

Discrete Ricci Flow applied to Complex Networks

Ollivier-Ricci Curvature and Its Relevance to Community Detection

**Lorenzo Fabbri
Giancarlo Oancia**

This essay is submitted for the exam of
Complex Networks



Department of Physics
Theoretical Physics Curriculum
University of Bologna
01/03/2025

ABSTRACT

This project uses Ricci Flow as a geometric approach to detect community structures in networks. It applies Ollivier-Ricci curvature to adjust the weights of edges in a graph, iterating the process to shrink intra-community edges and stretch inter-community edges. Having tested the Ricci Flow method, our goal was identifying the pre-labelled communities of a real dataset: Zachary's Karate Club graph [1].

CONTENTS

1	Overview of the Project	1
2	Introduction and Motivation	3
2.1	Poincaré Conjecture	3
2.2	Basics of Differential Geometry	6
2.3	Non-Linear Sigma Models and String Theory	16
2.4	Ricci Flow to Tackle Topological Problems	19
3	Ricci Flow and Oliver-Ricci Curvature	21
3.1	Riemannian Geometry and Curvature Notions	21
3.2	Optimal Transport and Ollivier’s Ricci Curvature	23
4	Application to Complex Networks	25
4.1	ARI, Modularity, and Performance	26
5	The Developed Code	33
5.1	The setup	34
5.2	Tests on Synthetic Graphs	34
5.3	Application to Zachary’s Karate Club Graph	41
6	Conclusions and Future Directions	45
Appendix A	Metric of a 2-sphere	47
References		49

CHAPTER 1

OVERVIEW OF THE PROJECT

The study of networks has gained considerable attention in various fields, ranging from sociology to biology, and beyond. One of the central problems in network science is community detection, where the objective is to identify groups of nodes (communities) that are more densely connected internally than with the rest of the network. Traditional methods for community detection often rely on statistical or combinatorial approaches. However, recent developments in geometric methods have introduced new approaches to this problem by leveraging concepts from differential geometry [2].

A powerful geometric tool, the Ricci Flow, originally developed in the context of smooth Riemannian manifolds, can be adapted to discrete network structures. In its original formulation, the Ricci Flow evolves the metric of a manifold according to the curvature (represented by the Ricci tensor), leading to a smoothing process over time. Ollivier-Ricci curvature, a discrete approximation of Ricci curvature for graphs, provides a framework to extend this idea to networks, where the “curvature” of edges encodes structural information about node connectivity. Specifically, positive curvature tends to shrink intra-community edges, while negative curvature expands inter-community edges [2].

In this project, we apply Ollivier-Ricci curvature and Ricci Flow to detect the two known communities in Zachary’s Karate Club graph. The approach follows the work of Ni et al. [2], where Ricci Flow is used to reshape edge weights iteratively, enhancing the separation between different communities. After applying Ricci Flow, we perform edge surgery to remove weakly connected edges and extract communities as the connected components of the resulting graph.

2 | Overview of the Project

The results will be compared to the predefined community labels, allowing for a direct comparison between the detected communities and the actual community structures.

The developed code can be accessed in the corresponding GitHub repository: [RicciFlowNetwork](#). Code documentation is accessible at [CodeDocumentation](#).

To conclude this introductory chapter, we provide here a concise map of how this document is organized and how each chapter contributes to the overarching discussion.

Chapter 2 opens with the historical motivations behind the Ricci Flow, viewing it as a geometrical tool to tackle topological problems. In particular, the Poincaré Conjecture is introduced as an example of such a problem, laying the foundations for why Ricci Flow became central in resolving deep topological questions. Given the need for a rigorous framework, the chapter proceeds with an overview of differentiable manifolds, needed for the formal definition of Ricci Flow. To complete the picture, a brief detour into string theory and non-linear sigma models highlights how Poincaré's Conjecture was eventually proved through Perelman's work, showing an instance of Ricci Flow in action.

Chapter 3 then moves to the formal definition of Ricci Flow itself, shifting from smooth differential settings to the discrete analog known as Ollivier-Ricci curvature. This transition is relevant for the application of Ricci Flow in a network-oriented context.

In Chapter 4, we shift focus to the application to complex networks. Exploiting curvature as a measure of structural cohesion, the chapter illustrates how Ricci Flow techniques can pinpoint communities, detect bridging edges, and reveal global organizational properties in graph-based data.

Chapter 5 is devoted to the code that implements these ideas. It analyses our code and the results obtained.

Finally, Chapter 6 is devoted to conclusions and future directions.

CHAPTER 2

INTRODUCTION AND MOTIVATION

2.1 Poincaré Conjecture

To understand the origins of Ricci Flow and its relevance to mathematics, we first need to examine how it applies to geometrical and topological problems. Hence, we begin by introducing the Poincaré conjecture and Perelman's proof.

In his 1895 paper *Analysis Situs* [3], Poincaré laid the foundations for what we now call *topology*. He clearly underlined his purpose in this article, stating:

...geometry is the art of reasoning well from badly drawn figures; however, these figures, if they are not to deceive us, must satisfy certain conditions; the proportions may be grossly altered, but the relative positions of the different parts must not be upset.

Because we don't need to trace every historical development, we now provide modern definitions and intuitions about topology [4], beginning with a formal definition of a topological space.

Definition 1 (Topological space). Let X be a non-empty set. A *topology* on X is a family \mathcal{A} of subsets $\mathcal{A} \ni A \subseteq X$, called *open sets*, such that

- The empty set and X are open sets,

$$\emptyset \in \mathcal{A}, \quad X \in \mathcal{A}. \tag{2.1}$$

- The union of any collection of open sets is again an open set,

$$A_i \in \mathcal{A}, i \in J \implies \bigcup_{i \in J} A_i \in \mathcal{A}, \tag{2.2}$$

with J a collection of indices.

- The intersection of any finite number of open sets is again an open set,

$$A_i \in \mathcal{A}, i = 1, 2, \dots, N \implies \bigcap_{i=1}^N A_i \in \mathcal{A}. \quad (2.3)$$

The set X , with the topology \mathcal{A} , is called a *topological space* (X, \mathcal{A}) .

We can then define a *closed set* simply by taking the complement of any open set:

Definition 2 (Closed sets). Given $A \in \mathcal{A}$, a subset $C \subseteq X$ is *closed* if

$$C \text{ closed} \iff X \setminus C \text{ open.} \quad (2.4)$$

One crucial property of topological spaces is *compactness*, which captures the idea of finiteness in this context:

Definition 3 (Compact). A topological space (X, \mathcal{A}) is called *compact* if every open cover,

$$X \in \bigcup_{\alpha \in C} U_\alpha, \quad (2.5)$$

contains a finite subcover,

$$X \in \bigcup_{i=1}^N U_i. \quad (2.6)$$

Another important property *connectedness*: intuitively, a connected space can't be partitioned into separate pieces with nothing in common. To make this intuition more precise, we first need to define a *path*.

Definition 4 (Path). Let (X, \mathcal{A}) be a topological space, and let $x, y \in X$. A *path* is a continuous function $f : [0, 1] \rightarrow X$ with $f(0) = x$ and $f(1) = y$. A *path-component* of X is an equivalence class of X where x and y belong to the same class if a path connects them.

Definition 5 (Connected). A topological space (X, \mathcal{A}) is *connected* if it can't be represented as the union of two or more disjoint non-empty open subsets.

Definition 6 (Path connected). A topological space (X, \mathcal{A}) is *path connected* if there is exactly one path-component. For non-empty spaces, this is equivalent to saying that any two points in X can be joined by a path.

Definition 7 (Simply connected). A topological space (X, \mathcal{A}) is *simply connected* if it is path connected and every path between two points can be continuously transformed into any other path connecting the same two points, while keeping the endpoints fixed.

Beyond these properties, we often define continuous maps from one topological space to another. A *homeomorphism* is a continuous, bijective map whose inverse is also continuous, making the two spaces essentially “the same” from a topological viewpoint.

We are particularly interested in manifolds, which are topological spaces locally resembling \mathbb{R}^n . To apply Ricci Flow in a topological setting, one endows these manifolds with a richer structure — namely, a Riemannian metric. For now, let’s define a general manifold:

Definition 8 (Manifold). A *topological manifold* is a topological space which locally resembles the real n -dimensional Euclidean space.

Definition 9 (Closed manifold). A *closed manifold* is a manifold without boundary that is also compact.

Classic examples are circles ($1D$) and spheres ($2D$). We are especially interested in the 3 -sphere, the direct generalization in three dimensions:

Definition 10 (3 -sphere). In coordinates¹, a 3 -sphere centred at (C_0, C_1, C_2, C_3) with radius r is the set of points $(x_0, x_1, x_2, x_3) \in \mathbb{R}^4$ satisfying

$$\sum_{i=0}^3 (x_i - C_i)^2 = (x_0 - C_0)^2 + (x_1 - C_1)^2 + (x_2 - C_2)^2 + (x_3 - C_3)^2 = r^2. \quad (2.7)$$

As a Manifold, the 3 -sphere is a compact, connected 3 -dimensional Manifold, with no boundary. Poincaré’s conjecture can thus be stated [3]:

A simply-connected closed manifold is homeomorphic to a sphere.

Initially, Poincaré did not consider this a conjecture because it seemed trivial, yet it became one of the greatest unsolved problems in mathematics. Refined over time, its final form [5] is:

¹We’ll define a chart on a manifold later on, for now let’s use our intuitive understanding of a set of coordinates in \mathbb{R}^n .

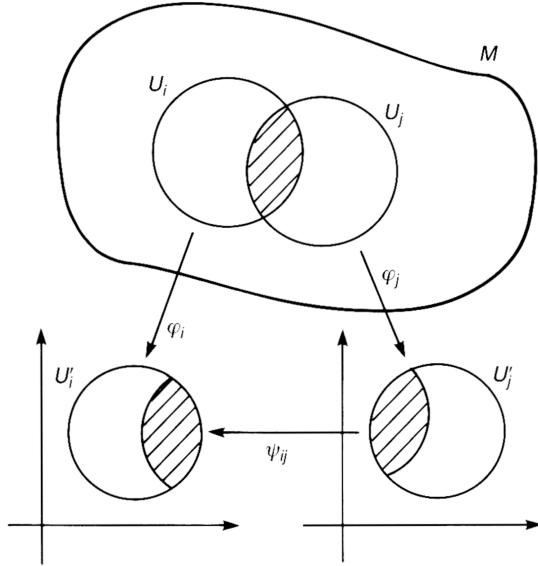


Figure 2.1 φ_i is a homeomorphism from U_i onto an open subset U'_i of \mathbb{R}^n .

Every three-dimensional topological manifold which is closed, connected, and has trivial fundamental group is homeomorphic to the three-dimensional sphere.

To avoid going into all the complexities of homology and fundamental groups, we cite a useful result:

Theorem T.1. *A path-connected topological space is simply connected if and only if its fundamental group is trivial.*

Understanding how to prove the Poincaré conjecture requires investigating specific manifold structures, ultimately leading to the application of Ricci Flow.

2.2 Basics of Differential Geometry

To understand the applications of Ricci Flow, we need to explore the properties of Riemannian Manifolds, especially how to define a flow on a manifold and how it relates to differential equations. We will see that the Ricci Flow can be viewed as a differential equation for a family of metrics on a Riemannian manifold.

2.2.1 Differentiable Manifolds

Let's begin by refining the definition 8 of a *differentiable Manifold*, [6, 7].

