Hypergraph p-Laplace operators for Image Denoising

Master-Arbeit

zur Erlangung des Grades

Master of Science (M.Sc.)

im Studiengang Computational and Applied Mathematics (CAM)

am Department Mathematik der
Friedrich-Alexander-Universität Erlangen-Nürnberg

vorgelegt am **19.Dezember 2023**

von **Kseniia Lukin**

Prüfer:	Dr. Daniel Tenbrinck
Prüfer:	Prof. Dr. Manuel Friedrich
Betreuer:	M.Sc. Ariane Fazeny

FAU
Friedrich-Alexander-Universität
Naturwissenschaftliche Fakultät

Department
MATHEMATIK

Acknowledgement

Firstly, I would like to express my sincere gratitude to Dr. Daniel Tenbrinck for his unwavering support and optimism throughout my master's thesis journey. His intellectual guidance, constructive feedback, and valuable advice motivated me to improve my thesis and challenge myself.

I am equally grateful to Ariane Fazeny for her exceptional assistance and the many important and meaningful discussions we had. Her insights significantly enriched my research.

Last but not least, I would like to thank my partner, Max, and my family. Their continuous support has inspired me to never give up when facing struggles in learning and in life.

Introduction

In this thesis we explore the field of hypergraph operators and their application to image denoising. The exploration unfolds in three distinct chapters, each contributing to the understanding and advancement of this approach to image processing. Building on fundamental work in graph structures, PDEs on graphs, and the widespread use of graph neural networks in machine learning, this work seeks to extend and generalise existing definitions of key operators such as gradients, adjoints, divergences, Laplacians, and p-Laplacians.

While graphs have proven invaluable in capturing pairwise interactions, their limitations become apparent when dealing with group relations, a crucial aspect in domains such as image correction. To address this limitation, we propose utilizing hypergraphs, which provide a natural framework of encoding group interactions. The adaptability of hypergraphs is highlighted through an exploration that examines both oriented and unoriented hypergraphs.

The first part of the thesis gives a thorough introduction to unoriented and oriented graphs. Key definitions, functions on graphs and differential operators including gradient, divergence and p-Laplacian operators are explained. In the following section, the thesis focuses on introducing the theory of hypergraphs. It establishes hypergraphs as a natural extension of traditional graphs and shows that a graph is a special case of a hypergraph.

Our theoretical foundations are complemented by practical applications in image processing using unoriented hypergraphs. Averaging operators for unoriented hypergraphs emerge as a new way to capture local and nonlocal relationships within images. With a special focus on image denoising, this chapter validates the effectiveness through numerical experiments that bridge the theoretical and empirical domains.

Related work

There is a significant amount of research in the field of graph and hypergraph theory and the integration of both into practical applications. The following studies cover a wide range of topics and provide valuable insights into both hypergraph theory and its applications in image processing.

1. Leonie Neuhauser, Renaud Lambiotte and Michael T. Schaub: Consensus dynamics and opinion formation on hypergraphs. Springer International Publishing, (2022). [NLS22]

This article investigates consensus dynamics and opinion formation using hypergraphs, with a particular focus on nonlinear interactions. The analysis takes into account factors such as initial state distribution, hypergraph structure, and nonlinear scaling, which provide valuable information about the evolution of consensus and opinion formation in hypergraphs.

2. Abderrahim Elmoataz, Matthieu Toutain and Daniel Tenbrinck: On the p-Laplacian and ∞-Laplacian on Graphs with Applications in Image and Data Processing, (2015). [ETT15]

This paper introduces a new family of partial difference operators on graphs, covering different variants of Laplace operators, and explores their applications in image and data processing. The proposed operators provide adaptive interpolation between p-Laplacian diffusion filtering and morphological filtering, providing a unified framework for different scenarios. The study established mathematical properties that prove the existence and uniqueness of solutions to parabolic and elliptic partial differential equations, including the intriguing case $p = \infty$. The adaptability and versatility of the formulation is illustrated with applications to segmentation, denoising, inpainting and clustering of various types of data. Thus, the article proposes a universal and unified approach to Laplacians based on graphs with gradient terms, enriching the field of mathematical modelling in image and data processing.

3. Eleonora Andreotti, Raffaella Mulas: Signless Normalized Laplacian for Hypergraphs, (2022). [AM22]

This paper explores the spectral theory of the unsigned normalized Laplacian for chemical hypergraphs, with a focus on bipartite cases. It is established that the spectrum of the unsigned normalized Laplacian for classical hypergraphs coincides with the spectrum of the normalized Laplacian for bipartite chemical hypergraphs. The study relates chemical hypergraphs to oriented hypergraphs. The article demonstrates that studying the spectrum of the normalized Laplacian in chemical hypergraphs is equivalent to studying oriented hypergraphs without catalysts. It also highlights the various applications of classical hypergraphs, including their relevance in modelling transportation networks, neural codes, social networks, and epidemiological networks. Structurally, the paper introduces and explores new types of hypergraphs with the aim of generalising known graph structures.

4. Yue Gao, Zizhao Zhang , Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou: Hypergraph Learning: Methods and Practices, (2022). [Gao+22]

The paper explores hypergraph learning, a technique renowned for its effectiveness in modelling complex data correlations. The authors conduct a systematic review of existing hypergraph generation methods, including distance-based, representation-based, attribute-based, and network-based approaches. The study introduces various learning methods for hypergraphs, including transductive and inductive hypergraph learning, hypergraph structure updating, and multi-modal hypergraph learning. Evaluations across applications, such as object and action recognition, and clustering, highlighting the effectiveness and efficiency of hypergraph generation and learning methods.

Main contributions

The main contributions of the thesis are as follows: I reviewed important definitions in graph and hypergraph theory from the works of [JMZ21], [Faz23] and [FTB23], then I modified the basic equations for first-order operators and Laplace operators using constant weight functions suitable for both oriented and unoriented hypergraphs. After that, I used an averaging operator to remove noise from an image using a dual analysis approach that includes a local and a non-local method. The latter method identifies vertex neighbours based on similar colour intensities.

I used hypergraph differential operators for image processing by representing pixel relationships through hypergraphs instead of a regular grid. This approach captures both local and nonlocal relationships based on the content of the image. I examined non-constant weight functions for hypergraphs and assigned different weights to hyperedges based on value perturbations within each hyperedge. I conducted two experiments. One experiment focused on local image processing using hyperedges from direct pixel neighbors. The second experiment focused on nonlocal processing. It considered pixel intensities across the entire image. The dynamic assignment of weights to hyperedges adds adaptability to different image characteristics. This method uses hypergraphs, non-constant weights, and dynamic processing to image denoising. This approach is effective in capturing both local and nonlocal structures, resulting in improved outcomes.

Overview of Chapters

Chapter 1 lays the foundation for our study by introducing the theory of oriented 1.2 and unoriented 1.1 graphs. The first section takes a thorough look at the basic definitions and properties of graphs, such as degree of vertices 1.3, incidence matrices 1.5 and others. This section also introduces the concept of a weighted graph in both the oriented 1.7 and unoriented 1.8 case. And then we explore functions on graphs in Section 1.2 and corresponding differential operators in Section 1.3 and Section 1.4. In particular, we consider first-order differential operators and p-Laplace operators in Section 1.3.2 and Section 1.4.2, which open up the potential for understanding and manipulating data in graph structures.

Chapter 2 extends our exploration to hypergraph notation. This chapter has a similar structure as the previous one, but now we are considering hypergraphs. Concepts such as unoriented 2.1 and oriented 2.2 hypergraphs are introduced in Section 2.1. We delve deeper into hypergraph theory by studying functions and differential operators on unoriented hypergraphs in Section 2.3. The Section 2.3.2 introduces Laplace and p-Laplace operators in the unoriented case. The next Section 2.4 is devoted to finding first and higher-order differential operators on an oriented hypergraph. Section 2.5 concludes the chapter with a description of the connections between graphs and hypergraphs.

Chapter 3 applies this knowledge to the field of image processing using unoriented hypergraphs. The study of averaging operators in Section 3.1 provides the basis for image denoising approaches. The chapter then explores local 3.3 and nonlocal 3.3 image processing techniques using unoriented hypergraphs, supported by numerical experiments.

Finally, Chapter 4 takes the discussion forward by summarising the findings and outlining future research directions.

Contents

Chapter 1

Graph notation

The first chapter examines the basic principles of graph theory, explaining key definitions for both oriented and unoriented cases. Section 1.1 explores fundamental concepts such as vertex degrees 1.3, adjacency 1.4 and incidence matrices 1.5 for describing graphs. This section also contains some examples of unoriented and oriented graphs.

The following Section 1.2 discusses functions on graphs. Concepts such as vertex 1.13 and edge 1.17 and arc 1.18 functions, weight functions for unoriented 1.8 and oriented 1.7 graphs, and spaces of functions are introduced in 1.15, 1.19 and 1.20.

Section 1.3 advances our understanding through the application of differential operators, including the computation of vertex-edge characteristic function 1.25, vertex 1.37 and edge 1.27 gradient operators, adjoint 1.40 and divergence 1.42 operators for them, and a smooth transition to the p-Laplacian operators Section 1.3.2 for unoriented graphs.

The chapter ends with a Section 1.4 on differential operators for oriented graphs. The list of definitions is similar to the previous section, but in this case the focus is on arcs 1.38 rather than edges, since for this type of graph connections between vertices have an orientation.

1.1 Graph theory: main definitions and properties

DEFINITION 1.1: Unoriented graph.

A graph is unoriented if it consists of a finite, nonempty set of vertices $V = \{v_1, v_2, ..., v_n\}$, $n \in \mathbb{N}$ and a set of edges $E_G = \{e_1, e_2, ..., e_k\}$, $k \in \mathbb{N}$ and each edge is a set $\{v_i, v_j\}$ of vertices.

$$G_u = (V, E_G),\ E_G \subseteq \{\{v_i, v_j\} \mid v_i, v_j \in V\}. \tag{1.1}$$

In simple words, the unoriented graph is a graph in which edges have no orientation. Therefore, the connections between vertices are symmetric.

Theoretical graph theory is concerned with simple graph and this is the reason why we continue with graphs without any self-loops $\{v_i, v_i\} \in E_G$ for $v_i \in V$.

Prior to introducing the next definition, let's establish the terminology of an "*oriented edge*" as an "*arc*".

DEFINITION 1.2: Oriented graph.

A graph is oriented if it consists of a finite, nonempty set of vertices $V = \{v_1, v_2, ..., v_n\}$, $n \in \mathbb{N}$ and a set of arcs $A_G = \{a_1, a_2, ..., a_q\}$, $k \in \mathbb{N}$ and each arc is an ordered tuple (v_i, v_j) of vertices:

$$G_o = (V, A_G), \quad A_G \subseteq \{(v_i, v_j) \mid v_i, v_j \in V, v_i \neq v_j\}. \tag{1.2}$$

This means each arc has an initial and terminal vertex and a specific direction from one vertex to another.

DEFINITION 1.3: Degree of vertex.

A degree of vertex v_i in an unoriented graph $G_u = (V, E_G)$ is the number of proper edges incident to vertex v_i and denoted as:

$$\deg(v_i) = |E_G(v_i)|, \tag{1.3}$$

where $E_G(v_i)$ represents the set of edges incident to vertex v_i. For an oriented graph $G_o = (V, A_G)$ the degree of vertices is defined as follows:

$$\deg(v_i) = |A_G(v_i)|, \tag{1.4}$$

where $A_G(v_i)$ represents the set of arcs incident to vertex v_i.

Hence, a vertex with no neighbours has degree 0 and is denoted as an isolated vertex.

DEFINITION 1.4: Adjacency matrix for graphs.

For an unoriented graph $G_u = (V, E_G)$ with vertex set $V = \{v_1, v_2, ..., v_n\}$ the adjacency matrix is defined as:

$$A_{G_u}(v_i, v_j) = \begin{cases} 1, & \text{if there is an edge from vertex } v_i \text{ to vertex } v_j \ , \\ 0, & \text{otherwise.} \end{cases} \tag{1.5}$$

The adjacency matrix of an unoriented graph is symmetric because edges between vertices have no direction.
For an oriented graph $G_o = (V, A_G)$ with vertex set $V = \{v_1, v_2, ..., v_n\}$ the adjacency matrix defined as:

$$A_{G_o}(v_i, v_j) = \begin{cases} 1, & \text{if there is an arc from vertex } v_i \text{ to vertex } v_j, \\ 0, & \text{otherwise.} \end{cases} \tag{1.6}$$

In contrast to the the adjacency matrix for oriented graphs, the matrix A_{G_o} is generally not symmetric, however in both cases, for unoriented graphs G_u and oriented graphs G_o the size of the matrices A_{G_u} and A_{G_o} is $n \times n$.

DEFINITION 1.5: Incidence matrix for graphs.

An incidence matrix for an oriented graph $G_o = (V, A_G)$ can be defined as a $|V| \times |A_G|$ matrix K_{G_o}, where:

$$K_{G_o}(v_i, a_q) = \begin{cases} 1, & \text{if arc } a_q \text{ is directed to vertex } v_i, \\ -1, & \text{if arc } a_q \text{ is directed from vertex } v_i, \\ 0, & \text{otherwise.} \end{cases} \tag{1.7}$$

For an unoriented graph $G_u = (V, E_G)$ the incidence matrix K_{G_u} is given as a $|V| \times |E_G|$ matrix, where:

$$K_{G_u}(v_i, e_q) = \begin{cases} 1, & \text{if edge } e_q \text{ is incident to vertex } v_i, \\ 0, & \text{otherwise.} \end{cases} \tag{1.8}$$

In either scenario for unoriented graphs G_u and oriented graphs G_o, the size of incidence matrices K_{G_u} and K_{G_o} is $n \times k$.

Example 1. Unoriented graph:
For an unoriented graph $G_u = (V, E_G)$ with a set of vertices V:

$$V = \{v_1, v_2, v_3, v_4, v_5\},$$

and set of edges E_G:

$$\begin{aligned} E_G &= \{\{v_1, v_2\}, \{v_2, v_5\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_3, v_5\}, \{v_2, v_4\}, \{v_4, v_5\}, \{v_1, v_5\}\} \\ &= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}, \end{aligned}$$

the vertex v_6 is isolated.

The adjacency matrix A_{G_u} can be constructed based on the connections between vertices and would look like this:

$$A_{G_u} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

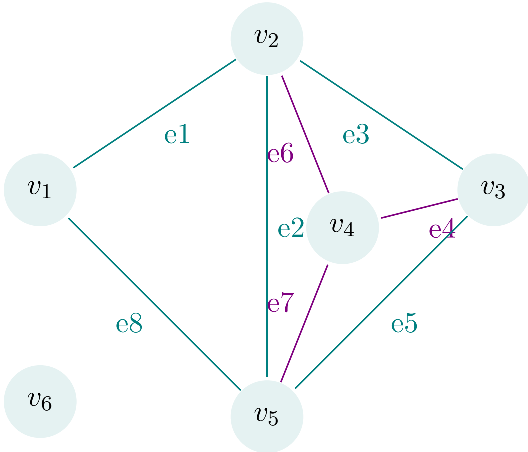


Figure 1.1: Unoriented Graph.

Explanation:

- $A_{G_u}(v_i, v_j) = 1$ if there is an edge between v_i and v_j, and 0 otherwise.
- For example, $A_{G_u}(1, 2) = 1$ because there is an edge e_1 between v_1 and v_2.
- The matrix is symmetric, so $A_{G_u}(v_i, v_j) = A_{G_u}(v_j, v_i)$ for all $v_i, v_j \in V$.

The incidence matrix for an unoriented graph is a matrix that represents which vertices are incident to which edges. Each row of the matrix corresponds to a vertex, and each column corresponds to an edge. The entry in row i and column q is 1 if vertex v_i is incident to edge e_q and 0 if the edge is not incident to the vertex.

$$K_{G_u} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} e1 & e2 & e3 & e4 & e5 & e6 & e7 & e8 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

Explanation:

- For example, $K_{G_u}(v_2, e_2) = 1$ because there is an edge e_2 between v_2 and v_5.
- The matrix is generally not square and not symmetric.

Example 2. Oriented graph:
We consider an oriented graph $G_o = (V, A_G)$ with a set of vertices V:

$$V = \{v_1, v_2, v_3, v_4, v_5\},$$

and set of arcs A_G:

$$\begin{aligned} A_G &= \{(v_2, v_1), (v_2, v_5), (v_3, v_2), (v_3, v_4), (v_3, v_5), (v_4, v_2), (v_5, v_4), (v_5, v_1)\} \\ &= \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}. \end{aligned}$$

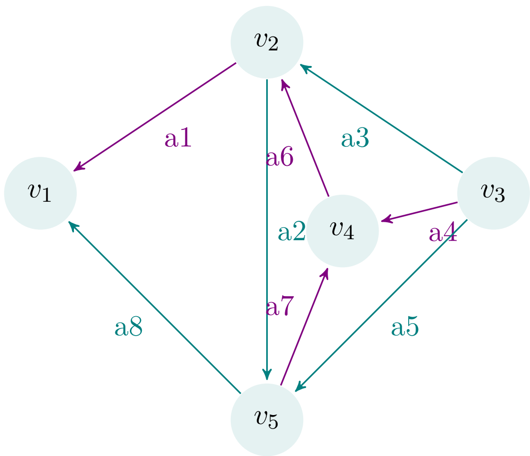


Figure 1.2: Oriented Graph.

Similar to the previous example, the adjacency matrix A_{G_o} can be constructed based on the connections between vertices. The entry is 1 if there is an arc from vertex v_i to v_j, and 0 otherwise.

$$A_{G_o} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Explanation:

- For example, $A_{G_o}(3, 2) = 1$ because there is an arc a_3 from v_3 to v_2.
- The adjacency matrix A_{G_o} for an oriented graph is not necessarily symmetric, so $A_{G_o}(v_i, v_j) \neq A_{G_o}(v_j, v_i)$ for $v_i, v_j \in V$.

As before, for an oriented graph, the incidence matrix K_{G_o} represents the relationship between vertices and arcs. Each row of the matrix corresponds to a vertex, and each column corresponds to an arc. The entry in row i and column q is 1 if the arc a_q ends at vertex v_i , -1 if it starts at vertex v_i, and 0 otherwise.

$$K_{G_o} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} a1 & a2 & a3 & a4 & a5 & a6 & a7 & a8 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & -1 & -1 \end{pmatrix} \end{array}$$

Explanation:

- For example, $A_{G_o}(v_3, a_4) = -1$ because arc a_4 goes from v_3 to v_4.
- The matrix is in general neither symmetric nor square.

DEFINITION 1.6: Weighted unoriented graph.

An unoriented graph is weighted, if G_u has weights associated to its edges. This means that each edge $e \in E_G$ connecting two vertices v_i and v_j carries a value, and these weights are non-negative, so

$$w : E_G \longrightarrow \mathbb{R}^+, \quad w_e \geq 0.$$

The full expression of an unoriented weighted graph is given by:

$$G_u = (V, E_G, w). \tag{1.9}$$

DEFINITION 1.7: Weighted oriented graph.

An oriented graph is weighted if G_o has weights associated to its arcs. This means that each arc $a \in A_G$ connecting two vertices v_i and v_j carries a value, and these weights are non-negative, so

$$w : A_G \longrightarrow \mathbb{R}^+, \quad w_a \geq 0.$$

The full expression of an oriented weighted graph is given by:

$$G_o = (V, A_G, w). \tag{1.10}$$

DEFINITION 1.8: Weighted adjacency matrices for graphs.

The adjacency matrix of a weighted unoriented graph $G_u = (V, E_G, w)$ and weighted oriented graph $G_o = (V, A_G, w)$ is denoted as

$$A(V, E_G, w) = (w_{ij})_{i,j=1,\dots,n},$$

and

$$A(V, A_G, w) = (w_{ij})_{i,j=1,\dots,n},$$

respectively with dimensions $|V| \times |V|$, where each element A_{ij} of the matrix contains the weight value w_e associated with the edge $e = \{v_i, v_j\}$ or arc $a = (v_i, v_j)$, if the vertices v_i and v_j are adjacent, otherwise it equals 0.

$$A(v_i, v_j) = \begin{cases} w_e, & \text{if there is an edge from vertex } v_i \text{ to vertex } v_j, \\ 0, & \text{otherwise.} \end{cases} \tag{1.11}$$

$$A(v_i, v_j) = \begin{cases} w_a, & \text{if there is an arc from vertex } v_i \text{ to vertex } v_j, \\ 0, & \text{otherwise.} \end{cases} \tag{1.12}$$

Moreover, for any unoriented graph $G_u = (V, E_G, w)$ we have $w_{ij} = w_{ji}$.

DEFINITION 1.9: Degree function in unoriented graphs.

