

Looking at the result of the dimensionality, we see that it achieved a much lower dimensionality ($d_H \approx 3.1$), relative to G3 ($d_H > 5$).

Rule G6

Given the minor differences between this rule and rule G5, we see that some characteristics such as the Hausdorff dimensionality are almost the same. However, the degree distribution has a cutoff at a higher degree than G5. This is because in G5, the end of the chosen edge that had a higher degree would had a higher probability to be the middle vertex. In G6, both ends had the same probability, therefore the high degree nodes in G5 had a higher chance to be destroyed than in G6.

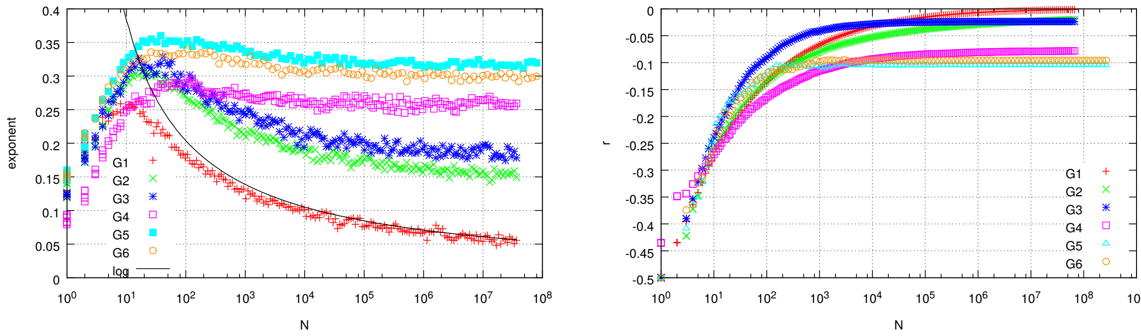
General results of the rules for growth

Growth rule	Network size (N)	Number of simulations	P(k)	$\langle d \rangle (N)$	Spectral dim. ($N = 2^{17}$)	Hausdorff dim.
G1	2^{26}	400	$\sim k^{-3.0(1)}$	$0.546(4) \ln N$	$2.2(2)$	∞
G2	2^{26}	400	$\sim k^{-3.95(5)}$	$\sim N^{0.17(2)}$	$2.1(2)$	> 5
G3	2^{26}	80	$\sim k^{-5.9(4)}$	$\sim N^{0.19(3)}$	$1.9(1)$	> 5
G4	2^{26}	400	$\sim k^{-5.0(4)}$	$\sim N^{0.27(1)}$	$1.9(1)$	$3.8(1)$
G5	2^{26}	400	n/a	$\sim N^{0.32(1)}$	$1.9(1)$	$3.1(1)$
G6	2^{26}	400	$\sim \exp(-0.40(1)k)$	$\sim N^{0.30(1)}$	$1.9(1)$	$3.3(1)$

Table 4.1: Rules for growth. Simulation results.

The results show that various degree distributions and mean distance evolutions could be achieved by simply using different rules for growth. Nevertheless, the spectral dimension for all rules is around 2, what is not so surprising given that the surfaces are 2D manifolds. The surfaces used in this measurement were very small, given the constraints of the method used to measure it, which needed to obtain the eigenvalues of a matrix with sides equal to the number of nodes.

In terms of structural evolution, we used the Pearson coefficient (see section 6.4) which for every rule showed an evolution that converges to a stable value for large networks.



(a) Exponent of power-law local fitting evolution with time.

(b) Pearson coefficient evolution with time

Figure 4.4: Rules for growing. Exponent of average distance and Pearson coefficient evolution for various rules.

Plot 4.4a shows the evolution of the measurement of the power-law exponent with system size (see details of this method in 6.6). For large system size, every network show an assimptotically evolution to a constant value. We see that rule G1 exponent goes to 0, that is almost the same behaviour that a log has.
The Pearson coefficient (4.4b) shows an initial strong anticorrelation from the initial surface. After enough steps we see this coefficient growing thowards 0 (G1 and G2) or stabilizing in a constast value (G3 and G4).

4.4 Variations of rule G3

G3.1

This rule was designed to observe the impact of selecting the 2nd neighbour, during the expansion procedure.

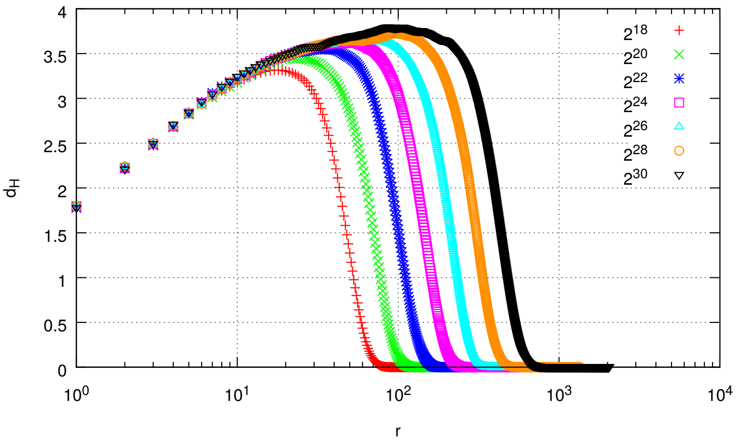


Figure 4.5: Rule G3.1. Hausdorff dimension d_H evolution with radius r for various system sizes N.

The dimension of the network clearly diminished from $d_H \approx 5$ (G3) to $d_H \approx 3.7$. So we see that this biased division mechanism has influenced the dimensionality, as expected.
Also, the high exponent $\gamma \approx 6.3$ of the degree distribution, clearly demonstrates the destruction of high degree nodes. In rule G3, given a less biased neighbour selection, the exponent was lower, $\gamma \approx 5.9$.

G3.2

This rule forced the choice of 2nd neighbour to be next to the 1st neighbour, given the order of neighbours around the middle vertex.

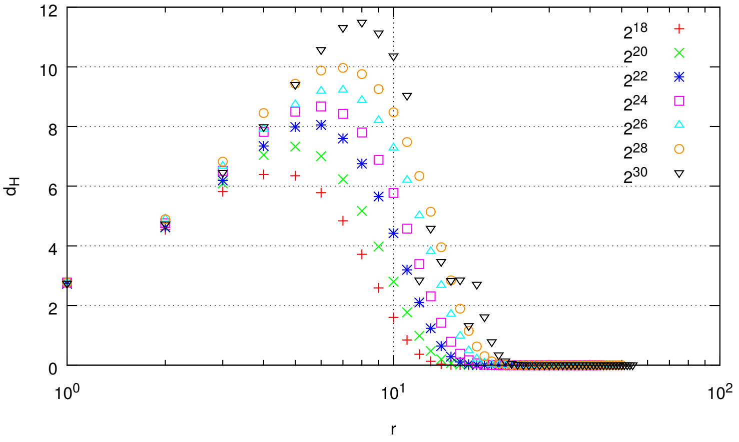


Figure 4.6: Rule G3.2. Hausdorff dimension d_H evolution with radius r for various system sizes N.

This rule is very similar to G1. We already know that the mean distance of G1 evolves with the logarithm of system size, so we expected that the same happened with this rule, what was confirmed by the experimental results. The growing dimension with system size indicates a possible infinite dimensionality.

