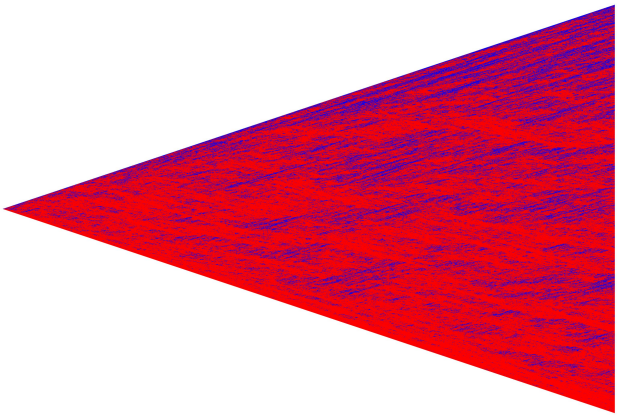
Radboud University Nijmegen
Department of High Energy Physics

Bachelor Thesis

THE GEOMETRY OF INFINITE CAUSAL DYNAMICAL TRIANGULATIONS IN TWO DIMENSIONS

Author:
Ludo van Alst

Supervisor:
Dr. Timothy Budd



July 9, 2018

Abstract

In the early 20th-century, Albert Einstein published his new theory for gravity, which gave a unifying description of space and time, called *spacetime*. Since then, gravity is nicely described for large structures within the universe, but if we probe space and time at smaller and smaller distances, this theory does not provide a good description. One approach to how gravity works on these scales is *Causal Dynamical Triangulation* (CDT). It is known that this model provides science with a good description of spacetime in many ways. An interesting and unexpected result about the geometry of causal dynamical triangulations is found in 2017 by Curien & Hutchcroft and deserves further research. It states that volumes in the two-dimensional version of this model do not scale with distance squared. This thesis contains methods to numerically determine the scaling of volume with respect to distance in the CDT model. Furthermore, a method to generate an infinite two-dimensional CDT is explained. It turns out that one can create this structure by adding two types of triangles in every layer in the CDT, each with probability 1/2. Lastly, it is shown that $\frac{\partial V}{\partial r}$ scales with $\log(r) \cdot r$, where r denotes a distance and V volume. The logarithmic factor is caused by the structure of the boundary of a ball with radius r.

Contents

Chapter 1

Introduction

From the 17th century on, gravity was described by a theory in which two masses induce a force on one another. This theory is written down in the *Philosophiæ Naturalis Principia Mathematica* by Isaac Newton. Approximately 150 years later, in 1915, another well-known physicist called Albert Einstein published his model for gravitation, called *General Relativity*. It provides a unifying description of space and time, which is referred to as *spacetime*. Masses determine the structure of spacetime, they curve it. This curvature is then directly related to many physical phenomena happening in the universe. One of those phenomena is that in a curved spacetime, light does not travel in a straight line.

As of today, there are four fundamental forces known to exist: the electromagnetic force, strong interactions, weak interactions and gravitational force. The first three forces can be described in a quantum theory, the framework of quantum mechanics and quantum field theory. Unfortunately, the last fundamental force, gravity, cannot be explained in a quantum framework (yet). This is because the general theory of relativity does only describe gravity on large scales. This problem is arguably the most fascinating in all of physics. The area in physics which is determined to solving the quantum gravity problem is called, no surprise, *Quantum Gravity*.

Within this area there are several approaches to come up with a solution to the problem. These include string theory, loop quantum gravity, effective field theory and causal dynamical triangulation. The latter is an approach which is used

frequently at Radboud University and it is also the one used in this thesis. Before causal dynamical triangulation will be explained in more detail, the mathematical foundation of quantum gravity will be highlighted.

An important property in all such theories describing quantum gravity is the Hausdorff dimension. For a well-defined theory, at scales much larger than the Planck scale, the Hausdorff dimension should be equal to the dimension of the spacetime you are working in. Which means that at this scale, the Hausdorff dimension of a (1+1)-dimensional spacetime should be approximately 2. For the model of causal dynamical triangulation it has already been proven that the Hausdorff dimension is almost surely equal to 2 [7]. The more realistic version of spacetime has (3+1) dimensions[1], hence the (1+1)-dimensional CDT is just a toy model to get a better understanding of the CDT model itself.

This thesis is written to get a better understanding of the geometry of spacetime on the smallest scales. More specifically, Curien et al. have shown [6] that distances within a two-dimensional CDT scale in a way which was not expected. Namely, volume scales with a larger factor than distance squared, which means that there is no exact power-law scaling. It will become clear that this is caused by quantum corrections. The correction factor will be further examined in this thesis and estimated using Monte Carlo simulations.

1.1 Path integral

An important tool to quantise a classical system is the so called *path integral*. This rather formal object is used to describe the quantum dynamics of some physical system by superposing all possible configurations of the system and giving each configuration a certain weight (quantum amplitude). When applying this tool to gravity, we speak of the *path integral of gravity*. Before explaining the details of the gravitational path integral, a more familiar and analogous object will be studied, namely the path integral for a free particle. Say a particle is in some initial state $|\psi(x_i, t_i)\rangle$ and wants to transit to some final state $|\psi(x_f, t_f)\rangle$, where (x, t) denotes a specific point in spacetime. When using the time evolution operator $\hat{U}(t_f, t_i)$,

[1] Three spatial dimensions and one time dimension.

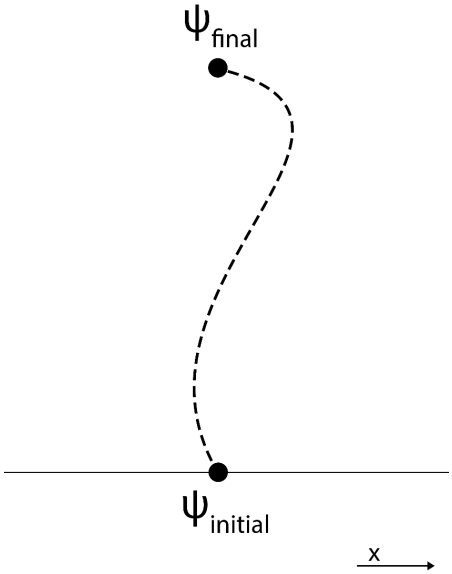


Figure 1.1: A free particle travelling in a two-dimensional spacetime.

one can write this transition in the following way:

$$|\psi(x_f, t_f)\rangle = \int_{-\infty}^{+\infty} dx_i \left[\hat{Z}(x_f, t_f; x_i, t_i)\,|\psi(x_i, t_i)\rangle\right],$$

where the propagator $\hat{Z}$ is

$$\hat{Z}(x_f, t_f; x_i, t_i) = \langle x_f|\,\hat{U}(t_f, t_i)\,|x_i\rangle\,.$$

The path integral $\hat{Z}$ is sometimes referred to as the propagator. It contains the transition amplitude from (x_i, t_i) to (x_f, t_f). Furthermore, because the particle could have been in any initial position, one has to integrate over all possible x_i to describe quantum state $|\psi(x_f, t_f)\rangle$. The time evolution operator in the path integral can be expanded further when concerning time steps of length Δt in between t_i and t_f. Then all possible paths can be approximated by the straight paths a particle can take in between time layers $t_i,\ t_i + \Delta t,\ t_i + 2\Delta t,\ \ldots,\ t_f$. When finally taking a continuum limit ($\Delta t \to 0$) one obtains the true propagator. [5]

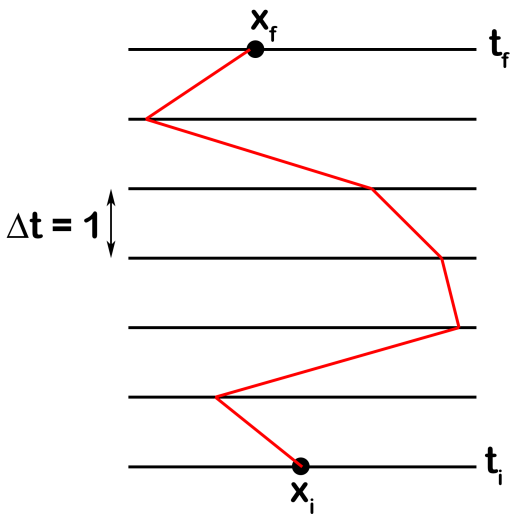


Figure 1.2: A free particle travelling in a two-dimensional spacetime where time is discretised.

