Simplicial Trend Filtering (Invited Paper)

Yang, Maosheng; Isufi, Elvin

DOI
10.1109/IEEECONF56349.2022.10051892

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 56th Asilomar Conference on Signals, Systems and Computers

Simplicial Trend Filtering

(Invited Paper)

Maosheng Yang and Elvin Isufi
Dept. of Intelligent Systems
Delft University of Technology

2022 56th Asilomar Conference on Signals, Systems, and Computers | 978-1-6654-5906-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/IEEECONF56349.2022.10051892

***Abstract*—Reconstructing simplicial signals, e.g., signals defined on nodes, edges, triangles, etc., of a network, from (partial) noisy observation is of interest in water/traffic flow estimation or currency exchange markets. Typically, this concerns solving a regularised problem w.r.t. the ℓ_2 norm of the divergence or the curl of the signal, i.e., the netflows at nodes and in triangles. Real-world simplicial signals are intrinsically divergence- or curl-free, which makes ℓ_2 regularizers inapplicable. To overcome this, we develop a simplicial trend filter (STF) by regularising the total divergence and the curl via their ℓ_1 norm. By tuning two scalars, the STF can reduce independently the divergence and curl much more than smooth filtering, leading to a better reconstructed signal. The SFT is a convex problem and can be solved by fast iterative algorithms. We apply the SFT to interpolation and denoising tasks in forex and music/artist transition recordings and show its superior performance to alternatives.**

I. Introduction

Nonparametric signal reconstruction has a long-lasting history in discrete signal [1], image [2], and more recently in graph signal processing (GSP) [3], [4]. This is typically an ill-posed problem and requires regularizing it with a term that induces particular biases into the solution. For instance, in GSP we use the Tikhonov regularizer as a way to recover smooth graph signals from noisy partial observations. The latter penalizes large node signal variations in adjacent nodes and can be represented via the ℓ_2 norm of signal differences in adjacent nodes. When the smoothness bias is invalid for the data at hand, other regularizes are needed such as diffusion [5] or bandlimitedness [6]. In other cases, the signals exhibit sharp transitions only at a few nodes, which is the case of e.g., a clustered opinion networks where nodes within a cluster share similar opinions but can have arbitrarily different opinions in different clusters [7]. In these cases, regularizers promoting sparsity in signal differences are more appropriate. The latter are known as graph trend filters and penalize the ℓ_1 norm of signal differences in adjacent nodes [8], [9].

The above regularizers work for signals defined on the nodes of a graph. But in practice, we often have signals defined on higher-order network structures such as edges, triangles, and so on [10], [11]. This is of interest for modeling flow type data e.g., flows in water, communication or traffic networks, exchange rates of foreign currency (forex), pairwise comparisons in statistical ranking, and so on [12]. The topological structure of these data can be represented through *simplicial complexes* and algebraically via their the Hodge Laplacian matrices; a direct extension of the graph Laplacian to the simplex [13]. Hodge Laplacians encode two types of simplicial adjacency: *lower adjacency* e.g., two edges are adjacent if they have a common node; and *upper adjacency* e.g., two edges are adjacent if they belong to the same triangle. This algebraic representation has given rise to a series of new techniques to analyze and process simplicial signals, e.g., simplicial Fourier transform [10], filtering [14]–[16] and neural networks [17].

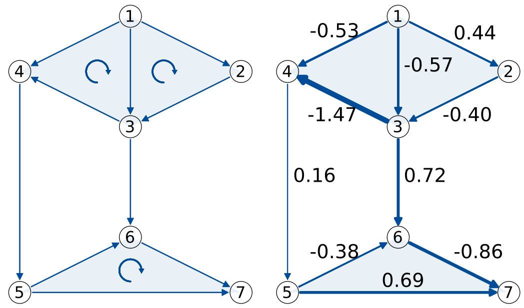
Nonparametric simplicial signal reconstruction is the equivalent problem discussed in GSP but now extended to the simplex. The Tikhonov regularizer has been extended in [18] for edge flow denoising and in [19] for interpolating missing values. Both works penalize the fitting part with a term based on the ℓ_2 norm of signal smoothness. But since in the simplex we have two types of adjacency, we penalize the smoothness on both the lower-connectivity – the ℓ_2 norm of the divergence of the flow, i.e., the netflow at the nodes – and the upper-connectivity – the ℓ_2 norm of the curl, i.e., the netflow circulating in triangles. The ℓ_2 regularizers only reduce the divergence or curl of the input but cannot preserve the divergence- or curl-free nature of the real-world edge flows, which is the case of arbitrage-free forex markets [12] where we want the divergence or the curl of the edge flow to be zero.

