The Duality Between (Scalar)

Ali Fele-Paranj

October 29, 2024

1 Introduction

In this document, we explore the analogies between vector calculus in continuum settings and graph-based flow analysis. Specifically, we delve into the concepts of gradient, divergence, and curl on graphs, as well as the role of Graham's cohomology in characterizing loops and flows in networks.

2 Gradient, Divergence, and Laplacian Operators

In both continuum and graph settings, we define various operators to analyze scalar and vector fields. These operators have close analogies in both domains.

2.1 Continuum Setting

In continuum space (e.g., $\mathbb{R}^n$), we define:

- **Scalar field:** A function $f\colon \mathbb{R}^n \to \mathbb{R}$ that assigns a scalar value to each point in space.

- **Vector field:** A function $F\colon \mathbb{R}^n \to \mathbb{R}^n$ that assigns a vector to each point in space.

- **Gradient** ∇: Converts a scalar field to a vector field. The gradient of f is defined as ∇f, which points in the direction of maximum increase of f.

- **Divergence** div: Converts a vector field to a scalar field, measuring the net flux exiting or entering a region. For a vector field F, $\operatorname{div} F$ is defined as the sum of partial derivatives $\sum_i \frac{\partial F_i}{\partial x_i}$.

In this setting, the Laplacian operator, $\Delta = \operatorname{div}\nabla$, acts on scalar fields:

$$\operatorname{div}(\nabla f) = \Delta f$$

where Δf measures the "smoothness" or rate of change of f across the space. Additionally, we have:

$$\operatorname{curl}(\nabla f) = 0$$

indicating that gradients are *exact fields* with no circulation around any point, a property central to cohomological interpretation.

2.2 Graph Setting

For a graph with nodes and edges:

- **Gradient operator** ∇: Maps scalar functions on nodes (e.g., pressure or potential) to functions on edges (e.g., flow along edges).

- **Adjoint gradient (Divergence)** ∇^*: Converts flows back into node-based values, effectively aggregating flow differences across edges.

The graph **Laplacian** operator, defined as $\nabla^*\nabla$, is often expressed as:

$$L = I - P$$

where I is the identity matrix and P is the Markov (or transition) operator, describing a random walk on the graph. This formulation relates to the "smoothness" of functions over the graph structure.

3 Boundary and Coboundary Operators

In both topology and graph theory, boundary operators help describe the structure and relationships of cycles.

3.1 Simplices and Boundaries

A **simplex** represents a fundamental building block in space. For instance:

- A *0-simplex* is a point.
- A *1-simplex* is a line segment between two points.
- A *2-simplex* is a triangle formed by three points.

The **boundary operator** ∂ maps an n-dimensional simplex to its $(n-1)$-dimensional boundary simplices. In graphs, this operator captures the notion of flux or flow across nodes.

3.2 Coboundary Operator

The **coboundary operator** δ, dual to the boundary operator, maps cochains (functions on simplices) to another cochain in a higher dimension. In graphs, this captures the idea of "change" along edges, forming the basis for defining differential forms and cohomology.

4 Graham's Cohomology and Loops in Networks

In network theory, **loops** or **cycles** are collections of edges forming closed paths. Graham's cohomology classifies cycles based on their ability to carry non-exact, harmonic flows, identifying loops that cannot be represented by a scalar potential.

4.1 Exact and Non-Exact Chains

Exact chains have no boundary, analogous to closed fields where the *curl of the gradient* is zero. In graph cohomology:

- Flows derived from a scalar potential lie in the **exact cochain**.
- Independent cycles correspond to non-exact cohomology classes.

These cohomology classes reveal the intrinsic "holes" in a space, capturing cycles independent of the boundary.

5 Defining a Notion of Curl on Graphs

To define a curl-like operator on graphs:

- A curl-like operation can detect non-gradient flows or circulation around cycles.
- This is achieved by evaluating flow on each cycle in the network; non-zero circulation indicates a curl-like structure.

We can analyze flows on each cycle using incidence matrices and cochain complexes. If the net "circulation" around a cycle is non-zero, the flow represents a **non-exact cohomology class**.