This propagator can be written as

$$\hat{Z}(x_f, t_f; x_i, t_i) = \int_{x(t_i)=x_i}^{x(t_f)=x_f} \mathcal{D}[x(t)] e^{\frac{i}{\hbar} S(t_f, t_i)} dx. \tag{1.1}$$

Where $x(t)$ denotes the path of the particle, $\mathcal{D}[x(t)]$ contains all possible paths from initial position to final position in spacetime and the complex powers of e are the quantum amplitudes assigning a weight to each path. The S is the action which in this case depends on the initial and final time of the particle:

$$S(t_f, t_i) = \int_{t_i}^{t_f} dt \left[\mathcal{L}(x(t), \dot{x}(t))\right].$$

Here $\mathcal{L}$ denotes the Lagrangian. Note that all of the previous notions are solitary for a free particle. In quantum gravity, it is not some path taken by a free particle that is important, but spacetime itself. The curvature of spacetime is captured by spacetime geometry $g_{\mu\nu}$. This tensor is one very important tool in general relativity and can also be studied using a path integral. The state of the system is now captured by a spacetime geometry instead of the position of a particle. The dynamics of spacetime geometries can then be captured in the path integral in the same way as before: [3]

$$\mathcal{Z} = \int \mathcal{D}[g_{\mu\nu}] e^{\frac{i}{\hbar} S_{\text{EH}}[g_{\mu\nu}]}. \tag{1.2}$$

Where this time, the action in a four-dimensional spacetime is the so called Einstein-Hilbert action

$$S_{\text{EH}}[g_{\mu\nu}] = \frac{1}{16\pi G}\int d^4x \left[\sqrt{-g}(R-2\Lambda)\right].$$

Where Λ denotes the cosmological constant, R the Ricci scalar and $g = \det(g_{\mu\nu})$. This path integral then acts the same way as the path integral does for a free particle (formula (1.1)). In this way, the propagator describes the transition amplitude between two different geometries. Using this propagator, expected values for observables $\mathcal{O}$ in the geometry can be obtained with

$$\frac{\int \mathcal{D}[g_{\mu\nu}]\ \mathcal{O}[g_{\mu\nu}]\ e^{\frac{i}{\hbar}S_{\text{EH}}[g_{\mu\nu}]}}{\mathcal{Z}}. \tag{1.3}$$

Here observable $\mathcal{O}[g_{\mu\nu}]$ assigns a value to a spacetime geometry $g_{\mu\nu}$. Before, in the free particle example, $\mathcal{D}$ was a factor which contained all possible paths between the initial and final position. Now, this quantity contains all possible transitions between some initial spacetime geometry and some final spacetime geometry. The question that arises is what this means and how one can calculate this.

1.2 Causal Dynamical Triangulation

In 1998, R. Loll and J. Ambjørn invented the model of Causal Dynamical Triangulation, CDT for short [4]. It is one of the approaches to the nonperturbative quantisation of gravity. The model should give a rigorous definition of the path integral over all causal geometries.

In the model of CDT, spacetime is triangulated in such a way that causality is preserved, there is a direction of time. In n-dimensional spacetime the name *triangle* can be misleading, instead we then speak of *simplices* which are the n-dimensional equivalent of triangles. These simplices build up spacetime in such a way that at most one of the (n-1)-dimensional sides of the simplex is spacelike and all others are timelike. All edges of the simplices are of length one.

Another model which tries to discretise spacetime is called Euclidean Dynamical Triangulation. As the name implies, this method also triangulates spacetime, but it does not preserve causality. In the two-dimensional variant of this model, the

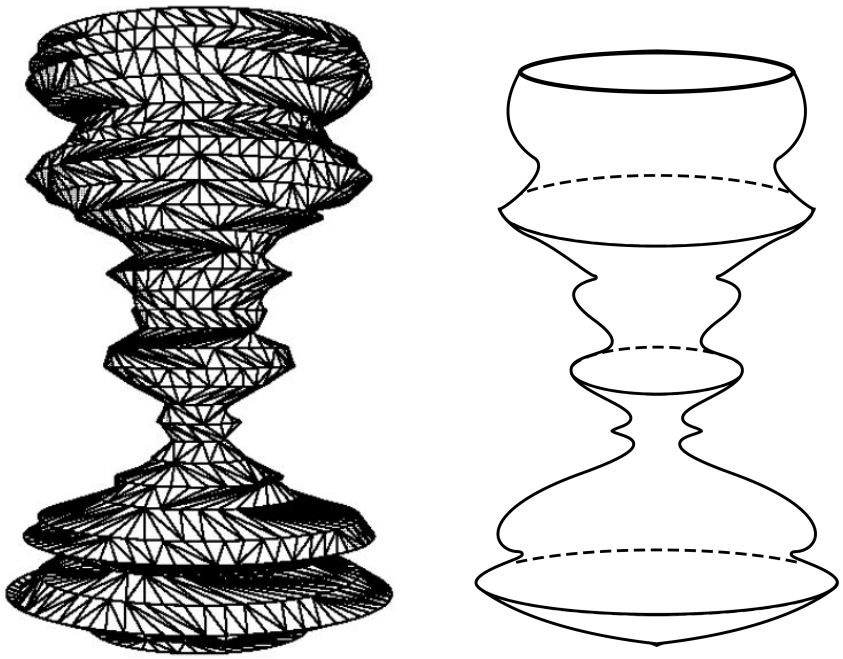


Figure 1.3: A CDT with its corresponding spacetime geometry. The picture on the left is taken from [2].

Hausdorff dimension is not as one would expect it to be, equal to 2. Instead, it turns out to be equal to 4 [4], which means that the path integral does not have a classical limit.

In this thesis we will only work with (1+1)-dimensional spacetime geometries, such that the simplices in the CDT used are just triangles. Figure 1.3 illustrates a two-dimensional CDT in a cylindrical shape. The horizontal lines in the CDT are called time layers (all points in that layer have equal time). Causality is then preserved because time increases upwards in the CDT. Distances within a CDT are described by the usual graph distance, which means that the distance between two vertices is the minimum length of the paths connecting them.

The curvature of spacetime is then described by the way the triangles are glued together. Keep in mind that all edges are of equal length, even if it does not look like this in figure 1.3. One has to add up all angles of triangles around a certain vertex. If these angles add up to 2π, there is locally flat curvature. If the angle is positive, then the resulting curvature at the vertex is positive and vice versa.

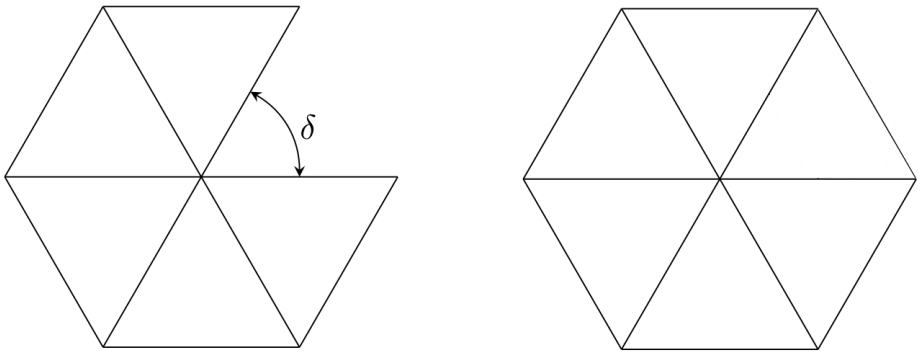


Figure 1.4: On the left there is a positive angle between two edges. On the right there is no angle between two edges. After gluing together the edges one obtains positive (respectively zero) curvature at the central vertex.

When the CDT is constructed in the end, we are still left with a discrete geometry, where on the surface of a triangle spacetime is locally flat. While scaling the size of individual triangles to zero, continuous spacetime is obtained.

1.3 Outline

In 2017, Curien et al. published a paper [6] containing non-trivial properties of two-dimensional CDTs. To understand what they have proven, we have to go through a few definitions. A Galton-Watson tree (GWT) is a tree (in the mathematical graph sense) in which there is some probability distribution $p(n)$ which determines the amount of children a vertex has, start with one vertex called the root. We call such a GWT *critical* if the expected value of the probability distribution is one. The *height* of a vertex v in the GWT is then the graph distance between v and the root. A GWT can be 'made' causal by adding horizontal edges between vertices which are at equal height.

Curien et al. have proven the following:

$$\frac{\text{width}_t(\mathcal{C}_\infty)}{t} \xrightarrow[t\to\infty]{\mathbb{P}} 0. \tag{1.4}$$

Where $\mathcal{C}_\infty$ denotes the set of causal critical Galton-Watson trees which survive

forever. The width is defined as

$$\text{width}_t(\tau) \equiv \max\{d(x, y) \mid x, y \in \delta[\tau]_t\}.$$

Where d denotes the graph distance within causal critical Galton-Watson tree τ which survives forever. $\delta[\tau]_t$ is the set of vertices in τ at height t.

A CDT contains one extra edge for almost every vertex in the Galton-Watson tree, which implies that convergence (1.4) also holds if we replace the GWTs by CDTs where the amount of vertices in the CDT approaches infinity. This can be interpreted as volume in this kind of CDT that increases with some larger factor than distance squared. Which at its turn implies that the Hausdorff dimension seems larger than 2 on a finite scale. The strategy is to simulate CDTs and determine the Hausdorff dimension using measures for volume and distance. In this thesis methods will be explained to numerically determine the speed of the convergence. This is done by first defining a volume and a distance within a CDT and generating CDTs uniformly at random. The defined quantities are then stored and analysed.