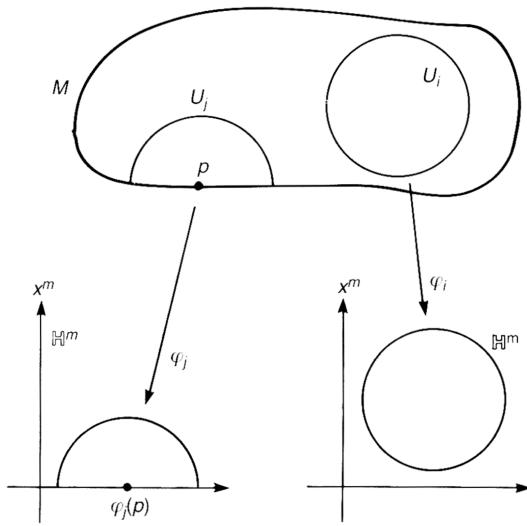


Figure 2.2 A Manifold \mathcal{M} with boundary. Here, $p \in \partial\mathcal{M}$.

Definition 11 (Differentiable manifold). An n -dimensional *differentiable manifold* \mathcal{M} is a topological space equipped with a family of pairs $\{(U_i, \varphi_i)\}$, called an *atlas*, where

- Each U_i is an open set in \mathcal{M} , and $\cup_i U_i = \mathcal{M}$.
- Each φ_i is a homeomorphism from U_i onto an open subset $U'_i \subseteq \mathbb{R}^n$, as showed in fig. 2.1
- For any U_i and U_j with $U_i \cap U_j \neq \emptyset$, the map

$$\varphi_i \circ \varphi_j^{-1}: \varphi_j(U_i \cap U_j) \rightarrow \varphi_i(U_i \cap U_j),$$

is infinitely differentiable.

Each pair (U_i, φ_i) is called a *chart*, with U_i the *coordinate neighbourhood* and φ_i the *coordinate map*. Basically, φ_i assigns n real *coordinates* $\{x_1(p), \dots, x_n(p)\}$ to each point of U_i .

A *Manifold with boundary* is defined similarly, except each chart maps into the closed half-space $H^n = \{(x^1, \dots, x^n) \in R^n | x^n \geq 0\}$, as showed in fig. 2.2.

2.2.2 Differentiable Maps

Due to the presence of the differentiable structure on a Manifold, we're able to use the calculus techniques developed for \mathbb{R}^n . We can define a map between manifolds.

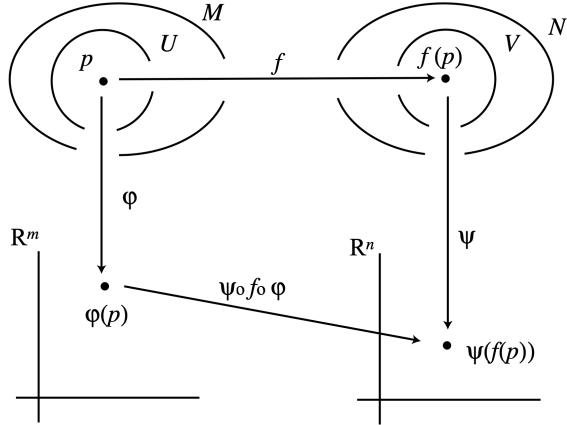


Figure 2.3 A map $f: \mathcal{M} \rightarrow \mathcal{N}$ has coordinates representation $\psi \circ f \circ \varphi^{-1}: \mathbb{R}^m \rightarrow \mathbb{R}^n$.

Let \mathcal{M} and \mathcal{N} be an m -dimensional and an n -dimensional Manifold, respectively. Let's then consider a map between them, i.e., $f: \mathcal{M} \rightarrow \mathcal{N}$, where $\mathcal{M} \ni p \mapsto f(p) \in \mathcal{N}$, see fig. 2.3 Taking the charts (U, φ) on \mathcal{M} and (V, ψ) on \mathcal{N} , the coordinate representation of f will be

$$\varphi \circ f \circ \varphi^{-1}: \mathbb{R}^m \rightarrow \mathbb{R}^n. \quad (2.8)$$

We can write $\varphi(p) = \{x^\mu\}$ and $\psi(f(p)) = \{y^\alpha\}$, which makes explicit that $y \equiv \psi \circ f \circ \varphi^{-1}(x)$ is an m -variables, vector-valued, usual function. Rendering the coordinate charts implicit, we may write $y^\alpha = f^\alpha(x^\mu)$, which makes clear why f is said to be *differentiable* at p if y is C^∞ . Notice, however, that the differentiability of f is independent of the chosen coordinates.

A very important type of maps between manifold is given by diffeomorphisms.

Definition 12 (Diffeomorphism). Let $f: \mathcal{M} \rightarrow \mathcal{N}$ be a homeomorphism and ψ and φ the same coordinate functions as before. Then, if $\psi \circ f \circ \varphi^{-1}$ is invertible and both $y \equiv \psi \circ f \circ \varphi^{-1}(x)$ and $x \equiv \varphi \circ f^{-1} \circ \psi^{-1}(y)$ are C^∞ , f is called a *diffeomorphism* and \mathcal{M} is said to be *diffeomorphic* to \mathcal{N} , $\mathcal{M} \equiv \mathcal{N}$.

Taking a diffeomorphism from a Manifold into itself, we can implement the concept of change of coordinates, in two different interpretation. First, the set of diffeomorphisms $f: \mathcal{M} \rightarrow \mathcal{M}$ form a group denoted by $\text{Diff}(\mathcal{M})$. Considering a particular $f \in \text{Diff}(\mathcal{M})$, and a chart (U, φ) , such that, for $p \in U$ and $f(p) \in U$, we get $\varphi(p) = x^\mu(p)$ and $\varphi(f(p)) = y^\mu(f(p))$, then y is a differentiable function of x and the above diffeomorphism can be thought as an *active transformation* for a change of coordinates.

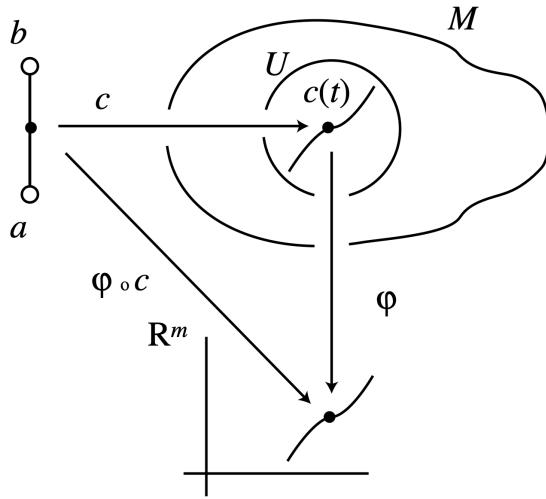


Figure 2.4 A curve c in \mathcal{M} and its coordinate representation $\psi \circ c$.

However, if (U, φ) and (V, ψ) are overlapping charts, for a point $p \in U \cap V$, there are two coordinates values, i.e., $x^\mu = \varphi(p)$ and $y^\mu = \psi(p)$. Then, the map $x \mapsto y$ is differentiable, and it represents a *passive transformation* for a change of coordinates.

Two important kinds of maps are curves and functions.

Definition 13 (Curve). An *open curve* in an n -dimensional Manifold \mathcal{M} is a map $c: (a, b) \rightarrow \mathcal{M}$, where (a, b) is an open interval such that $a < 0 < b$. A *closed curve* is a map $c: S^1 \rightarrow \mathcal{M}$. On a chart (U, φ) , a curve $c(t)$ has the coordinate representation $x = \varphi \circ c: \mathbb{R} \rightarrow \mathbb{R}^n$. See fig. 2.4

Definition 14 (Function). A *function* f on \mathcal{M} is a smooth map from \mathcal{M} to \mathbb{R} . On a chart (U, φ) , the coordinate representation of f is given by $f \circ \varphi^{-1}: \mathbb{R}^n \rightarrow \mathbb{R}$, which is a real-valued function of n variables. The set of functions is denoted by $\mathfrak{F}(\mathcal{M})$. See fig. 2.5

2.2.3 Vectors

The curves allow us to define a vector as a differential operator. So understand this more deeply, let's take a Manifold \mathcal{M} , a curve $c: (a, b) \rightarrow \mathcal{M}$ and a function $f: \mathcal{M} \rightarrow \mathbb{R}$, where (a, b) is an open interval containing $t = 0$, as showed in fig. 2.6. The tangent vector at $c(0)$ is defined to be the directional derivative of a function $f(c(t))$ along the

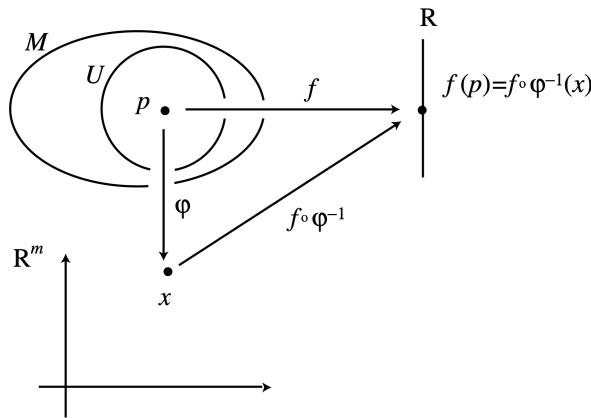


Figure 2.5 A function $f: \mathcal{M} \rightarrow \mathbb{R}$ and its coordinate representation $f \circ \varphi^{-1}$.

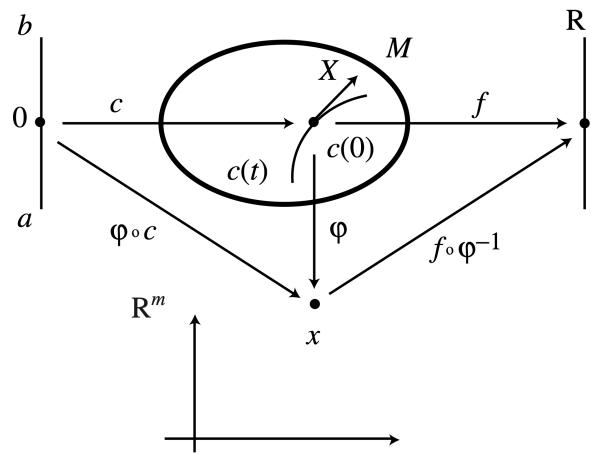


Figure 2.6 A curve c and a function f define a tangent vector along the curve in terms of directional derivatives.

curve $c(t)$ at $t = 0$. In particular,

$$X[f] := \frac{df(c(t))}{dt} \Big|_{t=0} = \frac{\partial f}{\partial x^\mu} \frac{dx^\mu(c(t))}{dt} \Big|_{t=0} = X^\mu \left(\frac{\partial f}{\partial x^\mu} \right), \quad (2.9)$$

where we defined

$$X = X^\mu \left(\frac{\partial}{\partial x^\mu} \right), \quad X^\mu = \frac{dx^\mu(c(t))}{dt} \Big|_{t=0} \quad (2.10)$$

Notice that we used abuse of notation, that is $\partial f / \partial x^\mu$ really means $\partial(f \circ \varphi^{-1}(x)) / \partial x^\mu$.

