

Simplicial Convolutional Filters

Yang, Maosheng ; Isufi, Elvin; Schaub, Michael T.; Leus, Geert

DOI

[10.1109/TSP.2022.3207045](https://doi.org/10.1109/TSP.2022.3207045)

Publication date

2022

Document Version

Final published version

Published in

IEEE Transactions on Signal Processing

Citation (APA)

Yang, M., Isufi, E., Schaub, M. T., & Leus, G. (2022). Simplicial Convolutional Filters. *IEEE Transactions on Signal Processing*, 70, 4633 - 4648. <https://doi.org/10.1109/TSP.2022.3207045>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Simplicial Convolutional Filters

Maosheng Yang ^{id}, *Graduate Student Member, IEEE*, Elvin Isufi ^{id}, *Member, IEEE*,
Michael T. Schaub ^{id}, *Member, IEEE*, and Geert Leus ^{id}, *Fellow, IEEE*

Abstract—We study linear filters for processing signals supported on abstract topological spaces modeled as simplicial complexes, which may be interpreted as generalizations of graphs that account for nodes, edges, triangular faces, etc. To process such signals, we develop simplicial convolutional filters defined as matrix polynomials of the lower and upper Hodge Laplacians. First, we study the properties of these filters and show that they are linear and shift-invariant, as well as permutation and orientation equivariant. These filters can also be implemented in a distributed fashion with a low computational complexity, as they involve only (multiple rounds of) simplicial shifting between upper and lower adjacent simplices. Second, focusing on edge-flows, we study the frequency responses of these filters and examine how we can use the Hodge-decomposition to delineate gradient, curl and harmonic frequencies. We discuss how these frequencies correspond to the lower- and the upper-adjacent couplings and the kernel of the Hodge Laplacian, respectively, and can be tuned independently by our filter designs. Third, we study different procedures for designing simplicial convolutional filters and discuss their relative advantages. Finally, we corroborate our simplicial filters in several applications: to extract different frequency components of a simplicial signal, to denoise edge flows, and to analyze financial markets and traffic networks.

Index Terms—Simplicial complexes, Hodge Laplacians, simplicial filters, filter design, Chebyshev polynomial.

I. INTRODUCTION

METHODS to process signals supported on non-Euclidean domains modeled as graphs have attracted substantial research interest recently. Most of these graph signal processing (GSP) methods focus on signals supported on nodes, e.g., temperature measurements in weather stations network or EEG signals in a brain network [2], [3]. Using linear shift operators that couple node signals to each other via the edges of a graph,

Manuscript received 27 January 2022; revised 14 July 2022; accepted 12 September 2022. Date of publication 15 September 2022; date of current version 30 September 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vincent Gripon. The work of Maosheng Yang was supported by the TU Delft AI Labs Programme. The work of Michael T. Schaub was supported by the Ministry of Culture and Science, MKW of the German State of North Rhine-Westphalia “NRW Rückkehrprogramm”. This work was presented in part at EUSIPCO 2021 [DOI: 10.23919/EUSIPCO54536.2021.9616185]. (*Corresponding author: Maosheng Yang.*)

Maosheng Yang and Elvin Isufi are with the Department of Intelligent Systems, Delft University of Technology, 2628 CD Delft, Netherlands (e-mail: m.yang-2@tudelft.nl; e.isufi-1@tudelft.nl).

Michael T. Schaub is with the Department of Computer Science, RWTH Aachen University, 52062 Aachen, Germany (e-mail: schaub@cs.rwth-aachen.de).

Geert Leus is with the Department of Microelectronics, Delft University of Technology, 2628 CD Delft, Netherlands (e-mail: g.j.t.leus@tudelft.nl).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TSP.2022.3207045>, provided by the authors.

Digital Object Identifier 10.1109/TSP.2022.3207045

e.g., in terms of an adjacency or a Laplacian matrix, we can design graph filters to process such node signals [2], [3], [4], [5].

However, we often encounter signals that are naturally associated with edges or sets of nodes in real-world applications. For example, blood flow between different areas in the brain [6], water flow in a hydrological network, data flow in a communication network, or traffic flow in a road network [7], [8]. We typically model these signals as a flow over the edges of a network. Edge flows have also been used in statistical ranking, to describe financial markets, to analyze games, etc. [9], [10], [11], [12]. Similarly, we may even encounter signals supported on sets of nodes [13], [14]. For instance, in a co-authorship network, the number of publications with more than two authors can be seen as such a signal [15].

In these cases, rather than focusing on utilizing the relationships between the nodes to process node signals, it can be fruitful to study relations between the node-relationships (edges, higher-order edges) themselves. In the case of edge-signals, we want to understand couplings between edges, e.g., mediated through a common incident node (lower adjacency) or because these edges contribute to a triadic relation (upper adjacency). To account for such relationships, we can model a (network) system as a simplicial complex (SC). Using this representation we can analyze signals associated to subsets of nodes, i.e., simplicial signals, with algebraic tools via so-called Hodge Laplacians [16], [17], [18], which generalize the familiar graph Laplacians.

In addition, the Hodge Laplacian admits a Hodge decomposition, which allows for an intuitive physical interpretation of signals supported on SCs [16], [17]. Namely, the Hodge decomposition states that any edge flow can be decomposed into gradient (curl-free), curl (divergence-free) and harmonic components, respectively. For instance, a water flow may contain a non-cyclic component which can be seen as the potential difference between water stations, a locally cyclic component with non-zero curl and a harmonic component being flow-conservative [8].

Previous works [18], [19] have established a framework to analyze simplicial signals and focused mostly on low-pass filtering applications. However, general linear filters for simplicial signals have not been considered in detail. In this paper, we propose a *simplicial convolutional filter* via the shift-and-sum operation as a matrix polynomial of the Hodge Laplacians to enable a flexible simplicial signal filtering. Our filter accounts for lower and upper adjacencies in an SC, e.g., the relationships between edges via a common node or a common 2-simplex, and allows to separately filter the three signal components provided by the Hodge decomposition.

Contributions: Our three main contributions include:

1) *Simplicial convolution.* We study simplicial shifting via the Hodge Laplacians as a basic operation to propagate signals locally using both lower- and upper-connectivities in an SC. Leveraging this shifting and the shift-and-sum operation, we

develop simplicial convolutional filters by aggregating multi-step shifted signals. Their local shifting operation allows a distributed implementation of the filter with a cost linear in the number of simplices. We show such filters are linear, shift-invariant, and equivariant to permutations of the labeling and the orientation of simplices.

2) *Filtering in the spectral domain.* Leveraging the simplicial Fourier transform (SFT) [18], we show that the principles of the convolutional theorem apply to the proposed filter, i.e., the filter output in the frequency domain operates as a point-wise multiplication between the filter frequency response and the SFT of the input signal. We further show how the simplicial frequencies act as measures of signal variations w.r.t. the lower and upper adjacencies and divide them into gradient, curl and harmonic frequencies. More precisely, the eigenmodes associated to these frequencies span the subspaces provided by the Hodge decomposition. Ultimately, this implies that the proposed filter can regulate signals independently in the three subspaces provided by the Hodge decomposition.

3) *Filter Design.* To implement a desired frequency response, we first consider a standard least-squares (LS) approach to design the filter. To avoid the eigenvalue computation, we then consider a grid-based universal design. As both strategies may suffer from numerical instability, we propose a numerically more stable Chebyshev polynomial design.

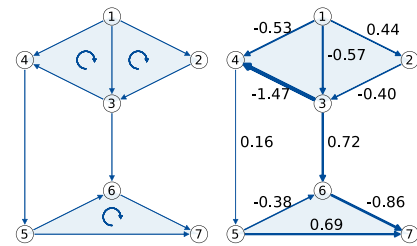
Related works: The idea of processing of signals defined on manifolds and topological spaces has been discussed in various areas, such as geometry processing [20], and topological data analysis [21]. For an introduction that is geared more towards a signal processing perspective see [17], [18], [22].

Filtering of simplicial signals has been partly approached from a regularization perspective. The works in [8], [19] proposed a regularized optimization framework based on (simplified variants of) the Hodge Laplacian, to promote flow conservation of the resulting estimated edge flows. The solution is a low-pass simplicial filter. The same regularizer was used in [19], [23] to perform edge flow interpolation by exploiting the divergence-free and curl-free behaviors of real-world flows. However, these assumptions do not always hold and the filters arising from the considered regularized optimization problems have limited degrees of freedom.

Filtering simplicial signals has also been analyzed in the Fourier domain, akin to how graph filters are analyzed via the graph Fourier transform [2]. The analogous SFT, defined via the eigendecomposition of the Hodge Laplacian was described in [18]. The eigenvectors provide a simplicial Fourier basis and the eigenvalues carry a notion of frequency.

In parallel, signal processing techniques have been extended from SCs to cell complexes [24], [25], [26]. However, the cell filters are essentially of the same form as the simplicial convolutional filters discussed here and in the preliminary conference version of this article [1]. Indeed, all the filters studied in the current paper can be generalized to cell complexes, by substituting the appropriate incidence matrices. Different neural network architectures on SCs have also been developed to learn from the simplicial data, e.g., [27], [28], [29], [30], [31], [32], [33]. Importantly, the linear operation in these different neural convolutional layers may be understood as a simplicial filter (as discussed here) with different filter parameters.

Outline: We begin by introducing some preliminaries in Section II. Then we propose the simplicial convolutional filter in Section III and investigate its properties. In Section IV, we



(a) A simplicial complex. (b) A random edge flow.

Fig. 1. Simplicial Complexes and Signals. (a) An SC of order 2 containing seven nodes, ten edges and three 2-simplex (the shaded filled triangles). Reference orientations of the simplices are indicate by corresponding arrows (the reference orientation of a node is trivial). (b) An arbitrary edge flow, where a negative flow indicates that the actual flow direction is opposite to the reference orientation and the magnitude is denoted by the edge width.

introduce the simplicial Fourier transform. We then analyze the simplicial filter in the spectral domain and study the notion of simplicial frequency. Different filter design methods are discussed in Section V. Finally, we use simplicial filters for subcomponent extraction and edge flow denoising, and consider applications to financial market and transportation networks in Section VI.

II. SIMPLICIAL COMPLEXES, SIGNALS AND HODGE LAPLACIANS

In this section, we review SCs [16], [34] and signals supported on simplices [18], [19]. We also introduce the Hodge Laplacians as an algebraic representation of a simplicial complex, that acts as a shift operator for simplicial signals.

Simplicial complexes: Given a finite set of vertices \mathcal{V} , a k -simplex \mathcal{S}^k is a subset of \mathcal{V} with cardinality $k + 1$. A subset of a k -simplex \mathcal{S}^k with cardinality k is called a *face* of \mathcal{S}^k ; hence, \mathcal{S}^k has $k + 1$ faces. A *coface* of a k -simplex is a simplex \mathcal{S}^{k+1} that includes it. A simplicial complex \mathcal{X} is a finite collection of simplices with an inclusion property: for any $\mathcal{S}^k \in \mathcal{X}$ all its faces $\mathcal{S}^{k-1} \subset \mathcal{S}^k$ are also part of the simplicial complex, i.e., $\mathcal{S}^{k-1} \in \mathcal{X}$. The order of an SC is the largest order of its simplices. For an SC of order K , we collect the k -simplices into a set $\mathcal{X}^k = \{\mathcal{S}_1^k, \dots, \mathcal{S}_{N_k}^k\}$ where $N_k = |\mathcal{X}^k|$ is the number of k -simplices [16], [18].

Based on its geometric realizations, we call a 0-simplex a node, a 1-simplex an edge and a 2-simplex a (filled) triangle. Note that an “empty triangle” formed by three vertices and three pairwise relations between them is not a 2-simplex. Henceforth, we refer to 2-simplices as triangles for simplicity. A graph is an SC of order 1 with N_0 nodes and N_1 edges. Fig. 1(a) shows an SC of order 2 including nodes, edges and triangles where edge $\{5, 6\}$ has nodes $\{5\}$ and $\{6\}$ as its faces and triangle $\{5, 6, 7\}$ as its coface.

Two k -simplices in an SC are *lower adjacent* if they have a common face and are *upper adjacent* if they are both faces of a common $(k + 1)$ -simplex. Thus, for the simplex \mathcal{S}_i^k , we define its *lower neighborhood* $\mathcal{N}_{i,\ell}^k$ as the set of its lower adjacent k -simplices and its *upper neighborhood* $\mathcal{N}_{i,u}^k$ as the set of its upper adjacent k -simplices. For the i th edge $\{5, 6\}$ in Fig. 1(a), we have that $\mathcal{N}_{i,\ell}^1 = \{\{4, 5\}, \{3, 6\}, \{5, 7\}, \{6, 7\}\}$, and the upper neighborhood is $\mathcal{N}_{i,u}^1 = \{\{5, 7\}, \{6, 7\}\}$.

Simplicial signals: For computational purposes, we fix an (arbitrary) reference orientation (see [16] and [35, p. 5] for more details) for each simplex according to the lexicographical ordering of its vertices. Given this reference orientation for k -simplices, we define a k -simplicial signal $\mathbf{s}^k = [s_1^k, \dots, s_{N_k}^k]^\top \in \mathbb{R}^{N_k}$ by attributing the value s_i^k to the i th k -simplex S_i^k . If the signal value s_i^k is positive, then the corresponding signal is aligned with the reference orientation; opposite otherwise. Fig. 1(b) illustrates an arbitrary edge flow on an SC, in which some flows are aligned with and other are opposite the reference orientation (negative). For convenience, henceforth, we denote a node signal by $\mathbf{v} = [v_1, \dots, v_{N_0}]^\top \in \mathbb{R}^{N_0}$ and an edge flow by $\mathbf{f} = [f_1, \dots, f_{N_1}]^\top \in \mathbb{R}^{N_1}$.

Hodge Laplacians: We can describe the relationships between $(k-1)$ -simplices and k -simplices by the k th incidence matrix $\mathbf{B}_k \in \mathbb{R}^{N_{k-1} \times N_k}$, which maps each k -simplex to the $(k-1)$ -simplices that are its faces; cf. [18], [34] for more details. Specifically, \mathbf{B}_1 is the node-edge incidence matrix, and \mathbf{B}_2 is the edge-triangle incidence matrix. By definition, incidence matrices have the property [16], [18]

$$\mathbf{B}_k \mathbf{B}_{k+1} = \mathbf{0}. \quad (1)$$

Upon defining the incidence matrices, we can describe an SC \mathcal{X} of order K via the Hodge Laplacians

$$\mathbf{L}_k = \mathbf{B}_k^\top \mathbf{B}_k + \mathbf{B}_{k+1} \mathbf{B}_{k+1}^\top, \quad k = 1, \dots, K-1, \quad (2)$$

with the graph Laplacian $\mathbf{L}_0 = \mathbf{B}_1 \mathbf{B}_1^\top$ and $\mathbf{L}_K = \mathbf{B}_K^\top \mathbf{B}_K$. The k th-Hodge Laplacian \mathbf{L}_k contains the lower Laplacian $\mathbf{L}_{k,\ell} \triangleq \mathbf{B}_k^\top \mathbf{B}_k$ and the upper Laplacian $\mathbf{L}_{k,u} \triangleq \mathbf{B}_{k+1} \mathbf{B}_{k+1}^\top$. The lower Laplacian encodes the lower adjacency relationships between simplices through faces while the upper one encodes the upper adjacency relationships through cofaces. In particular, $\mathbf{L}_{1,\ell}$ encodes the edge adjacencies through their incident nodes and $\mathbf{L}_{1,u}$ through the common triangles that they form.

III. SIMPLICIAL CONVOLUTIONAL FILTERS

In this section, we propose a simplicial convolutional filter based on the Hodge Laplacian. We study its basic building block, the simplicial shifting, and show how its local characteristics make it amenable to a distributed implementation. We then look into the properties of shift-invariance and permutation and orientation equivariance in the simplicial domain.

Given the k th-Hodge Laplacian \mathbf{L}_k , we define a *simplicial convolutional filter* to process a k -simplicial signal \mathbf{s}^k as

$$\mathbf{H}_k = h_0 \mathbf{I} + \sum_{l_1=1}^{L_1} \alpha_{l_1} (\mathbf{B}_k^\top \mathbf{B}_k)^{l_1} + \sum_{l_2=1}^{L_2} \beta_{l_2} (\mathbf{B}_{k+1} \mathbf{B}_{k+1}^\top)^{l_2}, \quad (3)$$

where $\mathbf{H}_k := \mathbf{H}(\mathbf{L}_{k,\ell}, \mathbf{L}_{k,u})$ is a matrix polynomial of the lower and upper Hodge Laplacians with filter coefficients $h_0, \boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{L_1}]^\top$, $\boldsymbol{\beta} = [\beta_1, \dots, \beta_{L_2}]^\top$ and filter orders L_1, L_2 . When $k=0$, we obtain the graph convolutional filter $\mathbf{H}_0 := \mathbf{H}(\mathbf{L}_0)$ built upon the graph Laplacian \mathbf{L}_0 [2], [4], [5].

In the following we will see that assigning two different sets of coefficients to the lower and upper Laplacian parts in \mathbf{H}_k enables the filter to treat lower and upper adjacencies differently which results in a more flexible control of the frequency response. Instead, the filter $\mathbf{H}_k = \sum_{l=0}^L h_l \mathbf{L}_k^l$, which is equivalent to setting $L_1 = L_2 = L$ and $\boldsymbol{\alpha} = \boldsymbol{\beta}$ in (3), cannot differentiate

between the two types of adjacencies and loses some expressive power.