Chapter 2

The geometry of a finite two-dimensional CDT

To investigate the structure of CDTs, one first has to think about a method to generate a CDT using a computer. This chapter focuses on CDTs with finite size, how to generate them and how large the CDT needs to be before the result in formula (1.4) is seen. A finite causal dynamical triangulation is illustrated in figure 2.1.

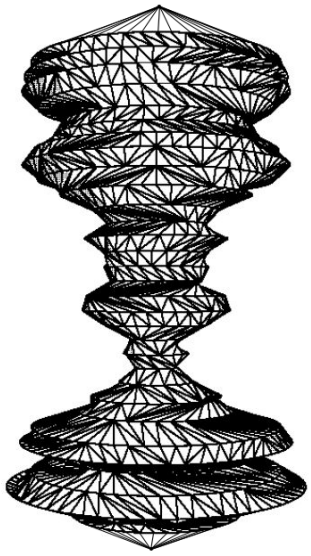


Figure 2.1: A finite CDT on the topology of a cylinder. [2]

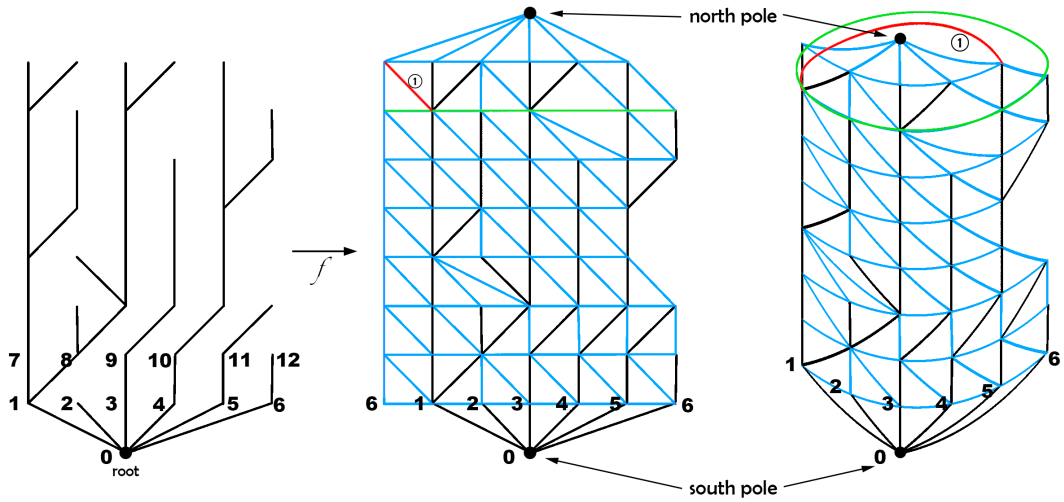


Figure 2.2: An illustration of the process to find the CDT corresponding to a plane tree. The tree shown on the left has already been relabelled.

2.1 Creation of a finite two-dimensional CDT

A two-dimensional causal dynamical triangulation in a cylindrical shape $S^1 \times [0, \mathcal{H}']$ has a correspondence with plane trees from graph theory. Plane trees differ from 'normal' trees by the spatial arrangement being fixed in the former case. We need plane trees in this case because the arrangement of the vertices influences the final structure of the CDT.

To simulate a random $S^1 \times [0, \mathcal{H}']$ CDT, a bijection [10] between plane trees and CDTs of the above form is used. Denote the set of plane trees with N vertices by T_N and the set of CDTs with $N+1$ vertices by $\mathcal{C}_{N+1}$. Denote the bijective function by

$$f : T_N \to \mathcal{C}_{N+1}. \tag{2.1}$$

2.1.1 Parentheses trees

A way to generate uniformly random plane trees is described by Knuth [9, algorithm W]. This algorithm generates a random ordered sequence $a_1, a_2, \ldots, a_{2n}$ of properly nested parentheses. From this list of (s and)s we can create the corre-

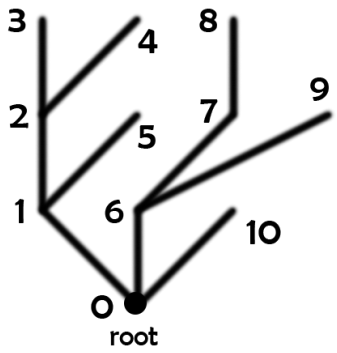


Figure 2.3: An illustration of parentheses tree (((()())())(((())())()) with its labels.

sponding labelled tree with $n+1$ vertices as follows. Start at the root of the tree and name this vertex $v_{\text{root}} = v_0$. All vertices have their label in subscript, label i runs from 1 to n such that label $l(v_i) = i$. Let k run from 1 to $2n$ and study a_k, the kth element in the sequence. If $a_k =$ '(', create a new vertex v_i and create an edge which connects v_i to the vertex which was just considered and is one layer below. If $a_k =$ ')', go down in the tree to the previous vertex. For example, list (((()())())(((())())()) results in the plane tree in figure 2.3.

The probability distribution of the nested parentheses is shown in the algorithm of Knuth. This algorithm works as follows. The arrows pointing to the left indicate that we give the variable a new value: $variable \leftarrow value$.

1. Start by defining three variables: $p \leftarrow n$, $q \leftarrow n$ and $m \leftarrow 1$.
2. $q = 0$? Yes $\rightarrow$ terminate the algorithm.
3. Let X be a random integer in the interval $[0, (q+p)(q-p+1)]$. Now there are two possibilities.
   (a) $X < (q+1)(q-p)$. Then set $q \leftarrow q-1$, $a_m \leftarrow$ ')'.
   (b) $X \geq (q+1)(q-p)$. Then set $p \leftarrow p-1$, $a_m \leftarrow$ '('.
4. End with $m \leftarrow m+1$ and return to step 2.

Using this algorithm, all plane trees with $n+1$ vertices have equal probability to be chosen.

2.1.2 Bijection between parentheses trees and CDTs

Bijective function (2.1) can be explained in words solely, but a clear mathematical description helps to clarify the programming that is done to create CDTs uniformly at random.

First some definitions are given. Take $\gamma \in T_N$ and denote $\gamma = (V, E)$ where V is the set of vertices and E is the set of edges, together describing (labelled) plane tree γ. Arrange the tree to the way it was generated using the method described in paragraph 2.1.1. This defines the spatial arrangement of the tree. Denote the root-vertex by $v_{\text{root}} = v_0$ and define $w_t = \{v \in V \mid d(v, v_0) = k\}$ where d is the usual graph distance function. This is the set of vertices at 'height' t, thus distance t from the root. Denote the height of the tree by $\mathcal{H} = \max\{t \mid w_t \neq \emptyset\}$. Relabel the vertices in the following way: let t run from 0 to $\mathcal{H}$. Then label the vertices in w_t from 'left' to 'right' as seen in figure 2.2.

Function $f : T_N \to \mathcal{C}_{N+1}$ is defined in the following way: add one vertex $v_{N+1} = v_{\text{np}}$ at height $\mathcal{H}' = \mathcal{H} + 1$: $V_G = V \cup \{v_{\text{np}}\}$, this vertex is called the *north pole*[1]. For every vertex $v \in V$: add two edges $e_{l(v),1}$ and $e_{l(v),2}$ to E. To know which edges to add, one more set has to be defined: $Q_t(v) = \{l(v_*) \mid v_* \in w_t \text{ and } l(v_*) < l(v)\}$. This set contains all labels at height t of vertices with a lower label than the label of vertex v. Now for every vertex $v \in V$ we want to add the following edges:

- If $Q_t(v) = \emptyset$, then set $v_{\text{neighbour}} = \max\{l(v) \mid v \in w_t\}$, otherwise $v_{\text{neighbour}} = \max(Q_t(v))$. This is the neighbouring vertex to the left at height t. $e_{l(v),1} = \{v, v_{\text{neighbour}}\}$;
- If $Q_{t+1}(v) = \emptyset$, then set $v_{\text{up}} = \max\{l(v) \mid v \in w_{t+1}\}$, otherwise $v_{\text{up}} = \max(Q_{t+1}(v))$. This is the vertex one height deeper and left of the branch in which v is located. $e_{l(v),2} = \{v, v_{\text{up}}\}$.

$E_G = E \cup \{e_{m,n} \mid 1 \leq m \leq N \text{ and } 1 \leq n \leq 2\}$ such that $\mathcal{C}_{N+1}$ is the set of all pairs (V_G, E_G) where V_G and E_G are constructed in the above described manner.

Now $f : T_N \to \mathcal{C}_{N+1}$ given by $f((V, E)) = (V_G, E_G)$ is a bijective function. Indeed, take any CDT (V_G, E_G) with $N + 1$ vertices, go through the steps given above in reverse to find the corresponding plane tree with N vertices, hence f is a surjective

[1] the root vertex is sometimes referred to as the *south pole*.

function. If two CDTs with N vertices are the same, $(V_{G_1}, E_{G_1}) = (V_{G_2}, E_{G_2})$, then clearly by the above construction the corresponding plane trees are the same, hence f is an injection and thus bijective.