G3.3

The choice mechanism in this rule is clearly a preferential choice of high degree vertices. Therefore, high degree nodes are being destroyed, leading to a more uniform degree distribution and a lower dimensionality than G3.1.

G3.4

This rule clearly chooses an vertex with preference proportional to its degree. The impact of this choice is almost the same as in G3.3 and the dimensionality and distance evolution reflects this similarity.

G3.5

The difference to rule G3.4 is that we forced the choice of the 2nd neighbour to be the opposite edge to the first chosen edge. This had almost no impact in dimensionality. Maybe the preferential choice by vertex degree is enough to achieve this dimensionality.

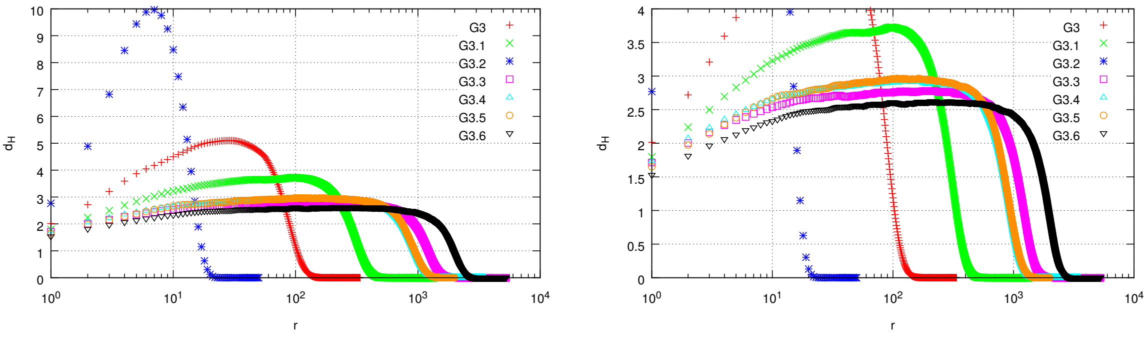
G3.6

In this last variation, we forced the choice of the middle vertex to be the edge's highest degree end. This had some impact in dimensionality, achieving $d_H \approx 2.7$, the lowest one of all experimented rules.

General results of the variations of rule G3

Growth rule	Network size (N)	Number of simulations	P(k)	$\langle d\rangle(N)$	Spectral dim. $(N=2^{17})$	Hausdorff dim.
G3.1	2^{26}	80	$\sim k^{-6.3(3)}$	$\sim N^{0.27(1)}$	$1.9(2)$	$3.7(2)$
G3.2	2^{26}	80	$\sim k^{-4.4(4)}$	$0.78(4)\ln N$	$2.2(2)$	∞
G3.3	2^{26}	80	$\sim \exp(-1.0(1)k)$	$\sim N^{0.36(1)}$	$1.9(2)$	$2.8(1)$
G3.4	2^{26}	80	$\sim \exp(-0.7(1)k)$	$\sim N^{0.34(1)}$	$1.9(3)$	$2.9(1)$
G3.5	2^{26}	80	$\sim \exp(-0.5(1)k)$	$\sim N^{0.34(1)}$	$1.9(2)$	$2.9(1)$
G3.6	2^{26}	80	$\sim \exp(-1.4(1)k)$	$\sim N^{0.38(1)}$	$1.9(2)$	$2.65(5)$

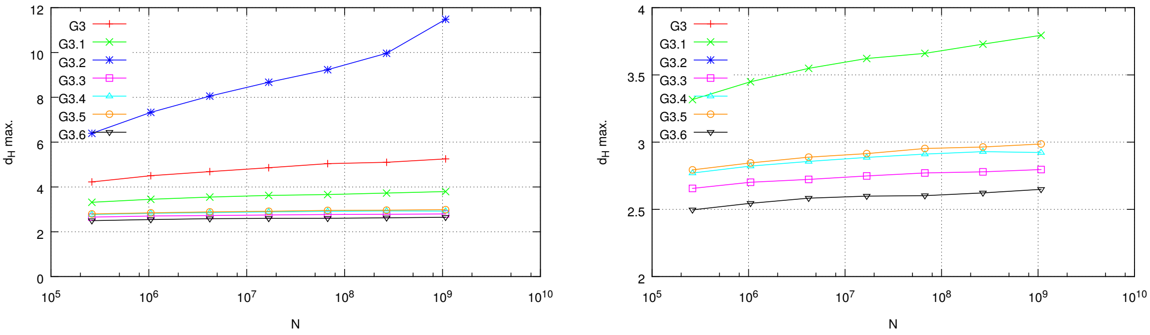
Table 4.2: Variations of rule G3. Simulation results.



(a) General plot.

(b) Detail plot.

Figure 4.7: Variations of G3. Hausdorff dimension variation with radius for various rules. System size $N = 2^{28}$.



(a) General plot.

(b) Detail plot.

Figure 4.8: Variations of G3. Maximum Hausdorff dimension variation with system size, for various rules.

What can we conclude from all these variations? It seems that destroying high degree nodes is a good method to achieve low dimensionalities. Nevertheless, the uniformity of degree, such in a regular surface (eg: icosphere), was far from being reached. Likewise, Hausdorff dimension 2 was also very difficult to achieve, being 2.7 the closest we obtained.

A interesting feature of all these variations of G3 (except for G3.2) was the uniformity of the spectral dimension, with almost all of the rules going around $d_S \approx 1.9$.

4.5 Rules for relaxation

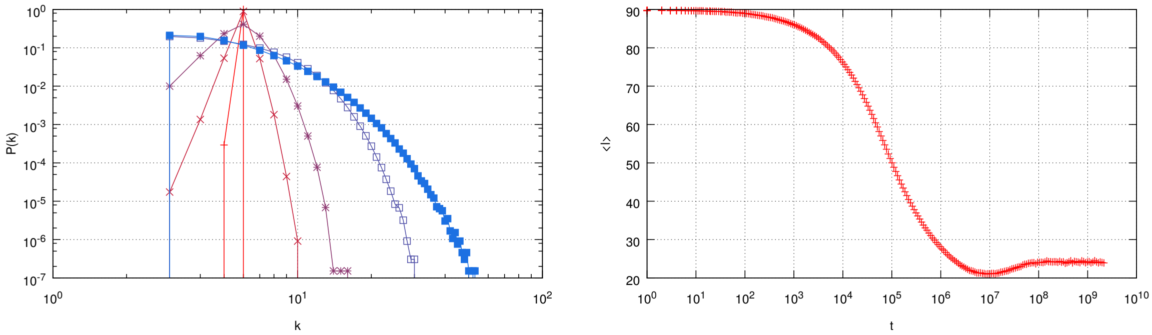
There are two main stages when making these simulations. The first stage is the evolution from an initial surface until the equilibrium stage, and the equilibrium stage itself.
Because we are using the rules for relaxation, the number of elements of a network is fixed, therefore, in order to obtain quantities that vary with system size, we had to run the rule for different system sizes until they reach the equilibrium stage.
Although ideally we would want to relax very large networks, the number of steps needed to reach the equilibrium stage also grow with system size, taking too long to simulate system sizes above $N > 10^5$.
Another problem is related to initial conditions, what should be the initial surface to relax? What we have observed observerd is that the inital surface does not matter. So, all rules begun with the icosphere (see section 6.8), except rule R1.

