Discrete Ricci Flow applied to Facebook Ego Network

Subtitle

Lorenzo Fabbri

Giancarlo Oancia

This essay is submitted for the exam of
Complex Networks

Graphics/UniversityCrest.png

Department of Physics
University of Bologna
17/11/2024

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. Planar graphs with different community structures are used as datasets, with the goal of identifying pre-labelled communities using the Ricci Flow method.

CONTENTS

CHAPTER 1

INTRODUCTION

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 ways to approach this problem by leveraging concepts from differential geometry [3].

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 Ricci tensor), leading to a smoothing process over time. Ollivier-Ricci curvature, a discretization 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 [3].

In this project, we explore the application of Ricci Flow in community detection by focusing on planar graphs, i.e. graphs that can be drawn on a plane without any edges crossing each other. By applying Ollivier-Ricci curvature to this network, we aim to assign and iteratively modify the curvature of edges, enhancing intra-community cohesion while stretching inter-community connections. Starting from graphs in which all the edges have weights set to 1, the flow will introduce weight diversity based on the topological characteristics of the graph, allowing observation of the natural curvature-driven "deformation" of the network.

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*.

CHAPTER 2

RICCI FLOW METHOD

Given a manifold M, a *Ricci Flow* is a smooth family $g(t)$, with $t \in [0, T)$, of Riemannian metrics satisfying the evolution equation

$$\partial_t g(t) = -2Ric_{g(t)} \tag{2.1}$$

where $Ric_{g(t)}$ is the Ricci tensor for the metric $g(t)$, i.e. the contraction of the Riemann tensor obtained with $g(t)$; in components $Ric^{ij}_{g(t)} \equiv g_{kl}(t)R^{iklj}$ [1].

The metric is deformed in a way which smooths out its irregularities, like in a heat diffusion process. Indeed, using harmonic local coordinates, the Ricci tensor approximates the Laplacian[1] of the metric up to lower-order terms in derivatives of the metric tensor:

$$Ric_g = -\frac{1}{2}\triangle g + \textit{lower-order terms} \tag{2.2}$$

which plugged into Equation 1 results in a nonlinear heat equation [2].

By viewing a network as a discrete analogue of a 3-manifold, we can interpret the process of community detection as a geometric decomposition analogous to how Ricci Flow decomposes a 3-manifold into distinct geometric regions. Each community in the network represents a cohesive region of the network, much like a geometric region in a manifold. Inter-community edges are analogous to the boundaries between geometric regions in the manifold while intra-community edges represent connections within a homogeneous geometric region.

[1] Actually the operator $\triangle$ in Equation 2 is the Laplace-Beltrami operator, a generalization of the Laplace operator to functions defined on Riemannian and pseudo-Riemannian manifolds.

2.1 Ollivier's Ricci Curvature

Ollivier's Ricci curvature is a discrete adaptation of the classical Ricci curvature defined for smooth manifolds, tailored for graph structures. In a network, nodes represent points, and edges between them denote "distances" or connections. Rather than relying on continuous geometry, Ollivier's Ricci curvature measures how well-connected neighboring nodes are by comparing the distances between their neighborhoods.

In a graph, each node x has a set of neighboring nodes, and we assign a probability distribution over this set, denoted as m_x. This distribution can be thought of as how much "mass" each neighbor carries. For example, the nodes x and y in Figure 1 each have three neighbors, and the respective probability distributions over their neighborhoods are denoted as m_x and m_y.

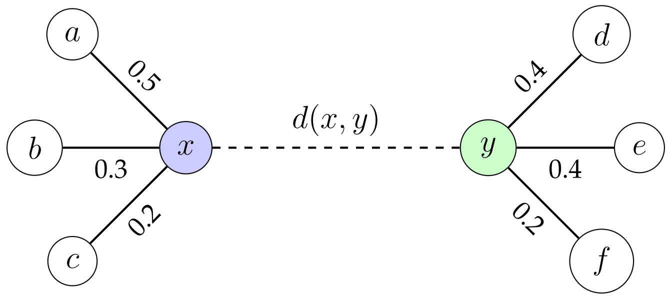

Figure 2.1 Nodes x and y with their respective neighborhoods and probability distributions. The weights on the edges represent the probability distribution m_x for node x's neighbors and m_y for node y's neighbors.

Formally, the Ollivier-Ricci curvature $\kappa(x, y)$ between two connected nodes x and y is defined using the *earthmover's distance*, also known as the *Wasserstein distance*, between probability distributions on the neighborhoods of x and y:

$$\kappa(x, y) = 1 - \frac{W_1(m_x, m_y)}{d(x, y)} \tag{2.3}$$

where

- $d(x, y)$ is the shortest path distance between nodes x and y.
- $W_1(m_x, m_y)$ is the Wasserstein-1 distance between the probability distributions of neighboring nodes around x and y:

$$W_1(m_x, m_y) = \mathit{inf}\left\{ \int_{X\times Y} d(x, y) d\gamma(x, y) \,\middle|\, \gamma \in \Gamma(m_x, m_y) \right\} \tag{2.4}$$

with γ being a probability measure on $X \times Y$ (the probability spaces of the neighboring nodes around x and y) and $\Gamma(m_x, m_y)$ indicating the collection of all possible transportation plans.