Note again that the construction of the function f is worked out to make clear how the simulation is programmed. The actual bijection between the set T_N and $\mathcal{C}_{N+1}$ can best be understood when looking at figure 2.2. In the algorithm above, edges are added between all vertices at certain height. This creates the time layers of the CDT. Furthermore, the triangles are created by the other edge which is added in the process.

2.2 Results

Using this method to generate a CDT with n vertices uniformly at random, statement (1.4) can numerically tested. Furthermore, since finite CDTs are used in this chapter, one has to be careful when studying the convergence. It could be that the height of the tree is less than the height t at which we would like to evaluate the width. To make up for this, we always evaluate the width at fractions of the height of the CDT. As the size of the structure increases, the t will of course also increase and thus we can study the convergence. Also, the t in the denominator is replaced by $\sqrt{n}$ where n denotes the amount of vertices in the CDT. This quantifies the volume of the system, hence we can write distance t as the square root of this volume.

The width as defined by Curien et al. is replaced by the diameter, which is the maximum distance from a randomly chosen vertex to any other vertex in the same layer of time. More formally:

$$\mathrm{diam}(v_*) \equiv \max\{d(v_*, v) \mid v \in w_*\}.$$

Where w_* is a set containing all vertices which have the same height as vertex v_*. This diameter scales in the same way the width does, but it takes way less time for the computer to determine, this is the reason this quantity is used. Now it is expected that the convergence relation is equivalent to

$$\frac{\mathrm{diameter}_t}{\sqrt{n}} \xrightarrow[n\to\infty]{\mathbb{P}} 0. \tag{2.2}$$

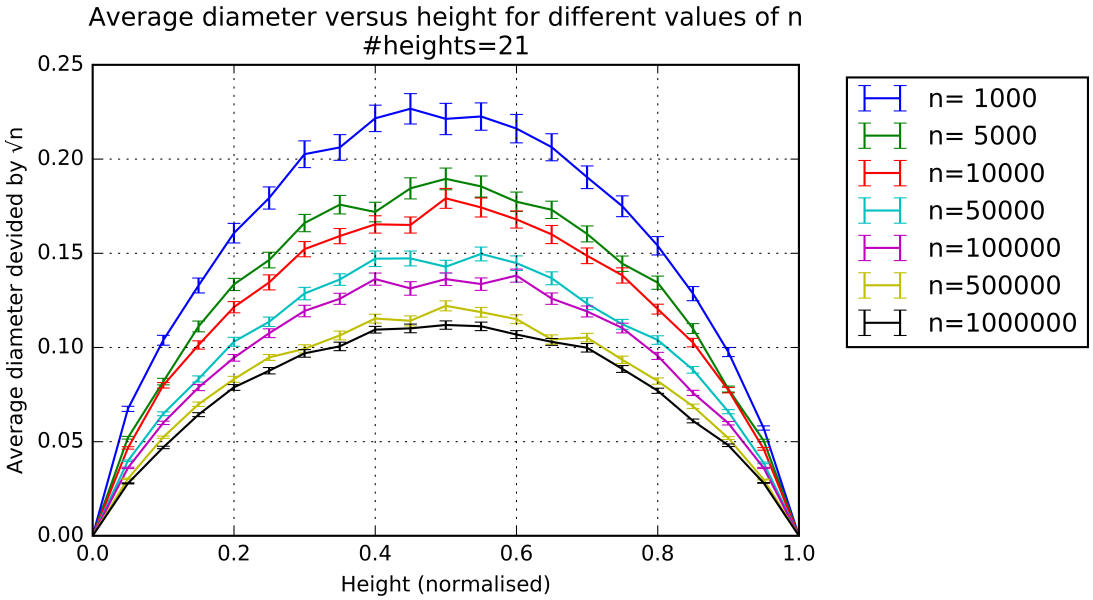


Figure 2.4: A plot of average width divided by $\sqrt{n}$ versus normalised height.

A plot is made of diameter$_t/\sqrt{n}$ versus normalised height. As can be seen in figure 2.4, the expected convergence is present. To get an idea of the speed of convergence, the diameter distributions for different values of n are collapsed. The collapse will then tell us the power at which volume n will scale with the average diameters:

$$\text{average diameter} \sim n^{\beta}.$$

As can be seen in the collapse in figure 2.5, the β for which the distributions collapse is 0.405. A hypothesis for the speed of convergence is $\log(n)^{\alpha}$, such that[2]

$$\frac{\text{average width}}{\sqrt{n}} \sim \log(n)^{\alpha}. \tag{2.3}$$

Of course, $\alpha < 0$ because we know that for $n \to \infty$ the term on the left-hand side converges to zero. $\alpha = -1$ would be the most natural value for α, there could be an explanation for this natural scaling. To determine the value for α, a log-log plot is made in which α will be the slope of the graph:

$$\log\left(\frac{\text{average width}}{\sqrt{n}}\right) \sim \alpha \cdot \log\log(n).$$

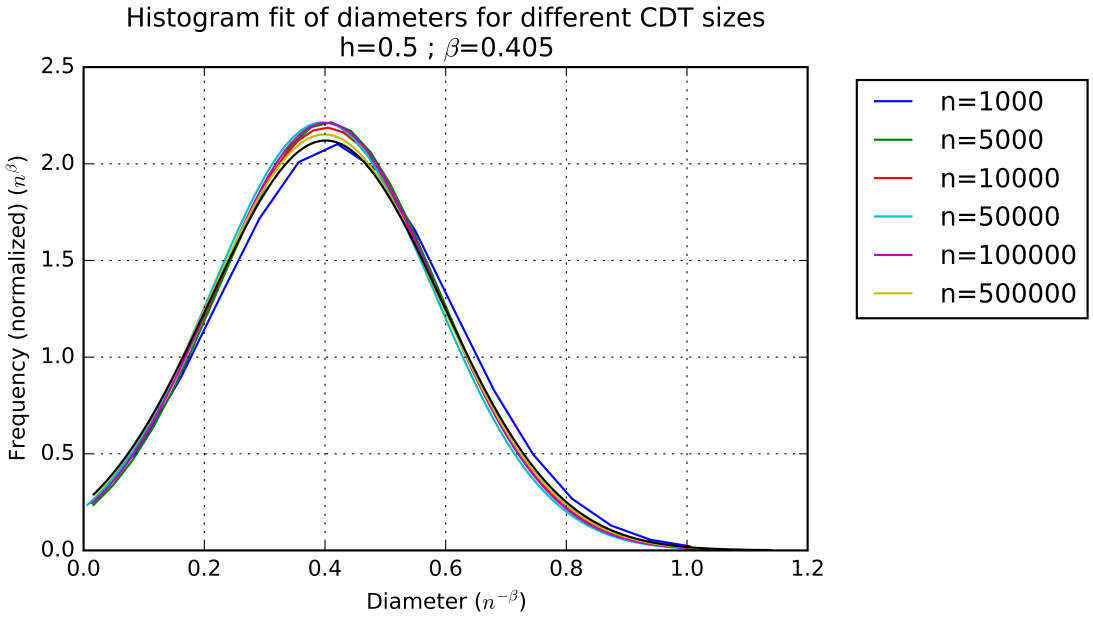


Figure 2.5: A collapse of the diameter distributions for different values of n. The β for which we see a collapse is 0.405.

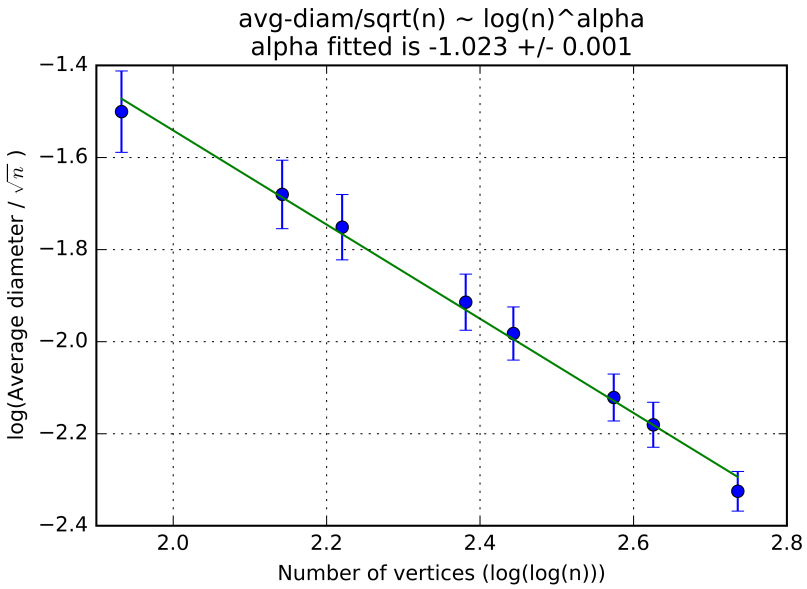


Figure 2.6: A plot of the log of the average diameter divided by $\sqrt{n}$ versus $\log\log(n)$.