The degree function $\deg(v) : V \to \mathbb{R}$ of a vertex $v_i \in V$ in an unoriented graph $G_u = (V, E_G, w)$ is given as:

$$\deg(v_i) = \sum_{j=1}^{n} w_{ij}. \tag{1.13}$$

For an isolated vertex v_i, $\deg(v_i) = 0$ and if vertices v_i and v_j are not directly connected thorugh an edge, we have $w_{ij} = 0$. Due to symmetric weights in the case of an unoriented hypergraph, the following equality holds true:

$$\sum_{j=1}^{n} w_{ij} = \sum_{j=1}^{n} w_{ji}. \tag{1.14}$$

DEFINITION 1.10: Degree function in oriented graphs.

The degree functions $\deg(v)^{in} : V \to \mathbb{R}$ and $\deg(v)^{out} : V \to \mathbb{R}$ of a vertex $v_i \in V$ in an oriented graph $G_o = (V, A_G, w)$ can be expressed as:

$$\deg(v_i)^{in} = \sum_{j=1}^{n} w_{ji}, \tag{1.15}$$

$$\deg(v_i)^{out} = \sum_{j=1}^{n} w_{ij}, \tag{1.16}$$

where $\deg(v_i)^{in}$ defined as the sum of the weights of all incoming arcs to vertex v_i and $\deg(v_i)^{out}$ is the sum of the weights of all outgoing arcs from v_i.
The total degree function is defined as the sum of the in-degree and out-degree of a vertex v_i:

$$\deg(v_i) = \deg(v_i)^{in} + \deg(v_i)^{out} \tag{1.17}$$

DEFINITION 1.11: Degree matrices.

The degree matrix $D(G_u)$ (size $n \times n$) for an unoriented graph represent the degrees of the vertices and is defined as:

$$D(G_u) = \mathrm{diag}(\deg(v_1), ..., \deg(v_n)). \tag{1.18}$$

In an oriented graph the degree matrix $D(G_o)$ can be divided into two matrices $D^{in}(G_o)$ and $D^{out}(G_o)$: one for $\deg(v)^{in}$ and one for $\deg(v)^{in}$.

DEFINITION 1.12: Neighbourhood of a vertex.

The neighbourhood of a vertex v in an unoriented graph $G_u = (V, E_G)$ or an oriented graph $G_o = (V, A_G)$ consists of all the vertices adjacent to v_i excluding itself:

$$N_u(v_i) = \{v_j \in V \backslash \{v_i\} \mid \{v_i, v_j\} \in E_G\},$$
$$N_o(v_i) = \{v_j \in V \backslash \{v_i\} \mid (v_i, v_j) \in A_G \ \text{ or } \ (v_j, v_i) \in A_G\}.$$

1.2 Functions on graphs

The concept of *real vertex functions* was first introduced by [ETT15] and then extended by [Faz23]. Based on these works, let us present the following definitions.

DEFINITION 1.13: Vertex function.

Let $f(v_i)$ assign a value to each vertex $v_i \in V$ for an unoriented graph $G_u = (V, E_G)$. Then we define a vertex function as:

$$f : V \to \mathbb{R} \qquad v_i \mapsto f(v_i).$$

The definition of a vertex function f for an oriented graph $G_o = (V, A_G)$ is exactly the same as for an unoriented graph $G_u = (V, E_G)$.

DEFINITION 1.14: Vertex weight function.

Let $w(v_i)$ assign a weight to each vertex $v_i \in V$ for an unoriented graph $G_u = (V, E_G)$. Then we define a weight function as:

$$w : V \to \mathbb{R}_{>0}, \qquad v_i \mapsto w(v_i).$$

The definition of a vertex weight function for an oriented graph $G_o = (V, A_G)$ is the same as for an unoriented graph.

DEFINITION 1.15: Space of vertex functions $S(V)$.

We define $S(V)$ as the space for all vertex functions f for an unoriented graph $G_u = (V, E_G)$ or an oriented graph $G_o = (V, A_G)$ and it can be written as:

$$S(V) = \{f \mid f : V \to \mathbb{R}\},$$

with the inner product for arbitrarily chosen vertex functions $f_1, f_2 \in S(V)$ and parameter $\gamma \in \mathbb{R}$:

$$\langle f_1, f_2 \rangle_{S(V)} = \sum_{v_i \in V} w(v_i)^\gamma f_1(v_i) f_2(v_i). \tag{1.19}$$

DEFINITION 1.16: L^p-norm on the space of vertex functions.

For an unoriented graph $G_u = (V, E_G)$ or for an oriented graph $G_o = (V, A_G)$ the $L^p(V)$-norm on the space of vertex functions is defines as:

$$\|\cdot\|_p : S(V) \to \mathbb{R}^+,$$
$$f \mapsto \|f\|_p = \begin{cases} \|f\|_p & \text{for} \quad 1 \leq p < \infty \\ \|f\|_\infty & \text{for} \quad p = \infty \end{cases},$$

where

$$\|f\|_p = \sum_{v_i \in V} (|f(v_i)|^p)^{\frac{1}{p}}, \tag{1.20}$$

$$\|f\|_\infty = \max_{v_i \in V}(|f(v_i)|), \tag{1.21}$$

with the absolute value of vertex functions $f \in S(V)$ as:

$$|\cdot| : \mathbb{R} \to \mathbb{R}^+,$$

$$f(v_i) \mapsto |f(v_i)| = \begin{cases} f(v_i) & \text{for} \quad f(v_i) \geq 0 \\ -f(v_i) & \text{for} \quad f(v_i) < 0 \end{cases} . \tag{1.22}$$

DEFINITION 1.17: Edge function F_E and edge weight function W_E.

Let $F_E(\{v_i, v_j\})$ assign a real value to each edge $\{v_i, v_j\} \in E_G$ for an unoriented graph $G_u = (V, E_G)$. Then an edge function is defined such that:

$$F_E : E_G \to \mathbb{R} \qquad e_q = \{v_i, v_j\} \mapsto F_E(e_q) = F_E(\{v_i, v_j\}).$$

Let $W_E(\{v_i, v_j\})$ assign a weight to each edge $\{v_i, v_j\} \in E_G$, so the edge weight function W_E is given by: ([ETT15], page 2418)

$$W_E : E_G \to \mathbb{R}_{>0} \quad e_q = \{v_i, v_j\} \mapsto W_E(e_q) = W_E(\{v_i, v_j\}).$$

Since the edges in an unoriented graph have no orientation, the symmetry property for any edge function F_E holds true $F_E(\{v_i, v_j\}) = F_E(\{v_j, v_i\})$ for all edges $\{v_i, v_j\} \in E_G$. For simplicity, we will write $F_E(v_i, v_j)$ instead of $F_E(\{v_i, v_j\})$ and hence the symmetry property implies $F_E(v_i, v_j) = F_E(v_j, v_i)$.

DEFINITION 1.18: Arc function F_A and arc weight function W_A.

Let $F_A((v_i, v_j))$ assign a real value to each arc $(v_i, v_j) \in A_G$ for an oriented graph $G_o = (V, A_G)$. Then an arc function is defined such that:

$$F_A : A_G \to \mathbb{R} \qquad a_q = (v_i, v_j) \mapsto F_A(a_q) = F_A((v_i, v_j)).$$

Let $W_A((v_i, v_j))$ assign a weight to each arc $(v_i, v_j) \in A_G$ so the arc weight function W_A is given by:

$$W_A : A_G \to \mathbb{R}_{>0}, \quad a_q = (v_i, v_j) \mapsto W_A(a_q) = W_A((v_i, v_j)).$$

For readability purposes, we will from now on omit the double brackets and just write $F_A(v_i, v_j)$ instead of $F_A((v_i, v_j))$.

DEFINITION 1.19: Space of edge functions $S(E_G)$.

We define $S(E_G)$ as the space of all edge functions F_E for an unoriented graph $G_u = (V, E_G)$, so:

$$S(E_G) = \{F \mid F : E_G \to \mathbb{R}\},$$

with the inner product for arbitrarily chosen edge functions $F, G \in S(E_G)$ and parameter $\chi \in \mathbb{R}$:

$$\langle F, G \rangle_{S(E_G)} = \sum_{e_q \in E_G} W(e_q)^{\chi} F(e_q) G(e_q). \tag{1.23}$$

DEFINITION 1.20: Space of arc functions $S(A_G)$.

We define $S(A_G)$ as the space of all arc functions F_A for an oriented graph $G_o = (V, A_G)$, so:

$$S(A_G) = \{F \mid F : A_G \to \mathbb{R}\},$$

with the inner product for arbitrarily chosen arc functions $F, G \in S(A_G)$ and parameter $\nu \in \mathbb{R}$:

$$\langle F, G \rangle_{S(A_G)} = \sum_{a_q \in A_G} W(a_q)^{\nu} F(a_q) G(a_q). \tag{1.24}$$

DEFINITION 1.21: L^p-norm on the space of edge functions.

For an unoriented graph $G_u = (V, E_G)$ the $L^p(E_G)$ - norm is defined as:

$$\|\cdot\|_p : S(E_G) \to \mathbb{R}^+,$$
$$F_E \mapsto \|F_E\|_p = \begin{cases} \|F_E\|_p & \text{for} \quad 1 \leq p < \infty \\ \|F_E\|_\infty & \text{for} \quad p = \infty \end{cases},$$

where

$$\|F_E\|_p = \sum_{e_q \in E_G} (|F_E(e_q)|^p)^{\frac{1}{p}}, \tag{1.25}$$

$$\|F_E\|_\infty = \max_{e_q \in E_G} (|F_E(e_q)|), \tag{1.26}$$

with the absolute value of edge functions $F_E \in S(E_G)$ as:

$$|\cdot| : \mathbb{R} \to \mathbb{R}^+,$$
$$F(e_q) \mapsto |F(e_q)| = \begin{cases} F(e_q) & \text{for} \quad F(e_q) \geq 0 \\ -F(e_q) & \text{for} \quad F(e_q) < 0 \end{cases}. \tag{1.27}$$

DEFINITION 1.22: L^p-norm on the space of arc functions.

For an oriented graph $G_o = (V, A_G)$ the $L^p(A_G)$ - norm is defined as:

$$\|\cdot\|_p : S(A_G) \to \mathbb{R}^+.$$

$$F_A \mapsto \|F_A\|_p = \begin{cases} \|F_A\|_p & \text{for} \quad 1 \leq p < \infty \\ \|F_A\|_\infty & \text{for} \quad p = \infty \end{cases},$$

where

$$\|F_A\|_p = \sum_{a_q \in A_G} (|F_A(a_q)|^p)^{\frac{1}{p}}, \tag{1.28}$$

$$\|F_A\|_\infty = \max_{a_q \in A_G} (|F_A(a_q)|), \tag{1.29}$$

with the absolute value of arc functions $F_A \in S(A_G)$ as:

$$|\cdot| : \mathbb{R} \to \mathbb{R}^+,$$

$$F(a_q) \mapsto |F(a_q)| = \begin{cases} F(a_q) & \text{for} \quad F(a_q) \geq 0 \\ -F(a_q) & \text{for} \quad F(a_q) < 0 \end{cases}. \tag{1.30}$$

So finally the unoriented graph with vertex and edge weights will be written as:

$$G_u = (V, E_G, w, W_E). \tag{1.31}$$

Similarly, the oriented graph with both weight functions will be written as:

$$G_o = (V, A_G, w, W_A). \tag{1.32}$$

1.3 Differential operators on unoriented graphs

1.3.1 First-order differential operators

For the following definitions of differential operators for an unoriented graph, including the gradient, adjoint and p-Laplacian operators, we need to introduce a new definition of a special vertex $v_{\tilde{q}}$, as in [FTB23]:

DEFINITION 1.23: Special vertex $v_{\tilde{q}}$.

For an unoriented graph $G_u = (V, E_G, w, W_E)$ we assign to each edge $e_q \in E_G$ a specific vertex $v_{\tilde{q}} \in e_q$. This is expressed as:

$$v_{\tilde{q}} := v_i, \quad \text{for} \;\; e_q = \{v_i, v_j\} \in E_G.$$

- **Choose the vertex $v_{\tilde{q}}$**:

  If the choice of $v_{\tilde{q}}$ is unclear, an alternative is to include each edge e_q exactly $|e_q| = 2$ times in the set of edges E_G. Where each version of the edge has a different special vertex $v_i \in e_q$, as proposed in [FTB23].

- **Transformation into an oriented graph**:

  This alternative approach allows the transformation of the unoriented graph into an oriented one. For each edge $e_q \in E_G$, $|e_q| = 2$ arcs are created in the oriented graph. Each arc has one output vertex $v_i \in e_q$ (corresponding to the special vertex $v_{\tilde{q}}$) and $|e_q| - 1 = 1$ input vertices.

DEFINITION 1.24: Vertex-edge characteristic function ψ.

The vertex-edge characteristic function $\psi : V \times E_G \to \{0, 1\}$ for an unoriented weighted graph $G_u = (V, E_G, w, W_E)$ is defined as:

$$\psi(v_i, e_q) = \begin{cases} 1 & \text{for} \quad v_i \in e_q, \\ 0 & \text{for} \quad v_i \notin e_q. \end{cases} \tag{1.33}$$

DEFINITION 1.25: Special vertex-edge characteristic function $\widetilde{\psi}$.

The special vertex-edge characteristic function $\widetilde{\psi} : V \times E_G \to \{0, 1\}$ for an unoriented weighted graph $G_u = (V, E_G, w, W_E)$ is defined as:

$$\widetilde{\psi}(v_i, e_q) = \begin{cases} 1 & \text{for} \quad v_i = v_{\tilde{q}}, \\ 0 & \text{for} \quad v_i \neq v_{\tilde{q}}, \end{cases} \tag{1.34}$$

and this function indicates if vertex $v_i \in V$ is the special vertex $v_{\tilde{q}}$ of edge $e_q \in E_G$. Moreover, the next equality holds true for $\forall v_i \in V$ and $\forall e_q \in E_G$:

$$\widetilde{\psi}(v_i, e_q) = 1 \quad \Rightarrow \quad \psi(v_i, e_q) = 1. \tag{1.35}$$

Future definitions of first-order differential operators for unoriented graphs $G_u = (V, E_G, w, W_E)$ will contain the following parameters:

- w_I and W_I represent vertex and edge weight functions, respectively, originating from the inner product of the space of vertex or edge functions.
- w_G and W_G are vertex and edge weight functions, respectively, introduced in the gradient definitions below.
- parameters $\alpha, \beta, \zeta, \gamma, \epsilon, \eta, \theta \in \mathbb{R}$ that can be chosen additionally to adapt to different cases.

The vertex gradient operator, as well as the derived vertex adjoint and vertex p-Laplacian operators for unoriented graphs are introduced in [FTB23], however this paper does not include a definition for an edge gradient, edge adjoint or edge p-Laplacian operator. Therefore, this thesis introduces novel operators for unoriented graphs based on edge functions inspired by the operators for vertex functions from [FTB23].

DEFINITION 1.26: Vertex gradient operator ∇_v^G.

Let $\nabla_v^G : S(V) \to S(E_G)$ be a vertex gradient operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$, which is defined as:

$$\nabla_v^G f : E_G \to \mathbb{R}, \quad e_q \mapsto \nabla_v^G f(e_q),$$

with

$$\begin{aligned}\nabla_v^G f(e_q) &= W_G(e_q)^\gamma \Bigg[\sum_{v_i \in V} \psi(v_i, e_q)\Big(w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}})\Big)\Bigg] \\ &= W_G(e_q)^\gamma \Bigg[\Big(\sum_{v_i \in V} \psi(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i)\Big) - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}})\Bigg].\end{aligned} \tag{1.36}$$

If you set the weights W_G, w_G and w_I equal to 1 in the vertex gradient operator (1.36), the simplified formula would be:

$$\nabla_v^G f(e_q) = \sum_{v_i \in V} \psi(v_i, e_q) f(v_i) - |e_q| f(v_{\tilde{q}}). \tag{1.37}$$

DEFINITION 1.27: Edge gradient operator ∇_e^G.

Let $\nabla_e^G : S(E_G) \to S(V)$ be an edge gradient operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$, which is defined as:

$$\nabla_e^G F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_e^G F(v_i),$$

$$\nabla_e^G F(v_i) = w_G(v_i)^\zeta \sum_{e_q \in E_G} \Big[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q)\Big] F(e_q) W_I(e_q)^\beta W_G(e_q)^\theta. \tag{1.38}$$

If the weights W_G, w_G and W_I are set to 1 in the edge gradient operator (1.38), the simplified formula would be:

$$\nabla_e^G F(v_i) = \sum_{e_q \in E_G} \Big[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q)\Big] F(e_q), \tag{1.39}$$

which corresponds to summing up the following terms:

$$\begin{cases} F(e_q) & \text{for} \quad v_i \neq v_{\tilde{q}}, \\ F(e_q) - 2F(e_q) = -F(e_q) & \text{for} \quad v_i = v_{\tilde{q}}, \end{cases} . \tag{1.40}$$

The edge gradient at vertex v_i with respect to the edge function F is the sum of differences between characteristic functions, multiplied by the function values $F(e_q)$.

DEFINITION 1.28: Vertex adjoint operator ∇_v^*.

Let $\nabla_v^* : S(E_G) \to S(V)$ be the vertex adjoint operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$ with constant weight functions $W_G = 1, w_G = 1$ and $W_I = 1$, then the vertex adjoint operator is defined as:

$$\nabla_v^* F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_v^* F(v_i),$$

$$\nabla_v^* F(v_i) = \sum_{e_q \in E_G} \Big[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \Big] F(v_i). \tag{1.41}$$

DEFINITION 1.29: Edge adjoint operator ∇_e^*.

Let $\nabla_e^* : S(V) \to S(E_G)$ be the edge adjoint operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$ with constant weight functions $W_G = 1, w_G = 1$ and $w_I = 1$, then the edge adjoint operator is defined as:

$$\nabla_e^* f : E_G \to \mathbb{R}, \quad e_q \mapsto \nabla_e^* f(e_q),$$

$$\nabla_e^* f(e_q) = \Big[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \Big] f(v_i). \tag{1.42}$$

The connection between gradient and adjoint operators holds true $\forall f \in S(V)$ and $\forall G \in S(E_G)$ in an unoriented weighted graph $G_u = (V, E_G, w, W)$:

$$\langle G, \nabla_v^G f \rangle_{S(E_G)} = \langle f, \nabla_v^* G \rangle_{S(V)}, \tag{1.43}$$

$$\langle f, \nabla_e^G G \rangle_{S(V)} = \langle G, \nabla_e^* f \rangle_{S(E_G)}. \tag{1.44}$$

Proof. For an unoriented weighted graph $G_u = (V, E_G, w, W)$ with a vertex function $f \in S(V)$ and an edge function $G \in S(E_G)$, we can prove the connection between the edge gradient ∇_e^G and the edge adjoint ∇_e^* as shown here (1.44). For this proof we will use the definitions of the inner product in $S(V)$ and the edge gradient operator ∇_e^G with constant (equal to 1) weight functions.

$$\begin{aligned}\langle f, \nabla_e^G G\rangle_{S(V)} &= \sum_{v_i\in V} f(v_i)\nabla_e^G(v_i)G(e_q)\\ &= \sum_{v_i\in V} f(v_i) \sum_{e_q\in E_G}\Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big]G(e_q)\\ &= \sum_{v_i\in V}\sum_{e_q\in E_G}\Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big]f(v_i)G(e_q)\end{aligned}$$

By exchanging the two sums, the following is obtained:

$$\begin{aligned}&= \sum_{e_q\in E_G}\sum_{v_i\in V}\Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big]f(v_i)G(e_q)\\ &= \sum_{e_q\in E_G} G(e_q)\sum_{v_i\in V}\Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big]f(v_i)\end{aligned}$$

And by using the definitions for the inner product on the space of all edge functions $S(E_G)$ and the edge adjoint operator ∇_e^* we have:

$$= \sum_{e_q\in E_G} G(e_q)\nabla_e^* f = \langle G, \nabla_e^* f\rangle_{S(E_G)}.$$

$\square$

Now we define the vertex divergence and edge divergence operators div_v^G and div_e^G based on the vertex and edge adjoint operators.

DEFINITION 1.30: Vertex divergence operator div_v^G .

Let $\mathrm{div}_v^G : S(E_G) \to S(V)$ be the vertex divergence operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$. Based on the equivalence from the continuum setting we have:

$$\mathrm{div}_v^G = -\nabla_v^* \qquad \Rightarrow \qquad \mathrm{div}_e^G F(v_i) = -\nabla_v^* F(v_i).$$

Therefore, the weighted divergence of an edge function $F \in S(E_G)$ at a vertex $v_i \in V$ and given that $W_G = 1, w_G = 1$ and $W_I = 1$, is equal to:

$$\mathrm{div}_v^G = \sum_{e_q\in E_G}\Big[|e_q|\widetilde{\psi}(v_i,e_q) - \psi(v_i,e_q)\Big]F(e_q). \tag{1.45}$$

DEFINITION 1.31: Edge divergence operator div_e^G .