Rule R1

This rule begins by relaxing a surface grown by rule G1. Comparing the results (table 4.3), we see there is a clear evolution from rule G1. This shuffling of vertices, has proven to be a effective way of changing the structure of the network.

Rule R2

The inital surface, the icosphere, aproximated delta degree distribution around 6. After relaxation, the network achieved a exponential distribution, revealing a structural change. In figure 4.9a, we see this evolution from initial(red) to final(blue) distribution.



(a) Rule G2. Degree distribution $P(k)$ evolution with time. Initial time in red, final time in blue.

(b) Rule G2. Mean distance $\langle l \rangle$ evolution with time.

Figure 4.9: Rule G2. Network characteristics evolution with time.

We see that the average distance evolution with time is not monotonic. There is a point where the average distance goes to a minimum, and then rises until it reaches equilibrium. This minimum is persistently present for various initial sizes.
If the starting surface was a small world surface, the evolution was somewhat different, but we see an inflexion about the same time.
The time of minimum also evolves, but at a different rate of the initial time for the equilibrium.
We can compare two speeds of growth with the size of system: initial time of equilibrium stage

and time of minimum distance.

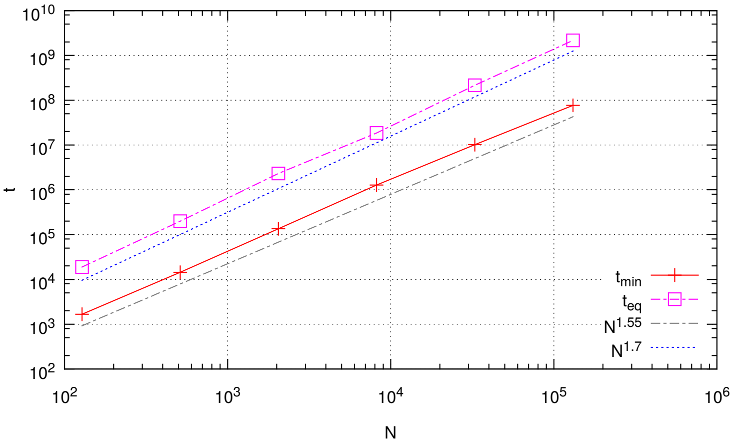


Figure 4.10: Rule R2. Evolution of time of minimum distance t_{min} and time of start of equilibrium t_{eq} with system size N. An icosphere is used as the initial surface.

Rule R3

This rule combines rules G3 with it's inverse process. Comparing with the simple growth of G3, we see that the degree distribution has a lower γ. This is related to the compressing process which creates high degree nodes.

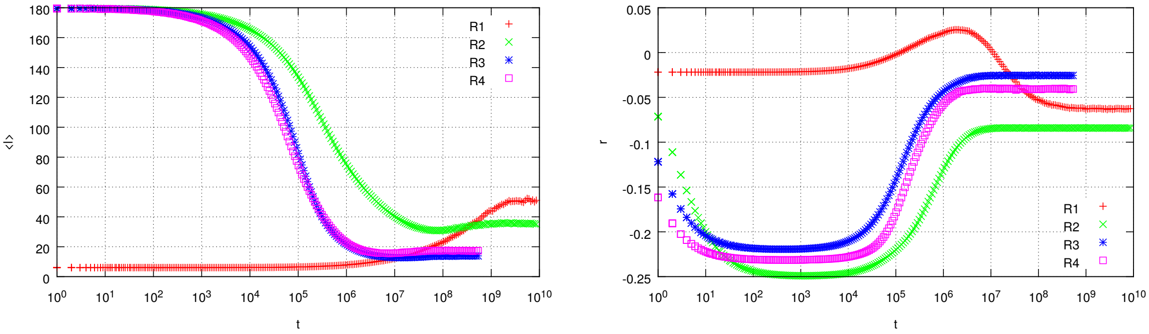
Rule R4

Rule R4 is quite similar to rule R3, except an extra flip in the expansion phase. The resulting degree distributions, distance evolutions and dimensionalities are quite similar. The extra flip gave R4 a higher γ and a lower dimensionality.

General results of the rules for relaxation

Growth rule	Network size (N)	Number of simulations	P(k)	$\langle d \rangle(N)$	Spectral dim. ($N = 2^{15}$)	Hausdorff dim.
R1	2^{15}	32	n/a	$\sim N^{0.47(6)}$	$1.3(1)$	$2.1(2)$
R2	2^{17}	20	$\sim \exp(-0.31(1)k)$	$\sim N^{0.30(2)}$	$1.5(2)$	> 3
R3	2^{17}	20	$\sim k^{-3.0(2)}$	$\sim N^{0.20(2)}$	$1.3(2)$	> 4
R4	2^{17}	20	$\sim k^{-3.3(1)}$	$\sim N^{0.25(3)}$	$1.5(2)$	> 3.5

Table 4.3: Rules for relaxation. Simulation results.



(a) Average distance $\langle l \rangle$ evolution with steps of relaxation t

(b) Pearson coefficient r evolution with steps of relaxation t

Figure 4.11: Rules for relaxation. Average distance and Pearson coefficient evolution with time for various rules.

The dimension of the relaxed networks is quite small, because the number of steps needed to reach the equilibrium phase increases above linearly with system size. This growth can be observed in plot 4.10.

Plots 4.11a and 4.11b show evolving characteristics during the relaxation process, which demonstrate the transition between a starting structure to another kind of structure. We see that rule R1 has a different evolution relative to the other ones, given that its initial structure is a network built by R1, instead of the icosahedron for the all the others.
After a large number of steps, both characteristics stabilize in constant values, which is a sign of an equilibrium phase.
Rules R2, R3 and R4, show an evolution of average distance which is non-mononotic, achieving an average distance lower that the average distance at the equilibrium phase.

One of the most surprising characteristics of these algorithms is the spectral dimension. Given the relaxing purpose of these rules, it was expected that these rules would "uniformize" the structure, leading to a spectral dimension that would reflect the dimension of the embedding space. But what we have found is that all of these relaxing algorithms obtained a dimension $d_S \approx 1.5$ and even lower. Due to computational limitation, the largest network in which we measured this dimension had size $N = 2^{15}$. So the measured d_S still might change significantly, and may even approach 2, for larger networks.

4.6 n-dimensional results

In order to derive the degree evolution of the n-dimensional versions of rules G1 and G2, we will assume that there are no correlations. Assuming that we can compare the edge introduction to a preferential attachment operation, we can compare this to the model presented in [20, 23]. In that model, the exponent γ in $P(k) \sim k^\gamma$ is given by $\gamma \approx 3 + \frac{A}{m}$ in which A is the initial attractiveness of a site, given by $P_{choice}(k) \sim k + A$, and m, which is the number of links introduced in the model at each step.

Let's call S_n the number of n-simplexes incident to a vertex. The starting object of a manifold composed of n-simplexes, will be an $n+1$-simplex, therefore each vertex of this initial manifold has degree $k = n + 1$ and $S_n = n + 1$.