When a fit is made through the data points we get $\alpha = -1.023 \pm 0.001$ as can be seen in figure 2.6. Unfortunately, $\alpha = -1$ is not within the error margin. One of the problems in this method is that the diameters stay below 150 most of the time and the convergence happens when this distance approaches infinity. One needs to generate a CDT in which large distances can be reached with ease.

A possible solution would be to generate the infinite structure from the start. To do this, one first has to define an infinite CDT and prove that this corresponds to a finite CDT where we let the number of vertices approach infinity. This definition is explained in the next chapter and we will see that it is quite easy to study very large systems with just few memory usage by the simulation.

[2]The *log* used is the natural logarithm, sometimes denoted by ln.

Chapter 3

The geometry of an infinite two-dimensional CDT

As discussed in chapter 2, the distances that are reached when simulating finite CDTs stay quite small. When a CDT is generated which contains one million vertices, the distances mostly stay between 100 and 150. One would have to go to very large systems to say something about the speed of convergence (1.4). A possible solution is explained in this chapter where we will generate the infinite structure from the start. First one needs to know how to randomly generate such infinite structure. The most important thing is that this definition of an infinite causal dynamical triangulation has to correspond to the limit $n \to \infty$ for a finite CDT. Any observable has to behave the same in both structures. We will see that the creation is actually really simple. Namely, starting with an initial setup, one can generate two kinds of triangles with equal probability between any two heights $t = T$ and $t = T + 1$.

3.1 Creation of an infinite two-dimensional CDT

Since the size of the CDT is infinite, where do you start generating the triangles? From one vertex in the triangulation, one can divide the two-dimensional CDT in four parts (see figure 3.1). Call this one vertex the *origin* $\mathcal{O}$ which we define to be located at $t = 0$ and $x = 0$. Define the *rightmost path* as follows. All paths from

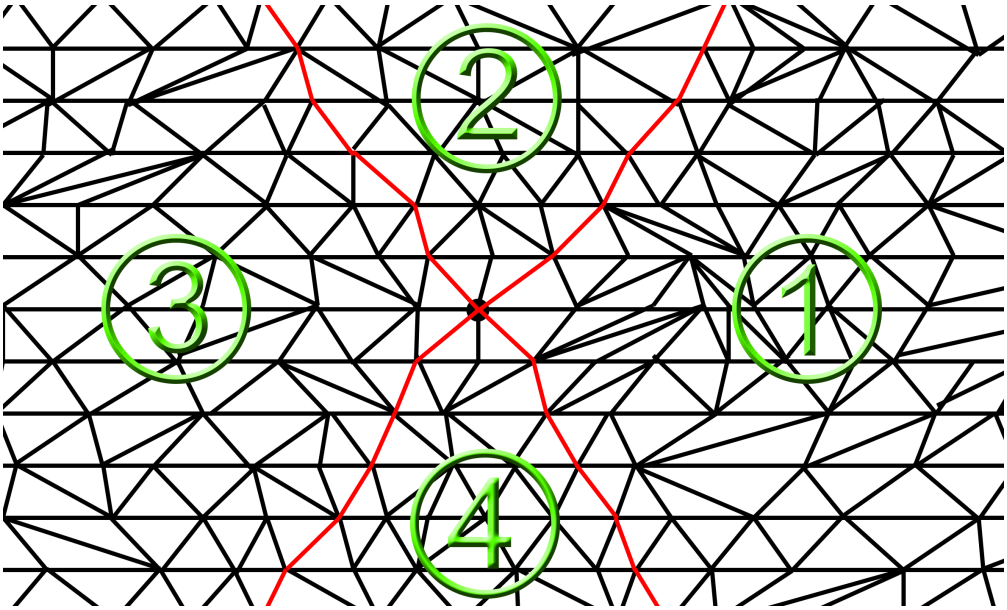


Figure 3.1: An infinite two-dimensional CDT divided in four areas. Area one is the one which is generated and then studied.

the origin to vertices in layer $t = 1$ are considered, then pick the path to the vertex with largest x value. Keep repeating this process for the new vertex. This can also be done for lowest x value and both can also be done downwards. This defines the four separated sections. Part one is bounded by the rightmost path going up and the rightmost path going down, this is the area which is simulated.

Denote the ordered list of triangles in between time layer t and $t+1$ by $(D_n(t))_{n=0}^{\infty}$, starting at the red boundary of area 1 in figure 3.1. There are two types of triangles present. A triangle can have two vertices in time layer t, we call these triangles *pointing up*. The other type has two vertices in time layer $t+1$, we call this type *pointing down*. Then for example in figure 3.1, we have $(D_n(0))_{n=0}^{\infty} = (\Delta, \nabla, \nabla, \Delta, \Delta, \Delta, \Delta, \ldots)$. Furthermore, one important observation has to be done in area 1, this will be used later. In area 1:

$$D_0(t) = \begin{cases} \Delta & \forall t \in \mathbb{Z}_{\geq 0} \\ \nabla & \forall t \in \mathbb{Z}_{<0}. \end{cases}$$

From this point on one would expect the two kinds of triangles to be created

with equal probability in each layer. Such that when simulating the system, the computer can pick one of these two possibilities each with probability 1/2. This is actually the case and a proof of this statement is given in the next section.

3.1.1 Bijection between unrooted plane trees and CDTs

To prove the triangles can be created each with probability 1/2, one has to do several things. First, the general setup will be explained, then the actual proof is given. Define $T_N^{\cdot} \equiv \{$causal triangulations with N triangles with three marked points, north and south pole and a vertex, the origin, not equal to a pole$\}$. Now pick $\tau_N^{\cdot} = (V, E) \in T_N^{\cdot}$ uniformly at random. The usual definition for graph distance is used. For any vertex $v \in V$, write d_v for the graph distance from the origin to this vertex. Further, write t_v for the time layer in which vertex v is located relative to the origin. Define the ball of radius r around the origin $\mathcal{O}$ as $\mathrm{B}_r(\tau_N^{\cdot}) \equiv \{$triangles incident to at least one vertex v with $d_v < r\}$.

Now in the local limit ($N \to \infty$), we wish to show that the finite CDT converges to an infinite CDT. More formally, let b be any finite triangulation with boundary, then

$$\lim_{N\to\infty} \mathbb{P}(\mathrm{B}_r(\tau_N^{\cdot}) = b) = \mathbb{P}(\mathrm{B}_r(\tau_\infty^{\cdot}) = b). \tag{3.1}$$

Instead of considering balls of radius r, it will be easier to use "diamonds" of radius r: $\Lambda_r(\tau_N^{\cdot}) \equiv \{$triangles incident to at least one vertex v with $d_v + |t_v| < 2r\}$. Since $\mathrm{B}_r(\tau_N^{\cdot}) \subseteq \Lambda_r(\tau_N^{\cdot})$, it is sufficient to prove that

$$\lim_{N\to\infty} \mathbb{P}(\Lambda_r(\tau_N^{\cdot}) = b) = \mathbb{P}(\Lambda_r(\tau_\infty^{\cdot}) = b). \tag{3.2}$$

To prove the latter, we will first find a bijection between finite CDTs and unrooted plane trees. Define $\mathcal{G}_N^{\cdot\cdot} \equiv \{$unrooted plane trees with two marked points v_1 and v_2 and N edges$\}$.

Lemma 1. *There exists a bijection $f : T_N^{\cdot} \to \mathcal{G}_N^{\cdot\cdot}$.*

An illustration of what happens is given in figure 3.2.

Proof. Pick $\tau_N^{\cdot} \in T_N^{\cdot}$ and write $\tau_N^{\cdot} = (V, E)$ where V is the set of vertices and E the set of edges. This $\tau_N^{\cdot}$ contains one marked point not equal to one of the

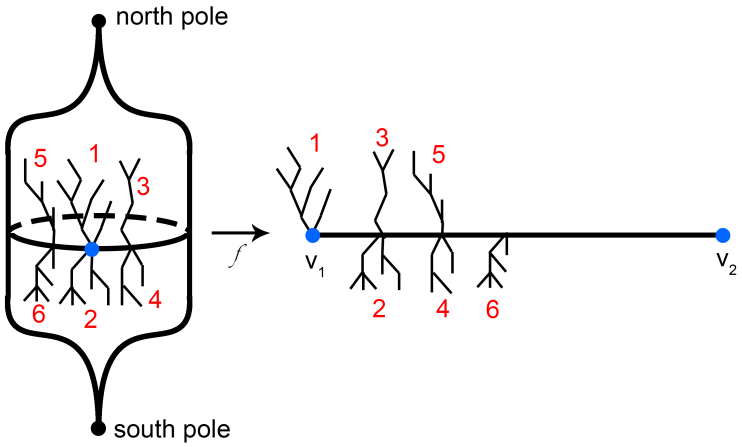


Figure 3.2: An illustration of the bijection from a CDT to a plane tree with two marked points.

poles. This marked point is the origin $\mathcal{O}$ of the CDT. Divide the CDT into two (overlapping) sections: one section $\tau_{N,1}^{\cdot} = (V_1, E_1)$ where $V_1 = \{v \in V \mid t_v \geq 0\}$ and $E_1 = \{\{v, v_*\} \in E \mid t_v, t_{v_*} \geq 0\}$, the other section $\tau_{N,2}^{\cdot} = (V_2, E_2)$ is defined similarly, just replace the $\geq$ signs by $\leq$ signs. Note that these sections *do* overlap, thus $\tau_N^{\cdot} \subset \tau_{N,1}^{\cdot} \cup \tau_{N,2}^{\cdot}$.