Trying to be more precise, as known from multivariate analysis, whenever we have a curve, we can define an equivalence class. This case is no different. Indeed, given

two curves $c_1(t)$ and $c_2(t)$ on \mathcal{M} , if they satisfy

$$c_1(0) = c_2(0) = p, \quad (2.11a)$$

$$\left. \frac{dx^\mu(c_1(t))}{dt} \right|_{t=0} = \left. \frac{dx^\mu(c_2(t))}{dt} \right|_{t=0}, \quad (2.11b)$$

then they yield the same differential operator X at p . Then, the above conditions define an equivalence relation \sim such that $c_1(t) \sim c_2(t)$. This allows us to define

Definition 15 (Vector). A *tangent vector* X is the *equivalence class of curves*

$$[c(t)] = \left\{ \tilde{c}(t) \mid \tilde{c}(0) = c(0) \wedge \left. \frac{dx^\mu(\tilde{c}(t))}{dt} \right|_{t=0} = \left. \frac{dx^\mu(c(t))}{dt} \right|_{t=0} \right\}. \quad (2.12)$$

All the tangent vectors at a particular point $p \in \mathcal{M}$ form a vector space called *tangent space* of \mathcal{M} at p , denoted by $T_p\mathcal{M}$. By means of equation (2.10), it's straightforward to see that the basis vectors of this vector space are

$$\left\{ e_\mu = \frac{\partial}{\partial x^\mu} \right\}, \quad \mu = 1, \dots, n. \quad (2.13)$$

Clearly, $\dim T_p\mathcal{M} = \dim \mathcal{M}$. Further, for each $V \in T_p\mathcal{M}$, we can expand it as $V = V^\mu e_\mu = V^\mu \partial_\mu$, and we call V^μ the components of V with respect to the basis.

Obviously, $\{e_\mu\}$ are not the only possible basis for the vector space $T_p\mathcal{M}$. Indeed, as known from linear algebra, an arbitrary linear combination $\hat{e}_i := A_i^\mu e_\mu$, with $A = (A_i^\mu) \in GL(n, \mathbb{R})$, is a basis as well. In this case, $\{\hat{e}_i\}$ is called a *non-coordinate basis*.

Further, given that a vector exists independently of its coordinates, we can take two coordinate charts, with a point in the intersection of their domains, i.e., $p \in U_i \cap U_j$, with $x = \varphi_i(p)$ and $y = \varphi_j(p)$. A generic vector $X \in T_p\mathcal{M}$ can be equivalently expanded as

$$X = X^\mu \frac{\partial}{\partial x^\mu} = \tilde{X}^\mu \frac{\partial}{\partial y^\mu}, \quad (2.14)$$

which shows the relation of the components of a vector in two different charts, i.e,

$$\tilde{X}^\mu = X^\nu \frac{\partial y^\mu}{\partial x^\nu}. \quad (2.15)$$

2.2.4 One-Forms

Given that $T_p\mathcal{M}$ is a vector space, we can define its dual vector space, whose elements are linear functional from $T_p\mathcal{M}$ to \mathbb{R} . This space is called *cotangent space*, is denoted by $T_p^*\mathcal{M}$, and their elements $\omega: T_p\mathcal{M} \rightarrow \mathbb{R} \in T_p^*\mathcal{M}$ are called *one-forms*.

The simplest one-form is the *differential* of a function $f \in \mathfrak{F}(\mathcal{M})$. Indeed, for a vector $V \in T_p\mathcal{M}$, its action on f is

$$V[f] = V^\mu \frac{\partial f}{\partial x^\mu} \in \mathbb{R}. \quad (2.16)$$

So, we define the action of $df \in T_p^*\mathcal{M}$ on a vector $V \in T_p\mathcal{M}$ by

$$\langle df, V \rangle \equiv V[f] = V^\mu \frac{\partial f}{\partial x^\mu} \in \mathbb{R}. \quad (2.17)$$

In coordinates $x = \varphi(p)$, the differential can be expanded as

$$df = \frac{\partial f}{\partial x^\mu} dx^\mu, \quad (2.18)$$

which clearly shows that $\{dx^\mu\}$ is a basis of $T_p^*\mathcal{M}$. In particular, it's the dual basis of $\{\partial_\mu\}$, since

$$\langle dx^\mu, \partial_\mu \rangle = \frac{\partial x^\nu}{\partial x^\mu} = \delta_\mu^\nu. \quad (2.19)$$

It's now easy to consider generic one-forms $\omega \in T_p^*\mathcal{M}$ and generic vectors $V \in T_p\mathcal{M}$. Picking a coordinate basis for the tangent space and its dual basis for the cotangent one, we can expand $\omega = \omega_\mu dx^\mu$ and $V = V^\mu \partial_\mu$. Then, as usual, we can define an *inner product* $\langle ., . \rangle: T_p^*\mathcal{M} \times T_p\mathcal{M} \rightarrow \mathbb{R}$ by

$$\langle \omega, V \rangle = \omega_\mu V^\mu \langle dx^\mu, \partial_\nu \rangle = \omega_\mu V^\nu \delta_\nu^\mu = \omega_\mu V^\mu. \quad (2.20)$$

Finally, analogously than before, we can pick two charts and a point in their domain's intersection, $p \in U_i \cap U_j$, so that

$$\omega = \omega_\mu dx^\mu = \tilde{\omega}_\nu dy^\nu, \quad (2.21)$$

with $x = \varphi_i(p)$ and $y = \varphi_j(p)$. Then, we can easily infer the transformation rule of a one-form under a change of coordinates, indeed,

$$dy^\nu = \frac{\partial y^\nu}{\partial x^\mu} dx^\mu \implies \tilde{\omega}_\nu = \omega_\mu \frac{\partial x^\mu}{\partial y^\nu}. \quad (2.22)$$

2.2.5 Tensors and Tensor Fields

From multilinear algebra, it should be familiar that a *tensor* of type (p, q) is a multilinear map which maps q elements of $T_p^*\mathcal{M}$ and r elements of $T_p\mathcal{M}$ to a real number:

$$T^{(q,r)} : \underbrace{T_p^*\mathcal{M} \otimes \cdots \otimes T_p^*\mathcal{M}}_{q \text{ times}} \otimes \underbrace{T_p\mathcal{M} \otimes \cdots \otimes T_p\mathcal{M}}_{r \text{ times}} \longrightarrow \mathbb{R}. \quad (2.23)$$

Picking a coordinate basis and its dual basis,

$$T^{(q,r)} = T^{\mu_1 \dots \mu_q}_{\nu_1 \dots \nu_r} \frac{\partial}{\partial x^{\mu_1}} \dots \frac{\partial}{\partial x^{\mu_q}} dx^{\nu_1} \dots dx^{\nu_r}. \quad (2.24)$$

The set of type (q, r) tensors at $p \in \mathcal{M}$ is denoted by $\mathcal{T}_{r;p}^q(\mathcal{M})$. Further, taking r -vectors $V_a = V_a^\mu \partial_\mu$ and q one-forms $\omega_a = \omega_{a\mu} dx^\mu$, the action of $T^{(q,r)}$ on them is given by

$$T(\omega_1, \dots, \omega_q; V_1, \dots, V_r) = T^{\mu_1 \dots \mu_q}_{\nu_1 \dots \nu_r} \omega_{1\mu_1} \dots \omega_{q\mu_q} V_1^{\nu_1} \dots V_r^{\nu_r}. \quad (2.25)$$

In order to probe a Manifold structure, it's necessary to have fields defined in it, i.e., structures which are smoothly defined for each point of \mathcal{M} . Knowing the tensor structure at a point, it's easy to generalize it to the entire Manifold. To begin with, let's consider the simplest tensor, i.e., a vector, which is a $(1, 0)$ tensor.

A *vector field* is a vector assigned smoothly to each point of \mathcal{M} . In other words, V is a vector field if $V[f] \in \mathfrak{F}(\mathcal{M})$, for any $f \in \mathfrak{F}(\mathcal{M})$. Then, each component of a vector field is itself a smooth function from \mathcal{M} to \mathbb{R} . We denote the set of vector fields with $\mathfrak{X}(\mathcal{M})$. Then, a vector $X \in \mathfrak{X}(\mathcal{M})$ computed at a point $p \in \mathcal{M}$ is a vector at $T_p\mathcal{M}$, i.e., $X|_p \in T_p\mathcal{M}$.

Similarly, a *tensor field* of type (q, r) is a smooth assignment of an element of $\mathcal{T}_{r;p}^q(\mathcal{M})$ at each point $p \in \mathcal{M}$. The set of tensor fields of type (q, r) on \mathcal{M} is denoted by $\mathcal{T}_r^q(\mathcal{M})$.

An important tensor field we're interested in is the metric tensor, which is tightly related to the definition of a Ricci Flow, in the context of Riemannian Manifolds. However, let's first study an important property of vector fields: they can generate a flow on a Manifold. Studying further the relation among flows and differential equation, allows us to better grasp the definition of a Ricci Flow.

2.2.6 Flow generated by a vector field

Let X be a vector field in \mathcal{M} , $X \in \mathfrak{X}(\mathcal{M})$. An *integral curve* $x(t)$ of X is a curve in \mathcal{M} , whose tangent vector at $x(t)$ is $X|_x$. In a chart (U, φ) , this is equivalent to

$$\frac{dx^\mu}{dt} = X^\mu(x(t)), \quad (2.26)$$

where $x^\mu(t)$ is the μ -th component of $\varphi(x(t))$ and $X = X^\mu \partial_\mu$. Pay attention to the abuse of notation. Indeed, we used x to denote both a point on \mathcal{M} and its coordinates.

Then, to finding the integral curve of a vector field X is equivalent to solving the system of ODEs (2.26), with initial condition $x_0^\mu = x^\mu(0)$, which are the coordinates of the integral curve at $t = 0$.

The existence and uniqueness theorem of ODEs guarantees that there is a unique solution to (2.26), at least locally, with the initial condition x_0^μ .

Let, then, $\sigma(t, x_0)$ be an integral curve of X passing through a point x_0 at $t = 0$ and let's denote its coordinates by $\sigma^\mu(t, x_0)$. Then, eq. (2.26) becomes

$$\frac{d}{dt} \sigma^\mu(t, x_0) = X^\mu(\sigma(t, x_0)), \quad (2.27)$$

with the initial condition

$$\sigma^\mu(0, x_0) = x_0^\mu. \quad (2.28)$$

The map $\sigma: \mathbb{R} \times \mathcal{M} \rightarrow \mathcal{M}$ si called *flow* generated by $X \in \mathfrak{X}(\mathcal{M})$. We can prove it satisfies the condition

$$\sigma(t, \sigma^\mu(s, x_0)) = \sigma(t + s, x_0), \quad (2.29)$$

for $s, t \in \mathbb{R}$.

Proof. Since

$$\begin{aligned} \frac{d}{dt} \sigma^\mu(t, \sigma^\mu(s, x_0)) &= X^\mu(\sigma(t, \sigma^\mu(s, x_0))), \\ \sigma(0, \sigma(s, x_0)) &= \sigma(s, x_0), \end{aligned}$$

and

$$\begin{aligned} \frac{d}{dt} \sigma^\mu(t + s, x_0) &= \frac{d}{d(t+s)} \sigma^\mu(t + s, x_0) = X^\mu(\sigma(t + s, x_0)), \\ \sigma(0 + s, x_0) &= \sigma(s, x_0), \end{aligned}$$

both sides of eq. (2.29) satisfy the same ODE and the same initial condition. Then, from the uniqueness theorem for the solution of ODEs, they must be the same. ■

This leads to the following theorem.

Theorem T.2. *For any point $x \in \mathcal{M}$, there exists a differentiable map $\sigma: \mathbb{R} \times \mathcal{M} \rightarrow \mathcal{M}$ such that*

- $\sigma(0, x) = x$;
- $t \mapsto \sigma(t, x)$ is a solution of (2.27) and (2.28);
- $\sigma(t, \sigma^\mu(s, x)) = \sigma(t + s, x)$.

We're now ready to introduce Riemannian Manifold and their most relevant properties.

2.2.7 Riemannian Manifolds

A Riemannian Manifold is defined as a smooth Manifold which admits a Riemannian metric.