Let $\mathrm{div}_e^G : S(V) \to S(E_G)$ be the edge divergence operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$. Based on the equivalence from the continuum setting we have:

$$\mathrm{div}_e^G = -\nabla_e^* \qquad \Rightarrow \qquad \mathrm{div}_e^G f(e_q) = -\nabla_e^* f(e_q).$$

Therefore, the weighted divergence of a vertex function $f \in S(V)$ at an edge $e_q \in E_G$ and given that $W_G = 1, w_G = 1$ and $w_I = 1$, is equal to:

$$\mathrm{div}_e^G f(e_q) = \Big[|e_q|\widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q)\Big] f(v_i). \tag{1.46}$$

1.3.2 p-Laplacian operators

DEFINITION 1.32: Vertex Laplacian operator Δ_v .

Let $\Delta_v : S(V) \to S(V)$ be the vertex Laplacian operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$. Derived from the connection within the continuum setting, we obtain:

$$\Delta_v f = \mathrm{div}_v^G(\nabla_v^G f), \tag{1.47}$$

so the weighted Laplacian of an vertex function $f \in S(V)$ at vertex $v_i \in V$ when W_G, W_I, w_G and w_I are equal to 1, is given by:

$$\Delta_v f(v_i) = \sum_{e_q \in E_G} \Big[\psi(v_i, e_q) - |e_q|\widetilde{\psi}(v_i, e_q)\Big] \sum_{v_i \in V} \psi(v_j, e_q) f(v_j) - |e_q| f(v_{\tilde{q}}), \tag{1.48}$$

DEFINITION 1.33: Edge Laplacian operator Δ_e.

Let $\Delta_e : S(E_G) \to S(E_G)$ be the edge Laplacian operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$. Derived from the connection within the continuum setting, we obtain:

$$\Delta_e F = \mathrm{div}_e^G(\nabla_e^G F), \tag{1.49}$$

so the weighted Laplacian of an edge function $F \in S(E_G)$ at an edge $e_q \in E_G$ when W_G, W_I, w_G and w_I are equal to 1, is given by:

$$\Delta_e F(e_q) = \left[|e_q|\widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q)\right] \sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q|\widetilde{\psi}(v_i, e_q)\right] F(e_q). \tag{1.50}$$

Proof. For an unoriented weighted graph $G_u = (V, E_G, w, W)$ with an edge function $F \in S(E_G)$, we can prove the connection between the edge Laplacian Δ_e and the edge divergence div_e^G as shown here (1.49). For this proof we will use the definitions of the edge divergence operator div_e and the edge gradient operator ∇_e^G with constant (equal to 1) weight functions.

$$\begin{aligned}
\mathrm{div}_e^G(\nabla_e^G F)(v_i) &= \left[|e_q|\widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q)\right] \nabla_e^G F(v_i) \\
&= \left[|e_q|\widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q)\right] \sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q|\widetilde{\psi}(v_i, e_q)\right] F(e_q) \\
&= \Delta_e F(e_q).
\end{aligned}$$

□

DEFINITION 1.34: Vertex p-Laplacian operator Δ_v^p .

Let $\Delta_v^p : S(V) \to S(V)$ be the vertex p-Laplacian operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$. Derived from the connection within the continuum setting, we obtain:

$$\Delta_v^p f = \mathrm{div}_v^G(|\nabla_v^G f|^{p-2} \nabla_v^G f). \tag{1.51}$$

Hence, the weighted p-Laplacian of a vertex function $f \in S(V)$ at vertex $v_i \in V$, considering that W_G, W_I, w_G and w_I all are equal to 1, given by:

$$\Delta_v^p f(v_i) = \sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q|\widetilde{\psi}(v_i, e_q)\right] \left|\sum_{v_j \in V} \psi(v_j, e_q) f(v_j) - |e_q| f(v_{\tilde{q}})\right|^{p-2} \left(\sum_{v_k \in V} \psi(v_k, e_q) f(v_k) - |e_q| f(v_{\tilde{q}})\right). \tag{1.52}$$

The full formula of the vertex p-Laplacian operator including the weight functions would look like this:

$$\Delta_v^p f(v_i) = \sum_{e_q \in E_G} \left[\psi(v_i, e_q) w_G(v_i)^\epsilon - |e_q| \widetilde{\psi}(v_i, e_q) w_G(v_i)^\eta \right] W_I(e_q)^\beta W_G(e_q)^{p\gamma}$$
$$\left| \sum_{v_j \in V} \psi(v_j, e_q) f(v_j) w_I(v_j)^\alpha w_G(v_j)^\epsilon - |e_q| f(v_{\tilde{q}})^\alpha w_I(v_{\tilde{q}}) w_G(v_{\tilde{q}})^\eta \right|^{p-2} \tag{1.53}$$
$$\left(\sum_{v_k \in V} \psi(v_k, e_q) f(v_k) w_I(v_k)^\alpha w_G(v_k)^\epsilon - |e_q| f(v_{\tilde{q}}) w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta \right).$$

DEFINITION 1.35: Edge p-Laplacian operator Δ_e^p .

Let $\Delta_e^p : S(E_G) \to S(E_G)$ be the edge p-Laplacian operator for an unoriented weighted graph $G_u = (V, E_G, w, W)$. Derived from the connection within the continuum setting, we obtain:

$$\Delta_e^p F = \text{div}_e^G (|\nabla_e^G F|^{p-2} \nabla_e^G F). \tag{1.54}$$

Hence, the weighted p-Laplacian of an edge function $F \in S(E_G)$ at an edge $e_q \in E_G$, considering that W_G, W_I, w_G and w_I all are equal to 1, given by:

$$\Delta_e^p F(e_q) = \left[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \right]$$
$$\left| \sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \right] F(e_q) \right|^{p-2} \tag{1.55}$$
$$\sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \right] F(e_q).$$

The full formula of the edge p-Laplacian operator including the weight functions would look like this:

$$\Delta_e^p F(e_q) = W_G(e_q)^\theta \Bigg(\left[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \right] w_I(v_i)^\alpha w_G(v_i)^{p\zeta}$$
$$\left| \sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \right] F(e_q) W_I(e_q)^\beta W_G(e_q)^\theta \right|^{p-2} \tag{1.56}$$
$$\sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \right] F(e_r) W_I(e_q)^\beta W_G(e_q)^\theta \Bigg).$$

Proof. For an unoriented weighted graph $G_u = (V, E_G, w, W)$ with an edge function $F \in S(E_G)$, we can prove the connection for the p-Laplacian operator Δ_e^p as shown here

(1.54). For this proof we will use the definitions of the edge divergence operator div_e^G and the edge gradient operator ∇_e^G with constant (equal to 1) weight functions.

$$
\begin{aligned}
\operatorname{div}_e^G(|\nabla_e^G F|^{p-2}\nabla_e^G F) = W_G(e_q)^\theta \Bigg(&\Big[|e_q|\widetilde{\psi}(v_i,e_q) - \psi(v_i,e_q)\Big] w_I(v_i)^\alpha w_G(v_i)^\zeta \\
&|\nabla_e^G F(v_i)|^{p-2}\nabla_e^G F(v_i)\Bigg) \\
&= W_G(e_q)^\theta \Bigg(\Big[|e_q|\widetilde{\psi}(v_i,e_q) - \psi(v_i,e_q)\Big] w_I(v_i)^\alpha w_G(v_i)^\zeta \\
&\left| w_G(v_i)^\zeta \sum_{e_q \in E_G} \Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big] F(e_q) W_I(e_q)^\beta W_G(e_q)^\theta \right|^{p-2} \\
&w_G(v_i)^\zeta \sum_{e_q \in E_G} \Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big] F(e_r) W_I(e_q)^\beta W_G(e_q)^\theta \Bigg) \\
&= W_G(e_q)^\theta \Bigg(\Big[|e_q|\widetilde{\psi}(v_i,e_q) - \psi(v_i,e_q)\Big] w_I(v_i)^\alpha w_G(v_i)^{\zeta+\zeta(p-2)+\zeta} \\
&\left| \sum_{e_q \in E_G} \Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big] F(e_q) W_I(e_q)^\beta W_G(e_q)^\theta \right|^{p-2} \\
&\sum_{e_q \in E_G} \Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big] F(e_r) W_I(e_q)^\beta W_G(e_q)^\theta \Bigg) \\
&= W_G(e_q)^\theta \Bigg(\Big[|e_q|\widetilde{\psi}(v_i,e_q) - \psi(v_i,e_q)\Big] w_I(v_i)^\alpha w_G(v_i)^{p\zeta} \\
&\left| \sum_{e_q \in E_G} \Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big] F(e_q) W_I(e_q)^\beta W_G(e_q)^\theta \right|^{p-2} \\
&\sum_{e_q \in E_G} \Big[\psi(v_i,e_q) - |e_q|\widetilde{\psi}(v_i,e_q)\Big] F(e_r) W_I(e_q)^\beta W_G(e_q)^\theta \Bigg) \\
&= \Delta_e^p F(e_q).
\end{aligned}
$$

Hence, the equation is valid (1.54) for all edges e_q in E_G and for all edge functions $F \in S(E_G)$.

□

1.4 Differential operators on oriented graphs

The gradient, adjoint and p-Laplacian operators for oriented graphs from the following sub-sections were introduced by [FTB23] and are summarised here in order to allow for a comparison with the differential operators for unoriented graphs of the previous section.

1.4.1 First-order differential operators

DEFINITION 1.36: Vertex-arc characteristic functions ψ_{out} and ψ_{in}.

Let $\psi_{out} : V \times A_G \to \mathbb{R}$ be the out-degree function and $\psi_{in} : V \times A_G \to \mathbb{R}$ be the in-degree function for an oriented weighted graph $G_o = (V, A_G, w, W)$, which are defined as:

$$\psi_{out}(v_i, a_q) = \begin{cases} 1 & \text{for} \quad a_q = (v_i, v_j) \text{ for } v_j \in V, \\ 0 & \text{for} \quad \text{otherwise.} \end{cases} \tag{1.57}$$

$$\psi_{in}(v_i, a_q) = \begin{cases} 1 & \text{for} \quad a_q = (v_j, v_i) \text{ for } v_j \in V, \\ 0 & \text{for} \quad \text{otherwise.} \end{cases} \tag{1.58}$$

Similar to the case of unoriented graphs, definitions of first-order differential operators for an oriented graph $G_o = (V, A_G, w, W_A)$ will contain the following parameters:

- w_I and W_I represent vertex and arc weight functions, respectively, originating from the inner product of the space of vertex or arc functions.
- w_G and W_G are vertex and arc weight functions, respectively, introduced in the gradient definitions below.
- parameters $\alpha, \beta, \zeta, \gamma, \epsilon, \eta, \theta \in \mathbb{R}$ that can be chosen additionally to adapt to different cases.

DEFINITION 1.37: Vertex gradient operator ∇_v^G.

Let $\nabla_v^G : S(V) \to S(A_G)$ be the vertex gradient operator for an oriented weighted graph $G_o = (V, A_G, w, W)$, which is given as:

$$\nabla_v^G f : A_G \to \mathbb{R}, \quad a_q \mapsto \nabla_v^G f(a_q),$$

$$\nabla_v^G f(a_q) = W_G(a_q)^\gamma \left[w_I(v_j)^\alpha w_G(v_j)^\epsilon f(v_j) - w_I(v_i)^\alpha w_G(v_i)^\eta f(v_i) \right]. \tag{1.59}$$

for an arc $a_q = (v_i, v_j)$. Moreover, the above gradient satisfies the expected property that for any constant vertex function $f \in S(V)$ it holds true that $\nabla_v^G f(a_q) = 0$, if $w_I(v_j)^\alpha w_G(v_j)^\epsilon = w_I(v_i)^\alpha w_G(v_i)^\eta$.

Let us set the weights W_G, w_G and w_I equal to 1 to simplify the formula, so that (1.59) becomes:

$$\nabla_v^G f(a_q) = f(v_j) - f(v_i). \tag{1.60}$$

This simplified expression implies that the vertex gradient of an oriented graph G_o for an arc $a_q = (v_i, v_j)$ is simply the difference between the function values of the vertices v_j and v_i.

DEFINITION 1.38: Arc gradient operator ∇_a^G.

Let $\nabla_a^G : S(A_G) \to S(V)$ be the arc gradient operator for an oriented weighted graph $G_o = (V, A_G, w, W)$, which is given as:

$$\nabla_a^G F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_a^G F(v_i),$$

$$\nabla_a^G F(v_i) = w_G(v_i)^\zeta \sum_{a_q \in A_G} \left[\frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} \right] W_I(a_q)^\beta W_G(a_q)^\theta F(a_q). \tag{1.61}$$

Similar to the definition above, we simplify the formula by setting the weights W_I, W_G and w_G equal to 1, so that (1.61) results in:

$$\nabla_a^G F(v_i) = \sum_{a_q \in A_G} \left[\frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} \right] F(a_q). \tag{1.62}$$

This simplified expression implies that the arc gradient at a vertex v_i in an oriented graph G_o is the sum of the differences between *in* and *out* contributions multiplied by the function values of the corresponding arcs.

The following definitions are given by default for the case where the weights W_G, W_I, w_G, w_I are equal to 1, unless stated otherwise.

DEFINITION 1.39: Additional vertex-arc characteristic function.

Let $\psi : V \times A_G \to \mathbb{R}$ be the vertex-arc characteristic function for an oriented weighted graph $G_o = (V, A_G, w, W)$, which are defined as:

$$\psi(v_i, a_q) = \begin{cases} 1 & \text{for} \quad a_q = (v_i, v_j) \text{ for } v_j \in V, \\ -1 & \text{for} \quad a_q = (v_j, v_i) \text{ for } v_j \in V, \\ 0 & \text{for} \quad \text{otherwise}. \end{cases} \tag{1.63}$$

DEFINITION 1.40: Vertex adjoint operator ∇_v^*.

Let $\nabla_v^* : S(A_G) \to S(V)$ be the vertex adjoint operator for an oriented weighted graph $G_o = (V, A_G, w, W)$, which is given as:

$$\nabla_v^* F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_v^* F(v_i),$$

$$\nabla_v^* F(v_i) = \sum_{v_j \in V} \Big[\psi(v_j, v_i) F(v_j, v_i) - \psi(v_i, v_j) F(v_i, v_j) \Big]. \tag{1.64}$$

DEFINITION 1.41: Arc adjoint operator ∇_a^*.

Let $\nabla_a^* : S(V) \to S(A_G)$ be the arc adjoint operator for an oriented weighted graph $G_o = (V, A_G, w, W)$, which is given as:

$$\nabla_a^* f : A_G \to \mathbb{R}, \quad a_q \mapsto \nabla_a^* f(a_q),$$

$$\nabla_a^* f(a_q) = \nabla_a^* f(v_i, v_j) = \frac{1}{\deg_{in}(v_j)} f(v_j) - \frac{1}{\deg_{out}(v_i)} f(v_i). \tag{1.65}$$

With the inner products defined for the space of vertex functions $S(V)$ and for the space of arc functions $S(A_G)$, the adjoint operators for oriented graphs $G_o = (V, A_G, w, W)$ satisfy $\forall f \in S(V)$ and $\forall G \in S(A_G)$:

$$\langle G, \nabla_v^G f \rangle_{S(A_G)} = \langle f, \nabla_v^* G \rangle_{S(V)}, \tag{1.66}$$

$$\langle f, \nabla_a^G G \rangle_{S(V)} = \langle G, \nabla_a^* f \rangle_{S(A_G)}. \tag{1.67}$$

Now we define the vertex divergence operators div_v^G and div_a^G based on the vertex and arc adjoint operators.

DEFINITION 1.42: Vertex divergence operator div_v^G.

Let $\mathrm{div}_v^G : S(A_G) \to S(V)$ be the vertex divergence operator for an oriented weighted graph $G_o = (V, A_G, w, W)$. Based on the equivalence from the continuum setting we have:

$$\mathrm{div}_v^G = -\nabla_v^* \quad \Rightarrow \quad \mathrm{div}_v^G F(v_i) = -\nabla_v^* F(v_i).$$

The weighted vertex divergence of arc function $F \in S(A_G)$ at a vertex $v_i \in V$ is equal to:

$$\mathrm{div}_v^G F(v_i) = \sum_{v_j \in V} \Big[\psi(v_i, v_j) F(v_i, v_j) - \psi(v_j, v_i) F(v_j, v_i) \Big]. \tag{1.68}$$

DEFINITION 1.43: Arc divergence operator div_a^G.

Let $\mathrm{div}_a^G : S(V) \to S(A_G)$ be the arc divergence operator for an oriented weighted graph $G_o = (V, A_G, w, W)$. Based on the equivalence from the continuum setting we have:

$$\mathrm{div}_a^G = -\nabla_a^* \quad \Rightarrow \quad \mathrm{div}_a^G f(a_q) = -\nabla_a^* f(a_q).$$

The weighted divergence of vertex function $f \in S(V)$ at an arc $a_q \in A_G$ with $a_q = (v_i, v_j)$ for $v_i, v_j \in V$ is equal to:

$$\mathrm{div}_a^G f(a_q) = \frac{1}{\deg_{out}(v_i)} f(v_i) - \frac{1}{\deg_{in}(v_j)} f(v_j). \tag{1.69}$$

1.4.2 p-Laplacian operators

DEFINITION 1.44: Vertex Laplacian operator Δ_v.

Let $\Delta_v : S(V) \to S(V)$ be the vertex Laplacian operator for an oriented weighted graph $G_o = (V, A_G, w, W)$. The vertex Laplacian operator is derived from the connection in the continuum setting and hence defined as:

$$\Delta_v f = \mathrm{div}_v^G (\nabla_v^G f).$$

The weighted Laplacian of vertex function $f \in S(V)$ at a vertex $v_i \in V$ is equal to:

$$\Delta_v f(v_i) = \sum_{v_j \in V} \Bigg[\Big(\psi(v_i, v_j) + \psi(v_j, v_i) \Big) f(v_j) - \Big(\psi(v_i, v_j) + \psi(v_j, v_i) \Big) f(v_i) \Bigg]. \tag{1.70}$$

DEFINITION 1.45: Arc Laplacian operator Δ_a.

Let $\Delta_a : S(A_G) \to S(A_G)$ be the arc Laplacian operator for an oriented weighted graph $G_o = (V, A_G, w, W)$. So the arc Laplacian operator derived from the connection of the continuum setting, is given by:

$$\Delta_a F = \mathrm{div}_a^G(\nabla_a^G F).$$

And the weighted Laplacian of arc function $f \in S(A_G)$ at an arc $a_q = (v_i, v_j) \in A_G$ is equal to:

$$\begin{aligned}\Delta_a F(a_q) = &\frac{1}{\deg_{out}(v_i)} \sum_{a_r \in A_G} \left(\frac{\psi_{in}(v_i, a_r)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_r)}{\deg_{out}(v_i)} \right) F(a_r) \\ &- \frac{1}{\deg_{in}(v_j)} \sum_{a_s \in A_G} \left(\frac{\psi_{in}(v_j, a_s)}{\deg_{in}(v_j)} - \frac{\psi_{out}(v_j, a_s)}{\deg_{out}(v_j)} \right) F(a_s).\end{aligned} \tag{1.71}$$

DEFINITION 1.46: Vertex p-Laplacian operator Δ_v^p.

Let $\Delta_v^p : S(V) \to S(V)$ be the p-Laplacian operator for an oriented weighted graph $G_o = (V, A_G, w, W)$, based on the connection in the continuum setting, we obtain:

$$\Delta_v^p f = \mathrm{div}_v^G(|\nabla_v^G f|^{p-2} \nabla_v^G f). \tag{1.72}$$

Hence, the weighted p-Laplacian of vertex function $f \in S(V)$ at a vertex $v_i \in V$ is equal to:

$$\begin{aligned}\Delta_v^p f(v_i) = \sum_{v_j \in V} \Bigg[&\psi(v_i, v_j) \Big| f(v_j) - f(v_i) \Big|^{p-2} \Big(f(v_j) - f(v_i) \Big) \\ &- \psi(v_i, v_j) \Big| f(v_i) - f(v_j) \Big|^{p-2} \Big(f(v_i) - f(v_j) \Big) \Bigg].\end{aligned} \tag{1.73}$$

The full formula for the vertex p-Laplacian operator including the weights W_I, W_G, w_G and w_I would look like this:

$$\begin{aligned}\Delta_v^p f(v_i) = \sum_{v_j \in V} \Bigg[&\psi(v_i, v_j) \Big| w_I(v_j)^\alpha w_G(v_j)^\epsilon f(v_j) - w_I(v_i)^\alpha w_G(v_i)^\eta f(v_i) \Big|^{p-2} \\ &\Big(w_G(v_i)^\eta w_I(v_j)^\alpha w_G(v_j)^\epsilon f(v_j) - w_I(v_i)^\alpha w_G(v_i)^{2\eta} f(v_i) \Big) \\ &- \psi(v_i, v_j) \Big| w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - w_I(v_j)^\alpha w_G(v_j)^\eta f(v_j) \Big|^{p-2} \\ &\Big(w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - w_I(v_j)^\alpha w_G(v_j)^\eta f(v_j) \Big) \Bigg] \\ &W_I(v_i, v_j)^\beta W_G(v_i, v_j)^{p\gamma}. \end{aligned} \tag{1.74}$$

DEFINITION 1.47: Arc p-Laplacian operator Δ_a^p.