Now denote the origin $\mathcal{O}$ in the new section by v_1 in $\tau_{N,1}^{\cdot}$ and v_2 in $\tau_{N,2}^{\cdot}$. Remove all edges where both vertices lay in the same time layer, except for $t = 0$, thus all edges $\{v, v_*\}$ for which $t_v = t_{v_*} \neq 0$. Also, for every vertex $v \in V$ except for the north and south pole, remove the leftmost edge to a vertex $v_* \in V$ with $t_{v_*} = t_v + 1$. Lastly, remove the previously called north pole from $\tau_{N,1}^{\cdot}$ and the south pole from $\tau_{N,2}^{\cdot}$.

Now we are left with some spine at $t = 0$ where every vertex v for which $t_v = 0$ is the root of a tree (in both sections separately). Label those trees in an alternating way as shown in figure 3.2.

Make a spine of length l where $l = \#\{v \in V \mid t_v = 0\}$ and at the end points of this spine we put v_1 and v_2. Now the labelled trees can be arranged on this spine

in the manner illustrated in the figure.

This uniquely determines the corresponding tree $g_N^{\cdot\cdot} \in \mathcal{G}_N^{\cdot\cdot}$. From this $g_N^{\cdot\cdot}$ we can reverse these steps to get the corresponding $\tau_N^{\cdot} \in T_N^{\cdot}$. □

Now we know how to unwind the CDT to a plane tree, we can look at the diamond structure. To study $\mathbb{P}(\Lambda_r(\tau_N^{\cdot}) = b)$ (write $\tau_N^{\cdot} = (V, E)$), one has to define the *span* w of the diamond. Start by defining some new quantities:

$$n_{(\geq)} \equiv \#\{v \in V_b \mid t_v = 0 \text{ and } x_v \geq 0\};$$
$$n_{(\leq)} \equiv \#\{v \in V_b \mid t_v = 0 \text{ and } x_v \leq 0\}.$$

Then the span will be $w = \max\{n_{(\leq)}, n_{(\geq)}\}$. Figure 3.3 illustrates what is happening in the next part.

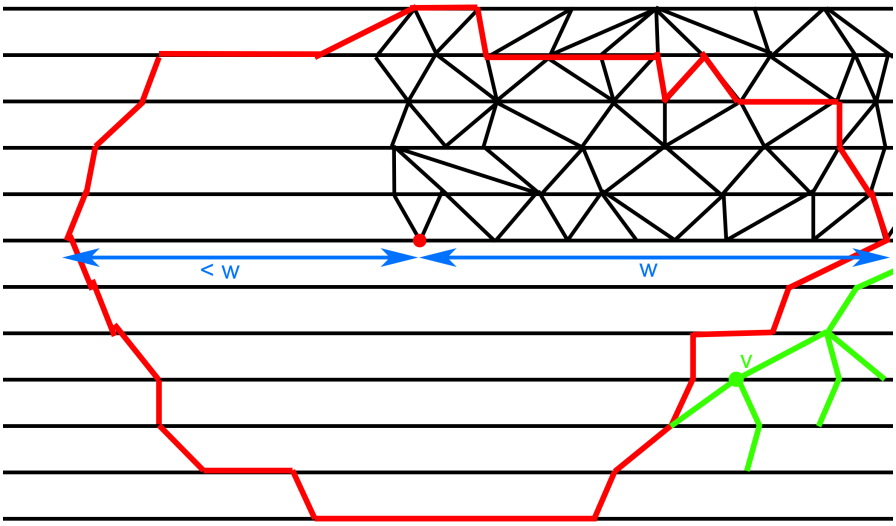


Figure 3.3: An illustration of a diamond b. The green branch has its root outside the diamond.

Lemma 2. *A diamond b with origin $\mathcal{O}$ and span w contains vertices belonging to at most the first $4w$ trees in the plane tree.*

Proof. To see this, one has to argue that all other trees which have their root on the spine do not share vertices with diamond $b = (V_b, E_b)$. Diamond b has total

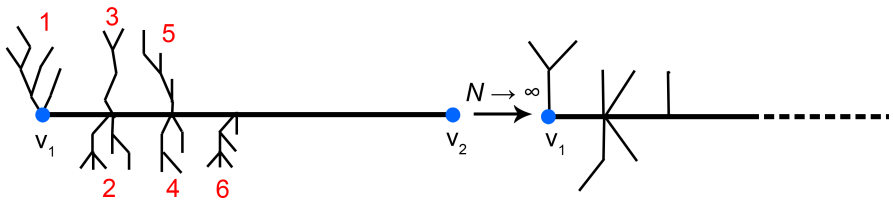


Figure 3.4: An illustration of the bijection between unrooted plane trees and a Galton-Watson tree.

span width $n_{(\leq)} + n_{(\geq)} - 1 \leq 2w$ and thus has at most $4w$ outgoing trees, on every vertex two. The placement of the trees in lemma 1 is essential, we know for sure that at most the first $2w$ roots are are contained in the diamond.

Assume there is a tree, further than the first $4w$ trees which has a branch inside diamond b, then there is an edge between a vertex v_* *in* the diamond and one vertex v *outside* the diamond. Because the root of the tree which is considered is further than $2w$ from vertex v_1, we know v is one step closer to $t = 0$ than v_*. We also know that the graph distance from the origin to vertex v outside the diamond is at most the graph distance from the origin to vertex v_* plus one.

More formally, we will prove that under the above conditions, v will be inside the diamond. Since $v_* \in V_b$, we know $d_v + |t_v| < 2r$. Also, there is an edge with end points v and v_*, thus the graph distance to the origin can at most differ by one. v is one layer closer to $t = 0$, which means that $|t_v|$ is one less than $|t_{v_*}|$. Thus $d_{v_*} + |t_{v_*}| \leq d_v + 1 + |t_v| - 1 < 2r$. Hence $v \in V_b$. □

3.1.2 Local limit in unrooted plane trees

Now for the last part we need $\mathcal{G}_N^{\cdot\cdot}$ to converge in the local limit ($N \to \infty$) to an infinite tree with an infinite spine from which Galton-Watson trees (GWTs) grow. The GWTs have offspring distribution $p_n = \mathbb{P}(\#\text{ children} = n) = 2^{-n-1}$. This convergence is beyond the scope of this thesis, but a paper by Abraham and Delmas [1] and one by Kennedy [8] roughly imply the convergence in the local limit.

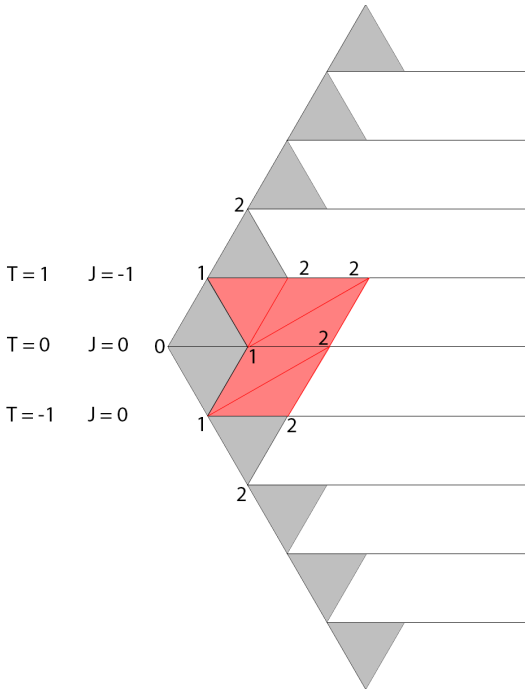


Figure 3.5: The field in which the infinite CDT structure is created is illustrated. The starting point is illustrated when only considering the grey triangles. The rest is created until we arive at a ball of radius 2.

Each of the trees growing from the spine are then independent and identically distributed as a Galton-Watson tree with the above offspring distribution.

Since the diamond only depends on the first $4w$ trees on the spine, formula (3.1) is valid. To generate an infinite CDT we can use the Galton-Watson tree and reverse all steps to arrive at the infinite CDT. The final thing that is left to prove is that the two types of triangles are created each with probability $1/2$.