Definition 16. Let \mathcal{M} be a differentiable Manifold. A *Riemannian metric* g on \mathcal{M} is a type $(0, 2)$ tensor field on \mathcal{M} such that, at each point $p \in \mathcal{M}$:

- $g_p(U, V) = g_p(V, U)$,
- $g_p(U, V) \geq 0$, where the equality holds only when $U = 0$.

Here, $U, V \in T_p \mathcal{M}$ and $g_p = g|_p$. Basically, g_p is a symmetric positive-definite bilinear form.

Recall the previous definition of inner product (2.20) between vectors and dual forms. For $V \in T_p \mathcal{M}$ and $\omega \in T_p^* \mathcal{M}$, then $\langle ., . \rangle: T_p^* \mathcal{M} \times T_p \mathcal{M} \rightarrow \mathbb{R}$. If there exists a metric tensor g , then, we can use it to define the inner product between two vectors $U, V \in T_p \mathcal{M}$, specifically by $g_p(U, V)$. Since $g_p: T_p \mathcal{M} \otimes T_p \mathcal{M} \rightarrow \mathbb{R}$, we may define a linear map $g_p(U, .): T_p \mathcal{M} \rightarrow \mathbb{R}$ by $V \mapsto g_p(U, V)$. Then, it's straightforward that $g_p(U, .) \in T_p^* \mathcal{M}$ is a one-form. Thus, the metric g_p gives rise to an isomorphism between $T_p \mathcal{M}$ and $T_p^* \mathcal{M}$.

Picking a chart (φ, U) , with coordinates $\{x^\mu\}$, we can expand the metric tensor in coordinates as

$$g_p = g_{\mu\nu}(p) dx^\mu \otimes dx^\nu, \quad g_{\mu\nu}(p) = g_p(\partial_\mu, \partial_\nu) = g_{\nu\mu}(p), \quad p \in \mathcal{M}. \quad (2.30)$$

It's usual convention to forget about the point p , denote the inverse metric as $g^{\mu\nu}$, and the determinant as $\det(g_{\mu\nu}) := g$ and $\det(g^{\mu\nu}) := g^{-1}$. Then, the isomorphism between $T_p\mathcal{M}$ and $T_p^*\mathcal{M}$ can be expressed as

$$\omega_\mu = g_{\mu\nu} U^\nu, \quad U^\mu = g^{\mu\nu} \omega_\nu. \quad (2.31)$$

The purpose of the next chapter will be to rigorously define the Ricci Flow. However, to conclude this motivation part, let's take a brief detour onto string theory, in which, with a simple example in the context of non-linear sigma models, it's possible to grasp the meaning of the Ricci Flow application to understand topological properties of a Manifold through renormalization group flows in a Riemannian Manifold.

2.2.8 Curvature and Ricci tensor

2.2.9 Covariant Derivative

2.3 Non-Linear Sigma Models and String Theory

The standard starting point of string theory is *Polyakov action*, which describes a bosonic classical, one-dimensional, string, which describes a two-dimensional worldsheet Σ on a 26-dimensional spacetime described by the spacetime coordinates $X^\mu(\xi)$, $\mu = 0, \dots, 25$, where $\xi^a = (\tau, \sigma)$, $a = 1, 2$, are the intrinsic coordinates on Σ . The metric on spacetime is denoted by $g_{\mu\nu}$, while the metric on the worldsheet is γ_{ab} . For a flat spacetime, with Minkowski metric $g_{\mu\nu} \equiv \eta_{\mu\nu} = \text{diag}(-1, +1, +1, +1)$, the action reads

$$S_P[X^\mu(\xi), \gamma_{ab}(\xi)] = -\frac{T}{2} \int_{\Sigma} d\tau d\sigma \sqrt{-\det(\gamma)} \gamma^{ab} \partial_a X_\mu(\xi) \partial_b X^\mu(\xi), \quad (2.32)$$

where T is a characteristic parameter of the string, related to the string length l .

The symmetries of this action allow us to consider a flat worldsheet metric, $\gamma_{ab} = \eta_{ab}$, considering the so-called *unit gauge*. Then, reintroducing explicitly the metric $g_{\mu\nu}$, even if it's flat in this case, we obtain

$$S_P = -T \int_{\Sigma} d^2\xi g_{\mu\nu}(X) \partial_a X^\mu \partial^a X^\nu. \quad (2.33)$$

Basically, it represents a 2-dimensional field theory on the worldsheet, where the coordinates X^μ are 26 dynamical fields in there. This allows us to quantize the theory with the usual quantization prescription, based on the substitution of the classical

Poisson brackets defined on a symplectic manifold with the commutators of operators acting on a Hilbert space.

After quantization, one notices that, for a closed string, defined by the periodicity condition $X^\mu(\tau, \sigma) = X^\mu(\tau, \sigma + l)$, the particle spectrum contains a *graviton* $\gamma_{\mu\nu}$, which resembles a gravitational wave at low energies, a scalar field φ called *dilaton* and an antisymmetric two-tensor $b_{\mu\nu}$ called *Kalb-Ramond tensor*.

Therefore, due to the presence of the graviton, one could wonder what happens for a non-flat spacetime. Then, after redefining the coordinates as a constant X_0^μ plus some other arbitrary fields Y^μ , i.e.,

$$X^\mu(\xi) = X_0^\mu(\xi) + \sqrt{\alpha'} Y^\mu(\xi), \quad (2.34)$$

we can expand the term in the Lagrangian as

$$\begin{aligned} & g_{\mu\nu}(X) \partial_a X^\mu \partial^a X^\nu \\ &= \alpha' \left[g_{\mu\nu}(X_0) + \sqrt{\alpha'} g_{\mu\nu,\rho}(X_0) Y^\rho(\xi) + \frac{\alpha'}{2} g_{\mu\nu,\rho\sigma}(X_0) Y^\rho(\xi) Y^\sigma(\xi) + \dots \right] \partial_a Y^\mu \partial^a Y^\nu \end{aligned} \quad (2.35)$$

where $g_{\mu\nu,\rho} \equiv \partial_\rho g_{\mu\nu}$.

We obtained an expansion in α' , where each term is an interaction term for the fields Y^μ , with couplings given by the derivatives of the metric. However, a crucial symmetry of the Polyakov action (2.32) is the invariance under *conformal transformations*. Those are diffeomorphisms on a Riemannian Manifold which preserve the metric up to rescaling, i.e.,

$$g(x) \rightarrow \tilde{g}(\tilde{x}) = e^{2\omega(\tilde{x})} g(\tilde{x}). \quad (2.36)$$

Without going into the details, the presence of this symmetry is considered as a consistency condition for the theory, as it allows for a perturbative interpretation of the interactions. In addition, the above interacting quantum field theory must undergo renormalization, in order to cure the divergences.

However, after renormalization, the conformal symmetry may be anomalous². Indeed, a particular example of conformal transformation (2.36) is provided by scale-invariance. That is to say, the theory can't depend on a scale to be conformal invariant. But, as the methods of *renormalization group* teach us, after renormalization the couplings run with the energy scale M of the system, dependence which is encapsulated

²An anomaly is a classical symmetry which is not preserved at the quantum level. In particular, it is due to a non-invariance of the measure of the path integral.

into the β -function

$$\beta(g_{\mu\nu}) = M \frac{\partial}{\partial M} g_{\mu\nu}. \quad (2.37)$$

As a consequence, for a quantum string theory to make sense, it must be conformal invariance, so the β -function associated to the metric, in the action (2.33) with the expansion (2.35), must vanish

$$\beta(g_{\mu\nu}) \stackrel{!}{=} 0. \quad (2.38)$$

A similar argument can be pursued for the other two particles in the closed string spectrum, i.e., the dilaton φ and the Kalb-Ramond form $b_{\mu\nu}$. The corresponding action will be

$$S_\sigma = -\frac{T}{2} \int_{\Sigma} d^2\xi \sqrt{-\det(\gamma)} [(\gamma^{ab}g_{\mu\nu}(X) + i\varepsilon^{ab}b_{\mu\nu}(X)) \partial_a X^\mu \partial_b^\nu + \alpha' \mathcal{R}\varphi(X)], \quad (2.39)$$

where ε^{ab} is the 2d Levi-Civita symbol, while $\mathcal{R} = \mathcal{R}(\gamma)$ is the Ricci scalar on the worldsheet.

Defining the Field Strength $H_{\mu\nu\rho} = \partial_\mu b_{\nu\rho} + \partial_\nu b_{\rho\mu} + \partial_\rho b_{\mu\nu}$, the vanishing of the beta functions reads

$$\beta(g_{\mu\nu}) = \alpha' \left(R_{\mu\nu} - \frac{1}{4} H_{\mu\lambda\rho} H^{\mu\lambda\rho} + 2\nabla_\mu \nabla_\nu \varphi \right) + O(\alpha'^2) \stackrel{!}{=} 0, \quad (2.40a)$$

$$\beta(b_{\mu\nu}) = \alpha' \left(\frac{1}{2} \nabla^\rho H_{\rho\mu\nu} + \nabla^\rho \varphi H_{\rho\mu\nu} \right) + O(\alpha'^2) \stackrel{!}{=} 0, \quad (2.40b)$$

$$\beta(\varphi) = \alpha' \left(\frac{1}{2} \nabla_\mu \varphi \nabla^\mu \varphi - \frac{1}{2} \nabla^2 \varphi - \frac{1}{24} H_{\mu\nu\rho} H^{\mu\nu\rho} \right) + O(\alpha'^2) \stackrel{!}{=} 0. \quad (2.40c)$$

The above equations are constraints for the spacetime fields (g, b, φ) , imposed to preserve conformal invariance of the quantum string. However, since those fields should be dynamical on spacetime, those must also be their equations of motion. This leads to the following *low-energy effective action*, which has (2.40) as equations of motion

$$S_{26} = \frac{1}{k_0^2} \int d^{26}x \sqrt{\det(g)} e^{-2\varphi} \left(\mathcal{R}(g) - \frac{1}{12} H_{\mu\nu\rho} H^{\mu\nu\rho} + 4\nabla_\mu \varphi \nabla^\mu \varphi \right). \quad (2.41)$$

To set this problem to a more general ground, and understand how this model is related to Ricci Flow, let's define more accurately what a σ -model is in field theory. A σ -model is a field theory for a field $\Phi: \Sigma \rightarrow \mathcal{M}$ that takes values in a manifold \mathcal{M} . Traditionally, Σ is the spacetime on which the field theory lives, and \mathcal{M} is called the target space. If the target space carries some linear structure, like a vector space, then

the whole physical system is called a linear σ -model. For general manifolds such as generic Riemannian ones, it is then called a non-linear σ -model. What we did above is to study the σ -model spacetime renormalization effects on the target space \mathcal{M} , and this is indeed an instance of Ricci Flow.

2.4 Ricci Flow to Tackle Topological Problems

CHAPTER 3

RICCI FLOW AND OLIVER-RICCI CURVATURE

Initially introduced by Richard S. Hamilton in the early 1980s, Ricci Flow arose historically as a powerful method in differential geometry. At its heart, it seeks to “smooth out” geometric irregularities of manifolds by evolving the underlying Riemannian metric through a partial differential equation (PDE) reminiscent of the classical heat equation. We recall that a manifold is a topological space locally resembling Euclidean space, and a *Riemannian manifold* is such a space equipped with an inner product on each tangent space, making it possible to measure angles, distances, and curvature.

Curvature, in particular, is fundamental to geometry: it describes how space bends or deviates from flatness. On a two-dimensional surface embedded in three-dimensional space, for example, curvature can be visualized by examining the deviation of geodesics (the generalization of “straight lines” in curved spaces) from parallelism, or by looking at how areas or angles are distorted compared to those in flat Euclidean geometry. The extension to higher dimensions and more abstract manifolds involves careful definitions but retains this key notion of “spatial bending.”