6 Gradient and Non-Gradient Flows on Graphs

- A **gradient flow** (or *exact flow*) is a flow derived from a scalar potential.
- A **circulatory flow** (or *solenoidal flow*) cannot be derived from a potential function and is associated with cycles.
- **Harmonic flows** are divergence-free, non-gradient flows that represent intrinsic topology; they cannot be reduced to any scalar potential.

6.1 Hodge Decomposition of Flows

The Hodge decomposition theorem for graphs states that any flow F can be decomposed into:

$$F = F_{\text{grad}} + F_{\text{circ}} + F_{\text{harmonic}}$$

where:

- F_{grad}: Gradient flow, derived from a scalar potential.

- F_{circ}: Circulatory flow, representing cycles.
- F_{harmonic}: Harmonic flow, a topological property associated with the graphs cohomology.

7 Summary and Conclusion

In summary:

- Graham's cohomology identifies non-trivial cycles in graphs that carry circulatory flow not derivable from a scalar potential.
- The **curl-like operator** in graph theory detects these circulatory flows.
- Flows are classified as **exact (gradient)**, **circulatory (non-gradient)**, and **harmonic**, with harmonic flows indicating topologically significant cycles.

8 Keywords for Further Research

To further explore these concepts, consider the following keywords:

- Graph signal processing
- Network analysis
- Discrete differential geometry
- Graph Cohomology
- Graham's Cohomology and Network Analysis
- Hodge Decomposition on Graphs
- Gradient and Circulatory Flows in Graph Theory
- Discrete Laplacian and Markov Operators
- Topological Flow Analysis on Networks
- Boundary and Coboundary Operators in Graphs

9 Introduction

In this document, we explore the analogies between vector calculus in continuum settings and graph-based flow analysis. Specifically, we delve into the concepts of gradient, divergence, and curl on graphs, as well as the role of Graham's cohomology in characterizing loops and flows in networks.

10 Gradient, Divergence, and Laplacian Operators

In both continuum and graph settings, we define various operators to analyze scalar and vector fields. These operators have close analogies in both domains.

10.1 Continuum Setting

In continuum space (e.g., $\mathbb{R}^n$), we define:

- **Scalar field:** A function $f\colon \mathbb{R}^n \to \mathbb{R}$ that assigns a scalar value to each point in space.
- **Vector field:** A function $F\colon \mathbb{R}^n \to \mathbb{R}^n$ that assigns a vector to each point in space.
- **Gradient** ∇: Converts a scalar field to a vector field. The gradient of f is defined as ∇f, which points in the direction of maximum increase of f.
- **Divergence** div: Converts a vector field to a scalar field, measuring the net flux exiting or entering a region. For a vector field F, $\operatorname{div} F$ is defined as the sum of partial derivatives $\sum_i \frac{\partial F_i}{\partial x_i}$.

In this setting, the Laplacian operator, $\Delta = \operatorname{div}\nabla$, acts on scalar fields:

$$\operatorname{div}(\nabla f) = \Delta f$$

where Δf measures the "smoothness" or rate of change of f across the space. Additionally, we have:

$$\operatorname{curl}(\nabla f) = 0$$

indicating that gradients are *exact fields* with no circulation around any point, a property central to cohomological interpretation.

10.2 Graph Setting

For a graph with nodes and edges:

- **Gradient operator** ∇: Maps scalar functions on nodes (e.g., pressure or potential) to functions on edges (e.g., flow along edges).

- **Adjoint gradient (Divergence)** ∇^*: Converts flows back into node-based values, effectively aggregating flow differences across edges.

The graph **Laplacian** operator, defined as $\nabla^*\nabla$, is often expressed as:

$$L = I - P$$

where I is the identity matrix and P is the Markov (or transition) operator, describing a random walk on the graph. This formulation relates to the "smoothness" of functions over the graph structure.

11 Boundary and Coboundary Operators

In both topology and graph theory, boundary operators help describe the structure and relationships of cycles.

11.1 Simplices and Boundaries

A **simplex** represents a fundamental building block in space. For instance:

- A *0-simplex* is a point.

- A *1-simplex* is a line segment between two points.

- A *2-simplex* is a triangle formed by three points.