Rule 1 (n-dimensional)
Let's suppose that incident to a vertex, we have a new vertex. That will insert n n-simplexes and remove 1 n-simplex. The variation of simplexes is then $\Delta S_n = n - 1$.
Each new vertex has degree $n + 1$, therefore we can write:

$$S_n(k) = n + 1 + (k - (n + 1))(n - 1) \tag{4.16}$$

The probability of choice is then $P_{choice}(k) \sim k - (n + 1) + \frac{n+1}{n-1}$, obtaining a function $A = \frac{n+1}{n-1} - (n + 1)$. The number of new edges is equal to the number of vertices in a n-simplex, therefore $m = n + 1$, leading to:

$$\gamma = 2 + \frac{1}{n - 1} \tag{4.17}$$

Rule 2 (n-dimensional)
In this case, we have two different mechanisms of attraction. One of them is related to the $n - 1$-simplex that is divided and another is for the two vertexes of the two n-simplexes that share a $n - 1$ simplex.

The number of $n - 1$ simplexes for each vertex, S_{n-1} is dependent on the number of n-simplexes incident to the vertex. Given that each n-simplex has $n + 1$ faces ($n - 1$-simplexes), and that each of it's vertexes is on n of those, then the number of faces incident to each vertex is:

$$S_{n-1} = S_n \frac{n}{2} \tag{4.18}$$

in which the $1/2$ factor comes from the repetition of faces, which are shared by pairs of n-simplexes. The probability of choice for each vertex is then $P_{choice} \sim S_n$, for n new edges.

In the other hand, the for each n-simplex incident to a vertex, there is one face that does not include that vertex, so the number of faces is equal to the number of n-simplexes. Therefore the probability of choice is $P_{choice} \sim S_n$ for 2 new edges.

We see that both choices of vertex are proportional to the number of n-simplexes incident to it. If we recall Rule 1 (n-dimensional), we will have $A = \frac{n+1}{n-1} - (n + 1)$. The number of new edges will be $n + 2$. Therefore the exponent will be:

$$\gamma = 2 + \frac{2n}{(n - 1)(n + 2)} \tag{4.19}$$

Recalling the results for $n = 2$ (rule G2), we know that $\gamma = 4$, but equation 4.19 gives $\gamma = 3$. In fact, the model we are using to obtain the exponent, assumes that each new vertex has a degree equal to the number of new connections introduced in the model, but for $n = 2$ that is false. This can be solved by considering that every new vertex has an initial attractiveness $A = 2$, so $\gamma = 3 + \frac{2}{2} = 4$ as it should be.

Chapter 5

Evolution of surface topology

Until now, the focus has been in surfaces homeomorphic to the sphere, in which the generated surfaces had Euler characteristic $\chi = 2$.
In this section, we are going to evolve surfaces in which ”wormholes” will be introduced, leading to $\chi \leq 0$. Two approaches were used to accomplish that. The first one, begins with a surface with very few vertices and a given number of wormholes. Like in previous growing evolutions, we simply let growing rules evolve this surfaces. (See construction of these surfaces in 6.7)

The other approach consists in letting a surface grow with a growing rule, but then, at certain times, adding new wormholes. This process is controlled, regulating the rhythm at which the wormholes are added.

Wormhole introduction mechanism
To introduce wormholes in a surface, two triangles are chosen at random, their vertices are merged together, the two triangles are eliminated, and the wormhole is therefore created. This process eliminates 2 triangles, 3 vertices and 3 edges, and if we recall Euler's formula (eq. 2.6), we see that $\Delta\chi = -2$.
The process is described in more detail in the appendix.

There are some constraints limiting the choice of the two triangles. When a pair of vertices are merged together, there can be no common neighbours between them. This will prevent the creation of double edges or self edges.
Normally this kind of constraints manifest at small system sizes, when the surface is not big enough to have a good separation between the two chosen triangles. Conversely, for very big networks, this should become an almost non-existing event, given the distance between them.

Not only spatial problems constraint the introduction of new wormholes, but time can be also a limiting factor. At small t, the surface has a small number of triangles, so finding triangles that are separated enough is almost impossible, evolving to evolutionary dead-ends, where no wormhole can be introduced. Also, if the introduction of wormholes if very frequent, there is no time to grow enough the surface to make the needed separation.
We introduced wormholes with two frequencies, one proportional to $\log t$ and another proportional to t. Both methods start with no wormholes at $t = 0$. For the logarithmic frequency, we

had a t_1 where the first wormhole was introduced. The next wormholes were introduced at times $t_i = \lfloor t_1.\alpha^{i-1} \rfloor$ [1] , we see that at each t, the number wormholes is about $H \approx \frac{\ln t}{\ln \alpha}$.
The other method is proportional to time, so a new wormhole is introduced at time $t_i = i\beta$.

5.1 Results

5.1.1 Fixed number of wormholes

We start by analysing the evolution of surfaces with fixed number of wormholes.
In order to compare the impact of the various topologies, we designed simulations that started with various numbers of wormholes and then evolved with a growing algorithm.
The parameter that we use here is the number of wormholes H. Figure 5.1 shows the degree distribution of systems that started with different number of wormholes, but evolved to the same number of vertices:

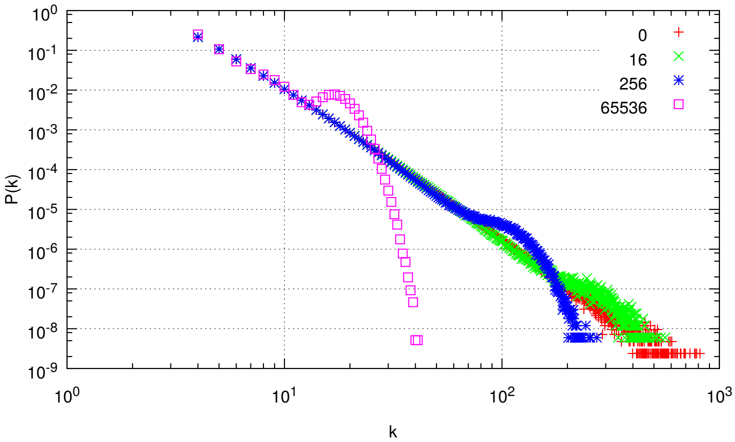


Figure 5.1: Fixed number of wormholes. Degree Distribution.
Rule G2 applied 2^{20} times to various initial number of wormholes H.

Clearly the number of wormholes induces a smaller degree cut-off and even an ”hump” in the degree distribution for high degrees. Next, we will fix the number of wormholes and test various system sizes.
Plot (5.2) shows the degree distribution for $H = 4096$ and various system sizes:

[1] $n = \lfloor x \rfloor$ is the largest integer, such that $n \leq x$.