3.1 Riemannian Geometry and Curvature Notions

In classical Riemannian geometry, curvature can be examined from multiple perspectives. One can study the *sectional curvature*, which measures how a two-dimensional plane (spanned by two tangent directions) curves. One can also investigate the *Ricci curvature*, which is obtained by summing or averaging the sectional curvatures over all planes containing a given direction. Ricci curvature has special importance: for instance, it appears in Einstein’s equations of General Relativity, linking geometry to matter and energy distributions in spacetime.

Concretely, let (M, g) be a Riemannian manifold, where M is a smooth manifold and g is the metric tensor. The Ricci curvature Ric is derived as a contraction of the Riemann curvature tensor, itself an operator capturing how much nearby geodesics converge or diverge. Positive Ricci curvature typically implies that geodesics tend to converge, reflecting a “crowded” or positively curved geometry akin to the sphere. Negative Ricci curvature implies geodesics tend to diverge, mirroring a hyperbolic or “saddle-like” structure. Zero Ricci curvature is the hallmark of Ricci-flat manifolds, with many implications for geometry and topology.

3.1.1 Hamilton’s Ricci Flow Equation

Hamilton introduced the Ricci Flow as the PDE:

$$\frac{\partial g_{ij}}{\partial t} = -2 R_{ij},$$

where g_{ij} are the components of the metric tensor g in local coordinates and R_{ij} are the components of the Ricci curvature tensor. Informally, each infinitesimal piece of the manifold changes in time, guided by curvature. Regions of *high positive* Ricci curvature shrink faster, while regions of *negative* Ricci curvature expand. This leads to a *flow* that tends to smooth out the geometric and topological features of M .

One of the most famous applications of Ricci Flow on manifolds is Grigori Perelman’s resolution of the Poincaré Conjecture and the more general Geometrization Conjecture for three-dimensional manifolds. Perelman’s work introduced the notion of *Ricci Flow with surgery*, a procedure to remove singular regions (places where curvature blows up to infinity) and continue the flow on the remaining parts. In 3D manifolds, these singularities can be visualized as “neck pinches” that effectively separate the manifold into topologically simpler pieces. Perelman showed that by performing a series of well-defined surgeries, one could decompose a three-dimensional manifold into model geometric pieces, completing Hamilton’s program toward a proof of the Geometrization Conjecture.

3.1.2 From Smooth Settings to Discrete Geometry

While Ricci Flow is classically defined on smooth manifolds, there has been considerable interest in transferring these ideas to *discrete* or combinatorial settings such as polyhedral surfaces, graphs, and complex networks. The general question is how to define concepts like “curvature” when one does not have a smooth manifold or a

Riemannian metric in the usual sense. Instead, discrete analogs focus on adjacency, distances along edges, and combinatorial properties that mimic or reflect continuum notions.

For surfaces composed of polygons (triangulations), one can define the curvature at a vertex via angle deficits, a concept dating back to classical differential geometry of polyhedral surfaces. However, for higher-dimensional graphs and networks that are not neatly embedded in any Euclidean space, a more general curvature definition is needed—one that depends mostly on the underlying distances and probability measures rather than an explicit embedding.

3.2 Optimal Transport and Ollivier's Ricci Curvature

A major breakthrough in defining a curvature notion for general metric spaces (including discrete networks) came via *optimal transport*. Historically, the optimal transport problem, originating in the work of Gaspard Monge in the 18th century, asks how to map one mass distribution into another with minimal transportation cost. Subsequent reformulations by Leonid Kantorovich turned this into a linear optimization problem known as the *Kantorovich relaxation*.

If (X, d) is a metric space, and we have two probability measures μ and ν on X , the *Wasserstein distance* (also called the earth mover's distance) measures how much "effort" is needed to move mass from μ to ν , given the metric d . Specifically, one solves an optimization problem that tries to minimize the total cost of moving infinitesimal amounts of mass from one location to another.

Yann Ollivier harnessed this framework to define a notion of *coarse Ricci curvature* on general metric measure spaces. Ollivier's definition, now widely called *Ollivier Ricci curvature*, is based on considering small probability balls of radius ε around points (or sometimes discrete probability measures concentrated on nearest neighbors in a graph) and calculating how much these small balls cost to move one onto the other under optimal transport. If moving these balls requires comparatively more effort than just their pairwise distance would suggest, the edge or connection between them is deemed negatively curved; if it requires less effort, it is positively curved. This emerges from an analog of the well-known statement in Riemannian geometry that Ricci curvature controls how geodesic balls deviate from each other, which can be recast in terms of mass transport.

CHAPTER 4

APPLICATION TO COMPLEX NETWORKS

To adapt Ollivier's construction to a graph $G = (V, E)$ where V is the set of nodes and E is the set of edges (possibly with weights), one often assigns to each node x a probability measure m_x . A typical choice is to concentrate the mass uniformly on x 's neighbors, possibly with some parameter α to keep a fraction of the mass at x itself. Let $W(m_x, m_y)$ denote the Wasserstein distance between m_x and m_y . The Ollivier Ricci curvature $\kappa(x, y)$ along the edge (x, y) is defined as

$$\kappa(x, y) = 1 - \frac{W(m_x, m_y)}{d(x, y)},$$

where $d(x, y)$ is the usual shortest-path distance (or a weight-based distance) between x and y . Intuitively:

- If many of the neighbors of x align well with the neighbors of y , then $W(m_x, m_y)$ is relatively small compared to $d(x, y)$, giving a larger curvature.
- If the neighbors do not overlap much, $W(m_x, m_y)$ will be comparatively large, implying smaller (or possibly negative) curvature.

This lines up with the idea that a highly "clustered" or cohesive set of nodes—often indicating an underlying community—acts more like a positively curved region in the manifold analogy, whereas edges bridging distant clusters reflect negative curvature. These insights lead to a method for analyzing and partitioning networks by focusing on edges with specific curvature characteristics.

4.1 ARI, Modularity, and Performance

In the study of complex networks, a fundamental task is to identify densely connected groups of nodes, commonly referred to as *communities*. These communities often correspond to meaningful substructures such as friend groups in social networks, functionally related proteins in biological networks, or topics in citation networks. There are numerous algorithms that aim to extract communities—ranging from graph partitioning heuristics and centrality-based edge removal to statistical and probabilistic methods.

The Ricci Flow-based technique for network community detection focuses on the geometric viewpoint: an edge with significantly negative Ricci curvature may signify a *bridge* between distinct communities, whereas edges with positive curvature are typically nestled within cohesive communities. Iteratively adjusting edge weights according to Ollivier Ricci curvature has the effect of “magnifying” bridging edges and “contracting” internal edges, ultimately making a subsequent threshold-based cut reveal the inherent clusters.

Here are the major steps (in broad terms) for using Ricci Flow to detect communities:

1. **Initialization:** Assign an initial weight to each edge (x, y) . Often, one starts with uniform weights or with weights based on an existing property such as adjacency or similarity.
2. **Probability Measures:** Choose how to define the measure m_x at each node x . A popular simple choice is to put uniform weight on all neighbors of x , ensuring $\sum_{v \in \text{neighbors}(x)} m_x(v) = 1$. Other weighting schemes (e.g., discounting more distant neighbors) can also be used.
3. **Curvature Computation:** For each edge (x, y) , compute the Ollivier Ricci curvature $\kappa(x, y)$ by solving the discrete optimal transport problem and using

$$\kappa(x, y) = 1 - \frac{W(m_x, m_y)}{d(x, y)}.$$

4. **Discrete Ricci Flow Update:** Adjust the edge weight according to

$$w_{xy}^{(i+1)} = w_{xy}^{(i)} - \eta \kappa_{xy}^{(i)} d_{xy}^{(i)}.$$

Often, one sets $\eta = 1$ and updates all edges simultaneously, then recomputes shortest path distances $d(\cdot, \cdot)$ for the next iteration. The total number of iterations can be chosen based on convergence criteria or practical heuristics.

5. **Network Surgery:** After a certain number of iterations, examine the distribution of edge weights. Typically, bridging edges (those connecting separate communities) will have grown in length. Choose a threshold T such that edges with $w_{xy} > T$ are considered “cuts,” removing them from the graph. The connected components that remain are taken as the identified communities.
6. **Post-Processing:** If a graph has hierarchical communities, additional steps (e.g., repeating the process within subcomponents) might be performed to further subdivide the clusters.

This pipeline is quite flexible in terms of parameter choices (e.g., the measure definition, the number of flow iterations, the threshold for edge cuts) and can adapt to various network topologies. The next subsections explain how to measure the resulting partition quality, focusing on two widely employed tools: the *Adjusted Rand Index (ARI)* and *Modularity*.

4.1.1 Adjusted Rand Index (ARI)

4.1.1.1 General Definition

The *Adjusted Rand Index (ARI)* is a popular external validation measure to compare a discovered clustering with a known ground-truth partition. Suppose you have a set of n items (in our case, the n nodes of a network). Let $C = \{C_1, \dots, C_r\}$ be a partition of these items into r clusters found by some method, and let $G = \{G_1, \dots, G_s\}$ be the ground-truth (or reference) partition into s clusters. The Rand Index (RI) measures the fraction of item pairs that are *consistently* assigned in both partitions (i.e., either assigned together in both or assigned to different clusters in both).

Formally, the Rand Index is given by:

$$\text{RI}(C, G) = \frac{a + d}{a + b + c + d},$$

where

- a is the number of pairs of items that are in the same cluster in C *and* in the same cluster in G ,

- b is the number of pairs of items that are in the same cluster in C but in different clusters in G ,
- c is the number of pairs that are in different clusters in C but in the same cluster in G ,
- d is the number of pairs that are in different clusters in C and in different clusters in G .

Because $a + d$ counts all the agreements (put together or kept apart) and $b + c$ counts the disagreements, $\text{RI}(C, G)$ is between 0 and 1, with 1 meaning a perfect match of the partitions.

However, the Rand Index does not correct for chance agreement. The *Adjusted Rand Index* refines this by subtracting the expected RI of random partitions and rescaling. One can define:

$$\text{ARI}(C, G) = \frac{\text{RI}(C, G) - \text{Expected}[\text{RI}]}{\max(\text{RI}) - \text{Expected}[\text{RI}]},$$

which yields a value that ranges from 0 (or negative, depending on definition) up to 1. Here, 1 indicates the clustering C exactly matches the ground-truth G , whereas an ARI near 0 suggests random agreement.

4.1.1.2 Applying ARI to Ricci Flow-based Community Detection

When applying Ricci Flow on a network, one typically obtains a final partition of the graph by removing edges beyond a certain length. If a ground-truth labeling exists, we compute the ARI to assess how well these communities align with the reference. If ARI is high, that means the geometric approach successfully captured the intended grouping. By examining the ARI as a function of the threshold, one can also decide the best cutoff for the “network surgery.” In practice, one might do a range of threshold values, measure the ARI for each, and pick the threshold that maximizes it (assuming the ground-truth is known). In contexts where the ground-truth partition is not known, we might rely on internal validation measures, such as modularity, to guess a suitable threshold.

4.1.2 Modularity

4.1.2.1 Definition of Modularity

Modularity is one of the most commonly used internal metrics for community detection in networks. Proposed initially by Newman and Girvan, it quantifies how

well a particular partition of the network divides the nodes into communities that are dense internally and sparse between each other.

Let us consider a network with n nodes and m edges (or total edge weight if it is a weighted graph). For a given partition of the network into k communities, the modularity Q is computed as:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j),$$

where A_{ij} is the adjacency matrix (or the weight matrix), d_i is the degree (or sum of weights) of node i , c_i is the community label of node i , and $\delta(c_i, c_j)$ is 1 if i and j are in the same community and 0 otherwise. The term $\frac{d_i d_j}{2m}$ approximates the expected number of edges (or expected weight) between i and j if edges are distributed randomly but respect node degrees. High modularity indicates that the actual number of intra-community edges is significantly above random expectation.