Motivated by the above setting, in this paper we extend the trend filtering concept to the simplex. The simplicial trend filtering (STF) has the following properties.

- **Sparse divergence/curl.** The STF penalizes the reconstruction problem with the ℓ_1 norm of the divergence and/or the curl when concerning edge flows. It naturally extends to account for higher-order simplicial structures (e.g., triangles and so on) but also multi-resolution information (neighboring edges that are non adjacent).
- **Controled sparsity.** By tuning two regularization weights we can independently control the sparsity w.r.t. the lower- (e.g., divergence sparsity) or upper-connections (e.g., curl sparsity). This is in contrast to the ℓ_2 regularizers which lead only to a global smooth behavior.
- **Convex problem.** The STF problem is a convex regularized least-squares problem that can be solved with of-the-shelf iterative methods such as ADMM [R11].

We corroborate the performance of SFT with real data from currency exchange and music transition recording.

Emails: m.yang-2, e.isufi-1@tudelft.nl. This work is supported by the TU Delft AI Labs Programme.



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

Fig. 1. (a): an SC of order 2 containing seven nodes, ten edges and three triangles (the shaded areas). The reference orientation of a node is trivial, that of an edge is indicated by the arrow on the edge and that of a triangle by the arc arrow. (b): A random edge flow, where a negative flow indicates that the actual flow direction is opposite to the reference edge orientation and the magnitude is denoted by the edge width.

II. BACKGROUD

Simplicial complexes. Given a finite set of vertices $\mathcal{V}$, a k-simplex $\mathcal{S}^k$ is a subset of $\mathcal{V}$ with cardinality $k+1$. For example, nodes, edges and triangles are 0-, 1- and 2-simplices. A *face* of $\mathcal{S}^k$ is a subset with cardinality k and thus a k-simplex has $k+1$ faces. A *coface* of $\mathcal{S}^k$ is a $(k+1)$-simplex that includes $\mathcal{S}^k$. A simplicial complex (SC) $\mathcal{X}^K$ of order K, is a collection of k-simplices $\mathcal{S}^k$, $k=0,\dots,K$, with an inclusion property–for any $\mathcal{S}^k \in \mathcal{X}^K$, then $\mathcal{S}^{k-1} \in \mathcal{X}^K$ if $\mathcal{S}^{k-1} \subset \mathcal{S}^k$. We denote the number of k-simplices in $\mathcal{X}^K$ by N_k. For computation, we fix an orientation for each simplex according to the lexicographical ordering of its vertices, e.g., a triangle $\{i,j,k\}$ is oriented as $[i,j,k]$ with $i<j<k$. See an SC example in Fig. 1a.

For the simplex $\mathcal{S}_i^k$, we define its *lower (upper) neighbourhood* $\mathcal{N}_{i,\ell}^k$ ($\mathcal{N}_{i,\mathrm{u}}^k$) as the set of k-simplices which share a common face (coface) with it. Moreover, we have that $\mathcal{N}_{i,\ell}^k = \mathcal{N}_{i,\ell}^{k+} \cup \mathcal{N}_{i,\ell}^{k-}$ with the *positive (negative)* neighborhood $\mathcal{N}_{i,\ell}^{k+}$ ($\mathcal{N}_{i,\ell}^{k-}$) containing the simplices that have a same (an opposite) orientation as $\mathcal{S}_i^k$ w.r.t. the common face; likewise, $\mathcal{N}_{i,\mathrm{u}}^k = \mathcal{N}_{i,\mathrm{u}}^{k+} \cup \mathcal{N}_{i,\mathrm{u}}^{k-}$. For the ith edge $[5,6]$ in Fig. 1a, we have that $\mathcal{N}_{i,\ell}^{1+} = \{[3,6],[6,7]\}$, $\mathcal{N}_{i,\ell}^{1-} = \{[4,5],[6,7]\}$ and $\mathcal{N}_{i,\mathrm{u}}^{1+} = \{[6,7]\}$, $\mathcal{N}_{i,\mathrm{u}}^{1-} = \{[5,7]\}$.