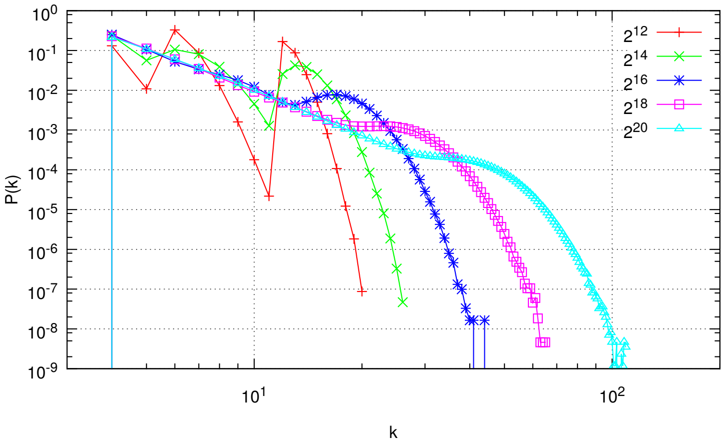
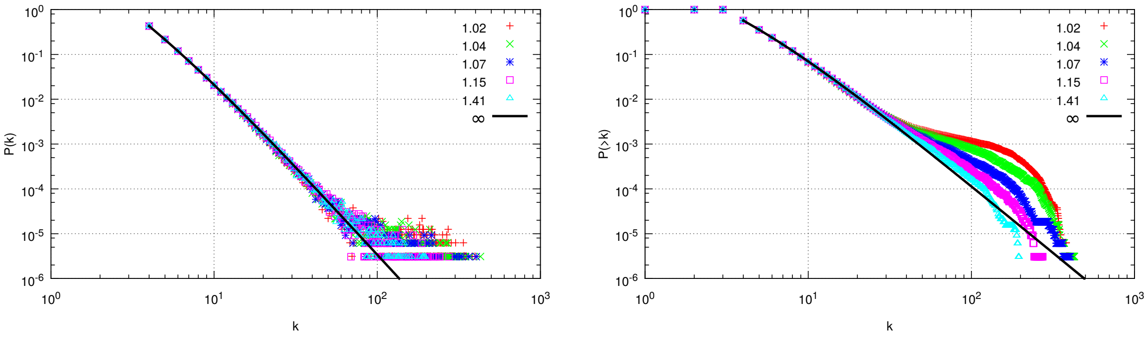


Figure 5.2: Fixed number of wormholes ($H = 4096$). Degree Distribution.
Comparison of various surface sizes, N, grown with rule G2.

For small sizes we see the manifestation of small size effects, but the increasing size of the model makes the initial surface almost irrelevant. In fact, for large system sizes we see that the degree distribution approximates the degree distribution of the surface without wormholes.

5.1.2 Dynamic wormhole introduction

In this approach, the surfaces are grown by rules G1 or G2. The two different methods to introduce wormholes have a very different impact on the evolution of the network.
In the case of the logarithmic approach, we had the α parameter [2] controlling the "logarithmic spacing" between introductions.



(a) Degree distribution

(b) Cumulative degree distribution

Figure 5.3: Logarithmic dynamic wormhole introduction. Degree distribution and cumulative degree distribution.
Rule G2 applied 2^{15} times for different α. $t_1 = 100$

Using rule G2, we see that the cumulative distribution has a little hump in the large degree region. This is expectable, because the fusion mechanism increases the degree of vertices of

[2] In logarithmic wormhole insertion, $\alpha = t_{i+1}/t_i$, where t_i is the time of insertion of wormhole i .

the merged triangles. The γ of the degree distribution is around 4, almost the same in the case without wormholes.

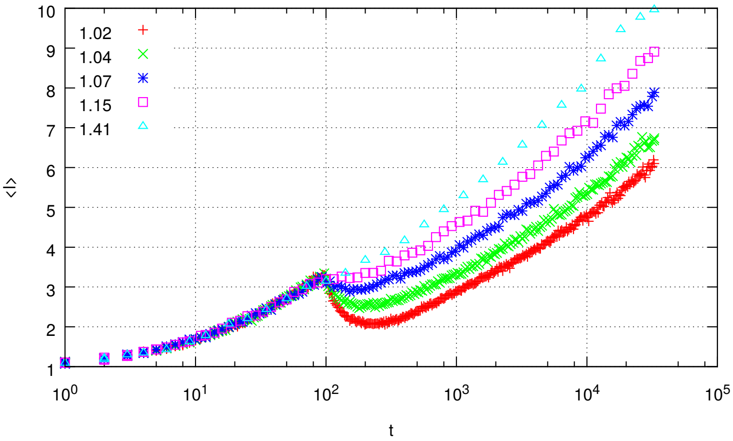
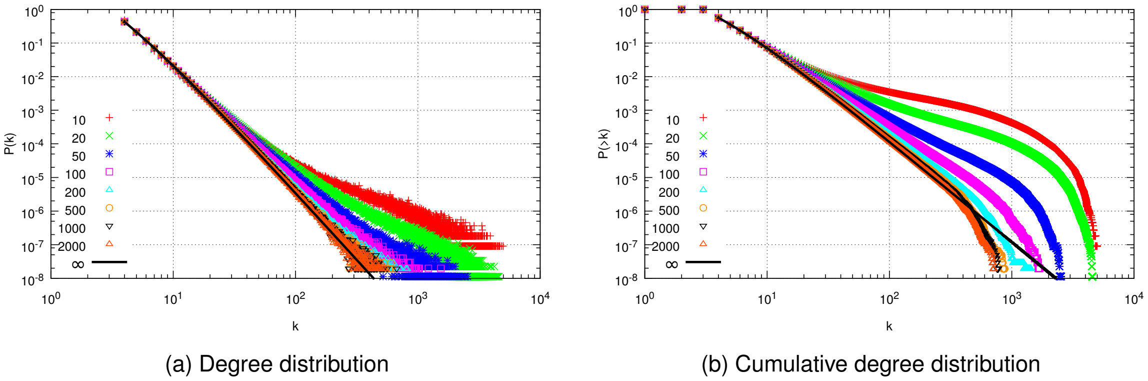


Figure 5.4: Logarithmic dynamic wormhole introduction. Mean distance. Rule G2 applied 2^{15} times for different α. $t_1 = 100$

The initial time where wormholes start to be inserted, t_1, can be chosen freely, as long it permits the surface to grow to a size large enough to sustain the wormhole insertion process. In the cases presented in fig. 5.4, $t_1 = 100$. Immediately after this point, the distance falls very fast, thanks to the introduction of shortcuts, but after some time, the normal distance growth of the growth rule resumes. The only difference between the different α values is an offset of the distance plot, moving to lower mean distance values for lower α.

The most interesting case, was the linear introduction of wormholes, in which the hole insertion frequency is uniform during all growth. To control the rate at which the wormholes are introduced, we have the β parameter [3] .



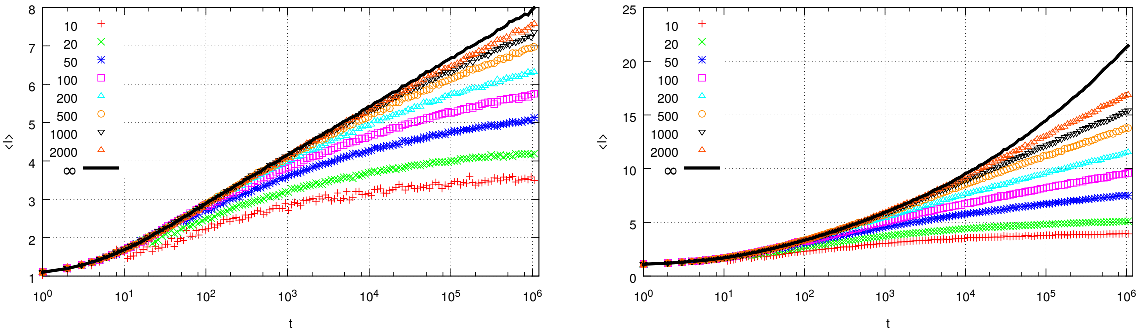
(a) Degree distribution

(b) Cumulative degree distribution

Figure 5.5: Linear dynamic wormhole introduction. Degree distribution and cumulative degree distribution. Rule G2 applied 2^{20} times for different β.