This curvature provides a natural way to quantify the coupling between two neighborhoods. High positive curvature implies that the neighborhoods of x and y are tightly connected, whereas negative curvature indicates sparse or loosely connected regions [3].

Edges with high positive curvature tend to represent intra-community connections, while edges with low or negative curvature correspond to inter-community links. This distinction allows us to leverage geometric intuition for community detection, adjusting edge weights based on their curvature values.

2.2 The Discrete Ricci Flow Algorithm

The discrete Ricci flow algorithm modifies the concept of Ricci Flow to suit network structures by iteratively updating the weights of the edges according to their curvature values. The steps of the algorithm are as follows:

1. **Initialization**: Begin with a graph $G(V, E)$, where V is the set of vertices and E is the set of edges. Each edge $(x, y) \in E$ is assigned an initial weight, which can be based on the shortest path distance between nodes x and y, or on a predefined edge weight.

2. **Curvature Computation**: For each edge (x, y), compute the Ollivier-Ricci curvature $\kappa(x, y)$. The Wasserstein distance here is defined as the minimum total weighted travel distance to move m_x to m_y, i.e.

$$W(m_x, m_y) = \mathit{inf} \left\{ \sum_{u,v \in V} A(u, v) d(u, v) \right\} \tag{2.5}$$

   where $A(u, v)$ is a map $V \times V \to [0, 1]$ called *discrete transport plan*. It corresponds to the amount of mass moved from vertex $u \in V_x$ (a neighbor of x) to vertex $v \in V_y$ (a neighbor of y). The transport plan satisfies $\sum_{v \in V} A(u, v) = m_x(u)$ and $\sum_{u \in V} A(u, v) = m_y(v)$, where $m_x(u)$ and $m_y(v)$ are the probability distributions on the neighborhoods of x and y respectively. These conditions ensure that the total mass leaving node x matches the distribution m_x, and the total mass arriving at node y matches m_y.

3. **Edge Weight Update**: Modify the weights of the edges based on their curvature values. If $\kappa(x, y)$ is positive, reduce the edge weight, simulating a contraction of intra-community edges. Conversely, if $\kappa(x, y)$ is negative, increase the edge weight, stretching inter-community edges. The updated weight $w'(x, y)$ is given by:
$$w'(x, y) = w(x, y) \times (1 - \alpha \cdot \kappa(x, y)) \tag{2.6}$$
where α is a step size parameter controlling the rate of change.

4. **Iteration**: Repeat the process of computing curvature and updating edge weights for a specified number of iterations or until convergence is achieved. Over time, intra-community edges shrink (strengthen) while inter-community edges stretch (weaken), causing communities to become more distinct.

5. **Surgery Process**: Remove edges that contribute to singularities. Edges with a curvature value below a certain threshold, typically set ≈ -0.1, are pruned. It is worth mentioning that there are other techniques to modify the graph structure to deal with problematic regions, effectively 'surgically' altering the graph to maintain the flow's stability. Common ones are edge contraction (i.e. reducing the weight of edges that are causing high curvature) and node splitting (i.e. dividing a single node into multiple nodes while redistributing the connected edges among them).

6. **Community detection**: After the edge weights have been updated through the Ricci Flow iterations, we apply the Louvain method for community detection, which returns a partition of the network based on the modified edge weights from the Ricci Flow process, where each node is assigned to a community.

2.3 Convergence Criteria and Stopping Conditions

The theoretical foundation of the discrete Ricci Flow method for community detection is supported by several key results, one of the most important being the convergence guarantees provided by Theorem 4.1 from [3].

THEOREMThe Ricci flow associated with the Ollivier K_0-Ricci curvature [2] detects the community structure on $G(a, b)$ if $a > b \geq 2$. Specifically, the algorithm asymp-

[2] K_0-Ricci curvature is a modification of Ollivier-Ricci curvature which incorporates a parameter K_0 in order to adjust the curvature calculation; emphasizing certain aspects of community structure.

totically shrinks the weights of intra-community edges faster than the weights of inter-community edges.

Theorem 1 guarantees the convergence of the discrete Ricci Flow algorithm, meaning that after a finite number of iterations, the edge weights stabilize and no further significant changes occur. This convergence ensures that the community structure revealed by the Ricci Flow process is robust and does not depend on further iterations.

While Ricci Flow has inherent convergence properties that can lead to well-separated communities, some practical stopping conditions will often be required for implementation:

1. **Change in Edge Weights**: Monitor the change in edge weights between iterations. Convergence is achieved when the change falls below a predefined threshold ϵ, indicating that the algorithm has stabilized:

$$\max_{(x,y)\in E} |w_{\text{new}}(x,y) - w_{\text{old}}(x,y)| < \epsilon$$

2. **Community Structure Stability**: Validate the stability of the detected community structure across iterations. Ensure that identified communities remain consistent and meaningful by assessing their internal cohesion and external separation.