Let $\Delta_a^p : S(A_G) \to S(A_G)$ be the arc p-Laplacian operator for an oriented weighted graph $G_o = (V, A_G, w, W)$, derived from the connection of the continuum setting, we obtain:

$$\Delta_a^p F = \mathrm{div}_a^G(|\nabla_a^G F|^{p-2} \nabla_a^G F). \tag{1.75}$$

And finally the weighted p-Laplacian of arc function $F \in S(A_G)$ at an arc $a_q = (v_i, v_j) \in A_G$ is equal to:

$$\begin{aligned}\Delta_a^p F(a_q) = &\frac{1}{\deg_{out}(v_i)} \left| \sum_{a_r \in A_G} \left(\frac{\psi_{in}(v_i, a_r)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_r)}{\deg_{out}(v_i)} \right) F(a_r) \right|^{p-2} \\ &\sum_{a_s \in A_G} \left(\frac{\psi_{in}(v_i, a_s)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_s)}{\deg_{out}(v_i)} \right) F(a_s) \\ &- \frac{1}{\deg_{in}(v_j)} \left| \sum_{a_t \in A_G} \left(\frac{\psi_{in}(v_j, a_t)}{\deg_{in}(v_j)} - \frac{\psi_{out}(v_j, a_t)}{\deg_{out}(v_j)} \right) F(a_t) \right|^{p-2} \\ &\sum_{a_u \in A_G} \left(\frac{\psi_{in}(v_j, a_u)}{\deg_{in}(v_j)} - \frac{\psi_{out}(v_j, a_u)}{\deg_{out}(v_j)} \right) F(a_u). \end{aligned} \tag{1.76}$$

The full formula for the arc p-Laplacian operator including weights W_I, W_G, w_G and w_I is given by:

$$
\begin{aligned}
\Delta_a^p F(a_q) = W_G(v_i, v_j)^\theta \Bigg[\\
&\frac{w_I(v_i)^\alpha w_G(v_i)^{p\zeta}}{\deg_{out}(v_i)} \left| \sum_{a_r \in A_G} \left(\frac{\psi_{in}(v_i, a_r)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_r)}{\deg_{out}(v_i)} \right) W_I(a_r)^\beta W_G(a_r)^\theta F(a_r) \right|^{p-2} \\
&\sum_{a_s \in A_G} \left(\frac{\psi_{in}(v_i, a_s)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_s)}{\deg_{out}(v_i)} \right) W_I(a_s)^\beta W_G(a_s)^\theta F(a_s) \\
&- \frac{w_I(v_j)^\alpha w_G(v_j)^{p\zeta}}{\deg_{in}(v_j)} \left| \sum_{a_t \in A_G} \left(\frac{\psi_{in}(v_j, a_t)}{\deg_{in}(v_j)} - \frac{\psi_{out}(v_j, a_t)}{\deg_{out}(v_j)} \right) W_I(a_t)^\beta W_G(a_t)^\theta F(a_t) \right|^{p-2} \\
&\sum_{a_u \in A_G} \left(\frac{\psi_{in}(v_j, a_u)}{\deg_{in}(v_j)} - \frac{\psi_{out}(v_j, a_u)}{\deg_{out}(v_j)} \right) W_I(a_u)^\beta W_G(a_u)^\theta F(a_u) \Bigg].
\end{aligned}
\tag{1.77}
$$

Chapter 2

Hypergraph notation

Since hypergraphs are a generalisation of graphs, many of the definitions of graphs apply to hypergraphs as well. In this chapter we introduce some basic notations for hypergraphs.

The structure and content of the following sections refer to the first chapter, but all definitions apply to both oriented and unoriented hypergraphs. So we start with the main definitions for hypergraphs in Section 2.1 and end with an illustration of them.

Then in 2.15 we consider functions defined on both types of hypergraphs, oriented and unoriented. The following concepts are introduced: vertex function 2.15, weight function 2.16, hyperedge 2.19 or hyperarc 2.19 function, and the spaces of these functions 2.21.

Section 2.3 is dedicated to unoriented hypergraphs only and is divided into two parts, Section 2.3.1 on first-order differential operators and Section 2.3.2 on p-Laplacian operators. In the next Section 2.4 we talk about differential operators in the case of an oriented hypergraph.

To conclude this chapter with Section 2.5, we show the relationship between graphs, consisting of vertices and edges or arcs, which play a fundamental role in representing pairwise relationships, and hypergraphs, which connect subsets of vertices and allow a more detailed representation of complex links.

2.1 Hypergraph theory: main definitions and properties

DEFINITION 2.1: Unoriented hypergraph.

An unoriented hypergraph H_u is denoted by a pair $H_u = (V, E_H)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices and $E_H = \{e_1, e_2, ..., e_k\}$ is the set of hyperedges:

$$H_u = (V, E_H),\ E_H \subseteq 2^V \backslash \{\emptyset\}, \tag{2.1}$$

where 2^V denotes the power set of the set of vertices V.

The hyperedges are subsets of connected vertices, and the hypergraph provides a framework for studying relationships and structures involving more than two vertices simultaneously.

DEFINITION 2.2: Oriented hypergraph.

An oriented hypergraph H_o is denoted by a pair $H_o = (V, A_H)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices and every hyperarc $a_q \in A_H$ is a tuple of two disjoint non-empty subsets of the set of vertices $a_q = (a_q^{out}, a_q^{in})$ (input and output vertices). $A_H = \{a_1, a_2, ..., a_q\}$ denotes the set of hyperarcs:

$$H_o = (V, A_H), A_H \subseteq (2^V \backslash \{\emptyset\}) \times (2^V \backslash \{\emptyset\}). \tag{2.2}$$

Therefore, hyperarcs consist of at least 2 and at most $|V|$ vertices and each vertex of the hyperarc is either an output or an input vertex, which give a hyperarc an orientation in contrast to the unoriented hyperedges in unoriented hypergraphs. Changing the orientation of a hyperarc a_q means exchanging its input and output vertices, leading to the pair (a_q^{out}, a_q^{in}).

DEFINITION 2.3: Set of incident hyperedges $\hat{E}_H$.

Let $\hat{E}_H$ be the set of incident hyperedges of a vertex $v_i \in V$ in an unoriented hypergraph $H_u = (V, E_H)$ which is given as:

$$\hat{E}_H(v_i) = \{e_q \in E_H \mid v_i \in e_q\}. \tag{2.3}$$

DEFINITION 2.4: Set of incident hyperarcs $\hat{A}_H$.

Let $\hat{A}_H$ be the set of incident hyperarcs of a vertex $v_i \in V$ in an oriented hypergraph $H_o = (V, A_H)$ which is given as:

$$\hat{A}_H(v_i) = \{a_q \in A_H \mid v_i \in a_q\} = \{a_q \in A_H \mid v_i \in a_q^{out} \text{ or } v_i \in a_q^{in}\}. \tag{2.4}$$

Moreover, a vertex $v_i \in V$ is called isolated, if it has no connections with any other vertex $v_j \in V$, which corresponds to $\hat{E}_H(v_i) = \emptyset$ or $\hat{A}_H(v_i) = \emptyset$.

DEFINITION 2.5: Degree of vertex.

The degree of vertex v_i in an unoriented hypergraph $H_u = (V, E_H)$ is the number of hyperedges incident to v_i and it is denoted by:

$$\deg(v_i) = |\hat{E}_H(v_i)|, \tag{2.5}$$

where $\hat{E}_H(v_i)$ is the set of hyperedges incident to vertex v_i.

For an oriented hypergraph $H_o = (V, A_H)$ the degree of a vertex $v_i \in V$ is defined as follows:

$$\deg(v_i) = |\hat{A}_H(v_i)|, \tag{2.6}$$

where $\hat{A}_H(v_i)$ is the set of hyperarcs incident to vertex v_i.

DEFINITION 2.6: Incidence matrix for hypergraphs.

An incidence matrix for an oriented hypergraph $H_o = (V, A_H)$ can be defined as a $|V| \times |A_H|$ matrix K_{H_o}, where:

$$K_{H_o}(v_i, a_q) = \begin{cases} 1, & \text{if hyperarc } a_q \text{ is directed to vertex } v_i, \\ -1, & \text{if hyperarc } a_q \text{ is directed from vertex } v_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

For an unoriented hypergraph $H_u = (V, E_H)$ the incidence matrix K_{H_u} is defined as a $|V| \times |E_H|$ matrix, where:

$$K_{H_u}(v_i, e_q) = \begin{cases} 1, & \text{if hyperedge } e_q \text{ is incident to vertex } v_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

DEFINITION 2.7: Weighted unoriented hypergraph.

An unoriented hypergraph is weighted if $H_u = (V, E_H)$ has weights associated to its hyperedges. This means that each hyperedge $e \in E_H$ carries a value, and these weights are non-negative, so:

$$w : E_H \longrightarrow \mathbb{R}^+, \quad w_e \geq 0.$$

The full expression of an unoriented weighted hypergraph is given by:

$$H_u = (V, E_H, w). \tag{2.9}$$

DEFINITION 2.8: Weighted oriented hypergraph.

An oriented graph is weighted if $H_o = (V, A_H)$ has weights associated to its hyperarcs. This means that each hyperarc $a \in A_H$ carries a value, and these weights are non-negative, so:

$$w : A_H \longrightarrow \mathbb{R}^+, \quad w_a \geq 0.$$

The full expression of an oriented weighted hypergraph is given by:

$$H_o = (V, A_H, w). \tag{2.10}$$

DEFINITION 2.9: Degree function in an unoriented hypergraph.

The degree function $\deg(v_i) : V \to \mathbb{R}$ of a vertex $v_i \in V$ in an unoriented hypergraph $H_u = (V, E_H, w)$ is given as:

$$\deg(v_i) = \sum_{e \in \hat{E}_H(v_i)} w_e. \tag{2.11}$$

For an isolated vertex v_i we have $\deg(v_i) = 0$.

DEFINITION 2.10: Degree function in an oriented hypergraph.

The degree functions $\deg(v)^{in} : V \to \mathbb{R}$ and $\deg(v)^{out} : V \to \mathbb{R}$ of a vertex $v_i \in V$ in an oriented hypergraph $H_o = (V, A_H, w)$ can be expressed as:

$$\deg(v_i)^{in} = \sum_{a \in \hat{A}_H(v_i):\ v_i \in a^{in}} w_a, \tag{2.12}$$

$$\deg(v_i)^{out} = \sum_{a \in \hat{A}_H(v_i):\ v_i \in a^{out}} w_a, \tag{2.13}$$

where $\deg(v_i)^{in}$ is defined as the sum of the weights of all hyperarcs which v_i is an input vertex of. And $\deg(v_i)^{out}$ is also the sum of the weights, but of all hyperarcs for which v_i is an output vertex.
The total degree function is defined as the sum of the in-degree and out-degree of a vertex v_i:

$$\deg(v_i) = \deg(v_i)^{in} + \deg(v_i)^{out}. \tag{2.14}$$

DEFINITION 2.11: Degree and hyperedge cardinality matrices.

The vertex degree matrix $D(H_u)$ for an unoriented weighted hypergraph $H_u = (V, E_H, w)$ is a diagonal $|V| \times |V|$ matrix which contains the degrees of all vertices and is defined as:

$$D(H_u) = \operatorname{diag}(\deg(v_1), ..., \deg(v_n)).$$

The hyperedge cardinality matrix $D(E_H)$ for an unoriented weighted hypergraph $H_u = (V, E_H, w)$ is a diagonal $|E_H| \times |E_H|$ matrix which contains the cardinality of all hyperedges and is defined as:

$$D(E_H) = \operatorname{diag}(|e_1|, ..., |e_k|).$$

DEFINITION 2.12: Degree and hyperarc cardinality matrices.

Similar to the unoriented case, the vertex degree matrix $D(H_o)$ for an oriented weighted hypergraph $H_o = (V, A_H, w)$ is a diagonal square matrix $|V| \times |V|$ which contains the the degrees of all vertices and is defined as:

$$D(H_o) = \mathrm{diag}(\deg(v_1), ..., \deg(v_n)).$$

The hyperarc cardinality matrix $D(A_H)$ for an oriented weighted hypergraph $H_o = (V, A_H, w)$ is given by two diagonal $|A_H| \times |A_H|$ matrices $D^{in}(A_H)$ and $D^{out}(A_H)$ which encode the number of output and input vertices of each hyperarc respectively and therefore they are defined as:

$$D^{in}(A_H) = \mathrm{diag}(|a_1^{in}|, ..., |a_k^{in}|),$$
$$D^{out}(A_H) = \mathrm{diag}(|a_1^{out}|, ..., |a_k^{out}|),$$
$$D(A_H) = D^{in}(A_H) + D^{out}(A_H).$$

Hence, the overall hyperarc cardinality matrix $D(A_H)$ encodes the total number of vertices for each hyperarc $a_q \in A_H$.

DEFINITION 2.13: Weight matrices for hypergraphs.

The weight matrix $W(H_u)$ of an unoriented weighted hypergraph $H_u = (V, E_H, w)$ is a diagonal $|E_H| \times |E_H|$ matrix which contains the weights of all hyperedges and is defined as:

$$W(H_u) = \mathrm{diag}(w_{e_1}, ..., w_{e_k}).$$

The weight matrix $W(H_o)$ of an oriented weighted hypergraph $H_o = (V, A_H, w)$ is a diagonal $|A_H| \times |A_H|$ matrix which contains the weights of all hyperarcs and is defined as:

$$W(H_o) = \mathrm{diag}(w_{a_1}, ..., w_{a_q}).$$

DEFINITION 2.14: Rank.

The rank $R(H_u)$ for an unoriented hypergraph $H_u = (V, E_H)$ is defined as the maximum cardinality of a hyperedge $e_q \in E_H$ in the hypergraph:

$$R(H_u) = \max_{e_q \in E_H} |e_q|. \tag{2.15}$$

Similarly, $R(H_o)$ for an oriented hypergraph $H_o = (V, A_H)$ is the maximum cardinality of a hyperarc $a_q \in A_H$:

$$R(H_o) = \max_{a_q \in A_H} |a_q|. \tag{2.16}$$

If for all hyperedges $e_q \in E_H$ of an unoriented hypergraph it holds true that $|e_q| = 2$, so in particular $R(H_u) = 2$, then we obtain a standard unoriented graph as a special case of an unoriented hypergraph. Similarly, if for all hyperarcs $a_q \in A_H$ it holds true that $|a_q| = 2$, especially $R(H_o) = 2$, then we can see that a standard oriented graph is a special case of an oriented hypergraph.

Example 3. Unoriented hypergraph:
The unoriented hypergraph $H_u = (V, E_H)$ below has 6 vertices, 3 hyperedges and 1 isolated vertex v_1. The rank or the hypergraph is $R(H_u) = 4$ and for example, the degree of vertex v_3 is $\deg(v_3) = 2$. The set of vertices is:

$$V = \Big\{v_1, v_2, v_3, v_4, v_5, v_6\Big\}.$$

The set of hyperedges is:

$$E_H = \Big\{e_1, e_2, e_3\Big\},$$

where $e_1 = \{v_2, v_3, v_5, v_6\}, e_2 = \{v_3, v_5\}, e_3 = \{v_4, v_6\}$.

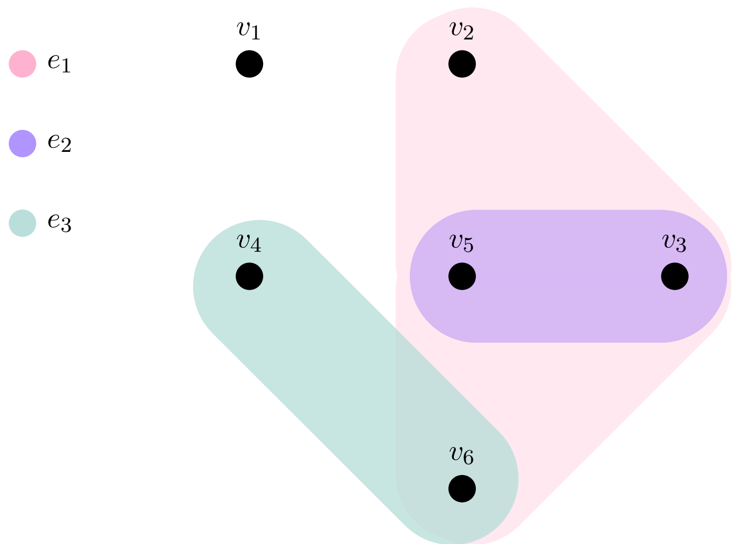


Figure 2.1: Unoriented Hypergraph.

The cardinality of the hyperedge e_3 is 2 and it contains the vertices v_4 and v_6. Vertex v_5 belongs to the hyperedges e_1 and e_2, and therefore the degree of the vertex is 2. ([Bre13], page 21)

$$D(H_u) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \qquad D(E_H) = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

For an unoriented hypergraph, the incidence matrix can be constructed similarly to the incidence matrix for an unoriented graph, but with some modifications to account for hyperedges connecting more than two vertices.

$$K_{H_u} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{ccc} e1 & e2 & e3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$

Example 4. Oriented hypergraph:
The oriented hypergraph $H_o = (V, A_H)$ has 6 vertices, 3 hyperarcs and 1 isolated vertex v_1. The rank of the hypergraph is $R(H_o) = 4$ and for example, the degree of vertex v_6 is $\deg(v_6) = 2$. The set of vertices is:

$$V = \Big\{v_1, v_2, v_3, v_4, v_5, v_6\Big\}.$$

The set of hyperarcs is:

$$A_H = \Big\{a_1, a_2, a_3\Big\},$$

where $a_1 = (\{v_2, v_3\}, \{v_5, v_6\})$, $a_2 = (\{v_5\}, \{v_3\})$, $a_3 = (\{v_4\}, \{v_6\})$.

For an oriented hypergraph in each hyperarc $a_q \in A_H$, the contained vertices can be divided into two sets: the output vertex set a_q^{out} and the input vertex set a_q^{in}. The incidence matrix K_{H_o} can therefore be split into two matrices, $K_{H_o}^{out}$ and $K_{H_o}^{in}$, encoding the output and input vertices of each hyperarc, respectively. ([DG23], page 20)

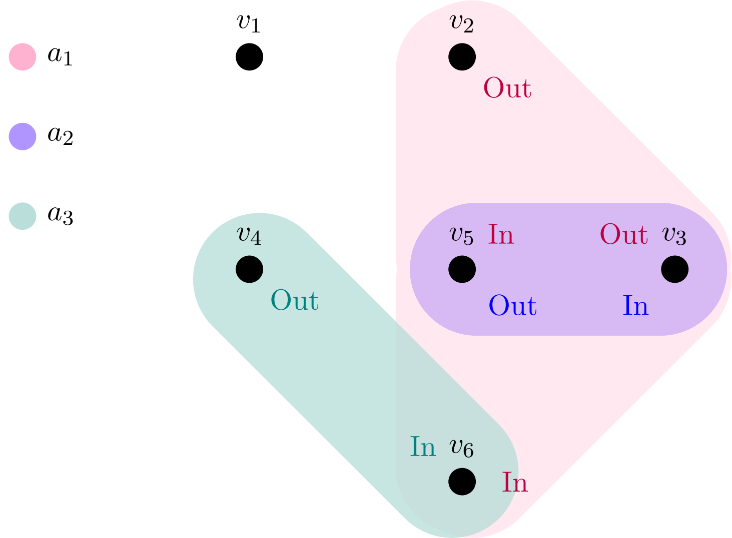


Figure 2.2: Oriented Hypergraph.

$$K_{H_u} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{ccc} e1 & e2 & e3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \quad K^{in}_{H_o} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{ccc} e1 & e2 & e3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \quad K^{out}_{H_o} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{ccc} e1 & e2 & e3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$

2.2 Functions on hypergraphs

DEFINITION 2.15: Vertex function.

Let $f(v_i)$ assign a value to each vertex $v_i \in V$ for an unoriented hypergraph $H_u = (V, E_H)$. Then we define a vertex function as:

$$f : V \to \mathbb{R} \qquad v_i \mapsto f(v_i). \tag{2.17}$$

The definition of a vertex function f for an oriented hypergraph $H_o = (V, A_H)$ is exactly the same as for an unoriented hypergraph $H_u = (V, E_H)$.

DEFINITION 2.16: Vertex weight function.