4.1.2.2 Modularity as a Stopping or Surgery Criterion

When using Ricci Flow for community detection, one can track how modularity evolves as edges get re-weighted and as one tries different thresholds for cutting. Typically, there is an intuitive sweet spot where further cutting does not substantially improve the modularity and might begin to over-segment the network.

A practical strategy is as follows:

1. Perform a fixed number of Ricci Flow iterations.
2. For a range of potential cut thresholds T_1, T_2, \dots, T_r , remove edges with weight above T_j .
3. Compute modularity Q_j for each T_j .
4. Select the threshold T_j that yields the maximum Q_j preceding a drop in modularity.

This threshold selection process is akin to the notion of “neck pinches” or “surgeries” in the manifold setting: a large weight often signifies a bridging structure (negative curvature region grown large) that is “pinching off” from the main components. If an external ground-truth is known, one may prefer the threshold that simultaneously optimizes ARI and modularity. In the absence of external labels, maximizing modularity is a common choice to define the “best” partition.

4.1.3 Interpreting Network “Surgery” in This Context

A hallmark of the classical Ricci Flow with surgery on manifolds is that when the flow develops singularities (often visualized as “neck pinches”), the manifold is physically separated into topologically distinct pieces. Drawing an analogy, in discrete Ricci Flow on graphs, the edges that grow large (due to negative curvature) can be viewed as “singularities” or bridging regions, reminiscent of the neck that pinches in a continuous manifold. The act of removing these edges at some iteration is the direct analog of performing surgery on the manifold. After removing these “necks,” the graph breaks into connected components, each presumably representing a dense or well-curved subregion, i.e., a community.

In practice, we carry out this network surgery step once or multiple times, balancing the preservation of meaningful connectivity with the desire to isolate truly separated clusters. Because many real networks can exhibit hierarchical or multi-level community structures, it is possible that each sub-component can be further refined if we continue the process within it. This multi-scale approach can be repeated if one suspects nested communities.

4.1.4 Algorithmic Complexity and Practical Considerations

While the continuous Ricci Flow PDE can be computationally demanding in the smooth case, the discrete version has its own set of computational challenges. The major cost typically arises in:

1. **Wasserstein Distances:** Computing $W(m_x, m_y)$ for each edge (x, y) . In a graph with n nodes and e edges, if we attempt an exact solution of the optimal transport problem, it can become computationally expensive. However, in many practical discrete settings, especially with local probability measures m_x that concentrate mass on immediate neighbors, $W(m_x, m_y)$ can often be computed with simpler combinatorial formulas or approximations.
2. **Repeated Distance Computation:** After each iteration, we update edge weights and potentially must recompute shortest-path distances $d(x, y)$ for all node pairs or for edges. If e is large, repeated all-pairs computations might be costly, unless efficient incremental shortest-path algorithms or approximate methods are used.

Despite these costs, modern computational resources and heuristics usually make discrete Ricci Flow feasible for networks of moderate size. For very large networks

(millions of edges), one might rely on approximations, sampling techniques, or simplified versions of curvature (e.g., proxies to Ollivier Ricci curvature).

In practice, one needs to choose three important parameters:

1. **Number of Ricci Flow Iterations:** Stopping too soon might not emphasize bridging edges enough to isolate communities; iterating too long might lead to degeneracies (e.g., certain edges become extremely large or extremely small). An empirical strategy is to run a moderate number (e.g., 10–20 iterations) and check if measures like ARI (if a ground truth is available) or modularity saturate.
2. **Cut Threshold for Network Surgery:** Typically chosen by scanning multiple thresholds and evaluating a measure of clustering quality (e.g., modularity) or ARI.
3. **Mass Distributions:** The simplest is uniform distribution on neighbors (sometimes with or without a fraction of mass at the node itself). More sophisticated distributions might downweight distant neighbors, especially in large or weighted networks.

4.1.5 Broader Theoretical Context and Future Directions

From a more theoretical vantage point, *why* does curvature—in the sense of Ollivier—detect communities so well? Intuitively, the geometry of a manifold with positive curvature is reminiscent of a cohesive, “ball-like” region, while negatively curved regions show hyperbolic expansions akin to branching. In network terms, cohesive subgraphs correspond to a “positive curvature signature” because local random walks or local mass distributions align more easily, whereas bridging edges or tree-like expansions create a negative curvature effect.

Future directions could include:

- Exploring *Forman-Ricci* curvature or other discrete curvature notions to see how they compare in community detection or anomaly detection tasks.
- Combining *higher-order* or *simplicial* network topologies, in which edges, triangles, and higher-dimensional simplices each carry a curvature notion, potentially refining the detection of substructures.
- Investigating multi-scale or hierarchical surgery, systematically refining each sub-community using iterative curvature-based updates.

All of these highlight that curvature-based community detection offers a distinctive geometric lens on networks, bridging rich ideas from differential geometry to the realm of data science and large-scale graph analysis.

CHAPTER 5

THE DEVELOPED CODE

The objectives of the developed code were

- **Implementation of a Ricci Flow method** able to evaluate the Ollivier-Ricci curvature of a given graph and update its edges' weights accordingly. Then we wanted the method to perform surgery on weakly connected edges to allow for community detection.

For these purposes we relied on Ollivier-Ricci library developed by Ni et al. [8], see [section 5.1](#).

- **Plotting graphs and communities** to appreciate the behavior of the method and obtain graphical results.
- **Testing the method** on synthetic graphs, trying to benchmark with an analogue test made by Ni et al. [2]. Results and further details are given in [section 5.2](#).
- **Evaluation of performances** by comparing the method with other commonly used community detection methods.
- **Application on real world data.** To do this we chose Zachary's Karate Club graph, which is directly accessible from Networkx library. Results and further details are given in [section 5.3](#).

In the this chapter we present the main ideas and tools related to the code. To get more insights on how the code as been constructed and subdivided (i.e. various classes and functions) we recommend consulting [CodeDocumentation](#) or the GitHub of the whole project: [RicciFlowNetwork](#).

5.1 The setup

To facilitate the computation of Ricci curvature within networks, we made use of the GraphRicciCurvature library as our starting point. This Python library is part of a more comprehensive library on Ricci curvature for networks. The latter provides tools to compute two discrete Ricci curvatures: Ollivier-Ricci and Forman-Ricci; it supports the analysis of both weighted and unweighted graphs. In addition, it offers basic methods for graph surgery and evaluation of the *adjusted rand index* (ARI).

Each graph we employed was generated with Networkx library. We also used its methods as a starting point for graphical representations. For plotting we implemented `GraphDrawer` class which, among other functionalities, allows us to see the detected communities separated in subgraphs. Nodes are colored with their corresponding community color, giving a visual indication of method's performance.

We used this setup to implement the code workflow depicted in fig. 5.1.

5.2 Tests on Synthetic Graphs

5.2.1 Stochastic Block Model Test Graph

To start testing our code we choose a Stochastic Block Model (SBM) graph. A SBM graph is a type of random graph model used to generate networks with a predefined community structure. It is an extension of the Erdos–Rényi, where nodes are divided into communities, and the probability of an edge between two nodes depends on their community membership. It is characterized by N nodes, k communities and a connectivity matrix P_{ij} which represents the probability of an edge between a node in community i and a node in community j . Since communities are predefined when generating the graph, we can directly compare the detected communities with the true ones.

For our test we set

$$N = 500, \quad k = 2, \quad P = \begin{bmatrix} 0.20 & 0.03 \\ 0.03 & 0.20 \end{bmatrix}$$

with every initial weight equal to one.

Curvature values for the initial SBM graph are shown in fig. 5.2a, while on the right we have the updated curvatures after 10 iterations of Ricci Flow. We can see that the community structure of this graph is already well established; Ricci Flow has however

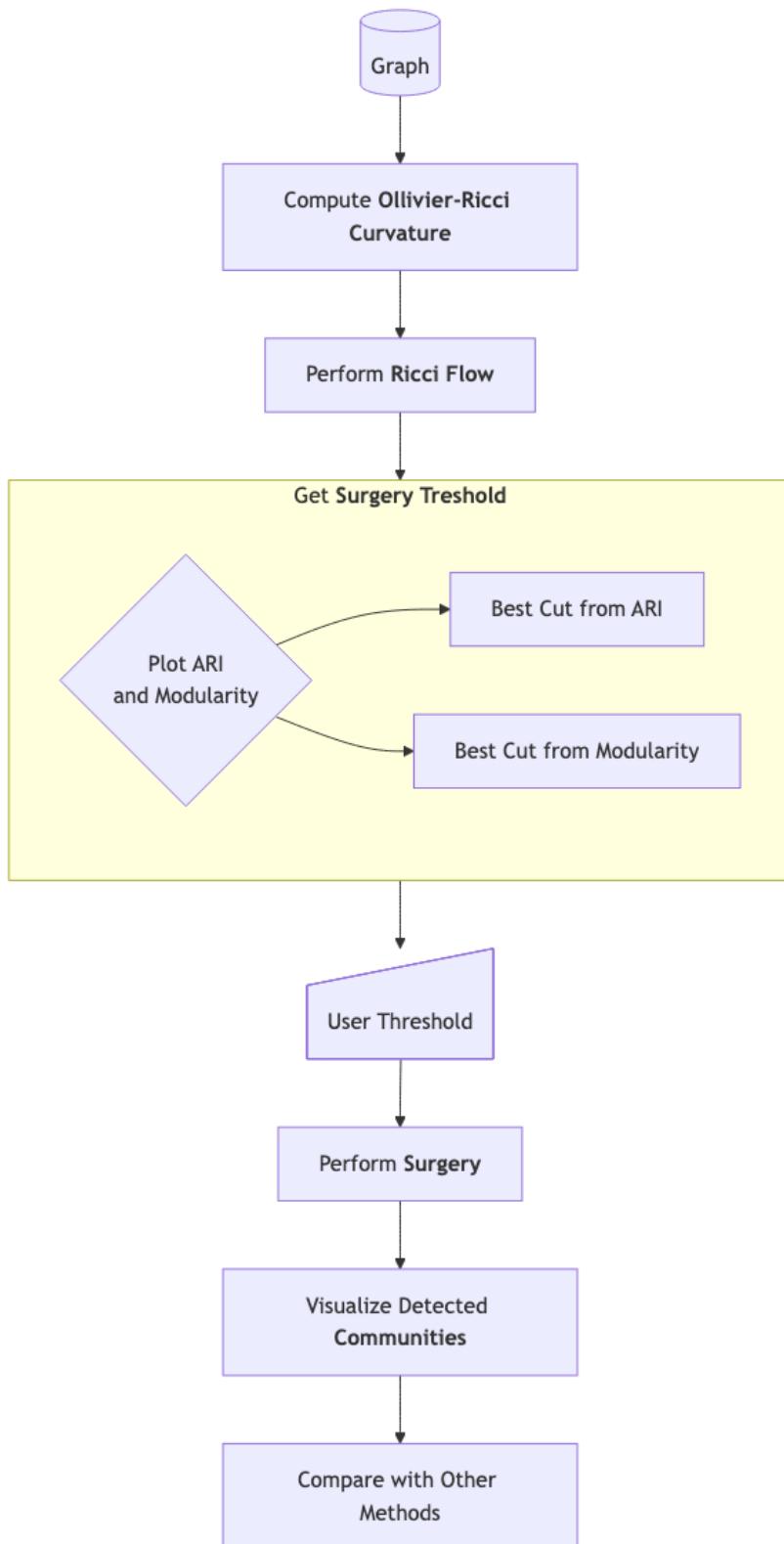
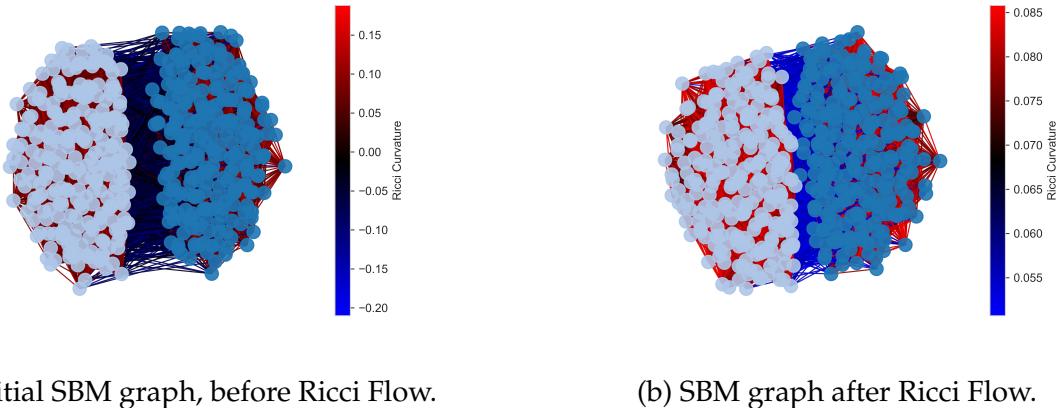


Figure 5.1 Code workflow.