When applying \mathbf{H}_k to a k -simplicial signal \mathbf{s}^k , it generates an output $\mathbf{s}_o^k = \mathbf{H}_k \mathbf{s}^k$ which is a *shift-and-sum* operation where the filter \mathbf{H}_k first shifts the signal L_1 times over the lower neighborhoods and L_2 times over the upper neighborhoods, and then sums the shifted results according the corresponding coefficients. This is analogous to the convolutions of graph signals, images and time series [2]. For ease of exposition, we study the filtering process of an edge flow \mathbf{f} via an edge filter \mathbf{H}_1 hereafter.

Simplicial shifting and local implementation: Consider an edge filter \mathbf{H}_1 applied to an edge flow \mathbf{f} with an output

$$\mathbf{f}_o = \mathbf{H}_1 \mathbf{f} = h_0 \mathbf{f} + \sum_{l_1=1}^{L_1} \alpha_{l_1} \mathbf{L}_{1,\ell}^{l_1} \mathbf{f} + \sum_{l_2=1}^{L_2} \beta_{l_2} \mathbf{L}_{1,u}^{l_2} \mathbf{f}, \quad (4)$$

where the basic operation consists of applying different powers of the lower/upper Hodge Laplacian to the edge flow. This basic operation is denoted *simplicial shifting*. Let us first consider the one-step lower shifting $\mathbf{f}_\ell^{(1)} \triangleq \mathbf{L}_{1,\ell} \mathbf{f}$ and one-step upper shifting $\mathbf{f}_u^{(1)} \triangleq \mathbf{L}_{1,u} \mathbf{f}$. We can express the one-step shifted results on the i th edge, $[\mathbf{f}_\ell^{(1)}]_i$ and $[\mathbf{f}_u^{(1)}]_i$, as

$$\begin{aligned} [\mathbf{f}_\ell^{(1)}]_i &= \sum_{j \in \{\mathcal{N}_{\ell,i}^1 \cup i\}} [\mathbf{L}_{1,\ell}]_{ij} [\mathbf{f}]_j, \\ [\mathbf{f}_u^{(1)}]_i &= \sum_{j \in \{\mathcal{N}_{u,i}^1 \cup i\}} [\mathbf{L}_{1,u}]_{ij} [\mathbf{f}]_j, \end{aligned} \quad (5)$$

which are the weighted linear combinations of the edge flows on the lower and upper neighborhoods, $\mathcal{N}_{\ell,i}^1$ and $\mathcal{N}_{u,i}^1$, of edge i . This implies that one-step shifting is a *local operation* in the edge space within the direct lower/upper neighborhoods.

Consider now the l -step lower shifting of an edge flow \mathbf{f} , $\mathbf{f}_\ell^{(l)} \triangleq \mathbf{L}_{1,\ell}^l \mathbf{f} = \mathbf{L}_{1,\ell} \mathbf{f}_\ell^{(l-1)}$, where the second equality indicates that the l -step lower shifting can be computed as a one-step shifting of the previously shifted result, $\mathbf{f}_\ell^{(l-1)}$. Accordingly, the l -step upper shifting follows $\mathbf{f}_u^{(l)} \triangleq \mathbf{L}_{1,u}^l \mathbf{f} = \mathbf{L}_{1,u} \mathbf{f}_u^{(l-1)}$. Thus, the simplicial shifting allows a recursive implementation. For example, the two-step shifted results, $\mathbf{f}_\ell^{(2)}$ and $\mathbf{f}_u^{(2)}$, can be computed from $\mathbf{f}_\ell^{(1)}$ and $\mathbf{f}_u^{(1)}$ via another shifting. Each edge thus collects the flows from its lower and upper neighbors two hops away. Fig. 2 illustrates such shifting operations. Likewise, the l -step shifted results $\mathbf{f}_\ell^{(l)}$ and $\mathbf{f}_u^{(l)}$ contain the information up to the l -hop lower and upper neighborhoods. Finally, we can express the output (4) as

$$\mathbf{f}_o = h_0 \mathbf{f}^{(0)} + \sum_{l_1=1}^{L_1} \alpha_{l_1} \mathbf{f}_\ell^{(l_1)} + \sum_{l_2=1}^{L_2} \beta_{l_2} \mathbf{f}_u^{(l_2)}, \quad (6)$$

which is a weighted linear combination of lower and upper shifted simplicial signals after different steps. Fig. 3 illustrates such a shift-and-sum operation.

Thus, the simplicial convolutional filter admits a *simplicial locality*, i.e., the output \mathbf{f}_o is localized in the L_1 -hop lower neighborhood and L_2 -hop upper neighborhood of an edge. If two edges are lower adjacent more than L_1 hops away or upper adjacent more than L_2 hops away, \mathbf{H}_1 does not mix signals

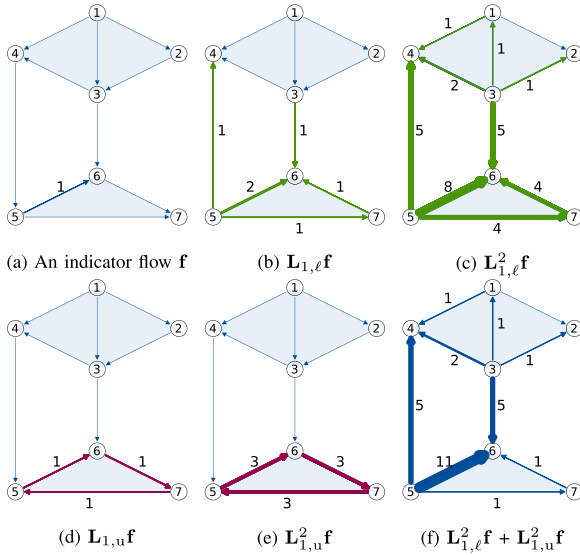


Fig. 2. Simplicial shifting. (a) An edge flow indicator \mathbf{f} of edge $\{5,6\}$. (b) One-step lower shifting $\mathbf{L}_{1,\ell}\mathbf{f}$. Edge $\{5,6\}$ and its direct lower neighboring edges (green) update their flows by aggregating information from their lower neighbors and themselves. (c) Two-step lower shifting $\mathbf{L}_{1,\ell}^2\mathbf{f}$. Lower neighboring edges (green) update their flows through faces within two hops away from edge $\{5,6\}$, which can be obtained by one-step shifting $\mathbf{L}_{1,\ell}\mathbf{f}$. (d) One-step upper shifting $\mathbf{L}_{1,u}\mathbf{f}$. Edge $\{5,6\}$ and its upper neighbors (red) update their flows through local information aggregation. (e) Two-step upper shifting $\mathbf{L}_{1,u}^2\mathbf{f}$. The output is localized within the one-hop upper neighborhood, as there is no upper neighboring edge two hops away from $\{5,6\}$. (f) Two-step shifting result $\mathbf{L}_{1,\ell}^2\mathbf{f}$, as the sum of (c) and (e).

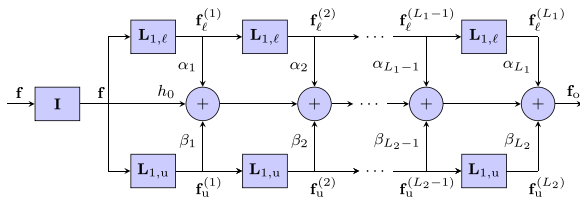


Fig. 3. Simplicial convolutional filtering is a shift-and-sum operation.

defined on such edges. Note, however, if two edges are lower or upper adjacent at distance d , a d -step lower or upper shifting via $\mathbf{L}_{1,\ell}^d$ or $\mathbf{L}_{1,u}^d$ does not necessarily cause an interaction between them: the aggregation might be cancelled out by the combination of the filter coefficients and the topology of the SC which leads to positive and negative entries in $\mathbf{L}_{1,\ell}$ or $\mathbf{L}_{1,u}$.

As a local operation within the simplicial neighborhood, the simplicial shifting allows for a distributed filter implementation, in which each edge updates its information only by a direct communication with its lower and upper neighbors. The communication complexity mainly comes from operation (5), which is of order $\mathcal{O}(D_\ell)$ with $D_\ell := \max\{|\mathcal{N}_{\ell,i}^1|_{i=1}^{N_1}\}$ for the lower simplicial shifting at each edge and $\mathcal{O}(D_u)$ with $D_u := \max\{|\mathcal{N}_{u,i}^1|_{i=1}^{N_1}\}$ for the upper simplicial shifting, i.e., the maximum number of lower and upper neighbors among all edges, respectively. Then, the total communication cost of the distributed implementation for each edge is $\mathcal{O}(D_\ell L_1 + D_u L_2)$ due to the L_1 lower shifting steps and L_2 upper ones.

Linearity and shift invariance: Simplicial convolutional filters are linear operators that are invariant to shifts.

Proposition 1: The simplicial filter \mathbf{H}_k [cf. (3)] is linear and shift-invariant. Specifically, in the edge space, given two edge

flows \mathbf{f}_1 and \mathbf{f}_2 and a simplicial filter \mathbf{H}_1 , we have

$$\begin{aligned} \text{Linearity: } & \mathbf{H}_1(a\mathbf{f}_1 + b\mathbf{f}_2) = a\mathbf{H}_1\mathbf{f}_1 + b\mathbf{H}_1\mathbf{f}_2, \\ \text{Shift-invariance: } & \mathbf{L}_{1,\ell}(\mathbf{H}_1\mathbf{f}_1) = \mathbf{H}_1(\mathbf{L}_{1,\ell}\mathbf{f}_1), \\ & \mathbf{L}_{1,u}(\mathbf{H}_1\mathbf{f}_1) = \mathbf{H}_1(\mathbf{L}_{1,u}\mathbf{f}_1). \end{aligned} \quad (7)$$

Proof: See Appendix A. \blacksquare

The shift-invariance implies that applying a lower or upper Hodge Laplacian to the simplicial output is equivalent to applying them to the input signal prior to filtering. Consequently, it holds that $\mathbf{H}_1\mathbf{H}'_1\mathbf{f} = \mathbf{H}'_1\mathbf{H}_1\mathbf{f}$ for any two filters \mathbf{H}_1 and \mathbf{H}'_1 .

Equivariance: In an SC, the labeling and the reference orientation of the simplices should not affect the filter output. We show that this is indeed the case. We can model the simplex relabeling of simplices by a set of permutation matrices [32]

$$\mathcal{P} = \{\mathbf{P}_k \in \{0,1\}^{N_k \times N_k} : \mathbf{P}_k \mathbf{1} = \mathbf{1}, \mathbf{P}_k^\top \mathbf{1} = \mathbf{1}, k \geq 0\}. \quad (8)$$

Let $\bar{\mathcal{P}} = (\mathbf{P}_0, \mathbf{P}_1, \dots) \subset \mathcal{P}$ denote a sequence of label permutations in an SC. After relabeling the simplices by $\bar{\mathcal{P}}$, signal s^k becomes $\mathbf{P}_k s^k$, i.e., a reordering of the entries of s^k . Similarly, the incidence matrix \mathbf{B}_k becomes $\bar{\mathbf{B}}_k = \mathbf{P}_{k-1} \mathbf{B}_k \mathbf{P}_k$, i.e., a reordering of the rows and columns of \mathbf{B}_k . Likewise, for \mathbf{L}_k , we get $\bar{\mathbf{L}}_k = \mathbf{P}_k \mathbf{L}_k \mathbf{P}_k^\top$.

Proposition 2 (Permutation equivariance): Consider the Hodge Laplacians \mathbf{L}_k and $\bar{\mathbf{L}}_k = \mathbf{P}_k \mathbf{L}_k \mathbf{P}_k^\top$ for a permutation sequence $\bar{\mathcal{P}}$. For the simplicial signals s^k and $\bar{s}^k = \mathbf{P}_k s^k$, with $\bar{\mathbf{H}}_k := \mathbf{H}(\bar{\mathbf{L}}_k)$ [cf. (3)], the simplicial filter outputs $s_o^k := \mathbf{H}_k s^k$ and $\bar{s}_o^k := \bar{\mathbf{H}}_k \bar{s}^k$ satisfy

$$\bar{s}_o^k := \bar{\mathbf{H}}_k \bar{s}^k = \bar{\mathbf{H}}_k (\mathbf{P}_k s^k) = \mathbf{P}_k \mathbf{H}_k s^k := \mathbf{P}_k s_o^k. \quad (9)$$

Proof: See Appendix B. \blacksquare

A new reference orientation of a k -simplex leads to a multiplication by -1 of the columns (or rows) of the incidence matrices \mathbf{B}_k and \mathbf{B}_{k+1} where the k -simplex appears and it also flips the sign of the corresponding simplicial signal. This can be modeled by a diagonal matrix \mathbf{D}_k from the set

$$\mathcal{D} = \{\mathbf{D}_k = \text{diag}(\mathbf{d}_k) : \mathbf{d}_k \in \{\pm 1\}^{N_k}, k \geq 1, \mathbf{d}_0 = \mathbf{1}\}, \quad (10)$$

where $\mathbf{d}_0 = \mathbf{1}$, as the orientation of the nodes is trivial [32]. Denote a sequence of orientation changes by $\bar{\mathcal{D}} = (\mathbf{D}_0, \mathbf{D}_1, \dots) \subset \mathcal{D}$. A k -simplicial signal s^k becomes $\mathbf{D}_k s^k$ after an orientation change by $\bar{\mathcal{D}}$. Accordingly, the incidence matrix \mathbf{B}_k becomes $\bar{\mathbf{B}}_k = \mathbf{D}_{k-1} \mathbf{B}_k \mathbf{D}_k$ and the Hodge Laplacian \mathbf{L}_k becomes $\bar{\mathbf{L}}_k = \mathbf{D}_k \mathbf{L}_k \mathbf{D}_k$.

Proposition 3 (Orientation equivariance): Consider the Hodge Laplacians \mathbf{L}_k and $\bar{\mathbf{L}}_k = \mathbf{D}_k \mathbf{L}_k \mathbf{D}_k$ for a sequence of orientation changes $\bar{\mathcal{D}}$. For the simplicial signals s^k and $\bar{s}^k = \mathbf{D}_k s^k$, with $\bar{\mathbf{H}}_k = \mathbf{H}(\bar{\mathbf{L}}_k)$, the simplicial filter outputs $s_o^k := \mathbf{H}_k s^k$ and $\bar{s}_o^k := \bar{\mathbf{H}}_k \bar{s}^k$ satisfy

$$\bar{s}_o^k := \bar{\mathbf{H}}_k \bar{s}^k = \bar{\mathbf{H}}_k (\mathbf{D}_k s^k) = \mathbf{D}_k \mathbf{H}_k s^k := \mathbf{D}_k s_o^k. \quad (11)$$

Proof: See Appendix C. \blacksquare

Intuitively, the two previous propositions state that for the simplicial filter \mathbf{H}_k the labeling and reference orientation of simplices are inconsequential for the filter output. These two properties have been previously reported in the context of neural network on SCs [27], [29], [32]. These equivariances imply that we can learn a filter to process a given simplex by seeing only permuted and reoriented versions of it: if two parts of an SC are topologically equivalent and the simplices support

corresponding flows, a simplicial convolutional filter yields equivalent outputs.

IV. SPECTRAL ANALYSIS OF SIMPLICIAL FILTERS

We now analyze the spectral properties of the simplicial convolutional filter. First, we introduce the Hodge decomposition and review the simplicial Fourier transform (SFT) [18]. Then, we investigate the simplicial frequency in terms of the Hodge decomposition. By defining three frequency types, we characterize the frequency response of the filter.

A. Hodge Decomposition

The Hodge decomposition in the edge space states that:

$$\mathbb{R}^{N_1} = \text{im}(\mathbf{B}_1^\top) \oplus \text{im}(\mathbf{B}_2) \oplus \ker(\mathbf{L}_1) \quad (12)$$

where $\text{im}(\cdot)$ and $\ker(\cdot)$ are the image and kernel of a matrix. This implies that the edge space is composed of three orthogonal subspaces, namely, the gradient space $\text{im}(\mathbf{B}_1^\top)$, the curl space $\text{im}(\mathbf{B}_2)$, and the harmonic space $\ker(\mathbf{L}_1)$ [16], [19]. Thus, any edge flow $\mathbf{f} \in \mathbb{R}^{N_1}$ can be decomposed into three orthogonal components $\mathbf{f} = \mathbf{f}_G + \mathbf{f}_C + \mathbf{f}_H$, which are the *gradient component* $\mathbf{f}_G \in \text{im}(\mathbf{B}_1^\top)$, the *curl component* $\mathbf{f}_C \in \text{im}(\mathbf{B}_2)$, and the *harmonic component* $\mathbf{f}_H \in \ker(\mathbf{L}_1)$, respectively. Furthermore, the incidence matrices \mathbf{B}_1 , \mathbf{B}_2 and their adjoints can be interpreted as follows [18], [19].

Divergence operator \mathbf{B}_1 : The incidence matrix \mathbf{B}_1 acts as a *divergence operator*. By applying it to an edge flow \mathbf{f} , we compute the divergence of the flow, $\text{div}(\mathbf{f}) = \mathbf{B}_1 \mathbf{f}$. The i th entry of $\text{div}(\mathbf{f})$ is the netflow passing through the i th vertex, i.e., the difference between the total inflow and outflow at vertex i . A vertex is a source or sink if it has a nonzero netflow.

Gradient operator \mathbf{B}_1^\top : The adjoint operator \mathbf{B}_1^\top is called the *gradient operator*, which takes the difference between node signals along the oriented edges to induce an edge flow, $\mathbf{f}_G = \mathbf{B}_1^\top \mathbf{v}$. We call $\mathbf{f}_G \in \text{im}(\mathbf{B}_1^\top)$ a *gradient flow* and subspace $\text{im}(\mathbf{B}_1^\top)$ the *gradient space*, as any gradient flow can be induced from a node signal via the gradient operator.