[3] In linear wormhole insertion, $\beta = t_{i+1} - t_i$, where t_i is the time of insertion of wormhole i .

The degree distribution of this method also shows a hump at high degrees, even bigger than in the logarithm case.



(a) Rule G1

(b) Rule G2

Figure 5.6: Linear dynamic wormhole introduction. Mean distance evolution with time. Rules applied 2^{20} times for different β.

The most important finding was the evolution of average distance $\langle l \rangle$. The increase of $\langle l \rangle$ with t is slower for smaller β, even reaching sub-logarithmic growth. Nevertheless, there is a limit below which β cannot go, because the surface has no more space for wormholes, has discussed before.
Rule G2 has a mean distance that grows with a power-law when there are no wormholes, so if we decrease β, we can go from power law, passing through logarithm and reaching sub-logarithm growth. Rule G1 also shows the same behaviour with the variation of β, but only varies between logarithm growth and sub-logarithmic growth.

5.1.3 General observations

As we have seen, both implementations, fixed and dynamic wormhole introduction have some differences.

In the case of fixed wormhole introduction, when the system becomes very large, the achieved degree distributions and mean distance evolutions, were very similar to the case without wormholes. The influence of the initial surface fades out as the system grows. [20]
In the case of dynamic wormhole introduction, the two used approaches, logarithmic and linear frequency, showed differences between them. In the logarithmic case, the degree distribution and distance evolution are similar to the case without wormholes. The reason for this similarity, is that the logarithm frequency of wormhole introduction becomes very slow for large times, so the effect of the wormholes becomes almost irrelevant.
The case of introduction of wormholes linearly in time, we see a hump in the number of the high degree nodes. We think that the ”hump” is caused by the wormhole construction mechanism that is choosing uniformly the two triangles. That will lead to high degree nodes having a higher change of being chosen (node degree = number of adjacent triangles). The fusing process will then eliminate 3 high degree nodes, but those connections will not be lost, because it will be used to create 3 even higher degree nodes.

Chapter 6

Methods and characteristics

6.1 Statistics

The data generated in these simulations was randomly chosen from the ensemble of possible realizations. This choice was made by using different seeds for the random number generator used in the simulations which was the default of the standard library of the c language. In order to obtain the ensemble average, every network realization has the same statistical weight.
In order to calculate a give characteristic, we simply consider the number of nodes with that characteristic for each sampled network, as the weight in the calculation of its expected value.

For any network, we can calculate the fraction of nodes of some degree, therefore to obtain the average degree distribution we just use:

$$P(k) = \frac{1}{S} \sum_{s} P_s(k) \tag{6.1}$$

in which $P_s(k)$ is the degree distribution of realization s and $S = |\{s\}|$ is the number of realizations.

Mean distance
First, for each run s, for each time t, we chose a given number of random initial vertices (normally 10) from where we measured the distance for every other vertex obtaining the distance distribution:

$$P_s(l,t) = \frac{2}{N(N-1)} N_s(l_{ij} = l) \tag{6.2}$$

in which $N_s(l_{ij} = l)$ is the number of distinct pairs of vertices i, j in the realization s, that are at a distance l.
The mean distance between two nodes is given by:

$$\langle l \rangle(t) = \frac{1}{S} \sum_{s} \sum_{l=1}^{\infty} P_s(l,t).l \tag{6.3}$$

Degree correlations (Nearest Neighbor):
To calculate the average neighbor degree, we considered that every vertex of degree k of the ensemble had the same statistical weight.

Given a number of runs S for some rule, the nearest neighbour used the following formula:

$$\langle k_{nn}\rangle(k) = \frac{\sum_s \langle k_{nn}\rangle_s(k) N_s(k)}{\sum_s N_s(k)} \tag{6.4}$$

6.2 Hausdorff dimensionality

One of the simplest definitions of a dimension d is related to the volume of a ball of radius r, which volume grows with $\sim r^d$. [24]

In the case of networks, we can consider volume as the number of vertices within a distance r from a given vertex, let's call this volume $N(\leq r, i)$ for a vertex i. Then, we can establish the equation:

$$N(\leq r, i) \sim r^d \tag{6.5}$$

To obtain the dimension of our networks, we chose uniformly a given number of vertexes. For each node i, we applied the "Bread-first search" algorithm [25]. This algorithm, starts at node i, and progressively increases the distance, determining the 1st, the 2nd, ..., the nth neighbours, until all vertices are exhausted. If we count the number r-neighbours, then we will obtain $N(r, i)$, that naturally is proportional to the shortest-path distance distribution.

Using all the sampled i vertices, we can obtain a cumulative distribution:

$$N(\leq r) = \sum_i N(\leq r, i) \tag{6.6}$$

Assuming equation 6.5, we can obtain the Hausdorff dimension of the network by:

$$d_H = \lim_{r\to\infty} \frac{\partial \log N(\leq r)}{\partial \log r} \tag{6.7}$$

If we plot the dimension in function of radius, we see that the value $\partial \log N(\leq r)/\partial \log r$ is far from a constant. In fact, we only achieve this constant behaviour for large values of r, and at the same time, much smaller than $\langle l\rangle$.

6.2.1 Relation between Hausdorff dimension and average distance

Suppose the number of vertices N within a radius r grows with a power law:

$$N \sim r^{d_H} \tag{6.8}$$

If the average distance grows with a power law of exponent α, we can write:

$$\langle l\rangle \sim N^\alpha \Rightarrow N \sim \langle l\rangle^{1/\alpha} \tag{6.9}$$

Then we can relate the two quantities taking $r \sim \langle l\rangle$, obtaining:

$$\langle l\rangle^{d_H} \sim \langle l\rangle^{1/\alpha} \tag{6.10}$$

We then conclude that $\alpha = 1/d_H$.

6.3 Spectral dimension

The spectral dimension characterizes the diffusion capability of a network. Let's supose there is a particle performing a random walk in a given network, that has started its path at some stating vertex i at time $t = 0$. The probability $p_0(t)$ of finding that particle at the starting point at time t becomes lower if the dimensionality of the network is higher, and are related by the formula [20] :

$$p_0(t) \sim t^{-d_S/2} \tag{6.11}$$

in which d_S is the spectral dimension.
We can calculate d_S by simulating a series of random walks in our networks, but if the network is expected to be high-dimensional, the simulations will take an enourmous time to complete.

An alternative method to measure the spectral dimension, uses the Laplacian Matrix of the network. To derive this matrix, we will start by building a simple diffusion equation. Suppose we have some value ϕ_i in location i, we will have this variation in time:

$$\partial_t \phi_i = c \sum_j (\phi_j - \phi_i) a_{i,j} \tag{6.12}$$

The second term of the sum can be expressed as:

$$\phi_i \sum_j a_{i,j} = \sum_j \phi_j \delta_{i,j} k_i \tag{6.13}$$

We can rewrite que initial equation as:

$$\partial_t \phi_i = c \sum_j (a_{i,j} - \delta_{i,j} k_i) \phi_j = c \sum_j L_{i,j} \phi_j \tag{6.14}$$

With this relation, we can define the Laplacian matrix as:

$$L_{i,j} = k_i \delta_{i,j} - a_{i,j} \tag{6.15}$$

This derivation was obtained from Newman [25].
If we obtain the matrix eigenvalues $\{\lambda_i\}$, we can construct the eigenvalue density function:

$$\rho_L(\lambda) = \sum_i \delta(\lambda - \lambda_i) \tag{6.16}$$

where $\delta(x)$ is the dirac delta.
The relation between the spectral dimension and the eigenvalue density is given by:

$$\rho_L(\lambda) \sim \lambda^{D_S/2-1} \tag{6.17}$$

for λ values near 0.
In practice, to measure d_s in our networks, it is easier to built a cumulative function of $\rho_L(\lambda)$:

$$C_L(\lambda) = \int_0^\infty \rho_L(\lambda) d\lambda \sim \lambda^{D_S/2} \tag{6.18}$$

giving us a power law. By measuring the slope of $C_L(\lambda)$ in loglog space, we can get the spectral dimension.