(a) Initial SBM graph, before Ricci Flow. (b) SBM graph after Ricci Flow.

Figure 5.2 Comparison of SBM graph before and after having applied 10 iterations of Ricci Flow on edges.

highlighted the weak connection of central inter community nodes as one can see in fig. 5.2b.

In fig. 5.3 we see that surgery with a cutoff between 1 and ≈ 1.5 leads to a perfect distinction between the two communities, i.e. an ARI of 1.

Lastly, fig. 5.4a depicts the graph after surgery with a cutoff between 1 and ≈ 1.5 . As expected we got a separation into two distinct clusters. Fig. 5.4b shows the two communities corresponding to the two connected components.

5.2.2 Lancichinetti-Fortunato-Radicchi Test Graph

As a more complex synthetic graph for testing we used a Lancichinetti-Fortunato-Radicchi (LFR) benchmark graph. LFR graphs are widely used for testing community-detection algorithms because they produce networks with heterogeneous (scale-free) degree distributions and community-size distributions, making them more realistic than simpler models. Nodes are assigned to communities according to specified power-law exponents, and a “mixing” parameter controls the fraction of edges that connect different communities.

For our test we built a graph with 500 nodes ($n = 500$) with a degree distribution exponent $\tau_1 = 3$ and a community-size exponent $\tau_2 = 1.5$. For the mixing parameter we chose $\mu = 0.2$ indicates a relatively strong community structure by limiting the proportion of inter-community edges. We set each community to have a minimum of 20 nodes and a maximum of 70 nodes. The expected average degree is set to 20, with a maximum node degree capped at 50.

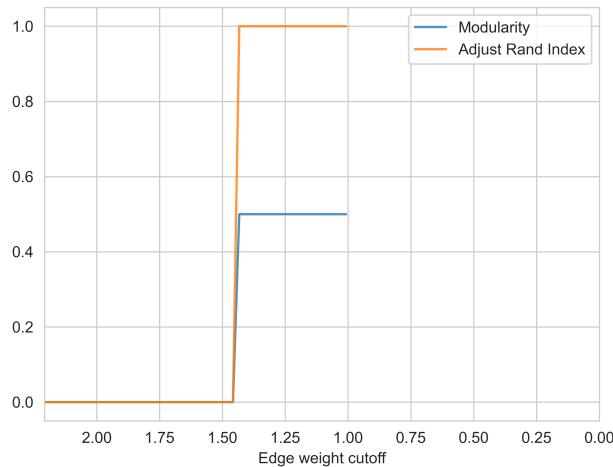


Figure 5.3 SBM graph's ARI and modularity behavior for different surgery cutoffs.

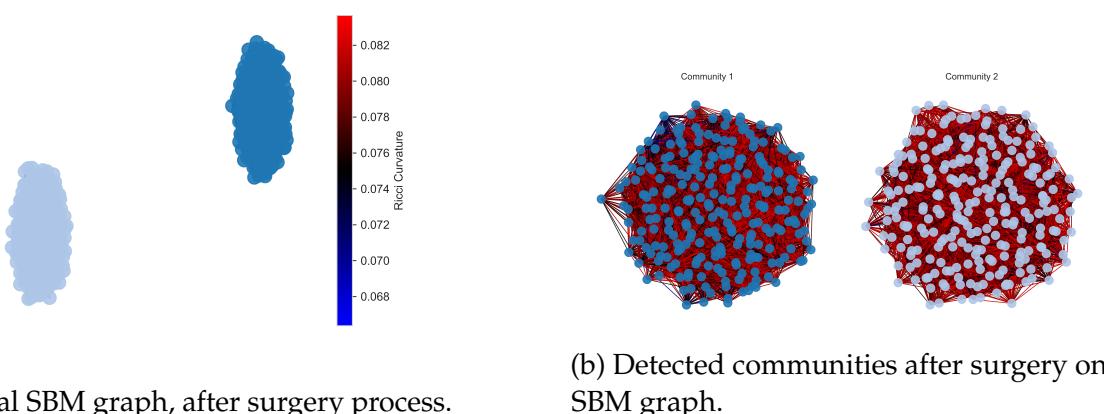


Figure 5.4 Comparison of SBM graph after surgery and corresponding connected components (i.e. the detected communities).

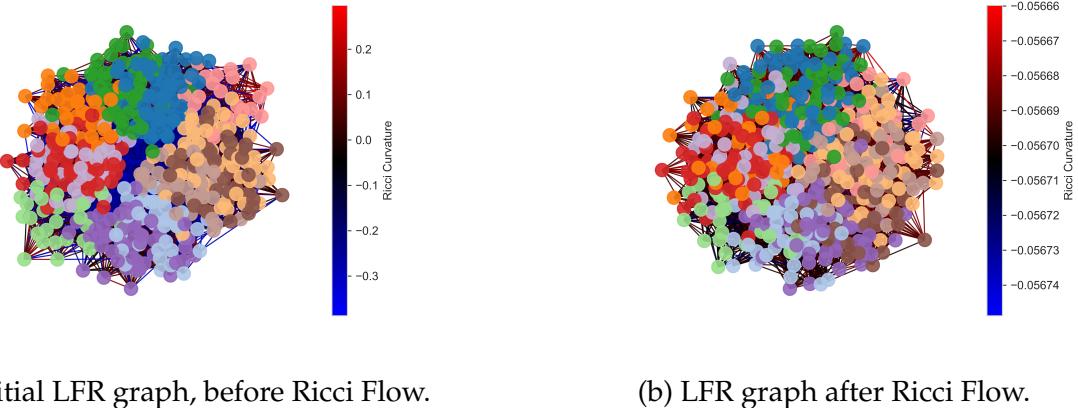


Figure 5.5 Comparison of LFR graph before and after having applied 40 iterations of Ricci Flow on edges.

For this graph we applied 40 iterations of Ricci Flow as its structure is more complex than the previous SBM graph (see fig. 5.5a). In fig. 5.5b we see a convergence of the curvature values after Ricci Flow; the high number of nodes and communities makes difficult to visualize properly the graph, for this reason we introduced also an histogram plot of weights and curvature values in the final code implementation for the Karate club graph (see section 5.3).

Fig. 5.6 shows the behavior of ARI and modularity depending on the chosen cutoff point. Also in this case we see that it is possible to obtain an ARI of 1.

In fig. 5.7 we have the graph after surgery (we chose as cutoff ≈ 1.12 to get an ARI of 1) and we can see that it got divided into connected components with all the nodes of the same color, i.e. same community. In this case we did not produce a plot community by community due to the high number of clusters.

5.2.3 Further Tests on Lancichinetti-Fortunato-Radicchi Graphs

As a final test for our code we decided to try reproducing a result presented in the work of Ni et al. [2] (see figure 8a at page 12). In particular, it is about applying Ricci Flow on different LFR graphs differing for the modularity parameter μ . Of course one expects the ARI to decrease with increasing values of μ as the community structure becomes less defined.

In fig. 5.8 we present our results with OTD method for curvature evaluation using first OTD and then ATD

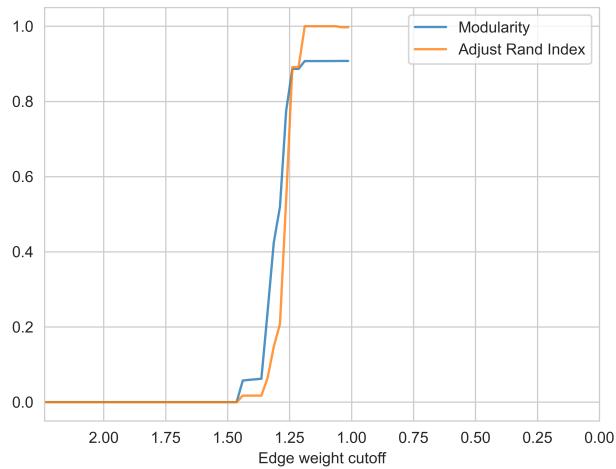


Figure 5.6 LFR graph's ARI and modularity behavior for different surgery cutoffs.

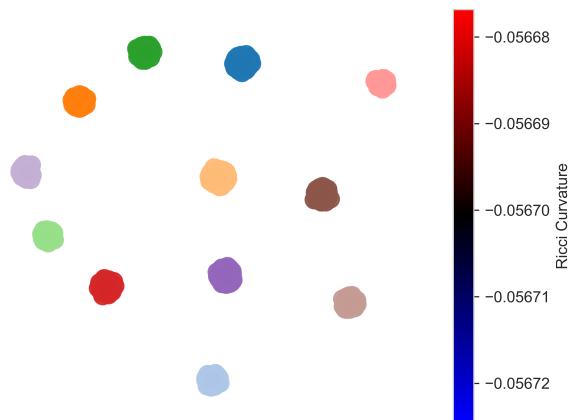


Figure 5.7 Final LFR graph, after surgery process.