The **boundary operator** ∂ maps an n-dimensional simplex to its $(n-1)$-dimensional boundary simplices. In graphs, this operator captures the notion of flux or flow across nodes.

11.2 Coboundary Operator

The **coboundary operator** δ, dual to the boundary operator, maps cochains (functions on simplices) to another cochain in a higher dimension. In graphs, this captures the idea of "change" along edges, forming the basis for defining differential forms and cohomology.

12 Graham's Cohomology and Loops in Networks

In network theory, **loops** or **cycles** are collections of edges forming closed paths. Graham's cohomology classifies cycles based on their ability to carry non-exact, harmonic flows, identifying loops that cannot be represented by a scalar potential.

12.1 Exact and Non-Exact Chains

Exact chains have no boundary, analogous to closed fields where the *curl of the gradient* is zero. In graph cohomology:

- Flows derived from a scalar potential lie in the **exact cochain**.
- Independent cycles correspond to non-exact cohomology classes.

These cohomology classes reveal the intrinsic "holes" in a space, capturing cycles independent of the boundary.

13 Defining a Notion of Curl on Graphs

To define a curl-like operator on graphs:

- A curl-like operation can detect non-gradient flows or circulation around cycles.
- This is achieved by evaluating flow on each cycle in the network; non-zero circulation indicates a curl-like structure.

We can analyze flows on each cycle using incidence matrices and cochain complexes. If the net "circulation" around a cycle is non-zero, the flow represents a **non-exact cohomology class**.

14 Gradient and Non-Gradient Flows on Graphs

- A **gradient flow** (or *exact flow*) is a flow derived from a scalar potential.
- A **circulatory flow** (or *solenoidal flow*) cannot be derived from a potential function and is associated with cycles.
- **Harmonic flows** are divergence-free, non-gradient flows that represent intrinsic topology; they cannot be reduced to any scalar potential.

14.1 Hodge Decomposition of Flows

The Hodge decomposition theorem for graphs states that any flow F can be decomposed into:

$$F = F_{\text{grad}} + F_{\text{circ}} + F_{\text{harmonic}}$$

where:

- F_{grad}: Gradient flow, derived from a scalar potential.
- F_{circ}: Circulatory flow, representing cycles.
- F_{harmonic}: Harmonic flow, a topological property associated with the graphs cohomology.

15 Summary and Conclusion

In summary:

- Graham's cohomology identifies non-trivial cycles in graphs that carry circulatory flow not derivable from a scalar potential.
- The **curl-like operator** in graph theory detects these circulatory flows.
- Flows are classified as **exact (gradient)**, **circulatory (non-gradient)**, and **harmonic**, with harmonic flows indicating topologically significant cycles.

16 Keywords for Further Research

To further explore these concepts, consider the following keywords:

- Graph Cohomology
- Graham's Cohomology and Network Analysis
- Hodge Decomposition on Graphs
- Gradient and Circulatory Flows in Graph Theory
- Discrete Laplacian and Markov Operators
- Topological Flow Analysis on Networks
- Boundary and Coboundary Operators in Graphs

17 Evolution on Edge Weights, Energy Minimization, and Theoretical Connections

In this section, we explore the theoretical implications of defining an evolution on the weights of the edges in a graph, such that it minimizes a given energy functional. This adjustment of edge weights affects the graphs loop structure, connectivity, and flow characteristics. We also consider analogous dynamics in continuum settings, vector bundle formalisms, and a categorical framework to generalize these notions.

17.1 Evolution on Edge Weights and Energy Functional

When defining a dynamic on the edge weights of a graph to minimize an energy functional, represented as a flow that adjusts weights toward lower energy, let $w_{ij}(t)$ denote the weight of the edge between nodes i and j at time t. Suppose that w_{ij} evolves according to a gradient descent on some energy functional $E(w_{ij})$:

$$\frac{dw_{ij}}{dt} = -\frac{\partial E}{\partial w_{ij}}.$$

This process modifies the edge weights, causing some to increase and others to decrease, which influences network connectivity and loop structures in the following ways:

- **Change in Loop Structure:** As weights change, loops (cycles) may form or dissolve depending on connectivity changes. Lighter weights may effectively remove edges from the network, while stronger weights may reinforce certain paths.