Curl adjoint \mathbf{B}_2 : By applying matrix \mathbf{B}_2 to a triangle signal $\mathbf{t} \in \mathbb{R}^{N_2}$, we can induce a *curl flow*, $\mathbf{f}_C = \mathbf{B}_2 \mathbf{t}$, corresponding to a flow locally circling along the edges of triangles. The space $\text{im}(\mathbf{B}_2)$ is the *curl space* as any flow in it can be induced from a triangle signal.

Curl operator \mathbf{B}_2^\top : By applying the operator \mathbf{B}_2^\top to an edge flow \mathbf{f} , we compute its curl as $\text{curl}(\mathbf{f}) = \mathbf{B}_2^\top \mathbf{f}$, where the i th entry is the netflow circulating along the i th triangle, i.e., the sum of the edge flows forming the triangle. This can be seen as a rotational variation measure of the edge flow.

The two incidence matrices and their adjoints provide insights into the three orthogonal subspaces and signal components given by the Hodge decomposition.

- i) If edge flow \mathbf{f} has zero divergence at each vertex, i.e., $\mathbf{B}_1 \mathbf{f} = \mathbf{0} \Leftrightarrow \mathbf{f} \in \ker(\mathbf{B}_1)$, then it is cyclic or *divergence-free*. The space $\ker(\mathbf{B}_1)$ is called the *cycle space*, orthogonal to the gradient space, i.e., $\mathbb{R}^{N_1} = \text{im}(\mathbf{B}_1^\top) \oplus \ker(\mathbf{B}_1)$. A gradient flow always has nonzero divergence, while a curl flow is divergence-free due to (1).
- ii) If edge flow \mathbf{f} has zero curl on each triangle, it is *curl-free* and $\mathbf{f} \in \ker(\mathbf{B}_2^\top)$. The curl space $\text{im}(\mathbf{B}_2)$ is orthogonal to the space $\ker(\mathbf{B}_2^\top)$ and we have $\mathbb{R}^{N_1} = \text{im}(\mathbf{B}_2) \oplus \ker(\mathbf{B}_2^\top)$. A gradient flow \mathbf{f}_G is curl-free due to (1).

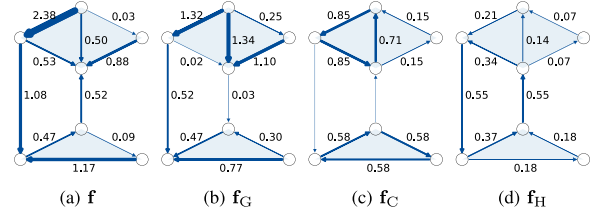


Fig. 4. Flow decomposition illustration (all numbers rounded to two decimal places). (a) A synthetic edge flow \mathbf{f} . (b) The gradient component \mathbf{f}_G has a nonzero netflow at each node, but a zero flow around each triangle. (c) The curl component \mathbf{f}_C has a zero netflow at each node, i.e., is divergence-free, but a nonzero flow around each triangle. (d) The harmonic component \mathbf{f}_H has a zero netflow at each node and zero circulation around each triangle, i.e., is divergence- and curl-free (see section VI-A).

- iii) The space $\ker(\mathbf{L}_1)$ is called the harmonic space. Any flow $\mathbf{f}_H \in \ker(\mathbf{L}_1)$ satisfies $\mathbf{L}_1 \mathbf{f}_H = \mathbf{0}$, which is harmonic, i.e., both divergence- and curl-free.

The decomposition of an edge flow into its three components reveals different properties of the flow, as shown in Fig. 4. For instance, we can study the effect of an external source or sink by extracting the gradient component of the edge flow [18]. In Section VI-A, we will discuss this subcomponent extraction problem and solve it with simplicial filters.

B. Simplicial Fourier Transform

The Hodge Laplacians are positive semidefinite matrices and admit an eigendecomposition

$$\mathbf{L}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^\top, \quad (13)$$

where the orthonormal matrix $\mathbf{U}_k = [\mathbf{u}_{k,1}, \dots, \mathbf{u}_{k,N_k}]$ collects the eigenvectors, and the diagonal matrix $\mathbf{\Lambda}_k = \text{diag}(\lambda_{k,1}, \dots, \lambda_{k,N_k})$ the associated eigenvalues. There exists a correspondence between \mathbf{U}_1 and the three orthogonal subspaces given by the Hodge decomposition (12), detailed in the following proposition.

Proposition 4: Given the 1-Hodge Laplacian of an SC $\mathbf{L}_1 = \mathbf{L}_{1,\ell} + \mathbf{L}_{1,u}$, the following holds.

- 1) Gradient eigenvectors $\mathbf{U}_G = [\mathbf{u}_{G,1}, \dots, \mathbf{u}_{G,N_G}] \in \mathbb{R}^{N_1 \times N_G}$ of $\mathbf{L}_{1,\ell}$ associated with nonzero eigenvalues span the gradient space $\text{im}(\mathbf{B}_1^\top)$ with dimension N_G , i.e., $\text{im}(\mathbf{B}_1^\top) = \text{im}(\mathbf{U}_G)$.
- 2) Curl eigenvectors $\mathbf{U}_C = [\mathbf{u}_{C,1}, \dots, \mathbf{u}_{C,N_C}] \in \mathbb{R}^{N_1 \times N_C}$ of $\mathbf{L}_{1,u}$ associated with nonzero eigenvalues span the curl space $\text{im}(\mathbf{B}_2)$ with dimension N_C , i.e., $\text{im}(\mathbf{B}_2) = \text{im}(\mathbf{U}_C)$.
- 3) Gradient eigenvectors \mathbf{U}_G are orthogonal to curl ones \mathbf{U}_C . Matrix $[\mathbf{U}_G \ \mathbf{U}_C]$ forms the eigenvectors of \mathbf{L}_1 associated with nonzero eigenvalues, which span the space $\text{im}(\mathbf{L}_1)$ with dimension $N_G + N_C$.
- 4) Harmonic eigenvectors $\mathbf{U}_H = [\mathbf{u}_{H,1}, \dots, \mathbf{u}_{H,N_H}] \in \mathbb{R}^{N_1 \times N_H}$ of \mathbf{L}_1 associated with zero eigenvalues span the harmonic space $\ker(\mathbf{L}_1)$ with dimension N_H , i.e., $\ker(\mathbf{L}_1) = \text{im}(\mathbf{U}_H)$. Matrices $[\mathbf{U}_H \ \mathbf{U}_C]$ and $[\mathbf{U}_H \ \mathbf{U}_G]$ provide the eigenvectors of $\mathbf{L}_{1,\ell}$ and $\mathbf{L}_{1,u}$ associated with zero eigenvalues, respectively.
- 5) The columns of \mathbf{U}_1 can be ordered such that $\mathbf{U}_1 = [\mathbf{U}_H \ \mathbf{U}_G \ \mathbf{U}_C]$. Matrix \mathbf{U}_1 forms an eigenvector basis for \mathbf{L}_1 , $\mathbf{L}_{1,\ell}$ and $\mathbf{L}_{1,u}$, and $N_1 = N_H + N_G + N_C$.

Proof: See Appendix D. ■

Proposition 4 shows that: i) the eigenvectors in \mathbf{U}_1 can fully span the three orthogonal subspaces given by the Hodge decomposition; ii) the Hodge Laplacian \mathbf{L}_1 and its lower and upper counterparts, $\mathbf{L}_{1,\ell}$ and $\mathbf{L}_{1,u}$, can be simultaneously diagonalized by \mathbf{U}_1 ; and, iii) from $\text{im}(\mathbf{B}_1^\top) = \text{im}(\mathbf{L}_{1,\ell})$, we have that the image of the lower Hodge Laplacian $\mathbf{L}_{1,\ell}$ coincides with the gradient space, and from $\text{im}(\mathbf{B}_2) = \text{im}(\mathbf{L}_{1,u})$, the image of $\mathbf{L}_{1,u}$ coincides with the curl space. These results are applicable to the k -Hodge Laplacian accordingly. See [19, Thm. 1] and [18, Prop. 1] for related discussions.

Thus, the lower shifting of a curl or harmonic flow leads to zero as the harmonic and curl space correspond to the null space of $\mathbf{L}_{1,\ell}$. Likewise, the upper shifting of a gradient or harmonic flow leads to zero, i.e.,

$$\mathbf{L}_{1,\ell}\mathbf{f}_C = \mathbf{0}, \quad \mathbf{L}_{1,u}\mathbf{f}_G = \mathbf{0}, \quad \mathbf{L}_{1,\ell}\mathbf{f}_H = \mathbf{L}_{1,u}\mathbf{f}_H = \mathbf{0}. \quad (14)$$

Given a k -simplicial signal \mathbf{s}^k , the *simplicial Fourier transform (SFT)* is given by its projection onto the eigenvectors \mathbf{U}_k , i.e., $\tilde{\mathbf{s}}^k \triangleq \mathbf{U}_k^\top \mathbf{s}^k$. Entry $[\tilde{\mathbf{s}}^k]_i$ represents the weight eigenvector $\mathbf{u}_{k,i}$ has on expressing \mathbf{s}^k . The inverse SFT is given by $\mathbf{s}^k = \mathbf{U}_k \tilde{\mathbf{s}}^k$. For $k = 0$, the SFT coincides with the GFT [18]. As for the GFT, the eigenvalues of \mathbf{L}_k carry the notion of simplicial frequencies. But in the simplex domain, this frequency notion is more involved. As we illustrate in the sequel for $k = 1$, the eigenvalues in the set $\mathcal{Q} = \{\lambda_{1,1}, \dots, \lambda_{1,N_1}\}$ of \mathbf{L}_1 measure three types of simplicial frequencies.

Gradient frequency: For any unit norm gradient eigenvector $\mathbf{u}_G \in \mathbf{U}_G$, associated to the gradient space $\text{im}(\mathbf{B}_1^\top)$, its corresponding eigenvalue follows

$$\lambda_G = \mathbf{u}_G^\top \mathbf{L}_1 \mathbf{u}_G = \|\mathbf{B}_1 \mathbf{u}_G\|_2^2 + \|\mathbf{B}_2^\top \mathbf{u}_G\|_2^2 = \|\mathbf{B}_1 \mathbf{u}_G\|_2^2, \quad (15)$$

where the last equality is due to the fact that \mathbf{u}_G is curl-free, i.e., $\mathbf{B}_2^\top \mathbf{u}_G = \mathbf{0}$. Thus, eigenvalue λ_G is the squared ℓ_2 -norm of the divergence $\mathbf{B}_1 \mathbf{u}_G$ of the eigenvector edge flow \mathbf{u}_G . The magnitude of λ_G measures the extent of total divergence, i.e., the nodal variation. The gradient eigenvectors associated with a large eigenvalue have a large total divergence. If the SFT $\tilde{\mathbf{f}} = \mathbf{U}_1^\top \mathbf{f}$ of an edge flow has a large weight on such an eigenvector, we say that it contains a high gradient frequency, corresponding to its large divergence. We call any eigenvalue λ_G associated to the gradient eigenvectors \mathbf{U}_G a *gradient frequency* and collect them in the set $\mathcal{Q}_G = \{\lambda_{G,1}, \dots, \lambda_{G,N_G}\}$.

Curl frequency: For any unit norm curl eigenvector $\mathbf{u}_C \in \mathbf{U}_C$ associated to the curl space $\text{im}(\mathbf{B}_2)$, its corresponding eigenvalue follows

$$\lambda_C = \mathbf{u}_C^\top \mathbf{L}_1 \mathbf{u}_C = \|\mathbf{B}_1 \mathbf{u}_C\|_2^2 + \|\mathbf{B}_2^\top \mathbf{u}_C\|_2^2 = \|\mathbf{B}_2^\top \mathbf{u}_C\|_2^2, \quad (16)$$

where the last equality is due to the fact that \mathbf{u}_C is divergence-free, i.e., $\mathbf{B}_1 \mathbf{u}_C = \mathbf{0}$. Eigenvalue λ_C is the squared ℓ_2 -norm of the curl $\mathbf{B}_2^\top \mathbf{u}_C$ of the eigenvector \mathbf{u}_C . Thus, the magnitude of λ_C measures the extent of total curl, i.e., the rotational variation. Any eigenvector in the curl space corresponding to a large eigenvalue has a large total curl. If the SFT of an edge flow contains large weights on such eigenvectors, we say that it has a high curl frequency. We name any eigenvalue λ_C associated to the curl eigenvectors \mathbf{U}_C a *curl frequency* and collect them in the set $\mathcal{Q}_C = \{\lambda_{C,1}, \dots, \lambda_{C,N_C}\}$.

Harmonic frequency: The remaining eigenvalues λ_H are associated to the eigenvectors $\mathbf{u}_H \in \mathbf{U}_H$ which span the harmonic

space $\text{ker}(\mathbf{L}_1)$. They can be expressed as

$$\lambda_H = \mathbf{u}_H^\top \mathbf{L}_1 \mathbf{u}_H = \|\mathbf{B}_1 \mathbf{u}_H\|_2^2 + \|\mathbf{B}_2^\top \mathbf{u}_H\|_2^2 = 0, \quad (17)$$

because \mathbf{u}_H is harmonic, i.e., both divergence- and curl-free. If the SFT of an edge flow has only nonzeros at the harmonic frequencies (which are all zeros), then it is a harmonic flow. We collect the harmonic frequencies (all zeros) in the set $\mathcal{Q}_H = \{\lambda_{H,1}, \dots, \lambda_{H,N_H}\}$.

With these three types of simplicial frequencies, low and high frequency notions in an SC are only meaningful with respect to a certain type. A higher gradient (curl) frequency indicates respectively a larger nodal (rotational) variability. This is different from the frequency notion in discrete and graph signal processing. A zero simplicial frequency does not correspond to a constant edge flow but a globally conservative flow, i.e., divergence- and curl-free. Fig. 5 shows examples of different eigenvectors and the associated eigenvalues, which would be the analogous of the complex exponentials in discrete-time signal processing for the edge space in Fig. 1(a).

Simplicial embeddings: From Proposition 4 and three types of simplicial frequencies, we can interpret the SFT by the following diagonalization of \mathbf{L}_1

$$\mathbf{L}_1 = \mathbf{U}_1 \text{blkdiag}(\mathbf{\Lambda}_H, \mathbf{\Lambda}_G, \mathbf{\Lambda}_C) \mathbf{U}_1^\top \quad (18)$$

with $\mathbf{U}_1 = [\mathbf{U}_H \ \mathbf{U}_G \ \mathbf{U}_C]$ and $\text{blkdiag}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ a block-diagonal matrix containing the square matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ as diagonal blocks. Similarly, we have

$$\mathbf{L}_{1,\ell} = \mathbf{U}_1 \text{blkdiag}(\mathbf{0}, \mathbf{\Lambda}_G, \mathbf{0}) \mathbf{U}_1^\top \quad (19a)$$

$$\mathbf{L}_{1,u} = \mathbf{U}_1 \text{blkdiag}(\mathbf{0}, \mathbf{0}, \mathbf{\Lambda}_C) \mathbf{U}_1^\top \quad (19b)$$

for the lower and upper Laplacians, where $\mathbf{0}$ is an all-zero matrix of appropriate dimensions. Such insightful eigendecompositions enable us to define the following three embeddings of an edge flow $\mathbf{f} \in \mathbb{R}^{N_1}$,

$$\begin{cases} \tilde{\mathbf{f}}_H = \mathbf{U}_H^\top \mathbf{f} = \mathbf{U}_H^\top \mathbf{f}_H \in \mathbb{R}^{N_H}, & \text{harmonic embedding} \\ \tilde{\mathbf{f}}_G = \mathbf{U}_G^\top \mathbf{f} = \mathbf{U}_G^\top \mathbf{f}_G \in \mathbb{R}^{N_G}, & \text{gradient embedding} \\ \tilde{\mathbf{f}}_C = \mathbf{U}_C^\top \mathbf{f} = \mathbf{U}_C^\top \mathbf{f}_C \in \mathbb{R}^{N_C}, & \text{curl embedding.} \end{cases} \quad (20)$$

They follow from the orthogonality of the three components given by the Hodge decomposition. Equivalently, we can write the SFT of \mathbf{f} as $\tilde{\mathbf{f}} = [\tilde{\mathbf{f}}_H^\top, \tilde{\mathbf{f}}_G^\top, \tilde{\mathbf{f}}_C^\top]^\top$. Each entry of an embedding represents the weight the flow has on the corresponding eigenvector (simplicial Fourier basis vector), e.g., entry $[\tilde{\mathbf{f}}_G]_i$ is the SFT of \mathbf{f} at the i th gradient frequency $\lambda_{G,i}$. Such an embedding provides a compressed representation of the edge flow if they present a degree of sparsity [18] and allows us to differentiate different types of edge flow, e.g., to cluster trajectories and analyze ocean drift data [35], [36].