Let $w(v_i)$ assign a weight to each vertex $v_i \in V$ for an unoriented hypergraph $H_u = (V, E_H)$. Then we define a vertex weight function as:

$$w : V \to \mathbb{R}_{>0} \qquad v_i \mapsto w(v_i). \tag{2.18}$$

The vertex weight function for an oriented hypergraph $H_o = (V, A_H)$ is the same as for an unoriented hypergraph.

DEFINITION 2.17: Space of vertex functions.

We define $S(V)$ as the space for all vertex functions f for an unoriented hypergraph $H_u = (V, E_H)$ or oriented hypergraph $H_o = (V, A_H)$ and it can be written as:

$$S(V) = \{f \mid f : V \to \mathbb{R}\},$$

with the inner product for arbitrarily chosen vertex functions $f_1, f_2 \in S(V)$ and parameter $\gamma \in \mathbb{R}$:

$$\langle f_1, f_2 \rangle_{S(V)} = \sum_{v_i \in V} w(v_i)^\gamma f_1(v_i) f_2(v_i). \tag{2.19}$$

DEFINITION 2.18: L^p-norm on the space of vertex functions.

For an unoriented hypergraph $H_u = (V, E_H)$ or an oriented hypergraph $H_o = (V, A_H)$ the $L^p(V)$-norm on the space of vertex functions is defined as:

$$\| \cdot \|_p : S(V) \to \mathbb{R}^+,$$

$$f \mapsto \|f\|_p = \begin{cases} \|f\|_p & \text{for} \quad 1 \leq p < \infty, \\ \|f\|_\infty & \text{for} \quad p = \infty, \end{cases}$$

where

$$\|f\|_p = \sum_{v_i \in V} (|f(v_i)|^p)^{\frac{1}{p}}, \tag{2.20}$$

$$\|f\|_\infty = \max_{v_i \in V} (|f(v_i)|), \tag{2.21}$$

with the absolute value of vertex functions $f \in S(V)$ as:

$$| \cdot | : \mathbb{R} \to \mathbb{R}^+,$$

$$f(v_i) \mapsto |f(v_i)| = \begin{cases} f(v_i) & \text{for} \quad f(v_i) \geq 0, \\ -f(v_i) & \text{for} \quad f(v_i) < 0. \end{cases} \tag{2.22}$$

DEFINITION 2.19: Hyperedge function F_E and hyperedge weight function W_E.

Let $F_E(e_q)$ assign a real value to each hyperedge $e_q \in E_H$ for an unoriented hypergraph $H_u = (V, E_H)$. Then a hyperedge function is defined such that:

$$F_E : E_H \to \mathbb{R} \qquad e_q \mapsto F_E(e_q).$$

Let $W_E(e_q)$ assign a weight to each hyperedge $e_q \in E_H$ so the hyperedge weight function W_E is given by:

$$W_E : E_H \to \mathbb{R}_{>0} \quad e_q \mapsto W_E(e_q).$$

Since the hyperedges in an unoriented hypergraph have no direction, the order of the vertices of a hyperedge e_q does not matter and hence all hyperedge functions F_E and especially also all hyperedge weight functions W_E are symmetric.

DEFINITION 2.20: Hyperarc function F_A and hyperarc weight function W_A.

Let $F_A(a_q)$ assign a real value to each hyperarc $a_q \in A_H$ for an oriented hypergraph $H_o = (V, A_H)$. Then a hyperarc function is defined such that:

$$F_A : A_H \to \mathbb{R} \qquad a_q \mapsto F_A(a_q).$$

Let $W_A(a_q)$ assign a weight to each hyperarc $a_q \in A_H$ so the hyperarc weight function W_A is given by:

$$W_A : A_H \to \mathbb{R}_{>0} \quad a_q \mapsto W_A(a_q). \tag{2.23}$$

So finally an unoriented hypergraph with vertex and hyperedge weight functions will be written as:

$$H_u = (V, E_G, w, W_E). \tag{2.24}$$

Similarly, an oriented hypergraph with both weight functions will be written as:

$$H_o = (V, A_G, w, W_A). \tag{2.25}$$

DEFINITION 2.21: Space of hyperedge functions $S(E_H)$.

We define $S(E_H)$ as the space for all hyperedge functions F_E for an unoriented hypergraph $H_u = (V, E_H)$, so:

$$S(E_H) = \{F \mid F : E_H \to \mathbb{R}\},$$

with the inner product for arbitrarily chosen hyperedge functions $F_1, F_2 \in S(E_H)$ and parameter $\omega \in \mathbb{R}$:

$$\langle F_1, F_2 \rangle_{S(E_H)} = \sum_{e_q \in E_H} W(e_q)^{\omega} F_1(e_q) F_2(e_q). \tag{2.26}$$

DEFINITION 2.22: Space of hyperarc functions $S(A_H)$.

We define $S(A_H)$ as the space for all hyperarc functions F_A for an oriented hypergraph $H_o = (V, A_H)$, so:

$$S(A_H) = \{F \mid F : A_H \to \mathbb{R}\},$$

with the inner product for arbitrarily chosen hyperarc functions $F_1, F_2 \in S(A_H)$ and parameter $\phi \in \mathbb{R}$:

$$\langle F_1, F_2 \rangle_{S(A_H)} = \sum_{a_q \in A_H} W(a_q)^{\phi} F_1(a_q) F_2(a_q). \tag{2.27}$$

DEFINITION 2.23: L^p-norm on the space of hyperedge functions.

For an unoriented hypergraph $H_u = (V, E_H)$ the $L^p(E_H)$-norm is defined as:

$$\| \cdot \|_p : S(E_H) \to \mathbb{R}^+,$$
$$F_E \mapsto \|F_E\|_p = \begin{cases} \|F_E\|_p & \text{for} \quad 1 \le p < \infty, \\ \|F_E\|_\infty & \text{for} \quad p = \infty, \end{cases}$$

where:

$$\|F_E\|_p = \sum_{e_q \in E_H} (|F_E(e_q)|^p)^{\frac{1}{p}}, \tag{2.28}$$

$$\|F_E\|_\infty = \max_{e_q \in E_H} (|F_E(e_q)|), \tag{2.29}$$

with the absolute value of hyperdege functions $F_E \in S(E_H)$ as:

$$| \cdot | : \mathbb{R} \to \mathbb{R}^+.$$
$$F(e_q) \mapsto |F(e_q)| = \begin{cases} F(e_q) & \text{for} \quad F(e_q) \ge 0, \\ -F(e_q) & \text{for} \quad F(e_q) < 0. \end{cases} \tag{2.30}$$

DEFINITION 2.24: L^p-norm on the space of hyperarc functions.

For an oriented hypergraph $H_o = (V, A_H)$ the $L^p(A_H)$-norm is defined as:

$$\| \cdot \|_p : S(A_H) \to \mathbb{R}^+,$$
$$F_A \mapsto \|F_A\|_p = \begin{cases} \|F_A\|_p & \text{for} \quad 1 \le p < \infty, \\ \|F_A\|_\infty & \text{for} \quad p = \infty, \end{cases}$$

where:

$$\|F_A\|_p = \sum_{a_q \in A_H} (|F_A(a_q)|^p)^{\frac{1}{p}}, \tag{2.31}$$

$$\|F_A\|_\infty = \max_{a_q \in A_H} (|F_A(a_q)|), \tag{2.32}$$

with the absolute value of hyperarc functions $F_A \in S(A_H)$ as:

$$| \cdot | : \mathbb{R} \to \mathbb{R}^+,$$
$$F(a_q) \mapsto |F(a_q)| = \begin{cases} F(a_q) & \text{for} \quad F(a_q) \geq 0, \\ -F(a_q) & \text{for} \quad F(a_q) < 0. \end{cases} \tag{2.33}$$

2.3 Differential operators on unoriented hypergraphs

2.3.1 First-order differential operators

In line with the preceding discussion for unoriented graphs in Section 1.3.1 we now aim at introducing a definition for a special vertex $v_{\tilde{q}}$ for each hyperedge $e_q \in E_H$ in an unoriented hypergraph as proposed in [FTB23]. This definition is needed for deriving meaningful gradient, adjoint and p-Laplacian operators..

DEFINITION 2.25: Special vertex $v_{\tilde{q}}$.

For an unoriented hypergraph $H_u = (V, E_H, w, W_E)$ we assign to each hyperedge $e_q \in E_H$ a specific vertex $v_{\tilde{q}} \in e_q$. This is expressed as:

$$v_{\tilde{q}} := v_i, \qquad \text{for } v_i \in e_q \in E_H.$$

- **Choose the vertex $v_{\tilde{q}}$**:

  If the choice of $v_{\tilde{q}}$ is unclear, an alternative is to include each hyperedge e_q exactly $|e_q|$ times in the set of hyperedges E_H, where each version of the hyperedge has a different special vertex $v_i \in e_q$.

- **Transforming into an oriented hypergraph**:

  This alternative approach allows the transformation of the unoriented hypergraph into an oriented one. For each hyperedge $e_q \in E_H$, $|e_q|$ hyperarcs are created in the oriented hypergraph. Each hyperarc has one output vertex $v_i = v_{\tilde{q}} \in e_q$ and $|e_q| - 1$ input vertices $v_j \in e_q \backslash \{v_i\}$.

DEFINITION 2.26: Vertex-hyperedge characteristic function ψ.

The vertex-hyperedge characteristic function $\psi : V \times E_H \to \{0, 1\}$ for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$ is defined as:

$$\psi(v_i, e_q) = \begin{cases} 1 & \text{for} \quad v_i \in e_q, \\ 0 & \text{for} \quad v_i \notin e_q. \end{cases} \tag{2.34}$$

DEFINITION 2.27: Special vertex-hyperedge characteristic function $\widetilde{\psi}$.

Let $\widetilde{\psi} : V \times E_H \to \{0, 1\}$ be the vertex-hyperedge characteristic function for an unoriented weighted hypergraph $H_u = (V, E_H, w, W_E)$, which is defined as:

$$\widetilde{\psi}(v_i, e_q) = \begin{cases} 1, & \text{for} \quad v_i = v_{\tilde{q}}, \\ 0, & \text{for} \quad v_i \neq v_{\tilde{q}}. \end{cases} \tag{2.35}$$

This function indicates if vertex $v_i \in V$ is the special vertex $v_{\tilde{q}}$ of hyperedge $e_q \in E_H$. Moreover, the next equality holds true $\forall v_i \in V$ and $\forall e_q \in E_H$.

$$\widetilde{\psi}(v_i, e_q) = 1 \quad \Rightarrow \quad \psi(v_i, e_q) = 1. \tag{2.36}$$

Analogously to the case with graphs, future definitions of first-order differential operators for the unoriented hypergraph $H_u = (V, E_H, w, W_E)$ will contain the following parameters:

- w_I and W_I represent vertex and hyperedge weight functions, respectively, originating from the inner product of the space of vertex or hyperedge functions.
- w_G and W_G are vertex and hyperedge weight functions, respectively, introduced in the gradient definitions of this subsection.
- parameters $\alpha, \beta, \zeta, \gamma, \epsilon, \eta, \theta \in \mathbb{R}$ that can be chosen additionally to adapt to different cases.

The presented vertex gradient, vertex adjoint and vertex p-Laplacian operators for unoriented hypergraphs were introduced in [FTB23], however as in the case of unoriented standard graphs, the paper does not include a definition for an hyperedge gradient, adjoint or p-Laplacian operator for unoriented hypergraphs. Hence, in this thesis we utilize the concept of assigning a special vertex to each hyperedge as proposed in [FTB23], but not only for operators for vertex functions, but also for hyperedge functions.

DEFINITION 2.28: Vertex gradient operator ∇_v^H.

Let $\nabla_v^H : S(V) \to S(E_H)$ be the vertex gradient operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$, which is given by:

$$\nabla_v^H f : E_H \to \mathbb{R}, \quad e_q \mapsto \nabla_v^H f(e_q),$$

$$\nabla_v^H f(e_q) =$$
$$W_G(e_q)^\gamma \left[\sum_{v_i \in V} \psi(v_i, e_q) \Big(w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\epsilon f(v_{\tilde{q}}) \Big) \right]$$
$$= W_G(e_q)^\gamma \left[\Big(\sum_{v_i \in V} \psi(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \Big) - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \Big) \right]. \tag{2.37}$$

This rewriting simplifies the expression by consolidating terms, providing a more concise representation of the gradient.
Moreover, the above gradient $\nabla_v^H f(e_q)$ is equal to zero for all constant $f \in S(v)$ if the equation:

$$w_I(v_i)^\alpha w_G(v_i)^\epsilon = w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta$$

holds true for all hyperedges $e_q \in E_H$ and vertices $v_i \in V$.
If all weight functions W_G, w_I and w_G are constant one, then the gradient (2.37) simplifies to:

$$\nabla_v^H f(e_q) = \Big(\sum_{v_i \in V} \psi(v_i, e_q) f(v_i) \Big) - |e_q| f(v_{\tilde{q}}). \tag{2.38}$$

DEFINITION 2.29: Hyperedge gradient operator ∇_e^H.

Let $\nabla_e^H : S(E_H) \to S(V)$ be the hyperedge gradient operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$, which is defined as:

$$\nabla_e^H F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_e^H F(v_i),$$

$$\nabla_e^H F(v_i) = w_G(v_i)^\zeta \sum_{e_q \in E_H} \Big(\psi(v_i, e_q) - \widetilde{\psi}(v_i, e_q)|e_q| \Big) W_I(e_q)^\beta W_G(e_q)^\theta F(e_q). \tag{2.39}$$

With the simplifying assumptions $W_G = 1, W_I = 1$ and $w_G = 1$, the formula (2.39) becomes:

$$\nabla_e^H F(v_i) = \sum_{e_q \in E_H} \Big(\psi(v_i, e_q) - \widetilde{\psi}(v_i, e_q)|e_q| \Big) F(e_q). \tag{2.40}$$

THEOREM 2.30: Generalization of the edge gradient operator.

The hyperedge gradient operator ∇_e^H is a generalisation of the edge gradient operator ∇_e^G. That is, for a given weighted unoriented graph $G_u = (V, E_G, w, W)$ and an edge function $F \in S(E_G)$, the hyperedge gradient operator ∇_e^H for the weighted oriented hypergraph $H_u = (V, E_H, w, W)$ corresponds to the edge gradient operator ∇_e^G for the graph from definition 1.27.

Proof. Given a weighted unoriented standard graph $G_u = (V, E_G, w, W)$ together with an edge function $F \in S(E_G)$, then applying the hyperedge gradient operator ∇_e^H for weighted unoriented hypergraphs to any vertex $v_i \in V$ leads to the definition of the edge gradient operator:

$$\nabla_e^G F(v_i) = w_G(v_i)^\zeta \sum_{e_q \in E_G} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \right] F(e_q) W_I(e_q)^\beta W_G(e_q)^\theta.$$

Therefore, the hyperedge gradient operator for an unoriented hypergraph is a valid generalisation of the edge gradient operator for an unoriented standard graph. □

The definitions below for an unoriented hypergraph are given by default for the case where the weights W_G, W_I, w_G, w_I are equal to 1, unless stated otherwise.

DEFINITION 2.31: Vertex adjoint operator ∇_v^*.

Let $\nabla_v^* : S(E_H) \to S(V)$ be the vertex adjoint operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$ which is defined as:

$$\nabla_v^* F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_v^* F(v_i),$$

$$\nabla_v^* F(v_i) = \sum_{e_q \in E_H} \left(\psi(v_i, e_q) - \widetilde{\psi}(v_i, e_q) |e_q| \right) F(e_q). \tag{2.41}$$

DEFINITION 2.32: Hyperedge adjoint operator ∇_e^*.

Let $\nabla_e^* : S(V) \to S(E_H)$ be the hyperedge adjoint operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$ which is defined as:

$$\nabla_e^* f : E_H \to \mathbb{R}, \quad e_q \mapsto \nabla_e^* f(e_q),$$

$$\nabla_e^* f(e_q) = \sum_{v_i \in V} \left(\psi(v_i, e_q) - \widetilde{\psi}(v_i, e_q) |e_q| \right) f(v_i). \tag{2.42}$$

The following connection between gradient and adjoint operators holds true $\forall f \in S(V)$ and $\forall G \in S(E_H)$ in an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$:

$$\langle G, \nabla_v^H f\rangle_{S(E_H)} = \langle f, \nabla_v^* G\rangle_{S(V)}, \tag{2.43}$$

$$\langle f, \nabla_e^H G\rangle_{S(V)} = \langle G, \nabla_e^* f\rangle_{S(E_H)}. \tag{2.44}$$

Proof. For an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$ with a vertex function $f \in S(V)$ and a hyperedge function $G \in S(E_H)$, we can prove the connection between the hyperedge gradient ∇_e^H and the hyperedge adjoint ∇_e^* in (2.44). For this proof we will use the definitions of the inner product in $S(V)$ and the hyperedge gradient operator ∇_e^H with constant (equal to 1) weight functions.

$$\begin{aligned}
\langle f, \nabla_e^G G\rangle_{S(V)} &= \sum_{v_i \in V} f(v_i) \nabla_e^H(v_i) G(e_q) \\
&= \sum_{v_i \in V} f(v_i) \sum_{e_q \in E_H} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q)\right] G(e_q) \\
&= \sum_{v_i \in V} \sum_{e_q \in E_H} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q)\right] f(v_i) G(e_q)
\end{aligned}$$

By exchanging the two sums, the following is obtained:

$$\begin{aligned}
&= \sum_{e_q \in E_H} \sum_{v_i \in V} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q)\right] f(v_i) G(e_q) \\
&= \sum_{e_q \in E_H} G(e_q) \sum_{v_i \in V} \left[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q)\right] f(v_i)
\end{aligned}$$

And by using the definitions for the inner product on the space of all hyperedge functions $S(E_H)$ and the hyperedge adjoint operator ∇_e^* we have:

$$= \sum_{e_q \in E_H} G(e_q) \nabla_e^* f = \langle G, \nabla_e^* f\rangle_{S(E_H)}.$$

□

Now we define the vertex divergence operator div_v^H and the hyperedge divergence operator div_e^H based on the vertex and hyperedge adjoint operators.

DEFINITION 2.33: Vertex divergence operator div_v^H.

Let $\mathrm{div}_v^H : S(E_H) \to S(V)$ be the vertex divergence operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$. Based on the equivalence from the continuum setting we have:

$$\mathrm{div}_v^H = -\nabla_v^* \qquad \Rightarrow \qquad \mathrm{div}_v^H F(v_i) = -\nabla_v^* F(v_i).$$

Therefore, the weighted divergence of a hyperedge function $F \in S(E_H)$ at a vertex $v_i \in V$, given that $W_G = 1, w_G = 1$ and $W_I = 1$, is equal to:

$$\mathrm{div}_v^H F(v_i) = \sum_{e_q \in E_H} \Big[\widetilde{\psi}(v_i, e_q)|e_q| - \psi(v_i, e_q) \Big] F(e_q). \tag{2.45}$$

DEFINITION 2.34: Hyperedge divergence operator div_e^H.

Let $\mathrm{div}_e^H : S(V) \to S(E_H)$ be the hyperedge divergence operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$. Based on the equivalence from the continuum setting we have:

$$\mathrm{div}_e^H = -\nabla_e^* \qquad \Rightarrow \qquad \mathrm{div}_e^H f(e_q) = -\nabla_e^* f(e_q).$$

Therefore, the weighted divergence of a vertex function $f \in S(V)$ at an edge $e_q \in E_H$, given that $W_G = 1, w_G = 1$ and $w_I = 1$, is equal to:

$$\mathrm{div}_e^H f(e_q) = \sum_{v_i \in V} \Big[\widetilde{\psi}(v_i, e_q)|e_q| - \psi(v_i, e_q) \Big] f(v_i). \tag{2.46}$$

2.3.2 p-Laplacian operators

DEFINITION 2.35: Vertex Laplacian operator Δ_v.

Let $\Delta_v : S(V) \to S(V)$ be the vertex Laplacian operator for an unoriented weighted hypergraph $H_u = (V, E_H, w)$. Derived from the connection in the continuum setting, we obtain:

$$\Delta_v f = \mathrm{div}_v^H(\nabla_v^H f).$$

so the weighted Laplacian of vertex function $f \in S(V)$ at a vertex $v_i \in V$ when W_G, W_I, w_G and w_I are equal to 1, is given by:

$$\Delta_v f(v_i) = \sum_{e_q \in E_H} \Big[\psi(v_i, e_q) - |e_q| \widetilde{\psi}(v_i, e_q) \Big] \sum_{v_j \in V} \Big[\psi(v_j, e_q) f(v_j) - |e_q| f(v_{\tilde{q}}) \Big]. \tag{2.47}$$

DEFINITION 2.36: Hyperedge Laplacian operator Δ_e.