If we evaluate the density of eigenvalues of a network's laplacian, $\rho(\lambda)$, we can obtain for values of $\lambda \to 0^+$, $\rho(\lambda) \sim \lambda^{d_s/2}$.

Examples of systems with exact spectral dimension are periodic regular n-dimensional lattices, in which every vertex has degree $2n$, and the network has spectral dimension equal to n. [5]
For regular periodic d-dimensional lattices, in which every element has n neighbours, the spectral dimension is exactly d.

(For a deeper understanding of this method, it can be consulted in [26])

6.4 Pearson Coefficient

The Pearson coefficient r is used to evaluate characteristics of structural correlations of a network. [27] This coefficient is defined as:

$$r = \frac{\langle jk \rangle_e - \langle k \rangle_e^2}{\langle k^2 \rangle_e - \langle k \rangle_e^2} \tag{6.19}$$

in which $\langle x \rangle_e$ denotes averaging x over edges and k is the degree of one end of an edge and j is the degree of the other end. This coefficient's value will show how much the degree correlations of this network deviates from the degree correlations of a random network.
If $r = 0$, it means the network is uncorrelated, like the classical random network.
If r is positive, it means that in average there is an assortative mixing of nearest neighbour's degree, but if is negative the mixing is dissortative.

6.5 Master equations and mean field theory

In order to obtain the mean evolution of a dynamic system, we can build a differential equation called the master equation [28]. If we assume that the system has no system memory of the previous state, then we can simply build a function that weights the transition from one state to another like:

$$\partial_t P(X,t) = \sum_{Y \neq X} [P(Y,t) p_{Y \to X} - P(X,t) p_{X \to Y}]$$

In which P(X,t) is the probability of the system being in state X at time t and $p_{X \to Y}$ is the transition probability from the configuration X to configuration Y. The master equation is made of two terms representing gain and loss contributions for the probability distribution of the system to be in a given configuration X. To obtain the transition probabilities, it is used a mean-field approximation, in which all the elements of the system in a certain state, feel an average interaction with the full system, ignoring correlations with elements in other states[26].

6.6 Fitting methods

In order to obtain the exponent of a quantity that obeys to a power law or an exponential function, the simplest way is to linearize the simulation data and fit a linear function in the region of interest. For degree distributions $P(k)$ that region is in the highest degree range, just before the cut-off and for the average distance $\langle l \rangle (N)$ that region is for the largest system sizes. If we measure this exponent systematically for various system sizes, we will obtain the trend of this exponent, as well the slight oscilations of this measurement, which will give us the error interval of this trend.

6.7 Topology evolution methods

To evolve the topology of a surface, the method is quite straightforward. We want to introduce a wormhole, so we choose two triangles at random and then we pair each vertex of one triangle with the vertices of the other triangle. The choice of pairing is random, only respecting orientation. Then, for each pair, we take the neighbours of one of them and add to the list of neighbours of the other.
Nevertheless, there are limitations to this method, if the two triangles are very close to each other. For each pair of vertices that will merge together, if they have common neighbours, the wormhole construction must be canceled, or else double edges would be created, which are forbidden in our model.
Even worse, if that common neighbour is a vertex of both triangles, that would create ”singularities” in the surface, that are vertices which their neighbours cannot define a closed path with a length equal to their number.

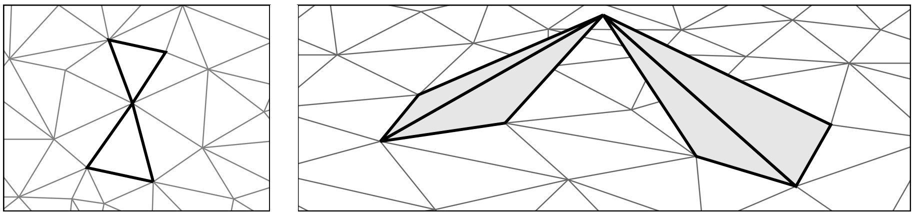


Figure 6.1: Singularities in the surface.
left: Selection of two triangles sharing a vertex.
right: Resulting singularity after merging (3D view).

Initial wormhole structures
In order to have a initial structure with a given number of wormholes, we have constructed a simple structure with the desired topology. The basic idea is that any topology can be achieved by gluing torus toghether. To minimize the impact of initial surface, a very simple ”donut” made of triangles was build. Then, by gluing H donuts to each other, it was obtained a surface with Euler’s characteristic $\chi = 2(1 - H)$

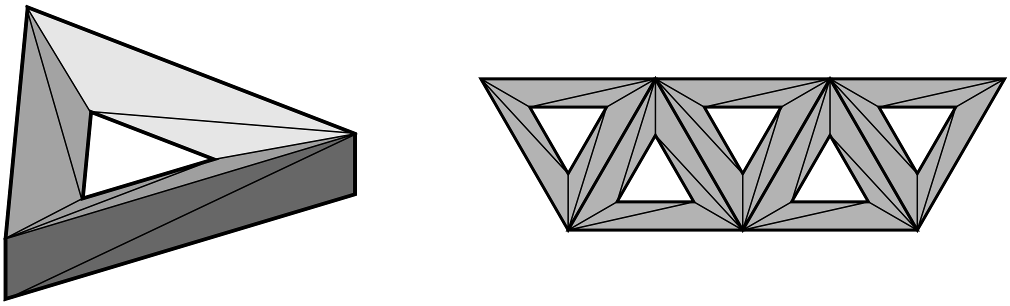


Figure 6.2: A ”donut” and a initial surface made of connected donuts.

6.8 Initial surfaces

To test the relaxing rules, we used surfaces generated by the growing rules, but also used ready-made surfaces. Initially we tried to build regular surfaces, but that is only possible for $\chi = 0$, the torus. To see that is so, let’s derive the formulas. Lets suppose that every vertex has an equal degree $k_i = c$, then we can write (for triangular meshes)

$$\sum_i k_i = Nc = 2E = 3T \tag{6.20}$$

Using Euler’s formula, we will get:

$$\frac{Nc}{3} + N - \frac{Nc}{2} = \chi \Leftrightarrow c = 6 - \frac{6\chi}{N} \tag{6.21}$$

For $\chi = 2$, we see that only the values $N = 4, 6, 12$ corresponding to the tetrahedron, octahedron and icosahedron can have all vertices with the same degree. Therefore for higher numbers of vertices, we can reach almost uniform degree, but not complete uniformity.

One of the initial surfaces is the recursive subdivision of an icosahedron, let’s call it an *icosphere*.