- **Impact on Flow and Cohomology:** The evolving weights affect cohomology classes, with new loops possibly introducing additional non-trivial cycles, and some disappearing as weights decrease.

17.2 Dual Notion in the Continuum Setting

This evolution resembles a **Ricci flow** in a continuum setting, where the manifold shape evolves to minimize a curvature-based energy functional. Similarly, the edge weights on a graph evolve to optimize network connectivity, and the graphs "curvature" (in terms of connectivity and loop structure) changes as weights evolve, analogous to the effects of Ricci flow on a manifold.

17.3 Tracking the Formation or Destruction of Loops

To analyze loop creation or destruction during weight evolution:

- **Persistent Homology:** This technique analyzes topological changes by quantifying the birth and death of features (loops, cavities) as a function of weight, revealing loop evolution over time.

- **Betti Numbers:** These numbers provide a measure of the graphs topological invariants (e.g., connected components and cycles). Calculating Betti numbers dynamically allows tracking of loop structure changes as weights evolve.

17.4 Theorems for Vector Fields in the Continuum Setting

Theorems from vector calculus offer insights into dynamics on graphs:

- **Stokes Theorem and Gauss Divergence Theorem:** These relate fluxes across boundaries to sources or sinks, providing tools to analyze how flows redistribute on weighted graphs.

- **Poincarés Lemma:** In a simply connected domain, a curl-free field can be expressed as a gradient of a scalar potential. For graphs without loops, conservative flows can be represented as gradient flows, with loop creation corresponding to non-exactness, yielding non-trivial cohomology classes.

17.5 Manifold Theorems and Their Applications

Theorems from manifold theory, such as the **Hairy Ball Theorem**, provide insights for graph theory:

- The Hairy Ball Theorem states that there is no continuous, non-vanishing vector field on even-dimensional spheres. In graph theory, this suggests constraints on global flow directionality, especially for high-dimensional graphs, pointing to inevitable singularities such as sources or sinks in complex graph structures.

17.6 Formalizing the Dynamics of the Vector Field on Graphs

The dynamics of a vector field on a graph can be formalized as an evolution on a **vector bundle** over the graph:

- A vector bundle on a graph assigns a vector space (such as the set of flows) to each node. Here, each edge represents a connection or transition, with weights modulating the flow across edges.

- As weights evolve, the "fiber" (possible flows) at each node changes, resulting in a dynamic vector field. This resembles a dynamical system on a vector bundle, driven by energy minimization on the edges.

17.7 Categorical Perspective: Categories and Functors

Category theory offers an abstract framework for formalizing the evolution of graph structure and flow dynamics:

- **Category of Graphs:** Define a category where objects are graphs with weighted edges, and morphisms are structure-preserving maps (e.g., homomorphisms) that respect edge weights.

- **Functorial Connections:** Define a functor from the category of weighted graphs to vector spaces, where each graph is mapped to its cohomology vector space, associating each graph with its flow space and topological features.

- **Homology Functor:** Define a homology functor H that maps each weighted graph to its set of homology classes, tracking changes in topological features (such as loops) over time. This functor captures changes in connectivity and flow dynamics as the graph evolves.

17.8 Summary of Key Concepts

- **Energy Minimization on Graphs:** Evolving edge weights alters the graphs loop structure and connectivity.

- **Continuum Dual:** The process has an analogue in Ricci flow, where space evolves to minimize curvature.

- **Tracking Loops:** Persistent homology and Betti numbers allow tracking of loop creation and destruction dynamically.

- **Theoretical Connections:** Vector calculus theorems offer insights into flux and circulation in graphs.

- **Vector Bundle Analogy:** Evolving edge weights form a dynamic system on a vector bundle over the graph.

- **Category Theory:** Categories and functors formalize the graphs structure and dynamics, giving a coherent framework for analyzing flow and topology changes.

17.9 Keywords for Further Research

For further exploration, consider the following keywords:

- Persistent Homology and Graphs
- Vector Bundle Dynamics in Graph Theory
- Ricci Flow and Network Evolution
- Category Theory for Graph Analysis
- Functorial Homology and Cohomology

References