Lemma 3. $\mathbb{P}(D_i(t) = a_1 \mid D_{i-1}(t) = a_2) = 1/2$ *for every* $t \in \mathbb{Z}$, $i \geq 1$, $a_1, a_2 \in \{\Delta, \nabla\}$.

Proof. First assume $D_{i-1}(t) = \Delta$. Evaluate the Galton-Watson tree with offspring distribution $p_n = 2^{-n-1}$ at a vertex v on the spine. Then

$$\begin{aligned}\mathbb{P}(D_i(t) = \nabla \mid D_{i-1}(t) = \Delta) &= \mathbb{P}(v \text{ has} \geq 1 \text{ child}) \\ &= 1 - \mathbb{P}(v \text{ has } 0 \text{ children}) \\ &= 1 - 2^{-0-1} = 1/2.\end{aligned}$$

Of course, $\mathbb{P}(D_i(t) = \Delta \mid D_{i-1}(t) = \Delta) = 1 - \mathbb{P}(D_i(t) = \nabla \mid D_{i-1}(t) = \Delta) = 1/2$ too. Now the other case where we start with a sequence of k ∇ triangles. Define $Y_{t,i}$ as the number of ∇ triangles after the ith triangle (including the ith triangle). So for example, if $D_i(t) = D_{i+1}(t) = D_{i+2}(t) = \nabla, D_{i+3}(t) = \Delta$, then $Y_{t,i} = 3$. Now

$$\begin{aligned}
\mathbb{P}(Y_{t,i} \geq k+1 \mid Y_{t,i} \geq k) &= \frac{\mathbb{P}(Y_{t,i} \geq k+1)}{\mathbb{P}(Y_{t,i} \geq k)} \\
&= \frac{1 - \mathbb{P}(Y_{t,i} \leq k)}{1 - \mathbb{P}(Y_{t,i} \leq k-1)} \\
&= \frac{1 - \mathbb{P}(v \text{ has at most } k+1 \text{ children})}{1 - \mathbb{P}(v \text{ has at most } k \text{ children})} \\
&= \frac{1 - \sum_{m=0}^{k+1} (\frac{1}{2})^{m+1}}{1 - \sum_{m=0}^{k} (\frac{1}{2})^{m+1}} \\
&= \frac{1}{2}.
\end{aligned}$$

Hence the probability for a Δ triangle at position $i + k$ is given by $\mathbb{P}(Y_{t,i} = k+1 \mid Y_{t,i} \geq k) = 1 - \mathbb{P}(Y_{t,i} \geq k+1 \mid Y_{t,i} \geq k) = 1/2$. □

3.1.3 Measurements in an infinite cdt

Now it is known how to generate the infinite CDT structure, the actual measurements on this type of system can be done. The benefit of this method compared to the method described in chapter 2 is that we can generate larger systems quicker and that the memory needed by the computer is less. When creating a ball of radius R, only the boundary needs to be stored at each radius $< R$. The boundary precisely determines how many triangles need to be created in each layer.

We will now examine figure 3.5. In layer $(D_n(1))_{n=0}^{\infty}$ we need at least one triangle pointing down and two triangles pointing up before all triangles are captured within radius 3. Thus the computer needs to generate triangles each with probability $1/2$ until both conditions are satisfied. The boundary is captured by a quantity we define as the *jump* J:

$$\begin{aligned}
J_t(r) = \#\{\Delta \in (D_n(t))_{n=0}^{\infty} \mid \Delta \in B(r)\} - \\
\#\{\nabla \in (D_n(t-1))_{n=0}^{\infty} \mid \nabla \in B(r)\}.
\end{aligned} \tag{3.3}$$

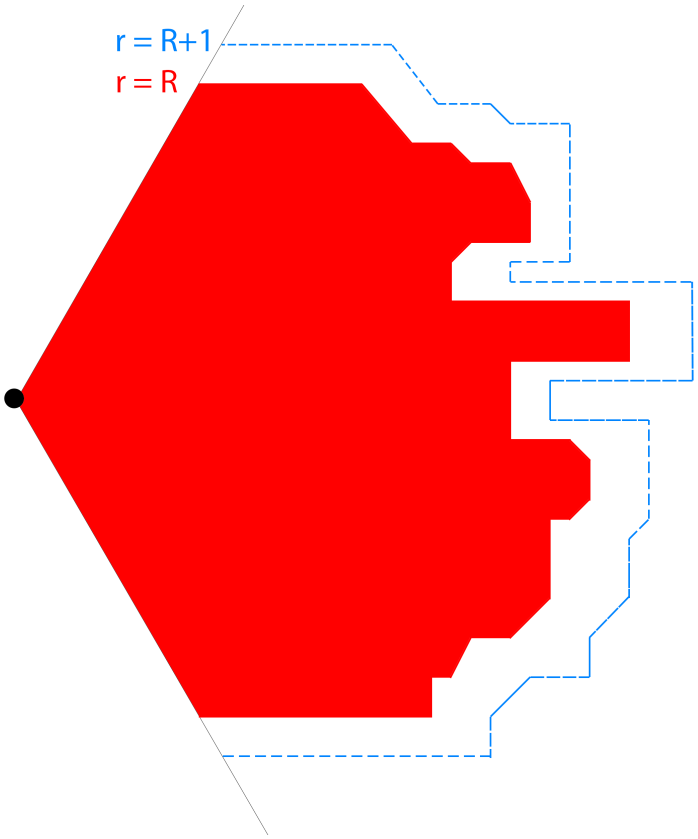


Figure 3.6: A ball of radius R together with the expected ball with radius $R+1$.

As mentioned earlier, only the boundary needs to be stored to determine the amount of triangles to be created in each layer. More precisely, to determine $J_t(R+1)$, one only needs to know $J_{t-1}(R)$, $J_t(R)$ and $J_{t+1}(R)$. Using this method to generate an infinite CDT, convergence (1.4) can be studied. In this structure, we still need to define quantities which are analogous to the quantities used by Curien & Hutchcroft. We are interested in a change in volume of the system as a function of some distance.

The increase in volume is now captured by the amount of triangles added in the step from the ball of radius R to the ball of radius $R+1$. As seen before, the amount of triangles to generate is directly related to the jumps. Observe in figure 3.6 that this change in volume scales with the jumps $J_t(R)$ times the radius R, such that

$$\frac{\partial V}{\partial r} \sim \langle J_t(r) \rangle \cdot r. \tag{3.4}$$

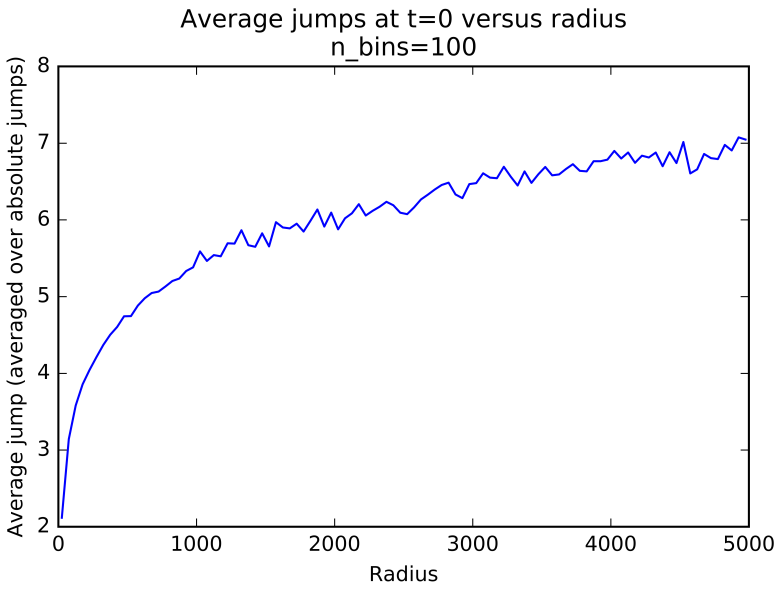


Figure 3.7: Jump at $t = 0$ versus r plotted.

Hence the jumps are influencing convergence (1.4).

3.2 Results

To examine these jumps, they are studied at $t = 0$. To get an idea of how the jumps scale as a function of r, a plot is made of the expectation value $\langle J_0(r) \rangle$ versus r.

As can be seen in figure 3.7, the jumps seem to scale with a logarithmic factor. The hypothesis then is

$$J_t(r) \sim (\log(r))^\alpha. \tag{3.5}$$

The most natural value for α would be one. It could be that there is some reason for the jumps to scale in this way. To numerically determine the α, a log-log plot is made such that the slope of the line is equal to α.