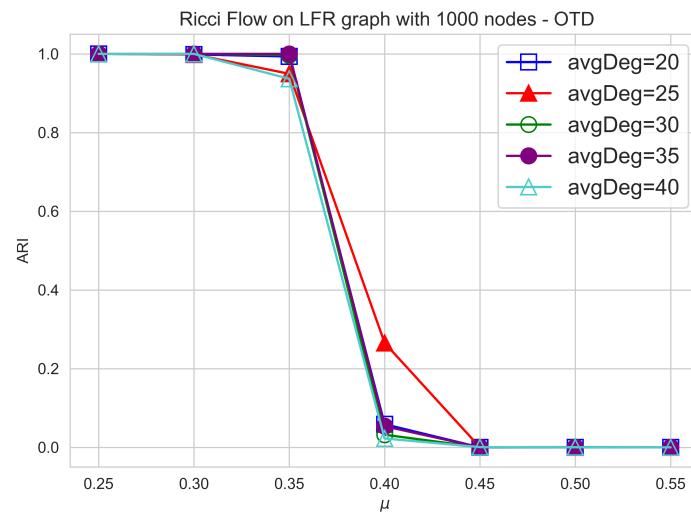


Figure 5.8 LFR OTD

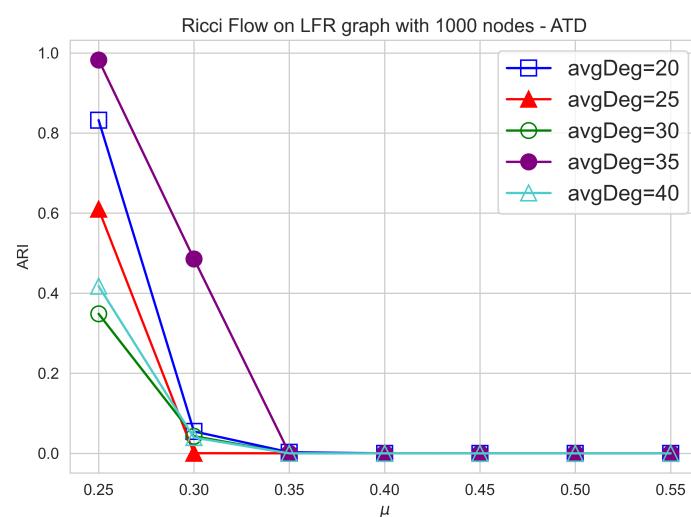
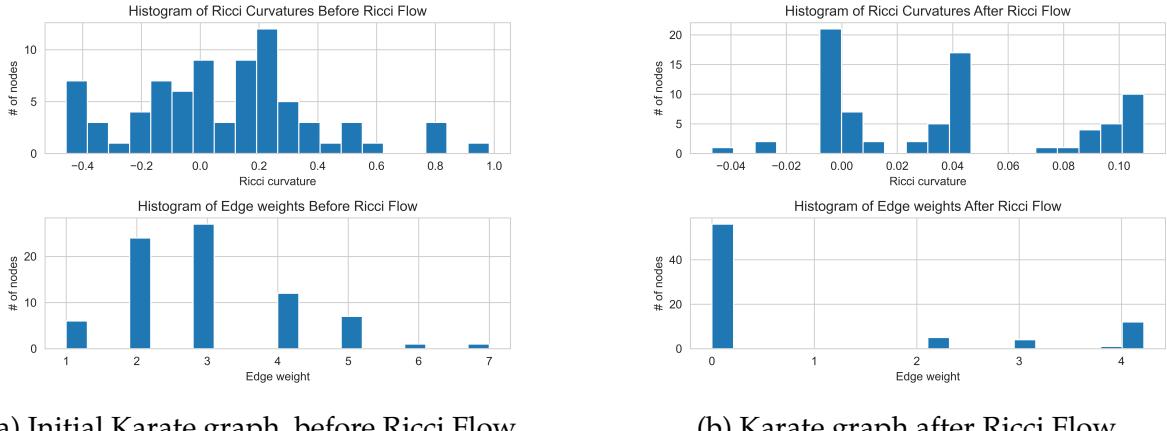


Figure 5.9 LFR ATD



(a) Initial Karate graph, before Ricci Flow.

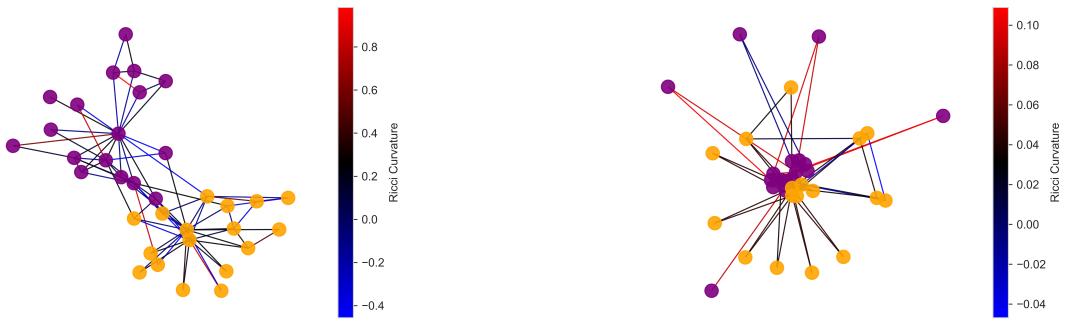
(b) Karate graph after Ricci Flow.

Figure 5.10 Comparison of curvature values and weights before and after having applied Ricci Flow on edges.

Our graphs do not match exactly the plots of Ni et al., as for them the loss of ARI starts from higher values of μ . This could be for various reasons:

- It is unclear which parameter they used to generate LFR graphs. For example, using Networkx it is not possible to control exactly the number of communities (only a minimum and maximum number can be set), so they probably generated their graphs in a different but unspecified way.
- They used an average ARI, probably generating the same graph multiple times. We generated the graph and applied the Ricci Flow just once due to an already high execution time.
- Their surgery process it is not specified in detail. They might have used multiple surgeries over the same Ricci Flow process to remove possible singularities affecting the final performance (even though this should not be the case for this kind of graph).

5.3 Application to Zachary's Karate Club Graph



(a) Initial Karate graph, before Ricci Flow.

(b) Karate graph after Ricci Flow.

Figure 5.11 Comparison of Karate graph before and after having applied Ricci Flow on edges.

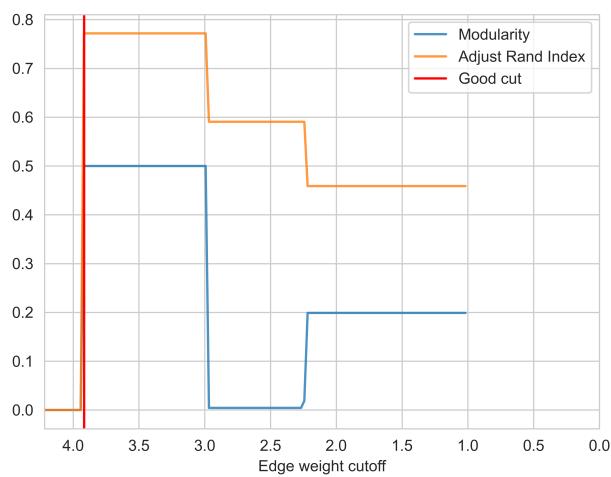


Figure 5.12 Karate acc

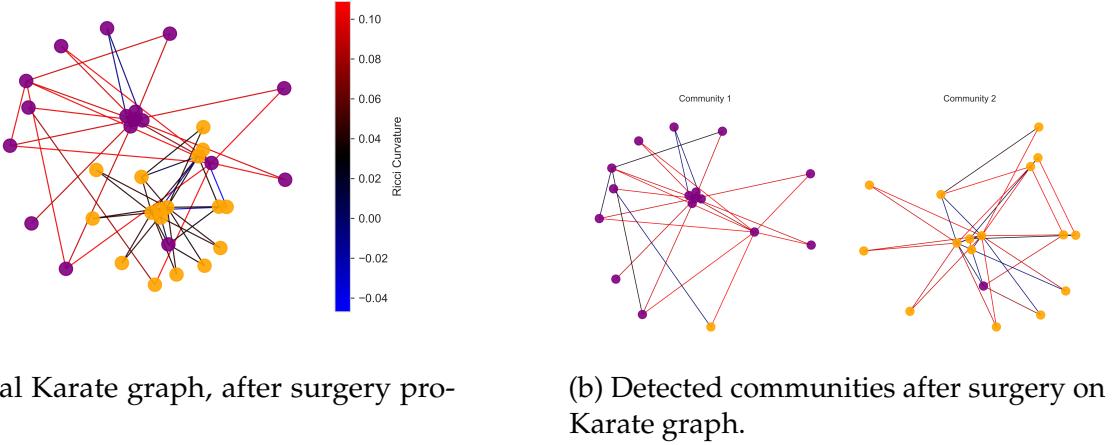


Figure 5.13 Comparison of Karate graph after surgery and corrensponding connected components (i.e. the detected communities).

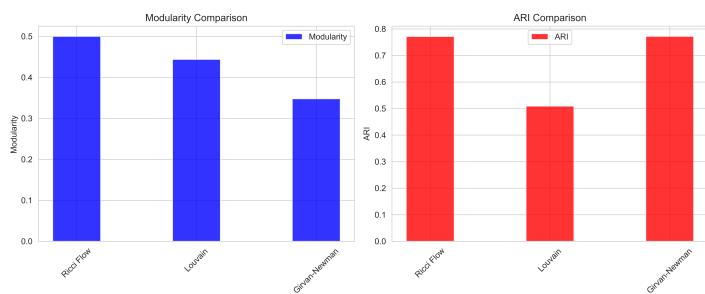


Figure 5.14 Karate acc

CHAPTER 6

CONCLUSIONS AND FUTURE DIRECTIONS

APPENDIX A

METRIC OF A 2-SPHERE

REFERENCES

- [1] Wayne W. Zachary. "An Information Flow Model for Conflict and Fission in Small Groups". In: *The University of Chicago Press Journals* (1997). URL: <https://doi.org/10.1086/jar.33.4.3629752>.
- [2] Chien-Chun Ni et al. "Community Detection on Networks with Ricci Flow". In: (2019). arXiv: [1907.03993 \[cs.SI\]](https://arxiv.org/abs/1907.03993). URL: <https://arxiv.org/abs/1907.03993>.
- [3] Henri Poincaré. "Analysis Situs". Trans. by John Stillwell. In: *Journal de l'École Polytechnique* 2 (1895).
- [4] Martin D. Crossley. *Essential Topology*. Springer, 2005. DOI: [10.1007/1-84628-194-6](https://doi.org/10.1007/1-84628-194-6).
- [5] Henri Poincaré. "Cinquième complément à l'analysis situs". Fifth Complement to Analysis Situs. Trans. by John Stillwell. In: *Rendiconti del Circolo Matematico di Palermo* 18 (1904 2008), pp. 45–110.
- [6] Lee J. M. *Introduction to Smooth Manifolds*. Second Edition. Springer, 2013. DOI: [10.1007/978-981-99-0565-2](https://doi.org/10.1007/978-981-99-0565-2).
- [7] Loring W. Tu. *An Introduction to Manifolds*. Second Edition. Springer, 2011. DOI: [10.1007/978-1-4419-7400-6](https://doi.org/10.1007/978-1-4419-7400-6).
- [8] Ni C.C., Cruceru C., and Barrett S. *GraphRicciCurvature*. <https://github.com/saibalmars/GraphRicciCurvature.git>. 2019.
- [9] Lee J. M. *Introduction to Riemannian Manifolds*. Second Edition. Springer, 2018. DOI: [10.1007/978-3-319-91755-9](https://doi.org/10.1007/978-3-319-91755-9).
- [10] Timo Weigand. "Introduction to String Theory". Lecture Notes. 2012.
- [11] J. Polchinski. *String theory. Vol. 1: An introduction to the bosonic string*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, Dec. 2007. DOI: [10.1017/CBO9780511816079](https://doi.org/10.1017/CBO9780511816079).
- [12] Mauro Carfora et al. "Ricci Flow from the Renormalization of Nonlinear Sigma Models in the Framework of Euclidean Algebraic Quantum Field Theory". In: (2019). arXiv: [1809.07652 \[math-ph\]](https://arxiv.org/abs/1809.07652). URL: <https://arxiv.org/abs/1809.07652v2>.
- [13] Richard H Bamler. "Recent developments in Ricci flows". In: (2021). arXiv: [2102.12615 \[math.DG\]](https://arxiv.org/abs/2102.12615). URL: <https://arxiv.org/abs/2102.12615>.
- [14] B. Chow and D. Knopf. *The Ricci Flow: An Introduction*. American Mathematical Society, 2004. ISBN: 0-8218-3515-7.

- [15] Ryan A. Rossi and Nesreen K. Ahmed. "The Network Data Repository with Interactive Graph Analytics and Visualization". In: AAAI. 2015. URL: <https://networkrepository.com>.