C. Filter Frequency Response

Upon defining the SFT, we can analyze the frequency response of the simplicial convolutional filter (3). From the diagonalizations (19a) and (20), we see that the lower simplicial shifting of an edge flow affects only the gradient component and the upper shifting affects only the curl component, i.e.,

$$\mathbf{L}_{1,\ell} \mathbf{f} = \mathbf{U}_G \mathbf{\Lambda}_G \tilde{\mathbf{f}}_G, \quad \mathbf{L}_{1,u} \mathbf{f} = \mathbf{U}_C \mathbf{\Lambda}_C \tilde{\mathbf{f}}_C. \quad (21)$$

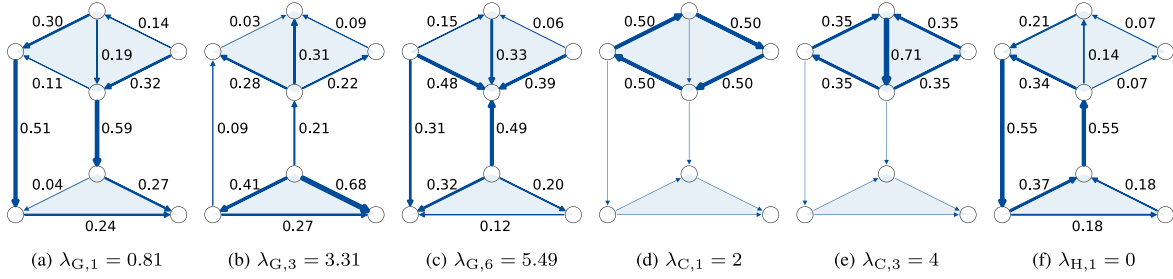


Fig. 5. Spectral analysis of the edge space in Fig. 1(a). The flow value (all numbers rounded to two decimal places) is indicated by the edge width and annotated next to the edge. It is zero if the edge is not annotated. If a flow orientation is opposite to the reference orientation, the corresponding flow value is negative. (a)-(c) The 1st, 3rd, and 6th eigenvectors in the gradient space \mathbf{U}_G with the corresponding gradient frequencies λ_G . The total divergence of the eigenvector increases with the eigenvalue. (d)-(e) The 1st and 3rd eigenvectors in the curl space \mathbf{U}_C with the corresponding curl frequencies λ_C . The total curl of the eigenvectors increases with the eigenvalue. (f) The only eigenvector in the harmonic space \mathbf{U}_H has frequency 0 and zero divergence and curl.

Through diagonalizing \mathbf{H}_1 by $\mathbf{U}_1 = [\mathbf{U}_H \ \mathbf{U}_G \ \mathbf{U}_C]$, we can find the frequency response of \mathbf{H}_1 as

$$\tilde{\mathbf{H}}_1 = \mathbf{U}_1^\top \mathbf{H}_1 \mathbf{U}_1 = \text{blkdiag} \left(\tilde{\mathbf{H}}_H, \tilde{\mathbf{H}}_G, \tilde{\mathbf{H}}_C \right), \quad (22)$$

where $\tilde{\mathbf{H}}_H = h_0 \mathbf{I}$, $\tilde{\mathbf{H}}_G = h_0 \mathbf{I} + \sum_{l_1=1}^{L_1} \alpha_{l_1} \Lambda_G^{l_1}$ and $\tilde{\mathbf{H}}_C = h_0 \mathbf{I} + \sum_{l_2=1}^{L_2} \beta_{l_2} \Lambda_C^{l_2}$. At an arbitrary frequency λ , the frequency response $\tilde{H}_1(\lambda)$ is given by

$$\begin{cases} \tilde{H}_H(\lambda) := h_0, & \text{for } \lambda \in \mathcal{Q}_H, \\ \tilde{H}_G(\lambda) := h_0 + \sum_{l_1=1}^{L_1} \alpha_{l_1} \lambda^{l_1}, & \text{for } \lambda \in \mathcal{Q}_G, \\ \tilde{H}_C(\lambda) := h_0 + \sum_{l_2=1}^{L_2} \beta_{l_2} \lambda^{l_2}, & \text{for } \lambda \in \mathcal{Q}_C, \end{cases} \quad (23)$$

which is the filter frequency response at the harmonic, gradient and curl frequencies, respectively. By the definition of spectral filtering, the filter \mathbf{H}_1 cannot distinguish the signal components belonging to the subspace spanned by the eigenvectors associated to an eigenvalue of multiplicity greater than one. For instance, the filter cannot respond differently to multiple harmonic components, but only scale them by a factor h_0 . In addition, we make the following three observations.

- 1) Filter \mathbf{H}_1 controls the different frequency types independently. The coefficient h_0 determines the harmonic frequency response and contributes to the whole simplicial spectrum. The coefficients α and β contribute only to the gradient and curl frequency response, respectively. This independent control on different signal subspaces corresponds to the different parameters imposed on the lower and upper adjacencies in the simplicial domain. In contrast, if setting $L_1 = L_2$ and $\alpha = \beta$ in \mathbf{H}_1 , the filter cannot regulate the gradient and curl spaces independently and has less flexibility.
- 2) The gradient frequency response is fully determined by the matrix polynomial in $\mathbf{L}_{1,\ell}$ [cf. (19a) and (22)]. Thus, if $L_2 = 0$, \mathbf{H}_1 has as responses h_0 for $\lambda \in \mathcal{Q}_H \cup \mathcal{Q}_C$, and $\tilde{H}_G = h_0 + \sum_{l_1=1}^{L_1} \alpha_{l_1} \lambda^{l_1}$, for $\lambda \in \mathcal{Q}_G$. This controls the gradient and non-gradient frequencies with a reduced design burden due to fewer parameters. But we give up the control on the curl frequencies. Likewise for the case of $L_1 = 0$, which is beneficial when only curl components need to be tuned.
- 3) At two *overlapping* frequencies $\lambda_1 = \lambda_2$ with $\lambda_1 \in \mathcal{Q}_G$, and $\lambda_2 \in \mathcal{Q}_C$, filter \mathbf{H}_1 responds differently as $\tilde{H}_G(\lambda_1)$

and $\tilde{H}_C(\lambda_2)$ [cf. (23)], respectively. This cannot be realized for the filter $\mathbf{H}_1 = \sum_{l=0}^L h_l \mathbf{L}_1^l$, i.e., setting $L_1 = L_2$ and $\alpha = \beta$, which follows $\tilde{H}_G(\lambda_1) = \tilde{H}_C(\lambda_2)$ instead. In the latter case, it will preserve the unwanted curl component at λ_2 when setting $\tilde{H}_G(\lambda) = 1$, for $\lambda \in \mathcal{Q}_G$, when the goal is to extract the gradient component.

V. FILTER DESIGN

Given a training set of input-output edge flow relations $\mathcal{T} = \{(\mathbf{f}_1, \mathbf{f}_{o,1}), \dots, (\mathbf{f}_{|\mathcal{T}|}, \mathbf{f}_{o,|\mathcal{T}|})\}$, we can learn the filter coefficients in a data-driven fashion by fitting the filtered output $\mathbf{H}_1 \mathbf{f}$ to the output \mathbf{f}_o . Specifically, consider a mean squared error (MSE) cost function and a regularizer $r(h_0, \alpha, \beta)$ to avoid overfitting, we formulate the problem as

$$\min_{h_0, \alpha, \beta} \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{f}_i, \mathbf{f}_{o,i}) \in \mathcal{T}} \|\mathbf{H}_1 \mathbf{f}_i - \mathbf{f}_{o,i}\|_2^2 + \gamma r(h_0, \alpha, \beta), \quad (24)$$

with $\gamma > 0$. A flow prediction based on (24) is detailed in [1].

In this section, we focus in detail on designing the simplicial filter given a desired frequency response. Specifically, we assume a desired frequency response g_0 at the harmonic frequency $\lambda = 0$, a gradient frequency response $g_G(\lambda)$ for $\lambda \in \mathcal{Q}_G$, and a curl frequency response $g_C(\lambda)$ for $\lambda \in \mathcal{Q}_C$. To design the coefficients h_0, α, β , our goal is then to approximate the desired response by the filter frequency response $\tilde{H}_1(\lambda)$ [cf. (23)], which can be formulated as

$$\begin{cases} h_0 \approx g_0, & \text{for } \lambda_i = 0, \\ h_0 + \sum_{l_1=1}^{L_1} \alpha_{l_1} \lambda_i^{l_1} \approx g_G(\lambda_i), & \text{for } \lambda_i \in \mathcal{Q}_G, \\ h_0 + \sum_{l_2=1}^{L_2} \beta_{l_2} \lambda_i^{l_2} \approx g_C(\lambda_i), & \text{for } \lambda_i \in \mathcal{Q}_C. \end{cases} \quad (25)$$

In the following, we first use a standard least-squares (LS) approach to solve (25). Later, we consider a universal design to avoid the eigenvalue computation when a continuous desired frequency response is given. In particular, we consider a grid-based and a Chebyshev polynomial approach.

A. Least-Squares Filter Design

Denote the number of distinct gradient frequencies in \mathcal{Q}_G by D_G , and that of distinct curl frequencies in \mathcal{Q}_C by D_C . Then, the three sets of equations in (25) contain respectively one, D_G and D_C distinct linear equations. Let $\mathbf{g} = [g_0, \mathbf{g}_G^\top, \mathbf{g}_C^\top]^\top$ collect the desired responses at distinct frequencies where

$[\mathbf{g}_G]_i = g_G(\lambda_{G,i})$, for $i = 1 \dots, D_G$, is the response at the i th distinct gradient frequency $\lambda_{G,i}$, and $[\mathbf{g}_C]_i = g_C(\lambda_{C,i})$, for $i = 1 \dots, D_C$, at the i th distinct curl frequency. Then, we can obtain the filter coefficients by solving the LS problem

$$\min_{h_0, \alpha, \beta} \left\| \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \Phi_G & \mathbf{0} \\ \mathbf{0} & \Phi_C \end{bmatrix} \begin{bmatrix} h_0 \\ \alpha \\ \beta \end{bmatrix} - \mathbf{g} \right\|_2, \quad (26)$$

where $\mathbf{1}$ ($\mathbf{0}$) is an all-one (all-zero) matrix or vector of an appropriate dimension, $\Phi_G \in \mathbb{R}^{D_G \times L_1}$ and $\Phi_C \in \mathbb{R}^{D_C \times L_2}$ are Vandermonde matrices with respective entries $[\Phi_G]_{ij} = \lambda_{G,i}^j$ and $[\Phi_C]_{ij} = \lambda_{C,i}^j$. We refer to problem (26) as the *LS design*, which can be solved either with a direct solver or with a decoupled solver, studied as follows.

Direct LS design: From the Cayley-Hamilton theorem [37] and [4, Thm. 3], we know that any analytical function of a matrix can be expressed as a matrix polynomial of degree less than its minimal polynomial degree, which equals the number of distinct eigenvalues for a positive semi-definite matrix. Thus, we can assume the filter orders $L_1 \leq D_G$ and $L_2 \leq D_C$. Under this condition, problem (26) admits a unique solution which can be obtained via the pseudo-inverse of the system matrix. Furthermore, when $L_1 = D_G$ and $L_2 = D_C$, Φ_G and Φ_C are square and any two rows in them are linearly independent, the solution leads to a zero cost in (26). We refer to this pseudo-inverse solution of (26) as the *direct LS design*.

In addition, given a desired edge operator \mathbf{G} , the following proposition states that it can be implemented by filter \mathbf{H}_1 .

Proposition 5: A desired linear operator $\mathbf{G} \in \mathbb{R}^{N_1 \times N_1}$ can be perfectly implemented by a simplicial filter \mathbf{H}_1 if the following three conditions hold true:

- i) Matrices \mathbf{G} and \mathbf{L}_1 are simultaneously diagonalizable, i.e., \mathbf{U}_1 forms an eigenvector basis for \mathbf{G} . Let $\mathbf{g} = [\mathbf{g}_H^\top, \mathbf{g}_G^\top, \mathbf{g}_C^\top]^\top$ collect the eigenvalues of \mathbf{G} .
- ii) If two eigenvalues of \mathbf{L}_1 are of the same frequency type and equal, the corresponding eigenvalues of \mathbf{G} are also equal. For $\lambda_{H,i} = \lambda_{H,j} = 0 \in \mathcal{Q}_H$, it holds that $[\mathbf{g}_H]_i = [\mathbf{g}_H]_j$, for $\lambda_{G,i} = \lambda_{G,j} \in \mathcal{Q}_G$, $[\mathbf{g}_G]_i = [\mathbf{g}_G]_j$, and for $\lambda_{C,i} = \lambda_{C,j} \in \mathcal{Q}_C$, $[\mathbf{g}_C]_i = [\mathbf{g}_C]_j$.
- iii) The filter orders of \mathbf{H}_1 fulfill $L_1 \geq D_G$, $L_2 \geq D_C$.

Proof: See Appendix E. \blacksquare

Decoupled LS design: We can reduce the complexity of solving (26) by decoupling the cost function for different frequency types. First, we rewrite problem (26) as

$$\min_{h_0, \alpha, \beta} \left\| \begin{bmatrix} \mathbf{1} & \Phi_G \end{bmatrix} \begin{bmatrix} h_0 \\ \alpha \end{bmatrix} - \mathbf{g}_G \right\|_2^2 + \left\| \begin{bmatrix} \mathbf{1} & \Phi_C \end{bmatrix} \begin{bmatrix} h_0 \\ \beta \end{bmatrix} - \mathbf{g}_C \right\|_2^2 + \|h_0 - g_0\|_2^2. \quad (27)$$

To approximate a solution, we ignore the dependence of the first two terms in (27) on coefficient h_0 and solve the last term separately to estimate h_0 . We then substitute the estimate \hat{h}_0 in the first two terms to obtain $\hat{\alpha}$ and $\hat{\beta}$, given by

$$\hat{h}_0 = g_0, \quad \hat{\alpha} = \Phi_G^\dagger (\mathbf{g}_G - g_0 \mathbf{1}), \quad \hat{\beta} = \Phi_C^\dagger (\mathbf{g}_C - g_0 \mathbf{1}), \quad (28)$$

which, referred to as the *decoupled LS design*, is suboptimal compared to the direct LS design. The following proposition discusses this suboptimality.

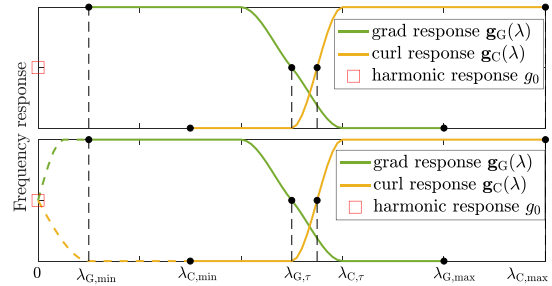


Fig. 6. An example of a continuous frequency response for the grid-based (top) and Chebyshev (bottom) filter design, which promotes low gradient frequencies and the high curl frequencies. The cut-off frequencies are subscripted by τ . The harmonic frequency response is given at frequency 0. For the Chebyshev polynomial design, we require that the gradient and curl are continuous functions starting from frequency 0 and $g_C(0) = g_G(0) = g_0$.

Proposition 6: The decoupled LS design (28) converges to the direct solution of (26) as $\|\Phi_G \Phi_G^\dagger - \mathbf{I}\|_F \rightarrow 0$ and $\|\Phi_C \Phi_C^\dagger - \mathbf{I}\|_F \rightarrow 0$ where $\|\cdot\|_F$ is the Frobenius norm.

Proof: See Appendix F. \blacksquare

Proposition 6 states that as the pseudo-inverses of Φ_G and Φ_C become closer to the true inverses, the decoupled solution converges to the direct solution. Moreover, the decoupled LS design reduces the computational cost to $\mathcal{O}(D_G^3 + D_C^3)$ from $\mathcal{O}((1 + D_G + D_C)^3)$ for the direct one. However, both designs require the computation of the eigenvalues of the Hodge Laplacians, which takes in general a computational complexity of $\mathcal{O}(N_1^3)$ [38]. In the sequel, we consider a universal design strategy to avoid the eigenvalue computation, specifically, a grid-based design and a Chebyshev polynomial design [39].

B. Grid-Based Filter Design

The grid-based filter design aims to match the desired frequency response in a continuous interval where the exact frequencies lie such that the eigenvalue computation of \mathbf{L}_1 can be avoided. Given a harmonic frequency response g_0 , a continuous gradient frequency response $g_G(\lambda)$, $\lambda \in [\lambda_{G,\min}, \lambda_{G,\max}]$ and a continuous curl frequency response $g_C(\lambda)$, $\lambda \in [\lambda_{C,\min}, \lambda_{C,\max}]$, we want that

$$\begin{cases} h_0 - g_0 \approx 0 \\ \int_{\lambda_{G,\min}}^{\lambda_{G,\max}} |h_0 + \sum_{l_1=1}^{L_1} \alpha_{l_1} \lambda^{l_1} - g_G(\lambda)|^2 d\lambda \approx 0 \\ \int_{\lambda_{C,\min}}^{\lambda_{C,\max}} |h_0 + \sum_{l_2=1}^{L_2} \beta_{l_2} \lambda^{l_2} - g_C(\lambda)|^2 d\lambda \approx 0, \end{cases} \quad (29)$$

which is a continuous version of (25). An example of the continuous frequency response is given in Fig. 6 (top).

By sampling M_1 and M_2 (grid)-points uniformly from the intervals $[\lambda_{G,\min}, \lambda_{G,\max}]$ and $[\lambda_{C,\min}, \lambda_{C,\max}]$, the problem (29) can then be formulated as an LS problem of form (26) but with the sampled frequencies as the entries of the system matrix instead of the true eigenvalues. We can again solve this LS problem either via a direct pseudo-inverse of the system matrix or via the decoupled solution method [cf. (28)]. Notice that the largest true eigenvalue can be approximated by efficient algorithms, e.g., power iteration, [38], [40]. For the smallest, we can set a small value greater than 0 as the lower bound.

C. Chebyshev Polynomial Filter Design

Both discussed filter designs rely on solving an LS problem, which has a Vandermonde matrix as the system matrix, and suffers from numerical instability. To tackle this issue, we consider a Chebyshev polynomial based filter design [39], [41]. As illustrated in Fig. 6 (bottom), consider a continuous gradient frequency response $g_G(\lambda)$, $\lambda \in [0, \lambda_{G,\max}]$ and a continuous curl frequency response $g_C(\lambda)$, $\lambda \in [0, \lambda_{C,\max}]$. We further require that $g_G(0) = g_C(0) = g_0$. That is, the continuous functions $g_G(\lambda)$ and $g_C(\lambda)$ are defined starting from frequency 0, at which they are equal to the harmonic frequency response g_0 .