The interval $[0, 5000]$ is split up in 200 bins in which the average of the jumps is taken. Since the scaling does not happen at low values for r, the right graph in figure 3.8 is fitted on the r interval $[3500, 5000]$. This results in $\alpha = 1.027 \pm 0.046$,

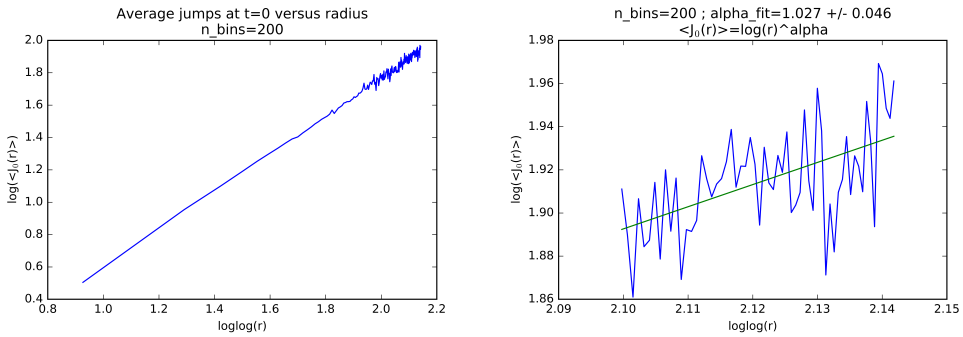


Figure 3.8: Two log-log plots of $J_0(r)$ versus r. On the left all data gathered is shown, radius interval $[0, 5000]$. The right figure is zoomed in on radius interval $[3500, 5000]$.

hence $\alpha = 1$ is within the error margin. One could gather more data on the jumps at $t = 0$ to get a more accurate value for α.

Another interesting observation can be seen in figure 3.6, at the boundary at $t = 0$ one observes an area which is in between two lobes. The most probable thing to happen there is that these lobes eventually merge, this can be seen in the blue dotted line in the figure, which is the probable evolution of the ball when increasing the radius by one.

To understand this phenomena better, a plot is made of *time* versus *radius* where the colours indicate the jumps at that point (r, t). Red indicates a positive jump and blue indicates a negative jump. What can be seen is that some, probably big, lobes survive for a long time, both positive and negative jump lobes.

In figure 3.10 it can be seen that at $t \approx -10$ the jumps can get large. They will 'eat' every counter lobe (a lobe where the jump has opposite sign) that comes in its way. This also supports the statement about the jump density plot, where certain lobes survive for a long time.

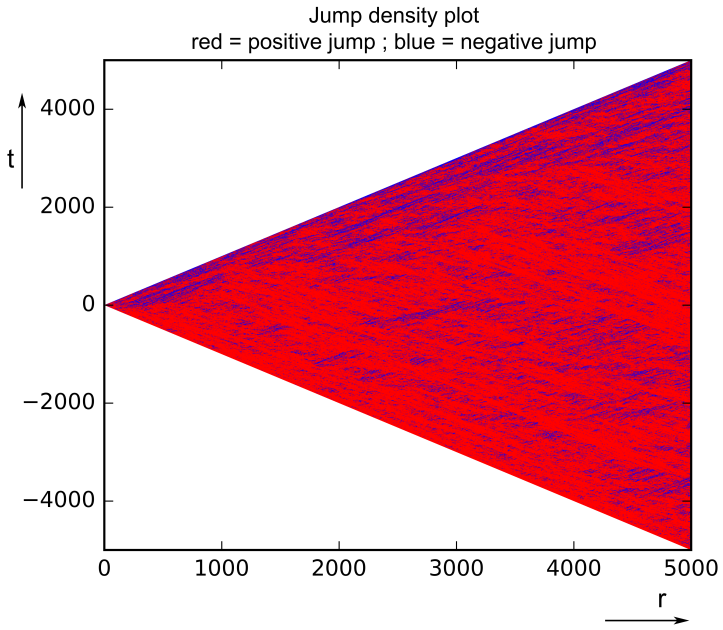


Figure 3.9: A density plot of time versus radius. The colours indicate the jumps, red is a positive jump, blue is a negative jump.

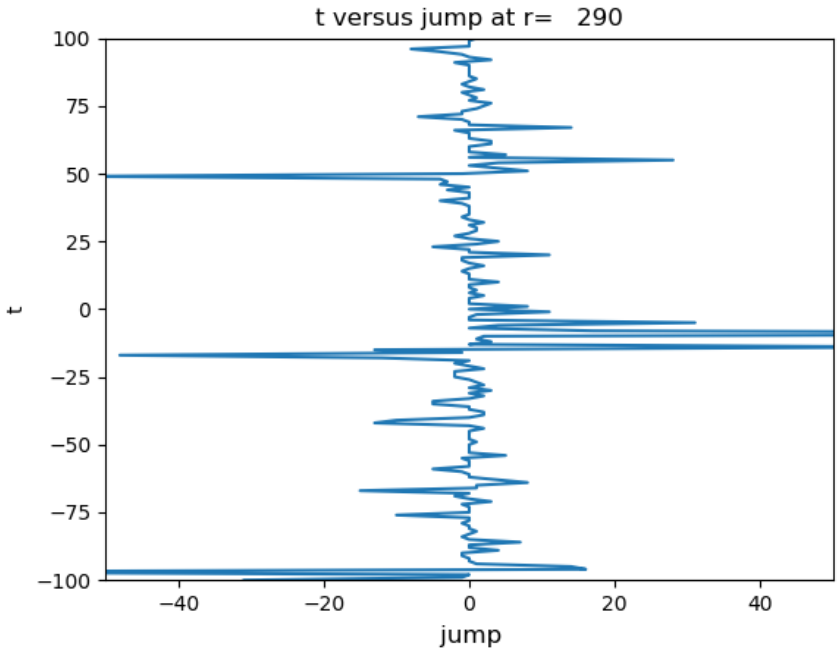


Figure 3.10: A plot of time versus jump at a fixed radius.

Chapter 4

Conclusion

In this thesis, the convergence as described by Curien & Hutchcroft is studied and further examined. First by scaling finite two-dimensional causal dynamical triangulations and later by generating infinite CDTs. The expected convergence is present and the goal was to determine the speed of this.

For the finite case, plane trees with n vertices are picked uniformly at random to determine the corresponding CDT. This finite two-dimensional CDT is then studied to get more precision on the statement of Curien & Hutchcroft. Unfortunately, even CDTs with one million vertices could not provide us with data at which the convergence can be studied. This is because we want to study what happens in the limit where the distance approaches infinity. In the simulated system, the average distance stays between 100 and 150 in a CDT with 1,000,000 vertices.

Later, infinite causal dynamical triangulations are studied with which much larger distances can be reached in shorter time. The definition of the infinite CDT is that one can generate the two kinds of triangles, up and down pointing, each with probability 1/2. This results in a structure in which observables behave the same as in the limit where amount of vertices reaches infinity in the finite CDTs. The structure which is then generated is a ball of a certain radius. In this way, it is really easy to generate the system. Also, the memory usage is way less compared to the finite CDT method because one only needs to store the boundary of the generated area. Using the boundary at radius r, one can determine how many triangles to generate in each layer to get a ball of radius $r + 1$.

This type of system is then studied and a relation is found between a change in volume of the ball and the boundary of the ball. As stated in the thesis, the boundary is described by what we call 'jumps' $J_t(r)$. These jumps seem to scale logarithmically with radius, thus $J_t(r) \sim \log(r)$. In the simulations ran for this thesis, it is also seen that these jumps survive for a long time, they stack up and become lobes of the generated ball. It would be interesting to know *why* the jumps scale in this way. A theoretical explanation of this phenomena would be a nice topic for future research.

Many more properties of the two-dimensional infinite CDT can be studied using the method described in the thesis. One could, in future research, study other kinds of distances within the system to get a better understanding of the geometry of infinite causal dynamical triangulations.

Bibliography

[1] R. Abraham and J-F Delmas. Local limits of conditioned galton-watson trees: the infinite spine case. *Electronic Journal of Probability 19*, pages 3–7, 2014.

[2] J. Ambjørn, A. Görlich, J. Jurkiewicz, and R. Loll. Nonperturbative quantum gravity. *Physics Reports 519*, page 48, 2012.

[3] J. Ambjørn, J. Jurkiewicz, and R. Loll. The universe from scratch. *Contemporary Physics*, pages 5–8, 2006.

[4] J. Ambjørn and L. Loll. Non-perturbative lorentzian quantum gravity, causality and topology change. *Nucl. Phys. B536*, pages 407–434, 1998.

[5] W. Beenakker. Kwantummechanica 2. 1.5.1 De afleiding van de padintegraal, 2015-2016.

[6] N. Curien, T. Hutchcroft, and A. Nachmias. Geometric and spectral properties of causal maps. *arXiv:1710.03137*, page 3, 2017.

[7] B. Durhuus, Jonsson T., and J.F. Wheater. On the spectral dimension of causal triangulations. *Journal of Statistical Physics*, pages 12–14, 2010.

[8] D.P. Kennedy. The galton-watson process conditioned on the total progeny. *Journal of Applied Probability*, pages 800–806, 1975.

[9] D.E. Knuth. *The Art of Computer Programming, volume 4, pre-fascicle 4A*. Pearson Education (Us), 2005.

[10] S. Zohren. *Uniform Infinite and Gibbs Causal Triangulations*. PhD thesis, Leiden University, 2012.