Let $\Delta_e : S(E_H) \to S(E_H)$ be the hyperedge Laplacian operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$. Derived from the connection in the continuum setting, we obtain:

$$\Delta_e F = \mathrm{div}_e^H(\nabla_e^H F), \tag{2.48}$$

so the weighted Laplacian of hyperedge function $F \in S(E_H)$ at a hyperedge $e_q \in E_H$ when W_G, W_I, w_G and w_I are equal to 1, is given by:

$$\begin{aligned}\Delta_e F(e_q) = \sum_{v_i \in V} \Big[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \Big] \\ \sum_{e_r \in E_H} \Big[\psi(v_i, e_r) - |e_r| \widetilde{\psi}(v_i, e_r) \Big] F(e_r).\end{aligned} \tag{2.49}$$

Proof. For an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$ with a hyperedge function $F \in S(E_H)$, we can prove the connection between the hyperedge Laplacian operator Δ_e and the hyperedge divergence operator div_e^H in (2.48). For this proof we will use the definitions of the hyperedge divergence operator div_e^H and the hyperedge gradient operator ∇_e^H with constant (equal to 1) weight functions.

$$\begin{aligned}\mathrm{div}_e^H(\nabla_e^H F)(v_i) &= \sum_{v_i \in V} \Big[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \Big] \nabla_e^H F(v_i) \\ &= \sum_{v_i \in V} \Big[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \Big] \sum_{e_r \in E_H} \Big[\psi(v_i, e_r) - |e_r| \widetilde{\psi}(v_i, e_r) \Big] F(e_r) \\ &= \Delta_e F(e_r).\end{aligned}$$

□

DEFINITION 2.37: Vertex p-Laplacian operator Δ_v^p.

Let $\Delta_v^p : S(V) \to S(V)$ be the p-Laplacian operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$. Derived from the connection in the continuum setting, we obtain:

$$\Delta_v^p f = \mathrm{div}_v^H(|\nabla_v^H f|^{p-2}\nabla_v^H f),$$

Hence, the weighted p-Laplacian of vertex function $f \in S(V)$ at a vertex $v_i \in V$, assuming that W_G, W_I, w_G and w_I are equal to 1, is given by:

$$\begin{aligned}\Delta_v^p f(v_i) = \sum_{e_q \in E_H} &\Big(\psi(v_i, e_q) - |e_q|\widetilde{\psi}(v_i, e_q)\Big)\\ &\left|\sum_{v_j \in V} \psi(v_j, e_q) f(v_j) - |e_q| f(v_{\tilde{q}})\right|^{p-2}\\ &\left[\sum_{v_k \in V} \psi(v_k, e_q) f(v_k) - |e_q| f(v_{\tilde{q}})\right].\end{aligned} \tag{2.50}$$

The full formula of the vertex p-Laplacian operator including the weight functions would look like this:

$$\begin{aligned}\Delta_v^p f(v_i) = \sum_{e_q \in E_H} &\Big[\psi(v_i, e_q) w_G(v_i)^\epsilon - |e_q|\widetilde{\psi}(v_i, e_q) w_G(v_i)^\eta\Big] W_I(e_q)^\beta W_G(e_q)^{p\gamma}\\ &\left|\sum_{v_j \in V} \psi(v_j, e_q) f(v_j) w_I(v_j)^\alpha w_G(v_j)^\epsilon - |e_q| f(v_{\tilde{q}}) w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta\right|^{p-2}\\ &\left[\sum_{v_k \in V} \psi(v_k, e_q) f(v_k) w_I(v_k)^\alpha w_G(v_k)^\epsilon - |e_q| f(v_{\tilde{q}}) w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta\right].\end{aligned} \tag{2.51}$$

DEFINITION 2.38: Hyperedge p-Laplacian operator Δ_e^p.

Let $\Delta_e^p : S(E_H) \to S(E_H)$ be the hyperedge p-Laplacian operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$. Derived from the connection in the continuum setting, we obtain:

$$\Delta_e^p F = \mathrm{div}_e^H(|\nabla_e^H F|^{p-2}\nabla_e^H F). \tag{2.52}$$

Hence, the weighted p-Laplacian of a hyperedge function $F \in S(E_H)$ at a hyperedge $e_q \in E_H$, assuming that W_G, W_I, w_G and w_I are equal to 1, is given by:

$$
\begin{aligned}
\Delta_e^p F(e_q) = \sum_{v_i \in V} & \Big[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \Big] \\
& \Big| \sum_{e_r \in E_H} \Big[\psi(v_i, e_r) - |e_r| \widetilde{\psi}(v_i, e_r) \Big] F(e_r) \Big|^{p-2} \\
& \sum_{e_s \in E_H} \Big[\psi(v_i, e_s) - |e_s| \widetilde{\psi}(v_i, e_s) \Big] F(e_s).
\end{aligned} \tag{2.53}
$$

The full formula of the hyperedge p-Laplacian operator including the weight functions would look like this:

$$
\begin{aligned}
\Delta_e^p F(e_q) = W_G(e_q)^\theta \sum_{v_i \in V} & \Bigg(\Big[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \Big] w_I(v_i)^\alpha w_G(v_i)^{p\zeta} \\
& \Big| \sum_{e_r \in E_H} \Big[\psi(v_i, e_r) - |e_r| \widetilde{\psi}(v_i, e_r) \Big] F(e_r) W_I(e_r)^\beta W_G(e_r)^\theta \Big|^{p-2} \\
& \sum_{e_s \in E_H} \Big[\psi(v_i, e_s) - |e_s| \widetilde{\psi}(v_i, e_s) \Big] F(e_s) W_I(e_s)^\beta W_G(e_s)^\theta \Bigg).
\end{aligned} \tag{2.54}
$$

Proof. For an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$ with a hyperedge function $F \in S(E_H)$, we can prove the connection between the hyperedge p-Laplacian operator Δ_e^p and the hyperedge divergence operator div_e^H in (2.52). For this proof we will use the definitions of the hyperedge divergence operator div_e^H and the hyperedge gradient operator ∇_e^H with constant (equal to 1) weight functions.

$$
\begin{aligned}
\mathrm{div}_e^H(|\nabla_e^H F|^{p-2} \nabla_e^H F) &= \sum_{v_i \in V} \Big[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \Big] (|\nabla_e^H F|^{p-2} \nabla_e^H F) \\
&= \sum_{v_i \in V} \Big[|e_q| \widetilde{\psi}(v_i, e_q) - \psi(v_i, e_q) \Big] \Big| \sum_{e_r \in E_H} \Big[\psi(v_i, e_r) - |e_r| \widetilde{\psi}(v_i, e_r) \Big] F(e_r) \Big|^{p-2} \\
&\quad \sum_{e_s \in E_H} \Big[\psi(v_i, e_s) - |e_s| \widetilde{\psi}(v_i, e_s) \Big] F(e_s) = \Delta_e^p F(e_s).
\end{aligned}
$$

□

2.4 Differential operators on oriented hypergraphs

2.4.1 First-order differential operator

DEFINITION 2.39: Vertex-hyperarc characteristic functions $\widetilde{\psi}$.

Let $\psi_{out} : V \times A_H \to \mathbb{R}$ be the out-degree function and $\psi_{in} : V \times A_H \to \mathbb{R}$ the in-degree function for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$, which are defined as:

$$\psi_{out}(v_i, a_q) = \begin{cases} 1 & \text{for} \quad v_i \in a_q^{out} \text{ or } v_i \in a_q^{in}, \\ 0 & \text{for} \quad otherwise. \end{cases} \tag{2.55}$$

$$\psi_{in}(v_i, a_q) = \begin{cases} 1 & \text{for} \quad v_i \in a_q^{out} \text{ or } v_i \in a_q^{in}, \\ 0 & \text{for} \quad otherwise. \end{cases} \tag{2.56}$$

In an oriented hypergraph, the in-degree of a vertex corresponds to the number of hyperarcs directed towards that vertex, while the out-degree represents the number of hyperarcs originating at the given vertex.

DEFINITION 2.40: Combined vertex-hyperarc characteristic function.

Let $\psi : V \times A_H \to \mathbb{R}$ be the combined vertex-hyperarc characteristic function for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$, which is defined as:

$$\psi(v_i, a_q) = \begin{cases} 1 & \text{for} \quad v_i \in a_q^{out}, \\ -1 & \text{for} \quad v_i \in a_q^{in}, \\ 0 & \text{for} \quad \text{otherwise}. \end{cases} \tag{2.57}$$

Similar to the case of unoriented hypergraphs, definitions of first-order differential operators for an oriented hypergraph $H_o = (V, E_H, w, W)$ will contain the following parameters:

- w_I and W_I represent vertex and hyperarc weight functions, respectively, originating from the inner product of the space of vertex or hyperarc functions.
- w_G and W_G are vertex and hyperarc weight functions, respectively, introduced in the gradient definitions below.
- parameters $\alpha, \beta, \zeta, \gamma, \epsilon, \eta, \theta \in \mathbb{R}$ that can be chosen additionally to adapt to different cases.

The following vertex and hyperarc gradient, adjoint and p-Laplacian operators were introduced in [Faz23].

DEFINITION 2.41: Vertex gradient operator ∇_v^H.

Let $\nabla_v^H : S(V) \to S(A_H)$ be the vertex gradient operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$, which is given as:

$$\nabla_v^H f : A_H \to \mathbb{R}, \quad a_q \mapsto \nabla_v^H f(a_q)$$

$$\begin{aligned} \nabla_v^H f(a_q) = W_G(a_q)^\gamma \sum_{v_i \in V} \Bigg[& \psi_{in}(v_i, a_q) \frac{w_I(v_i)^\alpha w_G(v_i)^\epsilon}{|a_q^{in}|} \\ & - \psi_{out}(v_i, a_q) \frac{w_I(v_i)^\alpha w_G(v_i)^\eta}{|a_q^{out}|} \Bigg] f(v_i). \end{aligned} \tag{2.58}$$

Moreover, the above gradient (2.58) is zero for all constant $f \in S(V)$, if for all $a_q \in A_H$ and all $v_k \in a_q^{in}$ and $v_l \in a_q^{out}$ it holds true that:

$$w_I(v_k)^\alpha w_G(v_k)^\epsilon = w_I(v_l)^\alpha w_G(v_l)^\eta.$$

We simplify the the vertex gradient by setting the weights $W_G = 1, w_I = 1, w_G = 1$, so that (2.58) results in:

$$\nabla_v^H f(a_q) = \sum_{v_i \in V} \left[\frac{\psi_{in}(v_i, a_q)}{|a_q^{in}|} - \frac{\psi_{out}(v_i, a_q)}{|a_q^{out}|} \right] f(v_i). \tag{2.59}$$

DEFINITION 2.42: Hyperarc gradient operator ∇_a^H.

Let $\nabla_a^H : S(A_H) \to S(V)$ be the hyperarc gradient operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$, which is given as:

$$\nabla_a^H F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_a^H F(v_i),$$

$$\nabla_a^H F(v_i) = w_G(v_i)^\zeta \sum_{a_q \in A_H} \left[\frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} \right] W_I(a_q)^\beta W_G(a_q)^\theta F(a_q). \tag{2.60}$$

Similar to the definition above, we simplify the formula by setting the weights $W_G = 1, w_I = 1, w_G = 1$, so that the previous formula (2.60) simplifies to:

$$\nabla_a^H F(v_i) = \sum_{a_q \in A_H} \left[\frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} \right] F(a_q). \tag{2.61}$$

The following definitions for an oriented hypergraph are defined under the assumption that the vertex and hyperedge weights w_G, w_I, W_I, W_G are equal to one, unless stated otherwise.

DEFINITION 2.43: Vertex adjoint operator ∇_v^*.

Let $\nabla_v^* : S(A_H) \to S(V)$ be the vertex adjoint operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$ which is defined as:

$$\nabla_v^* F : V \to \mathbb{R}, \quad v_i \mapsto \nabla_v^* F(v_i),$$

$$\nabla_v^* F(v_i) = \sum_{a_q \in A_H} \left[\frac{\psi_{in}(v_i, a_q)}{|a_q^{in}|} - \frac{\psi_{out}(v_i, a_q)}{|a_q^{out}|} \right] F(a_q). \tag{2.62}$$

DEFINITION 2.44: Hyperarc adjoint operator ∇_a^*.

Let $\nabla_a^* : S(V) \to S(A_H)$ be the hyperarc adjoint operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$ which is defined as:

$$\nabla_a^* f : A_H \to \mathbb{R}, \quad a_q \mapsto \nabla_a^* f(a_q),$$

$$\nabla_a^* f(a_q) = \sum_{v_i \in V} \left[\frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} \right] f(v_i). \tag{2.63}$$

The connection between gradient and adjoint operators holds true $\forall f \in S(V)$ and $\forall G \in S(A_H)$ in an oriented weighted hypergraph $H_o = (V, A_H, w, W)$:

$$\langle G, \nabla_v^H f \rangle_{S(A_H)} = \langle f, \nabla_v^* G \rangle_{S(V)}, \tag{2.64}$$

$$\langle f, \nabla_a^H G \rangle_{S(V)} = \langle G, \nabla_a^* f \rangle_{S(A_H)}. \tag{2.65}$$

Now we define the vertex divergence operator div_v^H and the hyperarc divergence operator div_a^H based on the vertex and hyperarc adjoint operators.

DEFINITION 2.45: Vertex divergence operator div_v^H.

Let $\mathrm{div}_v^H : S(A_H) \to S(V)$ be the vertex divergence operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$. Based on the definition in the continuum setting we have:

$$\mathrm{div}_v^H = -\nabla_v^* \qquad \Rightarrow \qquad \mathrm{div}_v^H F(v_i) = -\nabla_v^* F(v_i).$$

The weighted vertex divergence of hyperarc function $F \in S(A_H)$ at an vertex $v_i \in V$ is hence defined as:

$$\operatorname{div}_v^H F(v_i) = \sum_{a_q \in A_H} \left[\psi_{out}(v_i, a_q) \frac{1}{|a_q^{out}|} - \psi_{in}(v_i, a_q) \frac{1}{|a_q^{in}|} \right] F(a_q). \tag{2.66}$$

DEFINITION 2.46: Hyperarc divergence operator div_a^H.

Let $\operatorname{div}_a^H : S(V) \to S(A_H)$ be the hyperarc divergence operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$. Based on the definition in the continuum setting we have:

$$\operatorname{div}_a^H = -\nabla_a^* \qquad \Rightarrow \qquad \operatorname{div}_a^H f(a_q) = -\nabla_a^* f(a_q).$$

The weighted divergence of vertex function $f \in S(V)$ at a hyperarc $a_q \in A_H$ is defined as:

$$\operatorname{div}_a^H f(a_q) = \sum_{v_i \in V} \left[\frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} - \frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} \right] f(v_i). \tag{2.67}$$

2.4.2 p-Laplacian operators

DEFINITION 2.47: Vertex Laplacian operator Δ_v.

Let $\Delta_v : S(V) \to S(V)$ be the vertex Laplacian operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$. Derived from the connection in the continuum setting, we obtain:

$$\Delta_v f = \operatorname{div}_v^H (\nabla_v^H f).$$

Finally, the weighted Laplacian of vertex function $f \in S(V)$ at a vertex $v_i \in V$ is equal to:

$$\begin{aligned} \Delta_v f(v_i) = \sum_{a_q \in A_H} & \left[\frac{\psi_{out}(v_i, a_q)}{|a_q^{out}|} - \frac{\psi_{in}(v_i, a_q)}{|a_q^{in}|} \right] \\ & \sum_{v_j \in V} \left[\frac{\psi_{in}(v_j, a_q)}{|a_q^{in}|} - \frac{\psi_{out}(v_j, a_q)}{|a_q^{out}|} \right] f(v_j). \end{aligned} \tag{2.68}$$

DEFINITION 2.48: Hyperarc Laplacian operator Δ_a.

Let $\Delta_a : S(A_H) \to S(A_H)$ be the hyperarc Laplacian operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$. Derived from the connection in the continuum setting, we obtain:

$$\Delta_e F = \mathrm{div}_a^H(\nabla_a^H F).$$

So the weighted Laplacian of hyperarc function $F \in S(A_H)$ at a hyperarc $a_q \in A_H$ is defined as:

$$\begin{aligned}\Delta_a F(a_q) = \sum_{v_i \in V} &\left[\frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} - \frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)}\right] \\ &\sum_{a_r \in A_H}\left[\frac{\psi_{in}(v_i, a_r)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_r)}{\deg_{out}(v_i)}\right] F(a_r).\end{aligned} \tag{2.69}$$

DEFINITION 2.49: Vertex p-Laplacian operator Δ_v^p.

Let $\Delta_v^p : S(V) \to S(V)$ be the vertex p-Laplacian operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$. Based on the definition in the continuum setting, we obtain:

$$\Delta_v^p f = \mathrm{div}_v^H(|\nabla_v^H f|^{p-2} \nabla_v^H f). \tag{2.70}$$

Hence, the weighted p-Laplacian of vertex function $f \in S(V)$ at a vertex $v_i \in V$ is equal to:

$$\begin{aligned}\Delta_v^p f(v_i) = \sum_{a_q \in A_H} &\left[\frac{\psi_{out}(v_i, a_q)}{|a_q^{out}|} - \frac{\psi_{in}(v_i, a_q)}{|a_q^{in}|}\right] \\ &\left|\sum_{v_j \in V}\left[\frac{\psi_{in}(v_j, a_q)}{|a_q^{in}|} - \frac{\psi_{out}(v_j, a_q)}{|a_q^{out}|}\right] f(v_j)\right|^{p-2} \\ &\sum_{v_k \in V}\left[\frac{\psi_{in}(v_k, a_q)}{|a_q^{in}|} - \frac{\psi_{out}(v_k, a_q)}{|a_q^{out}|}\right] f(v_k).\end{aligned} \tag{2.71}$$

The complete formula for the vertex p-Laplacian operator including weights W_I, W_G, w_G and w_I is given by:

$$\Delta_v^p f(v_i) = \sum_{a_q \in A_H} \left[\psi_{out}(v_i, a_q) \frac{w_G(v_i)^\eta}{|a_q^{out}|} - \psi_{in}(v_i, a_q) \frac{w_G(v_i)^\epsilon}{|a_q^{in}|} \right] W_I(a_q)^\beta W_G(a_q)^{p\gamma}$$
$$\left| \sum_{v_j \in V} \left[\psi_{in}(v_j, a_q) \frac{w_I(v_j)^\alpha w_G(v_j)^\epsilon}{|a_q^{in}|} - \psi_{out}(v_j, a_q) \frac{w_I(v_j)^\alpha w_G(v_j)^\eta}{|a_q^{out}|} \right] f(v_j) \right|^{p-2}$$
$$\sum_{v_k \in V} \left[\psi_{in}(v_k, a_q) \frac{w_I(v_k)^\alpha w_G(v_k)^\epsilon}{|a_q^{in}|} - \psi_{out}(v_k, a_q) \frac{w_I(v_k)^\alpha w_G(v_k)^\eta}{|a_q^{out}|} \right] f(v_k). \tag{2.72}$$

DEFINITION 2.50: Hyperarc p-Laplacian operator Δ_a^p.

Let $\Delta_a^p : S(A_H) \to S(A_H)$ be the hyperarc p-Laplacian operator for an oriented weighted hypergraph $H_o = (V, A_H, w, W)$. Derived from the connection in the continuum setting, we obtain:

$$\Delta_a^p F = \operatorname{div}_a^H (|\nabla_a^H F|^{p-2} \nabla_a^H F).$$

And now finally, the weighted hyperarc p-Laplacian operator of hyperarc function $F \in S(A_H)$ at a hyperarc $a_q \in A_H$ is defined as:

$$\Delta_a^p F(a_q) = \sum_{v_i \in V} \left[\frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} - \frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} \right]$$
$$\left| \sum_{a_r \in A_H} \left[\frac{\psi_{in}(v_i, a_r)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_r)}{\deg_{out}(v_i)} \right] F(a_r) \right|^{p-2} \tag{2.73}$$
$$\sum_{a_s \in A_H} \left[\frac{\psi_{in}(v_i, a_s)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_s)}{\deg_{out}(v_i)} \right] F(a_s).$$

The complete definition of the hyperarc p-Laplacian operator including the weights W_I, W_G, w_G and w_I looks like this:

$$\Delta_a^p F(a_q) = W_G(a_q)^\theta \sum_{v_i \in V} \left[\frac{\psi_{out}(v_i, a_q)}{\deg_{out}(v_i)} - \frac{\psi_{in}(v_i, a_q)}{\deg_{in}(v_i)} \right] w_I(v_i)^\alpha w_G(v_i)^{2\zeta}$$
$$\left| \sum_{a_r \in A_H} \left[\frac{\psi_{in}(v_i, a_r)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_r)}{\deg_{out}(v_i)} \right] W_I(a_r)^\beta W_G(a_r)^\theta F(a_r) \right|^{p-2} \tag{2.74}$$
$$\sum_{a_s \in A_H} \left[\frac{\psi_{in}(v_i, a_s)}{\deg_{in}(v_i)} - \frac{\psi_{out}(v_i, a_s)}{\deg_{out}(v_i)} \right] W_I(a_s)^\beta W_G(a_s)^\theta F(a_s).$$

2.5 Connections between graphs and hypergraphs

Definition 2.14 suggests that a standard graph is a special case of a hypergraph for which the cardinality of each hyperedge e_q or hyperarc a_q is two. In other words, when all hyperedges or hyperarcs in a hypergraph contain exactly two vertices, the hypergraph reduces to an unoriented or oriented standard graph. Let's consider a simple example to illustrate the concept.