As the filter \mathbf{H}_1 is a sum of matrix polynomials of $\mathbf{L}_{1,\ell}$ and $\mathbf{L}_{1,u}$, our strategy is to first consider the Chebyshev polynomial design for each of them so to separately obtain the gradient and curl frequency responses, then sum these two polynomials to obtain the final filter. However, one type of frequency response could be affected unwantedly by the identity matrix term in the Chebyshev polynomial designed for the other frequency response, as detailed later. The requirement that $g_G(0) = g_C(0) = g_0$ allows a possible correction.

First, we approximate the gradient frequency response via a truncated series of shifted Chebyshev polynomials $\mathbf{H}_\ell := \mathbf{H}_\ell(\mathbf{L}_{1,\ell})$. Let $\bar{P}_l(\lambda)$, $\lambda \in [-1, 1]$ be the l th Chebyshev polynomial of the first kind [42]. We perform a transformation $P_l(\lambda) := \bar{P}_l(\frac{\lambda-\omega}{\omega})$ with $\omega := \frac{\lambda_{G,\max}}{2}$ to shift the domain to $[0, \lambda_{G,\max}]$. We then approximate the operator $g_G(\mathbf{L}_{1,\ell})$ that has the gradient frequency response $g_G(\lambda)$ by \mathbf{H}_ℓ of order L_1

$$\mathbf{H}_\ell = \frac{1}{2}c_{\ell,0}\mathbf{I} + \sum_{l_1=1}^{L_1} c_{\ell,l_1}P_{l_1}(\mathbf{L}_{1,\ell}) \quad (30)$$

where we have $P_0(\mathbf{L}_{1,\ell}) = \mathbf{I}$, $P_1(\mathbf{L}_{1,\ell}) = \frac{2}{\lambda_{G,\max}}\mathbf{L}_{1,\ell} - \mathbf{I}$, the l_1 th Chebyshev term, for $l_1 \geq 2$, is

$$P_{l_1}(\mathbf{L}_{1,\ell}) = 2P_{l_1-1}(\mathbf{L}_{1,\ell})P_{l_1-2}(\mathbf{L}_{1,\ell}) - P_{l_1-2}(\mathbf{L}_{1,\ell}), \quad (31)$$

and the Chebyshev coefficients c_{ℓ,l_1} can be computed as

$$c_{\ell,l_1} = \frac{2}{\pi} \int_0^\pi \cos(l_1\phi) g_G(\omega(\cos\phi + 1)) d\phi. \quad (32)$$

The frequency response $\tilde{H}_\ell(\lambda)$ of \mathbf{H}_ℓ can be found as

$$\begin{cases} p_{\ell,0} := \frac{1}{2}c_{\ell,0} + \sum_{l_1=1}^{\lfloor L_1/2 \rfloor} (c_{\ell,2l_1} - c_{\ell,2l_1-1}), & \text{for } \lambda \in \mathcal{Q}_H \cup \mathcal{Q}_C \\ \tilde{H}_{\ell,G}(\lambda) := \frac{1}{2}c_{\ell,0} + \sum_{l_1=1}^{L_1} c_{\ell,l_1}P_{l_1}(\lambda), & \text{for } \lambda \in \mathcal{Q}_G, \end{cases} \quad (33)$$

with coefficient $p_{\ell,0}$ on the identity term of \mathbf{H}_ℓ , which is the frequency response at the harmonic and curl frequencies, associated to the kernel of $\mathbf{L}_{1,\ell}$ [cf. (19a)]. For a reasonably large L_1 we have $p_{\ell,0} \approx g_0$ and $\tilde{H}_{\ell,G}(\lambda) \approx g_G(\lambda)$, $\lambda \in \mathcal{Q}_G$.

Second, to approximate the curl frequency response $g_C(\lambda)$, we follow the same procedure [cf. (30)-(32)] to obtain the Chebyshev polynomial $\mathbf{H}_u := \mathbf{H}_u(\mathbf{L}_{1,u})$ of order L_2

$$\mathbf{H}_u = \frac{1}{2}c_{u,0}\mathbf{I} + \sum_{l_2=1}^{L_2} c_{u,l_2}P_{l_2}(\mathbf{L}_{1,u}). \quad (34)$$

It has a frequency response $\tilde{H}_u(\lambda)$

$$\begin{cases} p_{u,0} := \frac{1}{2}c_{u,0} + \sum_{l_2=1}^{\lfloor L_2/2 \rfloor} (c_{u,2l_2} - c_{u,2l_2-1}), & \text{for } \lambda \in \mathcal{Q}_H \cup \mathcal{Q}_G \\ \tilde{H}_{u,C}(\lambda) := \frac{1}{2}c_{u,0} + \sum_{l_2=1}^{L_2} c_{u,l_2}P_{l_2}(\lambda), & \text{for } \lambda \in \mathcal{Q}_C. \end{cases} \quad (35)$$

with coefficient $p_{u,0}$ on the identity term of \mathbf{H}_u , which is the frequency response at the harmonic and gradient frequencies, associated to the kernel of $\mathbf{L}_{1,u}$ [cf. (19b)]. For a reasonably large L_2 we have $p_{u,0} \approx g_0$ and $\tilde{H}_{u,C}(\lambda) \approx g_C(\lambda)$, $\lambda \in \mathcal{Q}_C$.

Lastly, by summing \mathbf{H}_ℓ and \mathbf{H}_u , we obtain a filter that approximates the gradient and curl frequency responses. However, from (33), we see that \mathbf{H}_ℓ generates a response $p_{\ell,0}$ at both harmonic and curl frequencies. This will lift up the curl frequency response unwantedly by $p_{\ell,0}$. Similarly, \mathbf{H}_u has the effect of lifting the gradient frequency response by $p_{u,0}$ [cf. (35)]. By requiring that $g_G(0) = g_C(0) = g_0$, we can remove this unwanted influence by subtracting a term $g_0\mathbf{I}$ from the summation. Hence, the final Chebyshev polynomial design \mathbf{H}_1 of orders L_1 and L_2 is given by

$$\mathbf{H}_1 = \mathbf{H}_\ell + \mathbf{H}_u - g_0\mathbf{I}, \quad (36)$$

which has a frequency response $\tilde{H}_1(\lambda)$

$$\begin{cases} p_{\ell,0} + p_{u,0} - g_0, & \text{for } \lambda \in \mathcal{Q}_H \\ \tilde{H}_{1,G}(\lambda) + p_{u,0} - g_0, & \text{for } \lambda \in \mathcal{Q}_G \\ \tilde{H}_{1,C}(\lambda) + p_{\ell,0} - g_0, & \text{for } \lambda \in \mathcal{Q}_C. \end{cases} \quad (37)$$

The following proposition states that the approximation error of the Chebyshev polynomial design (36) is bounded.

Proposition 7: Let \mathbf{G} be the desired operator corresponding to the continuous gradient and curl frequency responses $g_G(\lambda)$ with $\lambda \in [0, \lambda_{G,\max}]$ and $g_C(\lambda)$ with $\lambda \in [0, \lambda_{C,\max}]$, as well as the harmonic one $g_G(0) = g_C(0) = g_0$. Let \mathbf{H}_1 [cf. (36)] be a truncated series of Chebyshev polynomials of orders L_1 and L_2 with frequency response $\tilde{H}_1(\lambda)$ [cf. (37)]. Define

$$\begin{aligned} B_1(L_1) &:= \sup_{\lambda \in [0, \lambda_{G,\max}]} \{ |\tilde{H}_\ell(\lambda) - g_0 - g_G(\lambda)| \}, \\ B_2(L_2) &:= \sup_{\lambda \in [0, \lambda_{C,\max}]} \{ |\tilde{H}_u(\lambda) - g_0 - g_C(\lambda)| \}, \end{aligned} \quad (38)$$

and $B := \max\{B_1(L_1), B_2(L_2)\}$. Then, we have that

$$\|\mathbf{G} - \mathbf{H}_1\|_2 := \max_{\mathbf{f} \neq \mathbf{0}} \frac{\|(\mathbf{G} - \mathbf{H}_1)\mathbf{f}\|_2}{\|\mathbf{f}\|_2} \leq B \quad (39)$$

Proof: See Appendix G. ■

When $g_G(\cdot)$ and $g_C(\cdot)$ are real analytic, a stronger bound can be found [39], [43]. In addition, we make the following three comments. i) When only the gradient frequency response $g_G(\lambda)$ is of interest, we can directly consider the Chebyshev polynomial \mathbf{H}_ℓ [cf. (30)] with $L_2 = 0$. Likewise we consider \mathbf{H}_u when only $g_C(\lambda)$ is of interest. ii) If a frequency response $g(\lambda)$ is given on the whole spectrum, we can build a filter $\mathbf{H} = \sum_{l=0}^L h_l \mathbf{L}_1^l$ based on a Chebyshev polynomial design analogous to the graph filter case [39]. iii) The Chebyshev polynomial design requires no eigenvalue computation of \mathbf{L}_1 . Thus, it does not suffer from numerical instability, allows to build simplicial filters with large L_1 and L_2 for an accurate design, and admits a recursive and distributed implementation due to the Chebyshev polynomial property (31).

VI. APPLICATIONS

In this section, we first discuss how to use a simplicial filter for subcomponent extraction and edge flow denoising. We then consider analyses of financial markets, street and traffic networks. These are similar to previous works [8], [9], [18], [44], [45], but we directly use our here developed filters instead of employing a regularized optimization problem (which implicitly defines a low-pass filter).

To gauge the performance in estimation tasks, we use the normalized root mean square error (NRMSE), $e = \|\hat{\mathbf{f}} - \mathbf{f}_0\|_2 / \|\mathbf{f}_0\|_2$ with the flow estimate $\hat{\mathbf{f}}$ and the true flow \mathbf{f}_0 . For filter design problems, we evaluate the spectral norm $\|\mathbf{H}_1 - \mathbf{G}\|_2$ with the designed filter \mathbf{H}_1 and the true operator \mathbf{G} . The Chebfun toolbox was used for the Chebyshev polynomial filter design [46].

A. Subcomponent Extraction

In pairwise ranking problems, we aim to rank alternatives by comparing their scores. The work [9] modeled the score differences between alternatives as edge flows on a pairwise comparison graph. The gradient component of these flows gives a global ranking and the curl components measures the inconsistency of the ranking. For further examples see [10], [11], [12]. A common approach to obtain the three components of the flow is to compute [9], [16], [18]:

$$\hat{\mathbf{f}}_G = \mathbf{P}_G \mathbf{f}, \quad \hat{\mathbf{f}}_C = \mathbf{P}_C \mathbf{f}, \quad \hat{\mathbf{f}}_H = \mathbf{P}_H \mathbf{f} = \mathbf{f} - \hat{\mathbf{f}}_G - \hat{\mathbf{f}}_C. \quad (40)$$

where $\mathbf{P}_G = \mathbf{B}_1^\top (\mathbf{B}_1 \mathbf{B}_1^\top)^\dagger \mathbf{B}_1$ is the projection onto the gradient space, the curl projector is $\mathbf{P}_C = \mathbf{B}_2 (\mathbf{B}_2^\top \mathbf{B}_2)^\dagger \mathbf{B}_2^\top$ and the harmonic projector $\mathbf{P}_H = \mathbf{I} - \mathbf{L}_1 \mathbf{L}_1^\dagger$. Notably, we can use a (polynomial) simplicial filter \mathbf{H}_1 to implement these operators, too.

Lemma 1: The projection operators (40) are equivalent to $\mathbf{P}_G = \mathbf{U}_G \mathbf{U}_G^\top$, $\mathbf{P}_C = \mathbf{U}_C \mathbf{U}_C^\top$ and $\mathbf{P}_H = \mathbf{U}_H \mathbf{U}_H^\top$. As \mathbf{U} requires the knowledge of global properties in this form the projections cannot be computed in a distributed way.

However, for $L_1 = D_G$, $L_2 = D_C$, there exists a unique $\{h_0, \alpha, \beta\}$ such that $\mathbf{H}_1 = \mathbf{P}_G$. These coefficients can be found by solving the system (26) with $g_0 = 0$, $\mathbf{g}_C = \mathbf{0}$ and $\mathbf{g}_G = \mathbf{1}$; analogous arguments lead to distributed implementations of $\mathbf{H}_1 = \mathbf{P}_C$ and $\mathbf{H}_1 = \mathbf{P}_H$.

Proof: See Appendix H. ■

For the gradient and curl components, this can be simplified.

Corollary 1: For the gradient projector \mathbf{P}_G , there exist a filter with $L_1 = D_G$, $L_2 = 0$ and a unique $\{h_0, \alpha\}$ such that $\mathbf{H}_1 = \mathbf{P}_G$. The solution is $h_0 = 0$ and $\alpha = \Phi_G^{-1} \mathbf{1}$. For the curl projector \mathbf{P}_C , there exist a filter with $L_1 = 0$, $L_2 = D_C$ and a unique $\{h_0, \beta\}$ such that $\mathbf{H}_1 = \mathbf{P}_C$. The solution is $h_0 = 0$ and $\beta = \Phi_C^{-1} \mathbf{1}$.

Proof: See Appendix I. ■

Corollary 1 shows a gradient projector can be build solely upon $\mathbf{L}_{1,\ell}$, since $\mathbf{L}_{1,u}$ has no effect in the gradient space [cf. (14) and (21)]. Similar arguments hold for the curl projector.

Fig. 7 reports the performance of the subcomponent extraction based on \mathbf{H}_1 via an LS design. We generated a synthetic edge flow $\mathbf{f} = \mathbf{U}_1 \mathbf{f}$ with a flat spectrum $\hat{\mathbf{f}} = \mathbf{1}$ on the SC in Fig. 1(a). We observe that as the filter order increases, the filter performs better as its expressive power increases. Setting $L_1 = L_2$ with $\alpha = \beta$ reduces the expressive power as expected. If filter orders obey $L_1 \geq 6$ and $L_3 \geq 3$, then the gradient (or curl) component

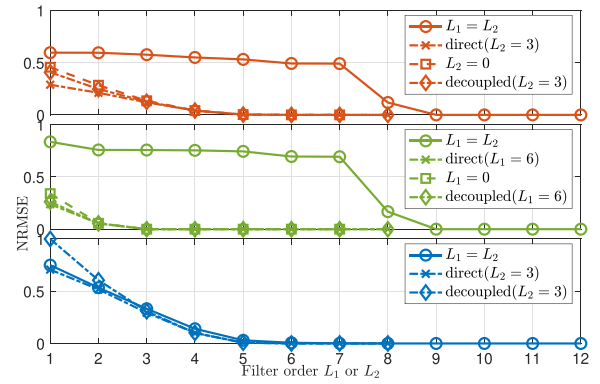


Fig. 7. Subcomponent extraction performance by filters \mathbf{H}_1 with different parameters based on an LS design. The extraction becomes better as the filter order increases. For the gradient and curl components, setting $L_1 = L_2$ and $\alpha = \beta$ worsens the performance because of the limited expressive power. For the general filter form, the direct and decoupled LS designs have smaller performance difference as the filter order grows (see Proposition 6).

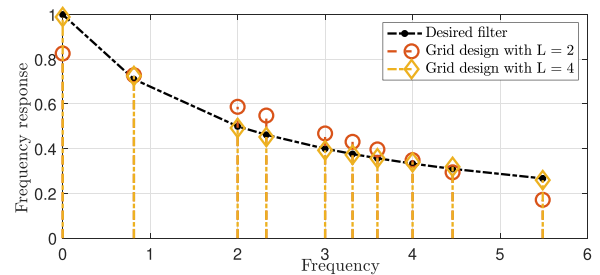


Fig. 8. Frequency responses of the denoising filter $\mathbf{H}_P = (\mathbf{I} + 0.5\mathbf{L}_1)^{-1}$ based on the grid design with different filter orders.

can be perfectly extracted, as shown in Fig. 4 and indicated by Lemma 1 and Corollary 1. In the general parameter setting, the decoupled LS design performs closer to the direct LS design as the filter order grows as shown in Proposition 6.

B. Edge Flow Denoising

Consider a noisy edge flow $\mathbf{f} = \mathbf{f}^0 + \epsilon$ with \mathbf{f}^0 the true edge flow and ϵ a zero-mean white Gaussian noise. To obtain an estimate $\hat{\mathbf{f}}$, we can solve the regularized optimization problem [8], [19]:

$$\min_{\hat{\mathbf{f}}} \|\hat{\mathbf{f}} - \mathbf{f}\|_2^2 + \mu \hat{\mathbf{f}}^\top \mathbf{P} \hat{\mathbf{f}}, \quad (41)$$

with an optimal solution $\hat{\mathbf{f}} = \mathbf{H}_P \mathbf{f} := (\mathbf{I} + \mu \mathbf{P})^{-1} \mathbf{f}$. For matrix \mathbf{P} we have two choices: (i) the edge Laplacian $\mathbf{L}_{1,\ell} = \mathbf{B}_1^\top \mathbf{B}_1$, leading to a regularizer $\|\mathbf{B}_1 \hat{\mathbf{f}}\|_2^2$ to penalize the flows with a nonzero divergence [8]; (ii) the Hodge Laplacian \mathbf{L}_1 , leading to a regularizer $\mathbf{f} \mathbf{L}_1 \hat{\mathbf{f}} = \|\mathbf{B}_1 \hat{\mathbf{f}}\|_2^2 + \|\mathbf{B}_2 \hat{\mathbf{f}}\|_2^2$ to penalize the flows with a nonzero divergence or curl [19].