3. **Maximum Iterations**: Define a maximum number of iterations N_{max} to prevent excessive computation and to facilitate early stopping if convergence is not achieved within this limit:

$$\text{If } k > N_{\text{max}}, \text{ stop algorithm}$$

CHAPTER 3

IMPLEMENTATION IN A TOY MODEL

Before implementing the method discussed above to planar graphs, we first investigate Ricci curvature flow on three types of simple graphs: a stochastic block model (SBM) graph, a Barbell graph, and a Caveman graph. Each graph type exhibits distinct structural properties, facilitating a comprehensive exploration of Ricci flow dynamics.

3.1 *GraphRicciCurvature* Library

In addition to the main library for networks, *Networkx*, the developed code relies on the *GraphRicciCurvature* library, which provides efficient algorithms for calculating the Ollivier-Ricci curvature on graphs.

Key features of the *GraphRicciCurvature* library include:

- **OllivierRicci Class**: This class is initialized with the graph and a parameter `alpha`, which controls the curvature's sensitivity to edge weights. Smaller alpha values (closer to 0) are often used when one focuses on local community detection, small subgraphs, or clusters. Larger alpha values (closer to 1) are used when the interest is in the global structure of the graph, such as long-range connections and global topology.

- **Curvature Computation**: The method `compute_ricci_curvature()` calculates the Ricci curvature for all edges in the graph.

- **Edge Curvature Assignment**: After computing the curvatures, the edge attributes of the original graph are updated with the curvature values calculated by the `OllivierRicci` instance.

3.2 Results

The Ricci Flow method was tested on three different types of graphs to evaluate its functionalities and performance. Each graph has distinct structural properties, making them ideal candidates for assessing the method's ability to identify both densely connected intra-community edges and sparser inter-community edges. For each graph type the Ricci Flow process has been simulated using 15 iterations and an alpha of 0.5. The results are summarized below.

Stochastic Block Model (SBM) graph: This graph is specifically designed to simulate community structures with predefined groups. Nodes within the same group are more likely to connect, while connections between groups are less frequent. We created an SBM graph with two communities of 50 nodes each, where the intra-community edge probability was set high to ensure clear separation between the communities. The Ricci Flow method (with threshold for surgery set to -0.05) successfully identified the two distinct communities, as illustrated in Figure 2.

Barbell graph: The barbell graph consists of two complete graphs (dense communities) connected by a path (sparse inter-community links). It is an excellent test case for methods that detect clear community boundaries with sparse connections between them. In our experiments, we used a barbell graph with two communities of 20 nodes each, connected by a 5-node path. The method (with threshold for surgery set to -0.1) correctly detected the two communities, with the inter-community path clearly stretched due to the low curvature, as seen in Figure 3.

Caveman graph: A caveman graph is another community-like structure where several cliques (complete subgraphs) are loosely connected. This type of graph is useful for testing models aimed at detecting clique-based communities. We tested the method on a caveman graph with 5 cliques, each consisting of 10 nodes. The method (with threshold for surgery set to 0) successfully separated the cliques, identifying each as a distinct community, as shown in Figure 4.

Across all three graph types, the developed Ricci Flow-based method behaved as expected, demonstrating its capability to detect both dense and sparse community structures. The method shrank intra-community edges and stretched inter-community edges, which aligns with the behavior predicted by the underlying theory of Ollivier-Ricci curvature. The results from the tests on the SBM graph, Barbell graph, and Caveman graph are displayed in Figures 2, 3, and 4 respectively; edges with different colours belong to different communities.

Additional results and details, such as plots of the networks after Ricci Flow and after surgery, can be found on the project's GitHub page.

Graphics/ToyModelResults/SBM_results/BeforeRicciFlow.png

(a) Initial SBM graph

Graphics/ToyModelResults/SBM_results/C

(b) Detected communities.

Figure 3.1 Comparison of the initial SBM graph and the community detection result.

Graphics/ToyModelResults/Barbell_resluts/BeforeRicciFlow.png

(a) Initial Barbell graph

Graphics/ToyModelResults/Barbell_reslu

(b) Detected communities

Figure 3.2 Comparison of the initial Barbell graph and the community detection result.

Graphics/ToyModelResults/Caveman_resluts/BeforeRicciFlow.png

(a) Initial Caveman graph

Graphics/ToyModelResults/Caveman_reslu

(b) Detected communities

Figure 3.3 Comparison of the initial Caveman graph and the community detection result.

Chapter 4

Implementation for Planar Graphs

4.1 Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

4.2 Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Appendix A

Metric of a 2-sphere

References

[1] Richard H Bamler. *Recent developments in Ricci flows*. 2021. arXiv: 2102.12615 [math.DG]. URL: https://arxiv.org/abs/2102.12615.

[2] B. Chow and D. Knopf. *The Ricci Flow: An Introduction*. American Mathematical Society, 2004. ISBN: 0-8218-3515-7.

[3] Chien-Chun Ni et al. *Community Detection on Networks with Ricci Flow*. 2019. arXiv: 1907.03993 [cs.SI]. URL: https://arxiv.org/abs/1907.03993.

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