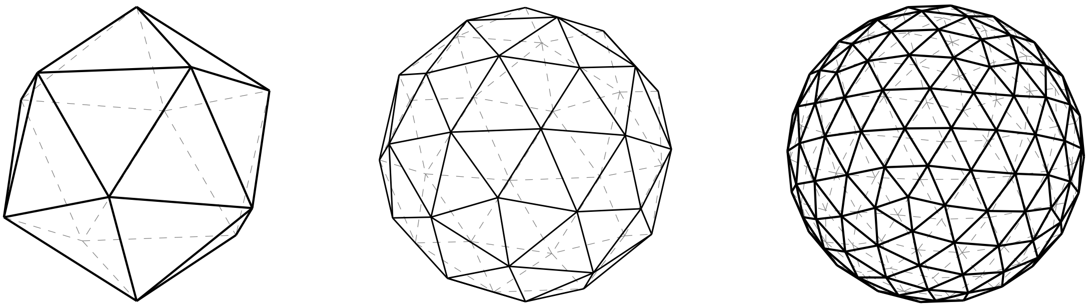


Figure 6.3: Stages of subdivision of an icosahedron (icosphere).

This surface is constructed by starting with an icosahedron and then recursively subdividing the triangles. A subdivision of a triangle consists in inserting a new vertex in each edge, dividing them, and creating a new triangle with the new vertices. All vertices will have degree 6, except the initial 12, that will have degree 5.

Another initial shape called the ”pillow”, consists in two planes of triangles, that are glued together at their borders. This object can achieve almost constant degree equal to 6, except the four corners of the pillow that must have degree 3.

Of course, any triangulated mesh, obeying to the desired topology and geometric constraints, can be used, but the objective of these initial surfaces is to be as uniform as possible, avoiding any unexpected structural propertie such as degree correlations.

6.9 Computational running costs

The first algorithms were written in python, in order to focus in writing correct code. To run the code faster, and with less memory, the code was then translated in C. Given the memory size and running time, for each rule, few networks were simulated for each run. Later the data of these small runs was joined to create the data.

Memory
The most practical way to deal with these network, was to build a vector of vectors of variable size, corresponding to the neighbourhood of each vertex. The neighbours of each vertex were ordered like in the surface (see section 3.1). The chosen data type was an `uint32_t`, (4 bytes), so the simulation size for $N = 2^{30}$ was at minimum 26GB. In fact given the overhead of statistics and pointers, the simulation was about 70-80 GB.

Time
In terms of time, the biggest simulations could run for 5 hours in a Intel Xeon CPU E5-2690 v3 for each network realization, only measuring the distance distribution in the final step. The measurement steps of distance were the most costly given that their complexity grows with $O(N^2)$. To minimize their impact, but still getting significant values, we measure the distance from around 10-50 random vertexes to all the remaining vertices.

Chapter 7

Conclusion

In this work, we use a network approach to produce evolving topologies. We focused mainly in 2D manifolds, and in a series of rules used to evolve them. We have found that by tunning these rules, we can obtain networks with distinct characteristics, such as degree distribution ranging from power-laws to exponential.

In contrast to regular random planar graphs which Hausdorff dimension is 4 [24], the networks in our work were not regular, revealing a variety of dimensionalities (from ≈ 2.0 to ∞). The series of rules we have tested were only a subset of possible combinations of selection strategies, so hypothetically dimension smaller than 2 can still be achieved.
We have also measured the spectral dimension for various networks, obtaining values between 1.5 and 2, contrasting to the variety of values of Hausdorff dimension.

We also found that changing the topology by adding wormholes at a constant rate, had a great impact in average distance, which had a sub-logarithmic growth with the number of nodes of the network.

The vast majority of this work was devoted to 2D surfaces, but the simple rules we developed could be generalized to higher dimensions, using simplexes of higher dimension. In section 4.6 shows that we can obtain power-law degree distributions $P(k) \sim k^{\gamma}$ with γ in the range between 2 and 4.

We can calculate analytical results by using techniques from network theory. For example, the exponent γ in the 2D case was calculated in section 4.3 with a master equation using the mean field approach. For larger dimensions, it was verified that the same approach produces the same results in section 4.6. The calculation of the mean distance $\langle l \rangle$ of G1 also used the mean field approach. The analytical results obtained for G1 and G2 confirmed the data from the simulations.

For future work, we could obtain a more systematic way of generating rules in order to completely relate the choice strategies with the obtained characteristics.
Another direction of work is the exploration of higher dimensional simplexes, which we already started in 4.6, by generating rules G1 and G2 for higher dimensions.

Bibliography

[1] Paul Erdős and Alfréd Rényi. On random graphs i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[2] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’networks. *nature*, 393(6684):440–442, 1998.

[3] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[4] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[5] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1):1–101, 2011.

[6] Robert Kleinberg. Geographic routing using hyperbolic space. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 1902–1909. IEEE, 2007.

[7] Marián Boguná, Dmitri Krioukov, and Kimberly C Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2009.

[8] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[9] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[10] G Petri, P Expert, F Turkheimer, R Carhart-Harris, D Nutt, PJ Hellyer, and Francesco Vaccarino. Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface*, 11(101):20140873, 2014.

[11] Luca Donetti et al. Detecting network communities: a new systematic and efficient algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(10):P10012, 2004.

[12] Li Daqing, Kosmas Kosmidis, Armin Bunde, and Shlomo Havlin. Dimension of spatially embedded networks. *Nature Physics*, 7(6):481–484, 2011.

[13] Wei Zeng, Rik Sarkar, Feng Luo, Xianfeng Gu, and Jie Gao. Resilient routing for sensor networks using hyperbolic embedding of universal covering space. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[14] Jan Ambjørn, Jerzy Jurkiewicz, and Renate Loll. Reconstructing the universe. *Physical Review D*, 72(6):064014, 2005.

[15] Diego Meschini, Markku Lehto, and Johanna Piilonen. Geometry, pregeometry and beyond. *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 36(3):435–464, 2005.

[16] Frank Antonsen. Random graphs as a model for pregeometry. *International journal of theoretical physics*, 33(6):1189–1205, 1994.

[17] Tomasz Konopka, Fotini Markopoulou, and Simone Severini. Quantum graphity: a model of emergent locality. *Physical Review D*, 77(10):104029, 2008.

[18] Ginestra Bianconi and Christoph Rahmede. Complex quantum network manifolds in dimension $d > 2$ are scale-free. *Scientific reports*, 5:13979, 2015.

[19] Zhihao Wu, Giulia Menichetti, Christoph Rahmede, and Ginestra Bianconi. Emergent complex network geometry. *Scientific reports*, 5:10073, 2015.

[20] S. Dorogovtsev. *Lectures on Complex Networks*. Oxford Master Series in Physics. OUP Oxford, 2010.

[21] Ciprian Manolescu. Triangulations of manifolds. *ICCM Not*, 2(2):21–23, 2014.

[22] Sergey N Dorogovtsev and José FF Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. OUP Oxford, 2013.

[23] Sergey N Dorogovtsev, José Fernando F Mendes, and Alexander N Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633, 2000.

[24] Pavel L Krapivsky, Sidney Redner, and Eli Ben-Naim. *A kinetic view of statistical physics*. Cambridge University Press, 2010.

[25] Mark Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[26] Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.

[27] SN Dorogovtsev, AL Ferreira, AV Goltsev, and JFF Mendes. Zero pearson coefficient for strongly correlated growing trees. *Physical Review E*, 81(3):031135, 2010.

[28] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633–4636, nov 2000.