Operator \mathbf{H}_P has the frequency response $\tilde{H}_P(\lambda_i) = 1/(1 + \mu \lambda_i)$ with (i) $\lambda_i = 0$ or $\lambda_i \in \mathcal{Q}_G$ for $\mathbf{P} = \mathbf{L}_{1,\ell}$, (ii) $\lambda_i = 0$ or $\lambda_i \in \mathcal{Q}_G \cup \mathcal{Q}_C$ for $\mathbf{P} = \mathbf{L}_1$. Thus, \mathbf{H}_P is a low pass filter which suppresses either the gradient frequencies or the non-harmonic frequencies. We can implement a simplicial filter to approximate \mathbf{H}_P . Fig. 8 shows the frequency responses of the filter \mathbf{H}_P with $\mathbf{P} = \mathbf{L}_1$, $\mu = 0.5$ based on the grid-based design, for which we considered 10 samples in the frequency interval $[0, 5.488]$ and

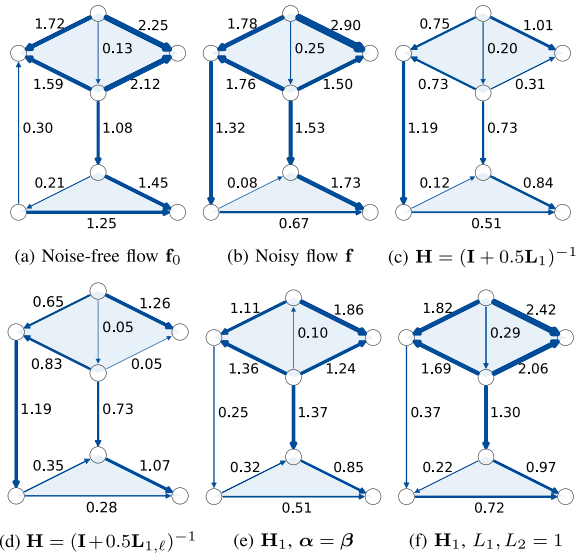


Fig. 9. Gradient flow denoising. (a) Edge flow \mathbf{f}_0 induced by a node signal with a flat spectrum. (b) The noisy observation \mathbf{f} with error $e = 0.46$. (c-d) Denoising with the low-pass filter \mathbf{H}_P with (c) $\mathbf{P} = \mathbf{L}_1$ [19], or (d) $\mathbf{P} = \mathbf{L}_{1,\ell}$ [8] leads to an even larger error of $e = 0.70$ or $e = 0.73$, respectively. (e) Denoising by a gradient based simplicial filter \mathbf{H}_1 with an order $L_1 = L_2 = 4$ and $\alpha = \beta$, yields a much better result with error $e = 0.39$. (f) Denoising by a general filter \mathbf{H}_1 with $\alpha \neq \beta$ provides an even smaller error with $e = 0.23$, even for a lower filter order $L_1 = L_2 = 1$.

the maximal eigenvalue is estimated with 50 steps of power iterations. The grid-based filter design errors compared to using the true eigenvalues are negligible, 0.023 and 0.004 for $L = 2$ and $L = 4$, respectively.

Note that the above optimization framework relies on the assumption that the true edge flow is either divergence-free or harmonic, which is not always true for real-world flows [9], [23]. When different spectral properties of the underlying edge flow are known, or certain spectral properties of noise are known, we are able to deal with various situations for denoising by properly designing the simplicial filters. To illustrate this we induced a gradient flow from a node signal with a flat spectrum and added Gaussian noise with an error of 0.46, shown in Figs. 9(a) and 9(b). Fig. 9 contrasts the denoising results of the regularized optimization methods with our filters that preserve the gradient component based on an LS design (Corollary 1). As Figs. 9(c) to 9(f), shown in this context the low-pass filters which are implicitly defined via the optimization procedures lead to large errors. In contrast the simplicial filters that preserve the gradient can denoise properly. Setting $\alpha = \beta$ again reduces the performance.

C. Currency Exchange Market

A currency exchange market can be described as a network where the vertices represent currencies, and the edges indicate the pairwise exchange rates. If all pairs of currencies are exchangeable, the vertex set \mathcal{V} and edge set \mathcal{E} make up a complete graph. For any currencies $i, j, k \in \mathcal{V}$, we expect an arbitrary-free condition, $r^{i/j}r^{j/k} = r^{i/k}$ with the exchange rate $r^{i/j}$ between i and j , i.e., the exchange path $i \rightarrow j \rightarrow k$ provides no gain or loss over a direct exchange $i \rightarrow k$. If we represent the exchange rates as edge flows $f_{ij} = \log(r^{i/j})$, this can be translated into the fact that \mathbf{f} is curl-free, i.e., $f_{ij} + f_{jk} + f_{ki} = 0$. Therefore,

TABLE I
CURRENCY EXCHANGE RATES CAPTURED FROM YAHOO!FINANCE, NOT ARBITRAGE-FREE

	USD	EUR	CNY	HKD	GBP	JPY	AUD
1 USD	1	0.8422	6.3739	7.7666	0.7207	110.1020	1.3377
1 EUR	1.1873	1	7.5681	9.2218	0.8557	130.7314	1.5883
1 CNY	0.1539	0.1321	1	1.2185	0.1131	17.2683	0.2099
1 HKD	0.1288	0.1085	0.8207	1	0.0928	14.1718	0.1723
1 GBP	1.3871	1.1685	8.8414	10.7732	1	152.6758	1.8557
1 JPY	0.0091	0.0077	0.0579	0.0706	0.0066	1	0.0122
1 AUD	0.7475	0.6299	4.7602	5.8001	0.5385	82.1837	1

TABLE II
THE GRADIENT COMPONENT, ARBITRAGE-FREE, PROVIDES A FAIR MARKET

	USD	EUR	CNY	HKD	GBP	JPY	AUD
1 USD	1	0.8422	6.3738	7.7665	0.7208	110.0171	1.3385
1 EUR	1.1874	1	7.5680	9.2216	0.8559	130.6292	1.5893
1 CNY	0.1569	0.1321	1	1.2185	0.1131	17.2608	0.2100
1 HKD	0.1288	0.1084	0.8207	1	0.0928	14.1656	0.1723
1 GBP	1.3873	1.1684	8.8425	10.7746	1	152.6286	1.8557
1 JPY	0.0091	0.0077	0.0579	0.0706	0.0066	1	0.0122
1 AUD	0.7471	0.6292	4.7618	5.8022	0.5385	82.1919	1

an ideal exchange edge flow is a gradient flow. This idea was exploited in [9], [23] to assess arbitrage possibilities in exchange markets, and provide arbitrage free exchange rates, respectively.

Here, we illustrate how we can analogously remove arbitrage opportunities via a simplicial filter that preserves only the gradient component of a given exchange rate flow. For a complete graph, there are two distinct eigenvalues, zero and N_0 , for the lower or upper Hodge Laplacian. Then, based on Corollary 1 we can extract the gradient component via $\mathbf{H}_1 = \frac{1}{N_0}\mathbf{L}_{1,\ell}$. Similarly, filter $\mathbf{H}_1 = \frac{1}{N_0}\mathbf{L}_{1,u}$ can extract the curl component, which indicate possible arbitrage opportunities.

In Table I, we show a real-world exchange market of seven currencies at 2021/07/12 10:30 UTC from the Currency Converter Yahoo!Finance. We built an SC formed by the seven currencies and all the 2-, 3-cliques, where the edge flow \mathbf{f} is the logarithm of the exchange rates in the upper triangular part without the diagonal in Table I. If one unit of currency yields more than 0.3% benefit or loss after two successive exchanges, we say the corresponding exchange rates are non-arbitrage-free. By computing the curl $\mathbf{B}_2^j \mathbf{f}$, we can identify six such triangles (up to machine precision) that are not arbitrage-free., e.g., USD-JPY-AUD (1 USD would yield 1.0041 USD), EUR-JPY-AUD, and HKD-GBP-JPY. By applying a filter $\mathbf{H}_1 = \frac{1}{N_0}\mathbf{L}_{1,\ell}$ on the exchange rate flows, we can extract its gradient flow, leading to the arbitrage-free exchange rate flow in Table II. This yields essentially the same results as solving the LS optimization problem considered in [9]. This simple example demonstrates the use of simplicial filters to generate an efficient financial market. Especially in a complete market, the form of the filters is trivial and the computational cost is much smaller compared to solving the LS problems.

D. London Street Network: Fast PageRank of Edges

PageRank, as a ranking scheme for web pages, can be studied in terms of a random walk on a graph, which can be used to measure the centrality of a node. PageRank was extended to the edge space to assess the topological importance of an edge in [35]. The input edge flow \mathbf{f} is an indicator vector

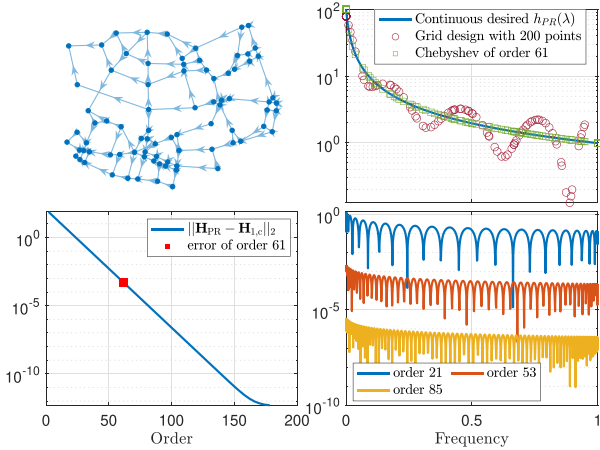


Fig. 10. PageRank analysis of the street network of London. Top left: network illustration. Top right: Frequency responses of the grid-based designed filter of order 9 and the Chebyshev filter of order 61 w.r.t. the desired continuous frequency response. Bottom left: Spectral norm error of the Chebyshev polynomial filter of different orders. Bottom right: Continuous frequency response errors of the Chebyshev filter of different orders.

which has value one on the edge of interest and zeros on the rest, then a PageRank vector π follows the linear system [35, Def. 6.2], $(\gamma\mathbf{I} + \mathbf{L}_{1,n})\pi = \mathbf{f}$, with $\mathbf{L}_{1,n}$ being the normalized 1-Hodge Laplacian¹ [35, Def. 3.3] and $\gamma > 0$. The solution is $\pi = (\gamma\mathbf{I} + \mathbf{L}_{1,n})^{-1}\mathbf{f}$ with the PageRank operator $\mathbf{H}_{\text{PR}} := (\gamma\mathbf{I} + \mathbf{L}_{1,n})^{-1}$, which does not require to construct the edge space random walk matrix [35, Thm. 3.4] compared to a power iteration implementation. For an indicator edge flow \mathbf{f} of edge i , the absolute values of the entries of π are the influence measures edge i has on the edges and the signs the influence orientations w.r.t. the reference orientations [35]. Furthermore, the overall importance of an edge can be assessed with the ℓ_2 -norm $\|\pi\|_2$ of its PageRank vector π . By extracting the gradient component π_G , we can study the importance of this edge w.r.t. the gradient space via its ℓ_2 -norm $\|\pi_G\|_2$ or relative norm $\frac{\|\pi_G\|_2}{\|\pi\|_2}$; likewise for the curl and harmonic components.

Experiment setup: The operator \mathbf{H}_{PR} can be interpreted as a low pass filter which attenuates the high gradient and curl frequencies. But to implement it we need to invert a matrix. Instead, we propose here a faster variant via a simplicial filter $\mathbf{H}_1 := \mathbf{H}_1(\mathbf{L}_{1,n})$ built on the normalized Hodge Laplacian. To find the filter coefficients, we considered a grid-based design and a Chebyshev polynomial with a desired response, $g(\lambda) = \frac{1}{\gamma+\lambda}$ with $\lambda \in [0, 1]$ and $\gamma = 0.01$.

We implemented the PageRank operator in the street network of London with 82 crossings (nodes), 130 streets (edges) and 12 triangles, as shown in the top left of Fig. 10 [44], [45]. We considered a grid-based design with 200 samples within the eigenvalue interval and a filter order of 9 and implemented the Chebyshev polynomials \mathbf{H}_1 of different orders.

Results: From Fig. 10, we see that the performance of the grid-based design decreases heavily when the filter order is larger than 9 due to the numerical instability, while a Chebyshev design allows a more accurate design as shown in Fig. 10.

We then computed the PageRank results of all edges with a Chebyshev filter of order 61 and obtained the norms of their three

¹ $\mathbf{L}_{1,n}$ admits a similarity transformation to its symmetric version, so its eigenvalues are real and carry the simplicial frequency notion.

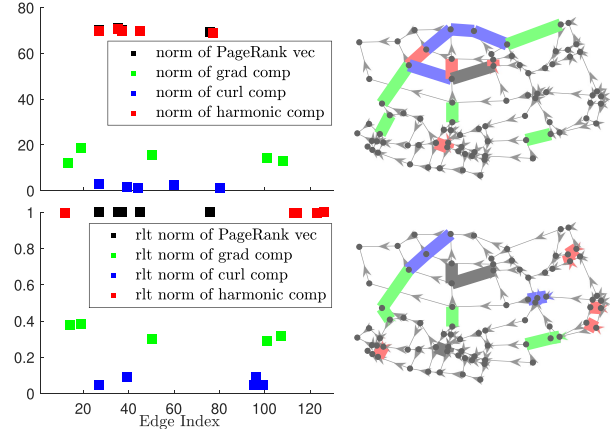


Fig. 11. PageRank vectors analysis. The scattered squares are the edges whose PageRank vectors contain the top-five largest absolute (Top) and relative (Bottom) total norm, gradient, curl and harmonic norms. The left figures show the PageRank values w.r.t. edge indices. The right figures show the highlighted edges in the network with shaded grey indicating the total norm, green the gradient norm, blue the curl norm and red the harmonic norm.

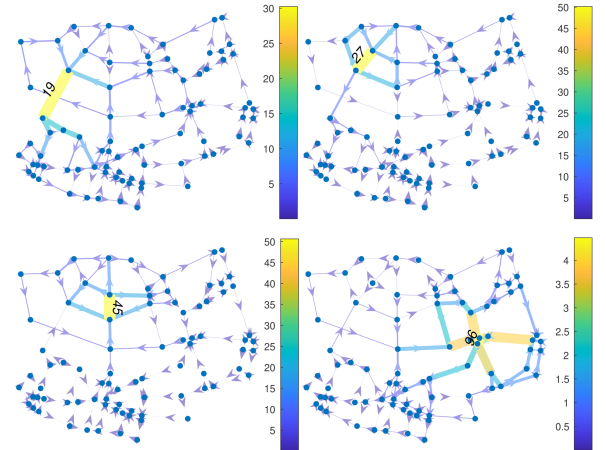


Fig. 12. Examples of the PageRank vectors of four edges. The edge width and color indicates the magnitude of the PageRank result on that edge. The labeled edges are the chosen edges, also with the largest PageRank results.

components in the absolute and relative senses. From Fig. 11, we can identify the most influential streets (dark grey) in the network, as the indicator flow on these streets induce a PageRank vector with the largest total norm. The streets (green) that have the biggest influence on the gradient space are the ones on which a traffic change leads to congestion on the intersections as the traffic flows on them have a large divergence. The red streets induce the most impact on the harmonic space and the blues ones on the curl space, where the traffic flows tend to induce a global or local cyclic flow, thus a small chance of congestion. The influences are measured in a relative sense in the bottom figures, and we notice that most streets would not cause a large influence on the curl space.

Finally, we show the simplicial PageRank vector of four edges to assess where their influences are concentrated in Fig. 12. Edge 19 (top left), sitting in the gradient space, has a large influence in terms of gradient flow components on the surrounding edges to which congestion on edge 19 would spread, as shown in Fig. 11. Edge 27 (top right) has a large influence on edges that form 1-dimensional “hole” [16],

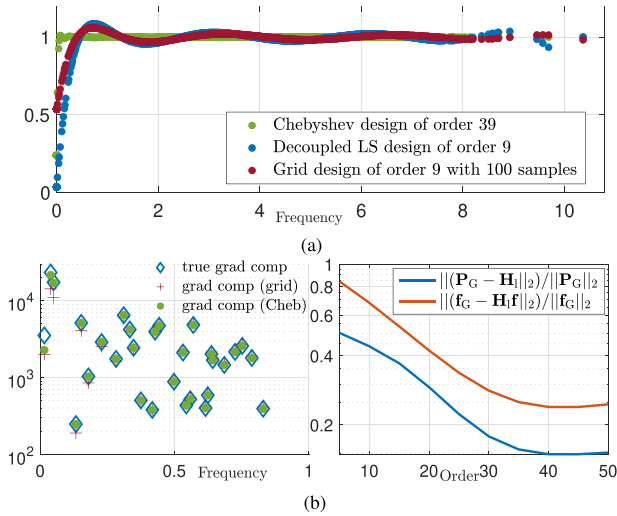


Fig. 13. Chicago road network gradient component extraction. (a): Filter frequency responses with different designs. (b): Left: SFT of the extracted gradient component in frequency range $[0,1]$. Right: Approximation errors of Chebyshev filters of different orders and the extracted gradient component.

containing mostly harmonic components. This may imply that a traffic change on edge 27 would less likely cause congestion. Edge 45 (bottom left), whose PageRank result has a large total norm, as seen in Fig. 11, acts similar as edge 27. Edge 96 (bottom right) induces smaller influences than the other three, but they reach further in the network. The most influenced edges are its direct upper neighbors, as also seen in the bottom (blue) of Fig. 11, where congestion would rather not happen.

E. Chicago Road Network: Gradient Component Extraction

We now conduct the subcomponent extraction on a real-world larger network. On the Chicago road network with 546 nodes, 1088 edges, and 112 triangles [47], we perform the gradient component extraction of the measured traffic flow, which is not divergence-free, via filter \mathbf{H}_1 built on the lower Hodge Laplacian. It is challenging to perform the filter design in this setting because some simplicial frequencies are close to each other, leading to the ill-conditioning of the LS design. This can be avoided by the Chebyshev polynomial design. This requires a continuous desired frequency response to perform the gradient component extraction, which ideally is an indicator function $\mathbf{1}_{\lambda>0}$ with $\lambda \in [0, \lambda_{G,\max}]$. Here we use the logistic function $g_G(\lambda) = \frac{1}{1+\exp^{-k(\lambda-\lambda_0)}}$ with the growth rate $k > 0$ and the midpoint λ_0 . If the smallest gradient frequency is close to 0, a large k and a small λ_0 are required to achieve a good approximation of the ideal indicator function.