Example 5. Unoriented hypergraph represented as unoriented graph .
The unoriented standard graph $G_u = (V, E_G)$ and unoriented hypergraph $H_u = (V, E_H)$ have the same set of vertices:

$$V = \{v_1, v_2, v_3, v_4\},$$

and also the same set of edges and hyperedges respectively:

$$E_G = E_H = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_3, v_4\}\}.$$

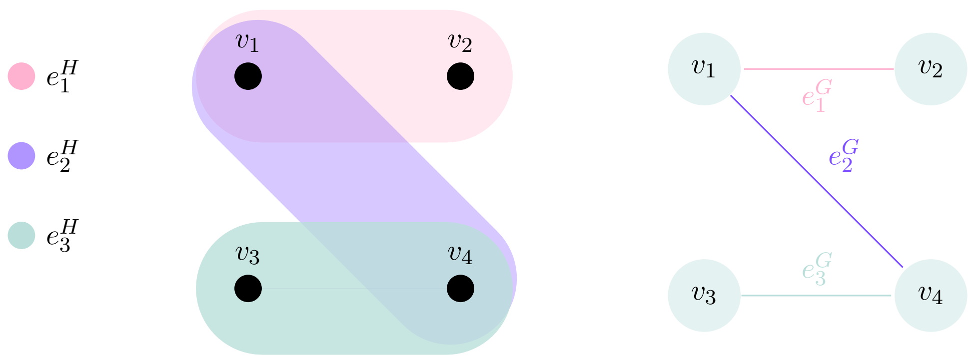


Figure 2.3: Unoriented hypergraph reduced to an unoriented graph .

In this example, each hyperedge contains exactly two vertices, so it can be interpreted as an edge from a standard graph.

DEFINITION 2.51: Bipartite graph G_B.

A bipartite standard graph $G_B = (V_1, V_2, E_G)$ is defined as a triple of two disjoint sets of vertices V_1 and V_2 and a set of edges E_G. [DG23] Within the two vertex sets V_1 and V_2, the vertices are not directly connected through any edge, therefore:

$$\forall e_q = \{v_i, v_j\} \in E_G : \quad v_i \in V_1, v_j \in V_2 \quad \text{or} \quad v_i \in V_2, v_j \in V_1 \tag{2.75}$$

Example 6. Unoriented hypergraph as bipartite graph.

In order to represent an unoriented hypergraph as a bipartite standard graph, we define two new sets of vertices V_1 and V_2 as well as a new set of edges E_G. The unoriented hypergraph $H_u = (V, E_H)$ of the example below consists of the vertex set $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and the hyperedge set $E_H = \{\{v_2, v_3, v_5, v_6\}, \{v_3, v_5\}, \{v_4, v_6\}\} = \{e_1^H, e_2^H, e_3^H\}$.

The two vertex sets of the bipartite graph are then defined in the following way:

$$V_1 = \{v_1, v_2, v_3, v_4, v_5, v_6\} = V$$

$$V_2 = \{v_{e_1}, v_{e_2}, v_{e_3}\} \mathrel{\widehat{=}} E_H$$

And the edge set is given by:

$$\begin{aligned} E_G &= \{\{v_i, v_{e_j}\} \mid v_i \in V, e_j \in E_H, v_i \in e_j\} \\ &= \{e_1^G, e_2^G, e_3^G, e_4^G, e_5^G, e_6^G, e_7^G, e_8^G\} \\ &= \{\{v_2, v_{e_1}\}, \{v_3, v_{e_1}\}, \{v_5, v_{e_1}\}, \{v_6, v_{e_1}\}, \{v_3, v_{e_2}\}, \{v_5, v_{e_2}\}, \{v_4, v_{e_3}\}, \{v_6, v_{e_3}\}\} \end{aligned}$$

Therefore, the newly created bipartite standard graph encodes the information which vertex $v_i \in V$ is part of which hyperedge $e_j \in E_H$.

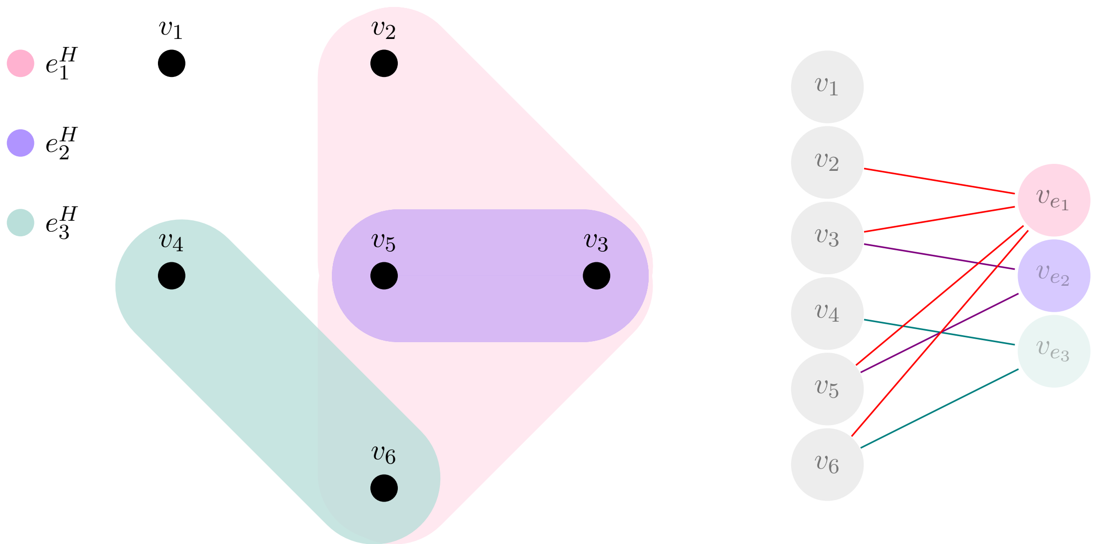


Figure 2.4: Connection between the hypergraph H_u and the bipartite graph G_B.

The incidence matrices for the unoriented hypergraph H_u and the bipartite graph G_B are:

Moreover, an unoriented bipartite graph G_B can also be transformed into an unoriented hypergraph, where one of the two vertex sets is interpreted as the set of vertices V and the other then corresponds to the set of hyperedges E_H. However, this could lead to hyperedges with cardinality 1 if a vertex of the second vertex set is only connected to one vertex of the first vertex set, which would constitute a loop.

$$K_{H_u} = \begin{array}{c|c} & \begin{array}{ccc} e_1^H & e_2^H & e_3^H \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} & \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \qquad K_{G_B} = \begin{array}{c|c} & \begin{array}{cccccccc} e_1^G & e_2^G & e_3^G & e_4^G & e_5^G & e_6^G & e_7^G & e_8^G \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_{e_1} \\ v_{e_2} \\ v_{e_3} \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

Example 7. Oriented hypergraph as bipartite graph.

Not only unoriented hypergraphs, but also oriented hypergraphs can be represented as a bipartite standard graph. As before, we define two new sets of vertices V_1 and V_2 as well as a set of arcs A_G. The given oriented hypergraph $H_o = (V, A_H)$ consists of the vertex set $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and the hyperarc set $A_H = \{(\{v_2, v_3\}, \{v_5, v_6\}), (\{v_5\}, \{v_3\}), (\{v_4\}, \{v_6\})\} = \{a_1^H, a_2^H, a_3^H\}$.

The two vertex sets of the bipartite graph are then defined in the following way:

$$V_1 = \{v_1, v_2, v_3, v_4, v_5, v_6\} = V$$

$$V_2 = \{v_{a_1}, v_{a_2}, v_{a_3}\} \mathrel{\widehat{=}} A_H$$

And the arc set is given by:

$$\begin{aligned} A_G &= \{(v_i, v_{a_j}) \mid v_i \in V, a_j \in A_H, v_i \in a_j^{out} \text{ or } v_i \in a_j^{in}\} \\ &= \{a_1^G, a_2^G, a_3^G, a_4^G, a_5^G, a_6^G, a_7^G, a_8^G\} \\ &= \{(v_2, v_{a_1}), (v_3, v_{a_1}), (v_{a_1}, v_5), (v_{a_1}, v_6), (v_{a_2}, v_3), (v_5, v_{a_2}), (v_4, v_{a_3}), (v_{a_3}, v_6)\} \end{aligned}$$

Therefore, the newly created bipartite standard graph encodes the information which vertex $v_i \in V$ is part of which hyperarc $a_j \in A_H$ and whether it is the an output or an input vertex of a_q.

Therefore, the newly created bipartite standard graph encodes the information which vertex $v_i \in V$ is part of which hyperarc $a_j \in A_H$ either as an output or an input vertex. The incidence matrices for the oriented hypergraph H_o and the bipartite graph G_B are:

Moreover, an oriented bipartite graph G_B can in many cases also be transformed into an oriented hypergraph, where one of the two vertex sets is interpreted as the set of vertices V and the other then corresponds to the set of hyperarcs A_H. However, issues with not well-defined hyperarcs can arise, if there are vertices in the second vertex set which are not both a starting vertex for an arc to the first vertex set and an ending vertex for another arc to the first vertex set. For an oriented hypergraph this would correspond to either the output vertex set or the input vertex set of a hyperarc being empty.

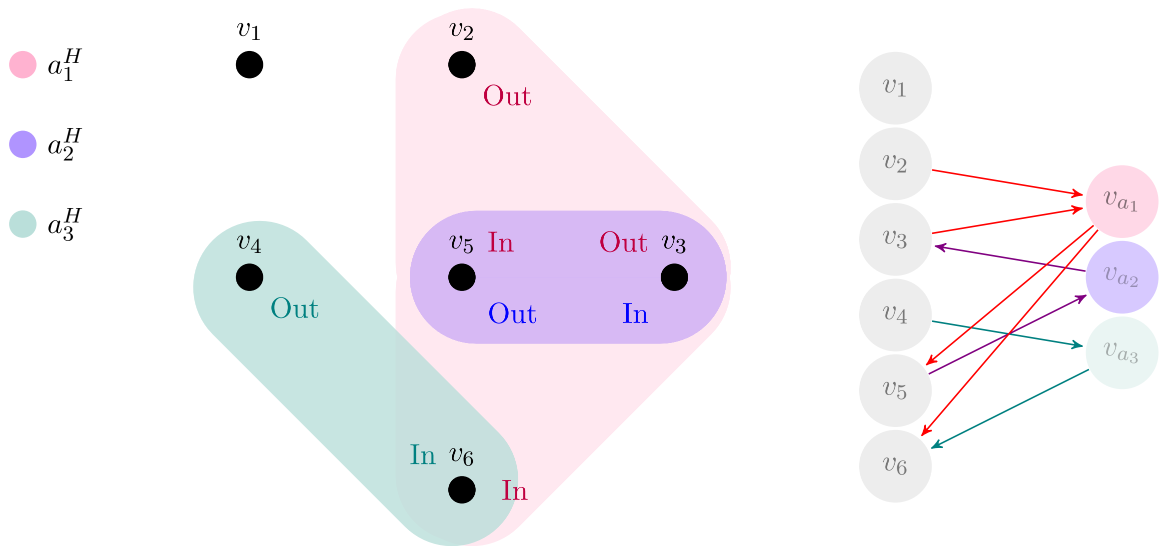


Figure 2.5: Connection between the hypergraph H_o and the bipartite graph G_B.

$$
K_{H_o} = \begin{array}{c c} & \begin{array}{ccc} a_1^H & a_2^H & a_3^H \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} & \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array}
\quad
K_{G_B} = \begin{array}{c c} & \begin{array}{cccccccc} a_1^G & a_2^G & a_3^G & a_4^G & a_5^G & a_6^G & a_7^G & a_8^G \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_{a_1} \\ v_{a_2} \\ v_{a_3} \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}
$$

Chapter 3

Image processing using unoriented hypergraphs

The application of hypergraph differential operators to image processing introduces a shift by moving away from the regular grid representation of pixels. By modelling pixels in terms of hypergraphs, the presented approach goes beyond the limitations of local relationships between adjacent pixels and instead, it takes advantage of nonlocal connections. As described in Section 3.3, hypergraphs allow for the representation of nuanced relationships, linking pixels that may be spatially distant but share similarities in terms of image intensities within their neighbourhood.

Thus, the first Section 3.1 introduces an averaging vertex operator (3.1), which will be used in the practical part of this paper. The next Section 3.2 presents the initial value problem (3.7) used for the image processing and its forward-Euler time discretisation (3.8). Finally, we will see results of image denoising with local and nonlocal hypergraphs with different parameters in Section 3.3.

3.1 Averaging operators on unoriented hypergraphs

A new Laplace operator introduced in [FTB23] and based on intuitive averaging is described below. The following definitions for vertex averaging operators assume that the vertex and hyperedge weights w_G, w_I, W_I and W_G are equal to one, unless stated otherwise. As introduced in definition 2.3, $\hat{E}_H(v_i)$ is the set of incident hyperedges of a vertex $v_i \in V$ and from now on, the special vertex $v_{\tilde{q}}$ for each hyperedge $e_q \in E_H$ is no longer necessary.

DEFINITION 3.1: Vertex averaging operator $\overline{\Delta v}$.

Let $\overline{\Delta_v} : S(V) \to S(V)$ be the vertex averaging operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$, which is defined as:

$$\overline{\Delta_v} f(v_i) = \frac{1}{|\hat{E}_H(v_i)|} \sum_{e_q \in E_H} \psi(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in V} \psi(v_j, e_q) f(v_j). \tag{3.1}$$

DEFINITION 3.2: Adjoint vertex averaging operator $\overline{\Delta_v}^*$.

Let $\overline{\Delta_v}^* : S(V) \to S(V)$ be the adjoint vertex averaging operator for an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$, which is given as:

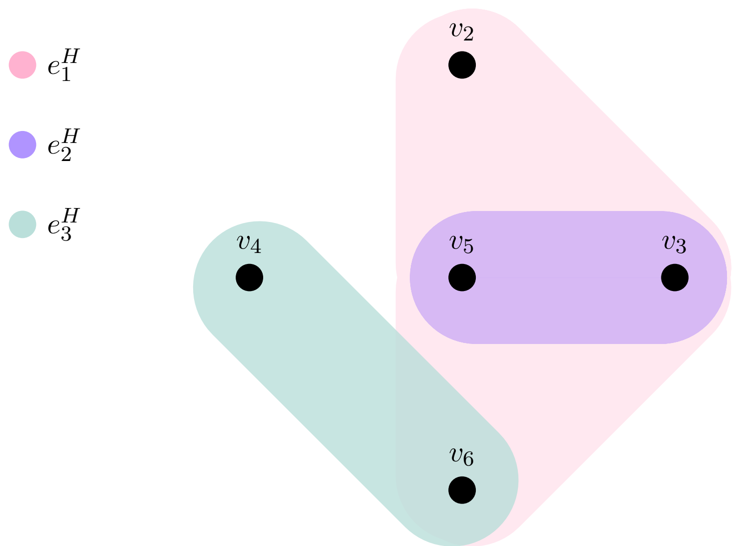
$$\overline{\Delta_v}^* f(v_i) = \sum_{e_q \in E_H} \psi(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in V} \psi(v_j, e_q) \frac{1}{|\hat{E}_H(v_j)|} f(v_j). \tag{3.2}$$

The next connection of the vertex averaging operator and the adjoint vertex averaging operator holds true for all vertex functions $f_1, f_2 \in S(V)$ in an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$:

$$\langle f_2, \overline{\Delta_v} f_1 \rangle_{S(V)} = \langle f_1, \overline{\Delta_v}^* f_2 \rangle_{S(V)}, \tag{3.3}$$

Before moving on to the next theorem, it is necessary to show that the vertex averaging operator $\overline{\Delta_v}$ does not conserve mean values. In order to illustrate this statement more clearly, an example will be given below.

Example 9. Vertex averaging operator does not conserve mean values. For our example we define an unoriented hypergraph $H_U = (V, E_H)$ with vertex set $V = \{v_2, v_3, v_4, v_5, v_6\}$ and hyperedge set $\{\{v_2, v_3, v_5, v_6\}, \{v_3, v_5\}, \{v_4, v_6\}\} = \{e_1^H, e_2^H, e_3^H\}$. Then we can calculate the vertex averaging operator $\overline{\Delta_v}$ for each vertex $v_i \in V$.



$$\overline{\Delta_v} f(v_2) = \frac{1}{1}\left[\frac{1}{4}\Big(f(v_2) + f(v_3) + f(v_5) + f(v_6)\Big)\right]$$

$$\overline{\Delta_v} f(v_3) = \frac{1}{2}\left[\frac{1}{4}\Big(f(v_2) + f(v_3) + f(v_5) + f(v_6)\Big) + \frac{1}{2}\Big(f(v_3) + f(v_5)\Big)\right]$$

$$\overline{\Delta_v} f(v_4) = \frac{1}{1}\left[\frac{1}{2}\Big(f(v_4) + f(v_6)\Big)\right]$$

$$\overline{\Delta_v} f(v_5) = \frac{1}{2}\left[\frac{1}{4}\Big(f(v_2) + f(v_3) + f(v_5) + f(v_6)\Big) + \frac{1}{2}\Big(f(v_3) + f(v_5)\Big)\right]$$

$$\overline{\Delta_v} f(v_6) = \frac{1}{2}\left[\frac{1}{4}\Big(f(v_2) + f(v_3) + f(v_5) + f(v_6)\Big) + \frac{1}{2}\Big(f(v_4) + f(v_6)\Big)\right]$$

$$\begin{aligned}\implies \ & \overline{\Delta_v} f(v_2) + \overline{\Delta_v} f(v_3) + \overline{\Delta_v} f(v_4) + \overline{\Delta_v} f(v_5) + \overline{\Delta_v} f(v_6) \\ & = \frac{5}{8} f(v_2) + \frac{9}{8} f(v_3) + \frac{3}{4} f(v_4) + \frac{9}{8} f(v_5) + \frac{11}{8} f(v_6).\end{aligned} \tag{3.4}$$

However, the sum of these values (3.4), does not equal to the sum vertex functions

$$f(v_2) + f(v_3) + f(v_4) + f(v_5) + f(v_6),$$

indicating that the vertex averaging operator does not conserve mean values for general vertex functions $f \in S(V)$. However, the adjoint vertex averaging operator conserves the overall energy for all vertex functions $f \in S(V)$.

THEOREM 3.3: Adjoint vertex averaging operator $\overline{\Delta_v}^*$ conserves mean values.

The adjoint vertex averaging operator $\overline{\Delta_v}^*$ for any unoriented hypergraph $H_u = (V, E_H, w, W)$ without isolated vertices conserves the mean value of all vertex function $f \in S(V)$, which is expressed through the following equality:

$$\sum_{v_i \in V} \overline{\Delta_v}^* f(v_i) = \sum_{v_i \in V} f(v_i), \tag{3.5}$$

for all $f \in S(V)$.

Proof. The following reformulations are valid for an unoriented hypergraph $H_u = (V, E_H, w, W)$ without any isolated vertices and with a vertex function $f \in S(V)$:

$$\begin{aligned}
\sum_{v_i \in V} \overline{\Delta_v}^* f(v_i) &= \sum_{v_i \in V} \sum_{e_q \in E_H} \psi(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in V} \psi(v_j, e_q) \frac{1}{|\hat{E}_H(v_j)|} f(v_j) \\
&= \sum_{v_i \in V} \sum_{e_q \in E_H} \sum_{v_j \in V} \psi(v_i, e_q) \frac{1}{|e_q|} \psi(v_j, e_q) \frac{1}{|\hat{E}_H(v_j)|} f(v_j) \\
&= \sum_{v_j \in V} \sum_{e_q \in E_H} \sum_{v_i \in V} \psi(v_i, e_q) \frac{1}{|e_q|} \psi(v_j, e_q) \frac{1}{|\hat{E}_H(v_j)|} f(v_j) \\
&= \sum_{v_j \in V} f(v_j) \frac{1}{|\hat{E}_H(v_j)|} \sum_{e_q \in E_H} \psi(v_j, e_q) \frac{1}{|e_q|} \sum_{v_i \in V} \psi(v_i, e_q) \\
&= \sum_{v_j \in V} f(v_j) \frac{1}{|\hat{E}_H(v_j)|} \sum_{e_q \in E_H} \psi(v_j, e_q) \\
&= \sum_{v_j \in V} f(v_j) \frac{1}{|\hat{E}_H(v_j)|} |\hat{E}_H(v_j)| \\
&= \sum_{v_j \in V} f(v_j).
\end{aligned}$$

Therefore, the adjoint vertex averaging operator $\overline{\Delta_v}^*$ presented here conserves the total energy of any given unoriented hypergraph $H_u = (V, E_H, w, W)$ for all vertex functions $f \in S(V)$.