We applied different filter design methods. For the LS-based methods, we set a filter order of 9 to avoid ill-conditioning and we considered the decoupled solver. Moreover, we treated the eigenvalues with a difference smaller than 0.3 as the same for the LS design, leading to 30 “different” eigenvalues. For the grid-based design, we uniformly sampled 100 points in the interval $[0, \lambda_{G,\max}]$ with $\lambda_{G,\max} = 10.8$ approximated by a 50-step power iteration. Lastly, we set $k = 100$ and $\lambda_0 = 0.01$ for the logistic function in the Chebyshev polynomial design.

As seen in Fig. 13(a), the Chebyshev polynomial of an order 39 only has one frequency response smaller than 0.9 at the smallest gradient frequency, while at the remaining

frequencies, it is able to well preserve the gradient component. The other methods have a poorer performance especially at small gradient frequencies. We then compared the gradient component extracted by above grid-based and Chebyshev polynomial filters. Fig. 13(b) (left) reports the SFT of the extracted flows at frequencies smaller than 1. The Chebyshev polynomial has a good extraction ability as it performs well even at the very small frequencies where the grid-based design fails. Fig. 13(b) (right) shows the filter design and the extraction errors of the Chebyshev polynomial of different orders. The extraction error cannot be further reduced because the traffic flow contains large components at the small frequencies and the logistic function, after all, is an approximate of the indicator function.

VII. CONCLUSION

We proposed a simplicial convolutional filter as a matrix polynomial of the Hodge Laplacians for simplicial signal processing. It generates an output as a linear combination of the shifted simplicial signals. This shift-and-sum operation is analogous to the convolutions of time series, images and graph signals and allows for a distributed filter implementation. In the frequency domain, the filter acts as a pointwise multiplication respecting the convolution theorem. Furthermore, the lower and upper Hodge Laplacians encode lower and upper adjacencies, respectively. For an edge flow, its lower shifting propagates the flow to its neighbors via the common incident nodes, while the upper one via the common triangles. By assigning two different sets of coefficients on the lower and upper parts in the filter, we can differentiate the lower and upper adjacencies. Via the Hodge decomposition, we see that this corresponds to an independent control of the filter on the gradient and curl spaces in the frequency domain. To achieve a desired frequency response, different filter design approaches are considered with pros and cons. The filter provides a faster and distributed solution for subcomponent extraction, simplicial signal denoising and other tasks in exploiting the higher-order connectivities of the network.

APPENDIX A

PROOF OF PROPOSITION 1

Due to the distributivity of matrix-vector multiplication, we have $\mathbf{L}_{1,\ell}(a\mathbf{f}_1 + b\mathbf{f}_2) = a\mathbf{L}_{1,\ell}\mathbf{f}_1 + b\mathbf{L}_{1,\ell}\mathbf{f}_2$, and $\mathbf{L}_{1,u}(a\mathbf{f}_1 + b\mathbf{f}_2) = a\mathbf{L}_{1,u}\mathbf{f}_1 + b\mathbf{L}_{1,u}\mathbf{f}_2$. Then, by working out the expression $\mathbf{H}_1(a\mathbf{f}_1 + b\mathbf{f}_2)$ and using the distributivity w.r.t the addition, the linearity proof completes. Since we have $\mathbf{L}_{1,\ell}^l \mathbf{L}_{1,\ell} = \mathbf{L}_{1,\ell} \mathbf{L}_{1,\ell}^l$, and similarly for $\mathbf{L}_{1,u}$, and $\mathbf{L}_{1,\ell} \mathbf{L}_{1,u} = \mathbf{0}$, the shift-invariance holds. The same proof applies to the general case with $k \neq 1$.

APPENDIX B

PROOF OF PROPOSITION 2

Since the permutation matrix \mathbf{P}_k is orthogonal, i.e., $\mathbf{P}_k \mathbf{P}_k^\top = \mathbf{P}_k^\top \mathbf{P}_k = \mathbf{I}$, we have that $(\mathbf{P}_k \mathbf{L}_{k,\ell} \mathbf{P}_k^\top)^{l_1} = \mathbf{P}_k \mathbf{L}_{k,\ell}^{l_1} \mathbf{P}_k^\top$, and similarly $(\mathbf{P}_k \mathbf{L}_{k,u} \mathbf{P}_k^\top)^{l_2} = \mathbf{P}_k \mathbf{L}_{k,u}^{l_2} \mathbf{P}_k^\top$. Thus, we can express the permuted simplicial filter as

$$\begin{aligned} \bar{\mathbf{H}}_k &= h_0 \mathbf{I} + \sum_{l_1=1}^{L_1} \alpha_{l_1} (\mathbf{P}_k \mathbf{L}_{k,\ell} \mathbf{P}_k^\top)^{l_1} + \sum_{l_2=1}^{L_2} \beta_{l_2} (\mathbf{P}_k \mathbf{L}_{k,u} \mathbf{P}_k^\top)^{l_2} \\ &= h_0 \mathbf{I} + \sum_{l_1=1}^{L_1} \alpha_{l_1} \mathbf{P}_k \mathbf{L}_{k,\ell}^{l_1} \mathbf{P}_k^\top + \sum_{l_2=1}^{L_2} \beta_{l_2} \mathbf{P}_k \mathbf{L}_{k,u}^{l_2} \mathbf{P}_k^\top \end{aligned}$$

$$\begin{aligned}
&= \mathbf{P}_k \left(h_0 \mathbf{I} + \sum_{l_1=1}^{L_1} \alpha_{l_1} \mathbf{L}_{k,\ell}^{l_1} + \sum_{l_2=1}^{L_2} \beta_{l_2} \mathbf{L}_{k,u}^{l_2} \right) \mathbf{P}_k^\top \\
&= \mathbf{P}_k \mathbf{H}_k \mathbf{P}_k^\top.
\end{aligned} \tag{42}$$

The output on the permuted SC can be written as $\bar{\mathbf{s}}_o^k := \bar{\mathbf{H}}_k \bar{\mathbf{s}}^k = \bar{\mathbf{H}}_k (\mathbf{P}_k \mathbf{s}^k) = \mathbf{P}_k \bar{\mathbf{H}}_k \mathbf{P}_k^\top \mathbf{P}_k \mathbf{s}^k = \mathbf{P}_k \mathbf{H}_k \mathbf{s}^k := \mathbf{P}_k \mathbf{s}_o^k$. The proof completes.

APPENDIX C

PROOF OF PROPOSITION 3

The diagonal matrix \mathbf{D}_k satisfies that $\mathbf{D}_k \mathbf{D}_k^\top = \mathbf{D}_k^\top \mathbf{D}_k = \mathbf{I}$. Following from that, we have that $(\mathbf{D}_k \mathbf{L}_{k,\ell} \mathbf{D}_k^\top)^{l_1} = \mathbf{D}_k \mathbf{L}_{k,\ell}^{l_1} \mathbf{D}_k^\top$, and similarly $(\mathbf{D}_k \mathbf{L}_{k,u} \mathbf{D}_k^\top)^{l_2} = \mathbf{D}_k \mathbf{L}_{k,u}^{l_2} \mathbf{D}_k^\top$. Following the same procedure in (42), we have

$$\begin{aligned}
\bar{\mathbf{H}}_k &= h_0 \mathbf{I} + \sum_{l_1=1}^{L_1} \alpha_{l_1} (\mathbf{D}_k \mathbf{L}_{k,\ell} \mathbf{D}_k^\top)^{l_1} + \sum_{l_2=1}^{L_2} \beta_{l_2} (\mathbf{D}_k \mathbf{L}_{k,u} \mathbf{D}_k^\top)^{l_2} \\
&= \mathbf{D}_k \mathbf{H}_k \mathbf{D}_k^\top.
\end{aligned} \tag{43}$$

Thus, the filter output on the reoriented simplices can be expressed as $\bar{\mathbf{s}}_o^k := \bar{\mathbf{H}}_k \bar{\mathbf{s}}^k = \bar{\mathbf{H}}_k (\mathbf{D}_k \mathbf{s}^k) = \mathbf{D}_k \mathbf{H}_k \mathbf{s}^k := \mathbf{D}_k \mathbf{s}_o^k$. The proof completes.

APPENDIX D

PROOF OF PROPOSITION 4

We show the proof for each item.

1) First, we show that the image of $\mathbf{L}_{1,\ell}$ is equivalent to the gradient space $\text{im}(\mathbf{B}_1^\top)$. (i) To show $\text{im}(\mathbf{L}_{1,\ell}) \subseteq \text{im}(\mathbf{B}_1^\top)$: From $\mathbf{L}_{1,\ell} = \mathbf{B}_1^\top \mathbf{B}_1$, for any non-zero $\mathbf{x} \in \text{im}(\mathbf{L}_{1,\ell})$, we have $\mathbf{x} = \mathbf{L}_{1,\ell} \mathbf{y}$, and we can always find a vector $\mathbf{z} = \mathbf{B}_1 \mathbf{y} \in \mathbb{R}^{N_0}$ such that $\mathbf{x} = \mathbf{B}_1^\top \mathbf{z}$; (ii) To show $\text{im}(\mathbf{B}_1^\top) \subseteq \text{im}(\mathbf{L}_{1,\ell})$: for every non-zero $\mathbf{x} \in \text{im}(\mathbf{B}_1^\top)$, we can find some $\mathbf{y} \perp \ker(\mathbf{B}_1^\top)$ such that $\mathbf{x} = \mathbf{B}_1^\top \mathbf{y} \neq \mathbf{0}$. This implies $\mathbf{y} \in \text{im}(\mathbf{B}_1)$, so there exists some $\mathbf{z} \in \mathbb{R}^{N_1}$ such that $\mathbf{y} = \mathbf{B}_1 \mathbf{z}$ and $\mathbf{x} = \mathbf{B}_1^\top \mathbf{B}_1 \mathbf{z} = \mathbf{L}_{1,\ell} \mathbf{z}$, and hence $\text{im}(\mathbf{B}_1^\top) \subseteq \text{im}(\mathbf{L}_{1,\ell})$. Combining (i) and (ii), we have that $\text{im}(\mathbf{L}_{1,\ell}) = \text{im}(\mathbf{B}_1^\top)$.

Second, we show that the eigenvectors \mathbf{U}_G of $\mathbf{L}_{1,\ell}$ associated with nonzero eigenvalues span the image of $\mathbf{L}_{1,\ell}$. As $\mathbf{L}_{1,\ell}$ is positive semidefinite (PSD), thus, diagonalizable, the geometric multiplicity of every eigenvalue equals to the algebraic multiplicity. That is, all the eigenvectors are linearly independent and form an eigenbasis. Then, matrix \mathbf{U}_G has a full column rank. Furthermore, for any $\mathbf{x} \in \text{im}(\mathbf{L}_{1,\ell})$, we have $\mathbf{x} = \mathbf{L}_{1,\ell} \mathbf{y} = \mathbf{U}_G \mathbf{A}_G \mathbf{U}_G^\top \mathbf{y}$, i.e., $\mathbf{x} \in \text{im}(\mathbf{U}_G)$ and $\text{im}(\mathbf{L}_{1,\ell}) \subseteq \text{im}(\mathbf{U}_G)$. Due to $\dim \text{im}(\mathbf{L}_{1,\ell}) = \text{rank}(\mathbf{L}_{1,\ell}) = N_G$, matrix \mathbf{U}_G spans the image of $\mathbf{L}_{1,\ell}$ and the gradient space $\text{im}(\mathbf{B}_1^\top)$.

2) The proof of 2) follows similarly as the proof of 1).

3) For arbitrary eigenvectors \mathbf{u}_G in \mathbf{U}_G and \mathbf{u}_C in \mathbf{U}_C , we have $\mathbf{u}_G = \frac{1}{\lambda_G} \mathbf{L}_{1,\ell} \mathbf{u}_G$ and $\mathbf{u}_C = \frac{1}{\lambda_C} \mathbf{L}_{1,u} \mathbf{u}_C$. Thus, from (1), their inner product follows $\mathbf{u}_G^\top \mathbf{u}_C = \frac{1}{\lambda_G \lambda_C} \mathbf{u}_G^\top \mathbf{L}_{1,\ell} \mathbf{L}_{1,u} \mathbf{u}_C = 0$, i.e., $\mathbf{U}_G \perp \mathbf{U}_C$. Moreover, from the definition of \mathbf{L}_1 , matrix $[\mathbf{U}_G \ \mathbf{U}_C]$ contains the eigenvectors of \mathbf{L}_1 associated with nonzero eigenvalues. The Hodge decomposition indicates that $\text{im}(\mathbf{B}_1^\top) \oplus \text{im}(\mathbf{B}_2) = \text{im}(\mathbf{L}_1)$. By combining with 1) and 2), we have that $\text{im}(\mathbf{B}_1^\top) = \text{im}(\mathbf{U}_G)$ and $\text{im}(\mathbf{B}_2) = \text{im}(\mathbf{U}_C)$.

4) As \mathbf{L}_1 is PSD, the eigenvectors associated to zero eigenvalues are linearly independent. Any vector $\mathbf{x} \in \ker(\mathbf{L}_1)$ follows $\mathbf{L}_1 \mathbf{x} = \mathbf{0}$, which means \mathbf{x} is an eigenvector associated with an eigenvalue 0, i.e., $\ker(\mathbf{L}_1) = \text{im}(\mathbf{U}_H)$ with dimension N_H . Moreover, we have $\ker(\mathbf{L}_1) = \ker(\mathbf{L}_{1,\ell}) \cap \ker(\mathbf{L}_{1,u})$ from the definition of \mathbf{L}_1 , then the columns of \mathbf{U}_H can be used as eigenvectors of $\mathbf{L}_{1,\ell}$ or $\mathbf{L}_{1,u}$ associated with zero eigenvalues. From 3) and $\ker(\mathbf{L}_1) = \text{im}(\mathbf{U}_H)$, we have $\mathbf{U}_H \perp \mathbf{U}_G$ and $\mathbf{U}_H \perp \mathbf{U}_C$. Thus, matrix $[\mathbf{U}_H \ \mathbf{U}_C]$ (or $[\mathbf{U}_H \ \mathbf{U}_G]$) provides the eigenvectors of $\mathbf{L}_{1,\ell}$ (or $\mathbf{L}_{1,u}$) associated with zero eigenvalues, and the proof completes.

5) From 3) and 4), we have that matrix \mathbf{U}_1 collects all eigenvectors of \mathbf{L}_1 . From 1), 2), and 4), we have that \mathbf{U}_1 provides all eigenvectors for $\mathbf{L}_{1,\ell}$ and $\mathbf{L}_{1,u}$.

APPENDIX E

PROOF OF PROPOSITION 5

With condition i), we can eigendecompose the operator \mathbf{G} as $\text{diag}(\mathbf{g}) = \mathbf{U}_1^\top \mathbf{G} \mathbf{U}_1$, then the equivalence between \mathbf{G} and \mathbf{H}_1 can be achieved through a set of linear equations in the spectral domain, i.e., $H_1(\lambda_i) = g_i$, for all $i = 1, \dots, N_1$. Based on condition ii), this set of linear equations is equivalent to linear system (26) and (27) with $1 + D_G + D_C$ equations. With the filter order requirement in condition iii), the Vandermonde matrices Φ_G and Φ_C have full row rank. Thus, there exist at least one solution to problem (26) and the proof completes.

APPENDIX F

PROOF OF PROPOSITION 6

The cost function J in problem (27) is convex w.r.t. variables, h_0 , α and β . Thus, we could find the optimality condition by setting the gradients of the cost function w.r.t. the three variables to zeros, given by

$$\begin{cases} \nabla_{h_0} J = h_0 - g_0 + (h_0 \mathbf{1} + \Phi_G \alpha - \mathbf{g}_G)^\top \mathbf{1} \\ \quad \quad \quad + (h_0 \mathbf{1} + \Phi_C \beta - \mathbf{g}_C)^\top \mathbf{1} = 0, \\ \nabla_{\alpha} J = \Phi_G^\top (h_0 \mathbf{1} + \Phi_G \alpha - \mathbf{g}_G) = \mathbf{0}, \\ \nabla_{\beta} J = \Phi_C^\top (h_0 \mathbf{1} + \Phi_C \beta - \mathbf{g}_C) = \mathbf{0}. \end{cases} \tag{44}$$

First, consider the case where we have that $\|\Phi_G \Phi_G^\dagger - \mathbf{I}\|_F = 0$ and $\|\Phi_C \Phi_C^\dagger - \mathbf{I}\|_F = 0$, i.e., $\Phi_G \Phi_G^\dagger = \mathbf{I}$ and $\Phi_C \Phi_C^\dagger = \mathbf{I}$, then the solution (28) results in that $(\hat{h}_0 \mathbf{1} + \Phi_G \hat{\alpha} - \mathbf{g}_G) = \mathbf{0}$ and $(\hat{h}_0 \mathbf{1} + \Phi_C \hat{\beta} - \mathbf{g}_C) = \mathbf{0}$, which satisfies the optimality condition (44).