□

Based on the fact that the adjoint vertex averaging operator $\overline{\Delta_v}^*$ conserves the mean in Euclidean space, and assuming the notation $\mathcal{Q} \in \mathbb{R}^{n \times m}$ as the original image, we can consider $\overline{\Delta_v} - \mathcal{Q}$ as a feasible operator for exploring scale spaces. Furthermore, the energy conservation of the adjoint shows that $\overline{\Delta_v}$ indeed has eigenvalue one with constant eigenfunction. Thus, the evolution equation using the operator $\overline{\Delta v} - \mathcal{Q}$ is expected to converge to a constant state and provide a suitable scale space, which we will explore further below.

LEMMA 3.4.

For an unoriented weighted hypergraph $H_u = (V, E_H, w, W)$ with constant vertex degrees the operator $\overline{\Delta_v} f - f$ is equivalent to the graph Laplacian on a weighted oriented graph $G_o = (V, A_G, w, W)$ with weight function:

$$w(a_q) = w(v_i, v_j) = \frac{1}{|\hat{E}_H(v_i)|} \sum_{e_q \in E_H} \frac{1}{|e_q|} \psi(v_i, e_q) \psi(v_j, e_q). \tag{3.6}$$

3.2 Image processing using unoriented hypergraphs

In the following we discuss PDEs based on the family of p-Laplacian and averaging operators on unoriented hypergraphs introduced before, which can be used for performing image processing.

- **Initial value problem**

Before formulating the initial value problem, it is necessary to introduce new parameters: The initial vertex function is given by $f_0 \in S(V)$ and an additional data fidelity term is defined as $\delta \in \mathbb{R}_0^+$. This parameter allows a balance between smoothing the perturbed image pixels through the hypergraph vertex averaging operator and maintaining the closeness to the original image. Considering that our averaging operator 3.1 is not self-adjoint, we do not use the classical regulation parameter $1/\delta$, since the corresponding term in (3.7) cannot occur in a gradient flow.

For grayscale images represented on regular grids, image processing is executed through a system of partial differential equations. This system can be viewed as an initial value problem associated with the vertex averaging operator introduced in 3.1. Our focus lies on solving the following initial value problem:

$$\begin{cases} \overline{\Delta_v} f(v_i,t) - f(v_i,t) + \delta \cdot \Big(f_0(v_i) - f(v_i,t) \Big) = 0, & v_i \in V, t \in (0,\infty) \\ f(v,0) = f_0(v_i) & v_i \in V. \end{cases} \tag{3.7}$$

- **Forward-Euler time discretisation**.

To solve the initial value problem (3.7), we use the forward-Euler time discretisation, which approximates the derivative of a function with the forward difference. The forward-Euler time discretization with a fixed time step size $\tau > 0$ involves updating the solution at each time step, and the chosen time step size is kept small enough to satisfy the Courant-Friedrichs-Lewy stability condition. The CFL stability criterion is a condition that ensures the stability of numerical methods for solving partial differential equations, particularly those involving convection terms:

$$f_{n+1}(v_i) = f_n(v_i) + \tau \cdot \Big[\overline{\Delta_v} f(v_i) - f_n(v_i) + \delta \cdot \Big(f_0(v_i) - f_n(v_i) \Big) \Big]. \tag{3.8}$$

3.3 Numerical experiments

- **Initial Image $\mathcal{Q}$.**

Our experiments are performed with a greyscale image $\mathcal{Q} \in \mathbb{R}^{n \times m}$ with a height of n pixels and a width of m pixels. In our numerical experiments, we confine our focus to greyscale images for the sake of simplicity.

In a greyscale image, each pixel has a value between 0 and 255, where zero corresponds to *black* and 255 corresponds to *white*. The values between 0 and 255 are different shades of grey, with values closer to 0 being darker and values closer to 255 being lighter.

We associate each pixel at position (i, j) with a vertex v_k of the hypergraph and the vertex function $f \in S(V)$ describes the colour intensity value of each pixel in the given image.

Figure 3.1: Initial image of the houses of Burano Island.

The next step is to generate a noisy image by adding Gaussian noise.

- **Create a noisy image $\widetilde{\mathcal{Q}}$ for image denoising.**

To perform an image denoising experiment, which can also be considered as an inverse problem, we need to add noise to the original image. Applying Gaussian noise to image pixels with a variance σ^2 corresponds to adding random values, sampled from a Gaussian distribution, to the pixel intensities of the image. The Gaussian distribution is characterized by its mean (in our case $\mu = 0$) and variance (here $\sigma^2 = 300$), where a bigger variance leads to greater variability in the noise.

So if $\mathcal{Q}(i, j) \in [0, 255]$ represents the original intensity of the pixel at position (i, j) in the original image, and $\mathcal{N}(i, j) \sim (0, \sigma^2)$ is the Gaussian noise with variance σ^2, the noisy pixel intensity $\widetilde{\mathcal{Q}}(i, j) \in [0, 255]$ can be expressed as:

$$\widetilde{\mathcal{Q}}(i, j) = \mathcal{Q}(i, j) + \mathcal{N}(i, j). \tag{3.9}$$

Image denoising is the process of removing noise from an image with the aim of recovering the original underlying structure, while simultaneously minimising the effect of unwanted artefacts introduced by the noise. By treating this as an inverse problem, the goal is to derive the original image from its noisy version.

The noisy image $\widetilde{\mathcal{Q}}$, which will be used to build a hypergraph with vertex function $f \in S(V)$, is shown below.

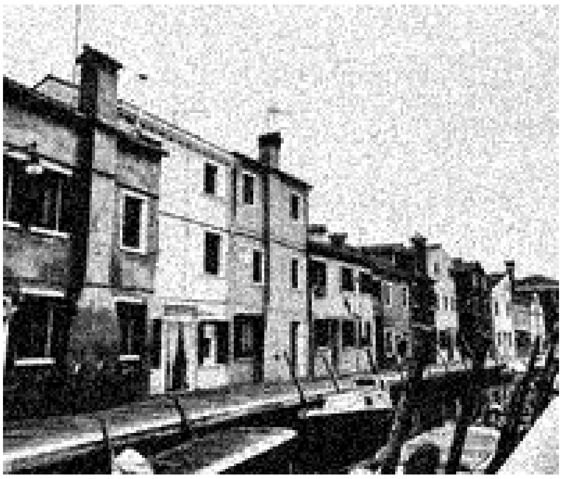


Figure 3.2: Noisy image of the houses of Burano Island.

- **Local image processing.**

Local image processing refers to the manipulation and analysis of images on a pixel-by-pixel basis, often performed within a specific region or neighborhood of an image.

1. **Hypergraph Construction**:

The unoriented hypergraph is constructed with vertices representing image pixels and hyperedges encoding the neighbourhood of each pixel. For each pixel at position (i, j), there is a vertex $v(i, j) \in V$ and a hyperedge $e^H(i, j)$ consisting of the pixel itself and its four direct neighbours. The hyperedge $e_H(i, j)$ is essentially a set of five vertices that represent the local structure around the pixel at (i, j).

Let us assume that, $v(i, j)$ is the some value associated with the central pixel and $e_H(i, j)$ is hyperedge containing the central pixel and its neighbouring pixels. For a pixel that is not on the boundary of the image, the hyperedge is given by:

$$e^H(i, j) = \{v(i, j), v(i-1, j), v(i, j-1), v(i, j+1), v(i+1, j)\}, \tag{3.10}$$

where $v(i-1, j), v(i, j-1), v(i, j+1), v(i+1, j)$ are the neighbours of $v(i, j)$.

2. **Neumann Zero Boundary Conditions**:

For pixels at the boundary of the image, Neumann zero boundary conditions are used. The image is assumed to be extended constantly, meaning the pixel values are considered constant beyond the image boundary. If the central pixel is on the boundary, the construction of the hyperedge has to be adjusted to account for the boundary.

The Neumann boundary conditions for a $m \times n$ image at the horizontal boundary lead to the following hyperedge definitions:

$$e^H(i, j) = \begin{cases} \{v(i, j), v(i, j-1), v(i+1, j), v(i, j+1)\} & \text{if } i = 1, j \notin \{1, n\}, \\ \{v(i, j), v(i, j-1), v(i-1, j), v(i, j+1)\} & \text{if } i = m, j \notin \{1, n\}. \end{cases} \tag{3.11}$$

Similarly for the vertical boundary, we use the following hyperedge definitions:

$$e^H(i,j) = \begin{cases} \{v(i,j), v(i-1,j), v(i,j+1), v(i+1,j)\} & \text{if } i \notin \{1,m\}, j = 1, \\ \{v(i,j), v(i-1,j), v(i,j-1), v(i+1,j)\} & \text{if } i \notin \{1,m\}, j = n. \end{cases} \tag{3.12}$$

And for the four corners we get:

$$e^H(i,j) = \begin{cases} \{v(i,j), v(i+1,j), v(i,j+1)\} & \text{if } i = 1, j = 1, \\ \{v(i,j), v(i,j-1), v(i+1,j)\} & \text{if } i = 1, j = n, \\ \{v(i,j), v(i-1,j), v(i,j+1)\} & \text{if } i = m, j = 1, \\ \{v(i,j), v(i,j-1), v(i-1,j)\} & \text{if } i = m, j = n. \end{cases} \tag{3.13}$$

3. **Numerical results**:
As initial data we have a greyscale image with dimensions 100 by 78 pixels, therefore by simple multiplication we get 7800 vertices and hyperedges. All these hyperedges are unique and part of the same unoriented hypergraph H_u.

From Fig. 3.3, we can see that with data fidelity parameter $\delta = 10$ the result of the denoised image is as close as possible to the original image. Gradually, as the δ parameter is decreased, the image becomes more and more blurry. We can conclude that the smaller the value of the data fidelity term parameter, the greater the smoothing effect of the vertex averaging operator.

The second numerical experiment Fig. 3.4 is dedicated to changing the time step τ while keeping a constant parameter $\delta = 0$. A bigger time step τ results in coarser image features, i.e. more pronounced smoothing, which leads to the loss of finer details in the image. We can therefore conclude that the vertex averaging operator $\overline{\Delta_v}$ has a more prominent effect for bigger time step values, resulting in less noise but also less details in the image.

(a) $\delta = 10$

(b) $\delta = 1$

(c) $\delta = 0.1$

(d) $\delta = 0.01$

(e) $\delta = 0.001$

(f) $\delta = 0.0001$

Figure 3.3: Results of local image processing with a constant time step $\tau = 0.1$ after 100 iterations.

(a) $\tau = 0.001$

(b) $\tau = 0.01$

(c) $\tau = 0.03$

(d) $\tau = 0.1$

(e) $\tau = 0.3$

(f) $\tau = 2$

Figure 3.4: Results of local image processing with a constant data fidelity term $\delta = 0$ after 100 iterations.

- **Nonlocal image processing.**

1. **Hypergraph construction:**
In this numerical experiment, we use a nonlocal image processing technique for an unoriented hypergraph, where hyperedges are constructed based on the similarity of pixel intensities rather than relying on a local neighborhood.

As before, for each pixel at position (i, j), there is a vertex $v(i, j) \in V$ and a hyperedge $e^H(i, j)$. We construct the hyperedge $e^H(i, j)$ by finding the closest pixels in terms of intensity value within a given radius. We add all vertices $v_k \in V$ to the hyperedge induced by vertex $v(i, j)$ if the distance between the vertex functions $f(v_k)$ and $f(v(i, j))$ satisfy the following statement:

$$|f(v_k) - f(v(i, j))| < \varepsilon. \tag{3.14}$$

This distance threshold $\varepsilon > 0$ ensures that vertices with similar values of the vertex function are connected in the hypergraph. The appropriate choice of ε depends on the image dimensions and the goals of the image processing application. Choosing ϵ too small can result in hyperedges $e^H(i, j)$ only containing the vertex $v(i, j)$ and choosing ϵ too big, can lead to hyperedges with very large cardinality and hence long computational running times.

It can occur that two different vertices $v(i, j)$ and $v(k, l)$ result in hyperedges containing the same set of vertices, so $e^H(i, j) = e^H(k, l)$, however we only consider unique hyperedges in our unoriented hypergraph.

2. **Assigning different weights to hyperedges:**
Up until now, all the weights related to hyperedges have been set to one, however in this experiment we will distribute the weights of hyperedges according to their homogeneous behaviour within itself. If the difference between the intensity values of the vertices contained in the hyperedge (and hence also the average between them) is close to or equal to zero, then the weight of this hyperedge will be big.

To measure the spread or variability of intensities within a hyperedge, we can calculate the variance based on the intensity values of the pixels included in the hyperedge. The formula below calculates the average squared difference between each intensity value $f(v_k)$ and the mean intensity $\overline{f} = \frac{1}{|e^H|} \sum_{v_k \in e^H} f(v_k)$ in the hyperedge:

$$\sigma^2 = \frac{1}{|e^H| - 1} \sum_{v_k \in e^H} (f(v_k) - \overline{f})^2. \tag{3.15}$$

Thus, based on the variance within a hyperedge e^H (3.15), a weight $w(e^H)$ is assigned to each hypergraph such that:

$$w(e^H) = \frac{1}{1 + \sigma^2}. \tag{3.16}$$

3. **Numerical results:**
To find the nearest neighbours of vertex $v_i \in V$, the KD-tree algorithm was used. This process essentially involves constructing a spatial tree structure that efficiently organises the image data in a way that allows for fast nearest neighbour queries using the Euclidean distance metric. As a result, we get an unoriented hypergraph $H_u = (V, E_H)$ containing unique hyperedges $E_H = \{e_1, ..., e_k\}$, for which we used the threshold $\varepsilon = 8$ in the presented numerical experiments.

Similar to the previous experiment for non-local image analysis, we first keep the step size $\tau = 1$ constant throughout and change the data fidelity parameter δ Fig. 3.5. In addition, the weights for each hyperedge are set to one for this experiment. As the data fidelity term δ decreases, noise gradually disappears from the image. This effect is particularly noticeable in the upper right corner of the image and dark lines also become more pronounced.

Keeping the data fidelity parameter constant $\delta = 0$, still not using the variable hyperedge weights and increasing the step size τ, gives positive results as well, as can be observed in Fig. 3.6. A continuous increase of τ leads to a denoising of the image, which again is particularly visible in the top right corner of the image.

The second experiment involves using different weights for the hyperedges, as described earlier, so that we can clearly see how the image changes with the data fidelity parameter δ. As δ decreases in Fig. 3.7, the resulting images become more and more different from the original image. This suggests that smaller values of the regularisation parameter lead to more significant changes in the image. The range of pixel intensities in the resulting images also decreases. This means that the images become more uniform or less varied in terms of pixel intensity, possibly converging towards a single grey level intensity for the whole image. This could indicate a high degree of regularisation or smoothing.

The results of keeping the data fidelity term constant $\delta = 0$ and increasing the step size τ together with the before introduced hyperedge weights are shown in Fig. 3.8. As τ increases from 0.001 to 0.1, the image eventually reaches an average grey colour and the diminishing number of different pixel intensities can be observed in this experiment as well.

(a) $\delta = 1$

(b) $\delta = 0.5$

(c) $\delta = 0.01$

(d) $\delta = 0.005$

(e) $\delta = 0.0001$

Figure 3.5: Results of nonlocal image processing with time step $\tau = 0$ after 100 iterations.

(a) $\tau = 0.01$

(b) $\tau = 0.1$

(c) $\tau = 0.5$

(d) $\tau = 0.7$

(e) $\tau = 1$

Figure 3.6: Results of nonlocal image processing with a constant data fidelity term $\delta = 0$ after 100 iterations.

(a) $\delta = 0.5$

(b) $\delta = 0.1$

(c) $\delta = 0.01$

(d) $\delta = 0.001$

(e) $\delta = 0.0001$

Figure 3.7: Results of nonlocal image processing with different weights for hyperedges at a constant time step $\tau = 1$ after 100 iterations.

(a) $\tau = 0.001$

(b) $\tau = 0.005$

(c) $\tau = 0.01$

(d) $\tau = 0.05$

(e) $\tau = 0.1$

Figure 3.8: Results of nonlocal image processing with different weights for hyperedges at a constant data fidelity term $\delta = 0.001$ after 100 iterations.

Chapter 4

Conclusion

In conclusion, this thesis has achieved its objectives in presenting the fundamentals of (hyper)graph theory, defining novel edge and hyperedge differential operators for unoriented graphs and hypergraphs, and applying the introduced differential operators to local and nonlocal image denoising. The thesis includes the definitions of oriented and unoriented graphs, as well as functions on them. It then explores differential operators and the family of p-Laplacian operators on graphs. Finally, the paper expands graph theory to the concept of hypergraphs, considering similar differential operators while taking into account differences between graphs and hypergraphs. The definitions of gradient, adjoint, and p-Laplacian operators both for vertex and hyperedge functions were examined for oriented and unoriented hypergraphs.

In the practical section, averaging operators for unoriented hypergraphs were described and used for local and nonlocal image processing. The impact of non-constant hypergraph weighting functions was also explored in numerical experiments. The weights of the hyperedges were determined based on the variance of the vertex function values of the vertices contained in the respective hyperedge.

Looking ahead, this thesis has established the basis for potential advancements and the future directions outlined cover a range of possibilities, including a more in-depth study of hypergraph gradients in relation to higher-order methods for partial differential equations and the impact of non-constant weight functions on hypergraphs.

Bibliography

Books

[Bre13] A. Bretto. *Hypergraph Theory: An Introduction*. Springer Publishing Company, Incorporated, 2013.

[DG23] Q. Dai and Y. Gao. *Mathematical Foundations of Hypergraph*. Springer Nature Singapore, 2023.

Articles

[NLS22] L. Neuhauser, R. Lambiotte, and M. Schaub. “Consensus Dynamics and Opinion Formation on Hypergraphs.” In: (2022). arXiv: 2105.01369.

[ETT15] A. Elmoataz, M. Toutain, and D. Tenbrinck. “On the p-Laplacian and ∞-Laplacian on graphs with Applications in Image and Data Processing.” In: (Nov. 2015), pp. 2412–2451. DOI: 10.1137/15M1022793.

[AM22] E. Andreotti and R. Mulas. “Signless normalized Laplacian for hypergraphs.” In: *Electronic Journal of Graph Theory and Applications* 10.2 (Sept. 2022), p. 485. DOI: 10.5614/ejgta.2022.10.2.11.

[Gao+22] Y. Gao et al. “Hypergraph Learning: Methods and Practices.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.5 (2022), pp. 2548–2566. DOI: 10.1109/TPAMI.2020.3039374.

[JMZ21] J. Jost, R. Mulas, and D. Zhang. “p-Laplace Operators for Oriented Hypergraphs.” In: *Vietnam Journal of Mathematics* 50 (Oct. 2021). DOI: 10.1007/s10013-021-00525-4.

[Faz23] A. Fazeny. “p-Laplacian Operators on Hypergraphs.Master thesis at FAU Erlangen-Nürnberg.” In: (2023).

[FTB23] A. Fazeny, D. Tenbrinck, and M. Burger. “Hypergraph p-Laplacians, Scale Spaces, and Information Flow in Networks.” In: (2023), 677â690. DOI: 10.1007/978-3-031-31975-4_52.

[vLux07] U. von Luxburg. “A Tutorial on Spectral Clustering.” In: (2007). arXiv: 0711.0189 [cs.DS].

[ZDC20] S. Zhang, Z. Ding, and S. Cui. “Introducing Hypergraph Signal Processing: Theoretical Foundation and Practical Applications.” In: *IEEE Internet of Things Journal* 7.1 (2020), pp. 639–660. DOI: 10.1109/JIOT.2019.2950213.

[JM19] J. Jost and R. Mulas. “Hypergraph Laplace operators for chemical reaction networks.” In: *Advances in Mathematics* 351 (2019), pp. 870–896. DOI: https://doi.org/10.1016/j.aim.2019.05.025.

[SMS18] S. Saito, D. Mandic, and H. Suzuki. “Hypergraph p-Laplacian: A Differential Geometry View.” In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11823.

[WKD16] M. M. Wolf, A. M. Klinvex, and D. M. Dunlavy. “Advantages to modeling relational data using hypergraphs versus graphs.” In: (2016), pp. 1–7. DOI: 10.1109/HPEC.2016.7761624.

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Erlangen, den 19.Dezember 2023 ....

Kseniia Lukin