Second, consider the general case that $\|\Phi_G \Phi_G^\dagger - \mathbf{I}\|_F \neq 0$ and $\|\Phi_C \Phi_C^\dagger - \mathbf{I}\|_F \neq 0$. By substituting the solution (28) into the optimality condition (44), we have

$$\begin{cases} \nabla_{h_0} J = \left((\Phi_G \Phi_G^\dagger - \mathbf{I})(\mathbf{g}_G - g_0 \mathbf{1}) \right)^\top \mathbf{1} \\ \quad \quad \quad + \left((\Phi_C \Phi_C^\dagger - \mathbf{I})(\mathbf{g}_C - g_0 \mathbf{1}) \right)^\top \mathbf{1}, \\ \nabla_{\alpha} J = \Phi_G^\top (\Phi_G \Phi_G^\dagger - \mathbf{I})(\mathbf{g}_G - g_0 \mathbf{1}), \\ \nabla_{\beta} J = \Phi_C^\top (\Phi_C \Phi_C^\dagger - \mathbf{I})(\mathbf{g}_C - g_0 \mathbf{1}). \end{cases} \tag{45}$$

For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with singular values σ_i , $i = 1, \dots, \min\{m, n\}$, we have that $\|\mathbf{A}\|_F = \left(\sum_{i=1}^{\min\{m, n\}} \sigma_i^2 \right)^{\frac{1}{2}}$. If

it holds $\|\Phi_G \Phi_G^\dagger - \mathbf{I}\|_F \rightarrow 0$ and $\|\Phi_C \Phi_C^\dagger - \mathbf{I}\|_F \rightarrow 0$, i.e., the Frobenius norms approach to zero, the number of trivial (zero) singular values of $\Phi_G \Phi_G^\dagger - \mathbf{I}$ and $\Phi_C \Phi_C^\dagger - \mathbf{I}$ increases. Accordingly, the number of trivial entries in $(\Phi_G \Phi_G^\dagger - \mathbf{I})(\mathbf{g}_G - g_0 \mathbf{1})$ and $(\Phi_C \Phi_C^\dagger - \mathbf{I})(\mathbf{g}_C - g_0 \mathbf{1})$ increases, which corresponds to a suboptimal condition of (44), i.e., the gradients $\nabla_{h_0} J(\hat{h}_0, \hat{\alpha}, \hat{\beta}) \rightarrow 0$, $\nabla_{\alpha} J(\hat{h}_0, \hat{\alpha}) \rightarrow \mathbf{0}$ and $\nabla_{\beta} J(\hat{h}_0, \hat{\beta}) \rightarrow \mathbf{0}$. The proof completes.

APPENDIX G

PROOF OF PROPOSITION 7

Since the operator \mathbf{G} corresponds to the desired continuous harmonic, gradient and curl frequency responses, it can be diagonalized by \mathbf{U}_1 . Therefore, we have that

$$\begin{aligned} \|\mathbf{G} - \mathbf{H}_c\|_2 &= \|\mathbf{U}_1(g(\boldsymbol{\Lambda}) - \tilde{H}_1(\boldsymbol{\Lambda}))\mathbf{U}_1^\top\|_2 \\ &= \|g(\boldsymbol{\Lambda}) - \tilde{H}_1(\boldsymbol{\Lambda})\|_2 = \max_{i=1, \dots, N_1} |g(\lambda_i) - \tilde{H}_1(\lambda_i)| \end{aligned} \quad (46)$$

where the diagonal matrix $g(\boldsymbol{\Lambda})$ has entries $g(\lambda_i) = g_0$ for $\lambda_i \in \mathcal{Q}_H$, $g(\lambda_i) = g_G(\lambda_i)$ for $\lambda_i \in \mathcal{Q}_G$ and $g(\lambda_i) = g_C(\lambda_i)$ for $\lambda_i \in \mathcal{Q}_C$. The frequency response $\tilde{H}_1(\lambda_i)$ for $\lambda_i \in \mathcal{Q}$ is given in (37). Moreover, based on the definition of $B_1(L_1)$ and $B_2(L_2)$ we have that

$$\max_{i=1, \dots, N_1} |g(\lambda_i) - \tilde{H}_1(\lambda_i)| \leq \max\{B_1(L_1), B_2(L_2)\} = B. \quad (47)$$

The proof completes.

APPENDIX H

PROOF OF LEMMA 1

We first show the equivalence between the two projection operator forms. From the Hodge decomposition, the gradient, the curl and the harmonic components are in the subspaces $\text{im}(\mathbf{B}_1^\top)$, $\text{im}(\mathbf{B}_2)$ and $\text{ker}(\mathbf{L}_1)$, respectively. Furthermore, from Proposition 4, we have that $\text{im}(\mathbf{B}_1^\top) = \text{im}(\mathbf{U}_G)$, $\text{im}(\mathbf{B}_2) = \text{im}(\mathbf{U}_C)$ and $\text{ker}(\mathbf{L}_1) = \text{im}(\mathbf{U}_H)$. Thus, each subcomponent can be obtained as the orthogonal projection of \mathbf{f} onto the subspace spanned by the eigenbasis, i.e., the LS estimate. The gradient projection is $\mathbf{P}_G := \mathbf{U}_G \mathbf{U}_G^\top$, the curl one $\mathbf{P}_C := \mathbf{U}_C \mathbf{U}_C^\top$ and the harmonic one $\mathbf{P}_H := \mathbf{U}_H \mathbf{U}_H^\top$. This can be shown via the SFT as well, i.e., $\mathbf{f}_G = \mathbf{U}_G \tilde{\mathbf{f}}_G = \mathbf{U}_G \mathbf{U}_G^\top \mathbf{f}$, likewise for the other two.

Second, the simplicial filter \mathbf{H}_1 can implement the gradient projector $\mathbf{P}_G = \mathbf{U}_G \mathbf{U}_G^\top$, if and only if we have that

$$\mathbf{H}_1 = \mathbf{U}_G \mathbf{U}_G^\top = [\mathbf{U}_G^\perp \ \mathbf{U}_G] \begin{bmatrix} \mathbf{0} & \\ & \mathbf{I}_{N_G} \end{bmatrix} \begin{bmatrix} (\mathbf{U}_G^\perp)^\top \\ \mathbf{U}_G^\top \end{bmatrix}, \quad (48)$$

with $\mathbf{U}_G^\perp = [\mathbf{U}_H \ \mathbf{U}_C]$. We have that $\mathbf{H}_1 = \mathbf{U}_1 \tilde{\mathbf{H}}_1 \mathbf{U}_1^\top$ from (22). Thus, (48) is equivalent to problem (26) with $g_0 = 0$, $\mathbf{g}_C = \mathbf{0}$ and $\mathbf{g}_G = \mathbf{1}$. With $L_1 = D_G$ and $L_2 = D_C$, there admits a unique solution $\{h_0, \alpha, \beta\}$ to system (26). Similar procedure can be followed for the implementation of the curl and harmonic projectors. The proof completes.

APPENDIX I

PROOF OF COROLLARY 1

To implement the projector \mathbf{P}_G via \mathbf{H}_1 with $L_2 = 0$ ($\beta = \mathbf{0}$), from (48), it is equivalent to set $g_0 = 0$ for $\lambda_i \in \mathcal{Q}_H \cup \mathcal{Q}_C$ and $g(\lambda_i) = 1$ for $\lambda_i \in \mathcal{Q}_G$ in (23), i.e., $\mathbf{g} = [0 \ \mathbf{1}_{D_G}^\top]^\top$. This returns problem (26) without Φ_C and β , i.e.,

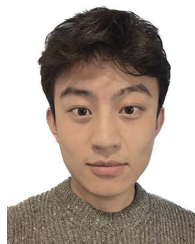
$$\min_{h_0, \alpha} \left\| \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \Phi_G \end{bmatrix} \begin{bmatrix} h_0 \\ \alpha \end{bmatrix} - \mathbf{g} \right\|_2^2. \quad (49)$$

If $L_1 = D_G$, the system matrix is square and any two rows are linearly independent, it admits a unique solution of $h_0 = 0$ and $\alpha = \Phi_G^{-1} \mathbf{1}$. Similar procedure can be followed for the curl projector \mathbf{P}_C . The proof completes.

REFERENCES

- [1] M. Yang, E. Isufi, M. T. Schaub, and G. Leus, "Finite impulse response filters for simplicial complexes," in *Proc. 29th Eur. Signal Process. Conf.*, 2021, pp. 2005–2009.
- [2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
- [3] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [4] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
- [5] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.
- [6] W. Huang, T. A. Bolton, J. D. Medaglia, D. S. Bassett, A. Ribeiro, and D. Van De Ville, "A graph signal processing perspective on functional brain imaging," *Proc. IEEE*, vol. 106, no. 5, pp. 868–885, May 2018.
- [7] K. K. Leung, W. A. Massey, and W. Whitt, "Traffic models for wireless communication networks," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 8, pp. 1353–1364, Oct. 1994.
- [8] M. T. Schaub and S. Segarra, "Flow smoothing and denoising: Graph signal processing in the edge-space," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2018, pp. 735–739.
- [9] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, "Statistical ranking and combinatorial Hodge theory," *Math. Program.*, vol. 127, no. 1, pp. 203–244, 2011.
- [10] O. Candogan, I. Menache, A. Ozdaglar, and P. A. Parrilo, "Flows and decompositions of games: Harmonic and potential games," *Math. Operations Res.*, vol. 36, no. 3, pp. 474–503, 2011.
- [11] T. Gebhart, X. Fu, and R. J. Funk, "Go with the flow? A large-scale analysis of health care delivery networks in the United States using Hodge theory," in *Proc. IEEE Int. Conf. Big Data*, 2021, pp. 3812–3823.
- [12] A. Mock and I. Volić, "Political structures and the topology of simplicial complexes," *Math. Social Sci.*, vol. 114, pp. 39–57, 2021.
- [13] W. Huang and A. Ribeiro, "Metrics in the space of high order networks," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 615–629, Feb. 2016.
- [14] C. Bick, H. A. Harrington, E. Gross, and M. T. Schaub, "What are higher-order networks?," submitted for publication.
- [15] A. Patania, G. Petri, and F. Vaccarino, "The shape of collaborations," *EPJ Data Sci.*, vol. 6, pp. 1–16, 2017.
- [16] L.-H. Lim, "Hodge Laplacians on graphs," *SIAM Rev.*, vol. 62, no. 3, pp. 685–715, 2020.
- [17] L. J. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*, vol. 3. Berlin, Germany: Springer, 2010.
- [18] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Trans. Signal Process.*, vol. 68, pp. 2992–3007, 2020.
- [19] M. T. Schaub, Y. Zhu, J.-B. Seby, T. M. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin' on the edge... and beyond," *Signal Process.*, vol. 187, 2021, Art. no. 108149.
- [20] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing*. Boca Raton, FL, USA: CRC Press, 2010.

- [21] L. Wasserman, "Topological data analysis," *Annu. Rev. Statist. Appl.*, vol. 5, pp. 501–532, 2018.
- [22] M. Robinson, *Topological Signal Processing*, vol. 81. Berlin, Germany: Springer, 2014.
- [23] J. Jia, M. T. Schaub, S. Segarra, and A. R. Benson, "Graph-based semi-supervised & active learning for edge flows," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 761–771.
- [24] T. M. Roddenberry, M. T. Schaub, and M. Hajji, "Signal processing on cell complexes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 8852–8856.
- [25] S. Sardellitti and S. Barbarossa, "Topological signal representation and processing over cell complexes," 2022, *arXiv:2201.08993*.
- [26] S. Sardellitti, S. Barbarossa, and L. Testa, "Topological signal processing over cell complexes," in *Proc. IEEE 55th Asilomar Conf. Signals, Syst., Comput.*, 2021, pp. 1558–1562.
- [27] M. Yang, E. Isufi, and G. Leus, "Simplicial convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 8847–8851.
- [28] S. Ebli, M. Defferrard, and G. Spremann, "Simplicial neural networks," in *Proc. NeurIPS Workshop Topological Data Anal. Beyond*, 2020. [Online]. Available: <https://openreview.net/forum?id=nPct39DVlfk>
- [29] C. Bodnar et al., "Weisfeiler and Lehman go topological: Message passing simplicial networks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1026–1037.
- [30] C. Bodnar et al., "Weisfeiler and Lehman go cellular: CW networks," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 2625–2640, 2021.
- [31] E. Bunch, Q. You, G. Fung, and V. Singh, "Simplicial 2-complex convolutional neural networks," in *Proc. NeurIPS Workshop Topological Data Anal. Beyond*, 2020. [Online]. Available: <https://openreview.net/forum?id=TLbnsKrt6J>
- [32] T. M. Roddenberry, N. Glaze, and S. Segarra, "Principled simplicial neural networks for trajectory prediction," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9020–9029.
- [33] T. M. Roddenberry and S. Segarra, "Hodgenet: Graph neural networks for edge data," in *Proc. IEEE 53rd Asilomar Conf. Signals, Syst., Comput.*, 2019, pp. 220–224.
- [34] J. R. Munkres, *Elements of Algebraic Topology*. Boca Raton, FL, USA: CRC Press, 2018.
- [35] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie, "Random walks on simplicial complexes and the normalized Hodge 1-Laplacian," *SIAM Rev.*, vol. 62, no. 2, pp. 353–391, 2020.
- [36] T. M. Roddenberry, F. Frantzen, M. T. Schaub, and S. Segarra, "Hodgelets: Localized spectral representations of flows on simplicial complexes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5922–5926.
- [37] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [38] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*. Philadelphia, PA, USA: SIAM, 2007.
- [39] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 736–751, Dec. 2018.
- [40] G. L. Sleijpen and H. A. Van der Vorst, "A Jacobi–Davidson iteration method for linear eigenvalue problems," *SIAM Rev.*, vol. 42, no. 2, pp. 267–293, 2000.
- [41] V. L. Druskin and L. A. Knizhnerman, "Two polynomial methods of calculating functions of symmetric matrices," *USSR Comput. Math. Math. Phys.*, vol. 29, no. 6, pp. 112–121, 1989.
- [42] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: CRC Press, 2002.
- [43] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.
- [44] H. Youn, M. T. Gastner, and H. Jeong, "Price of anarchy in transportation networks: Efficiency and optimality control," *Phys. Rev. Lett.*, vol. 101, no. 12, 2008, Art. no. 128701.
- [45] M. T. Schaub, J. Lehmann, S. N. Yaliraki, and M. Barahona, "Structure of complex networks: Quantifying edge-to-edge relations by failure-induced flow redistribution," *Netw. Sci.*, vol. 2, no. 1, pp. 66–89, 2014.
- [46] T. A. Driscoll, N. Hale, and L. N. Trefethen, *Chebfun Guide*. Pafnuty Publications, 2014. [Online]. Available: <http://www.chebfun.org/docs/guide/>
- [47] B. Stabler, H. Bar-Gera, and E. Sall, "Transportation networks for researcher core team," Dec. 2021. [Online]. Available: <https://github.com/bstabler/TransportationNetworks>



Maosheng Yang (Graduate Student Member, IEEE) received the M.Sc. degree in electrical engineering in August 2020 from TU Delft, Delft, The Netherlands, where he is currently working toward the Ph.D. degree with the group of multimedia computing. His research interests include signal processing and learning over higher-order networks, e.g., graphs and simplicial complexes. He is a Member of the TU Delft AI Lab on water networks.



Elvin Isufi (Member, IEEE) received the B.Sc. and M.Sc. in electronic and telecommunication engineering from the University of Perugia, Perugia, Italy, in 2014 and 2012, respectively, and the Ph.D. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 2019. He is currently an Assistant Professor with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, and Co-director of AldroLab -the TU Delft AI Lab on water networks. Prior to this, he was a Postdoctoral

Researcher with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. He was also an Associate Editor for Elsevier's *Signal Processing*. His research interests include intersection of signal processing, mathematical modeling, machine learning, and network theory with applications in recommender systems, biological networks, and sensor networks. He was the recipient of the paper awards at the IEEE CAMSAP 2017, IEEE DSLW 2021, and IEEE DSLW 2022. He was with the organising committee of EURASIP EUSIPCO 2020 and will also be part for the 2024 edition.



Michael T. Schaub (Member, IEEE) received the B.Sc. degree in electrical engineering and information technology from ETH Zurich, Zurich, Switzerland, in 2009, the M.Sc. degree in biomedical engineering and the Ph.D. degree in applied mathematics from Imperial College London, London, U.K., in 2010 and 2014, respectively. After a Postdoctoral stay in Belgium with the Université catholique de Louvain, Ottignies-Louvain-la-Neuve, Belgium and University of Namur, Namur, Belgium, he was a Postdoctoral Researcher with the Institute of Data,

Systems and Society, the Massachusetts Institute of Technology (MIT), and Marie-Sklodowska Curie Fellow with MIT and the Department of Engineering Science, University of Oxford, Oxford, U.K. Since summer 2020, he has been with the Department of Computer Science, RWTH Aachen University, Aachen, Germany, as a (W1) Assistant Professor. He obtained an ERC Starting Grant in 2022. He is broadly interested in interdisciplinary applications of applied mathematics in engineering, social and biological systems. His research interests include network theory, data science, machine learning, and dynamical systems.



Geert Leus (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the KU Leuven, Leuven, Belgium, in June 1996 and May 2000, respectively. He is currently a Full Professor with the Faculty of Electrical Engineering, Mathematics and Computer Science with the Delft University of Technology, Delft, The Netherlands. He was the recipient of the 2021 EURASIP Individual Technical Achievement Award, a 2005 IEEE Signal Processing Society Best Paper Award, and a 2002 IEEE Signal Processing Society Young Author Best

Paper Award. He is a Fellow of the EURASIP. Prof. Leus was a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society, the Chair of the IEEE Signal Processing for Communications and Networking Technical Committee, and Editor in Chief of the *EURASIP Journal on Advances in Signal Processing*. He is also the Chair of the EURASIP Technical Area Committee on Signal Processing for Multisensor Systems and Editor in Chief of *EURASIP Signal Processing*.