Hodge Laplacians. We can describe the relationships between $(k-1)$-simplices and k-simplices by the kth incidence matrix $\mathbf{B}_k \in \mathbb{R}^{N_{k-1}\times N_k}$, which maps each k-simplex to its faces, $(k-1)$-simplices. By definition, we have that $\mathbf{B}_k\mathbf{B}_{k+1} = \mathbf{0}$ [13]. Specifically, $\mathbf{B}_1$ is the node-edge incidence matrix, and $\mathbf{B}_2$ is the edge-triangle incidence matrix.

We can also represent an SC $\mathcal{X}$ of order K via the Hodge Laplacians $\mathbf{L}_k = \mathbf{B}_k^\top\mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top, \quad k=1,\dots,K-1$ 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 *lower Laplacian* $\mathbf{L}_{k,\ell} \triangleq \mathbf{B}_k^\top\mathbf{B}_k$ and the *upper Laplacian* $\mathbf{L}_{k,\mathrm{u}} \triangleq \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$ encode the lower and upper adjacency relationships between simplices through faces and cofaces, respectively. In particular, $\mathbf{L}_{1,\ell}$ and $\mathbf{L}_{1,\mathrm{u}}$ encode the edge adjacencies through their incident nodes and the common triangles that they form, respectively. They have entries

$$[\mathbf{L}_{1,\ell}]_{i,j} = \begin{cases} 2, & \text{for } i=j, \\ 1, & \text{for } j \in \mathcal{N}_{i,\ell}^{1+}, \\ -1, & \text{for } j \in \mathcal{N}_{i,\ell}^{1-}, \end{cases} \tag{1}$$

where the diagonal entry 2 of $\mathbf{L}_{1,\ell}$ indicates that each edge has two incident nodes, and

$$[\mathbf{L}_{1,\mathrm{u}}]_{i,j} = \begin{cases} d_{i,\mathrm{u}}, & \text{for } i=j, \\ 1, & \text{for } j \in \mathcal{N}_{i,\mathrm{u}}^{1+}, \\ -1, & \text{for } j \in \mathcal{N}_{i,\mathrm{u}}^{1-}, \end{cases} \tag{2}$$

where the upper degree $d_{i,\mathrm{u}}$ is the number of triangles that edge i participates in.

Simplicial signals. By attributing value s_i^k to the ith k-simplex $\mathcal{S}_i^k$, we define a *k-simplicial signal* $\mathbf{s}^k = [s_1^k,\dots,s_{N_k}^k]^\top \in \mathbb{R}^{N_k}$. If the signal value s_i^k is positive, then the corresponding signal is along the reference orientation; otherwise, opposite. In this paper, we work with node signal $\mathbf{v} = [v_1,\dots,v_{N_0}]^\top \in \mathbb{R}^{N_0}$, edge flow by $\mathbf{f} = [f_1,\dots,f_{N_1}]^\top \in \mathbb{R}^{N_1}$, e.g., Fig. 1b, and triangle flow $\mathbf{t} = [t_1,\dots,t_{N_2}]^\top \in \mathbb{R}^{N_2}$. We could apply incidence matrices on simplicial signals to achieve some physical operations. Specifically, we discuss the following.

Divergence. By applying matrix $\mathbf{B}_1$ to an edge flow $\mathbf{f}$, we compute its netflow at node i as

$$[\mathbf{B}_1\mathbf{f}]_i = \textstyle\sum_{j<i}\mathbf{f}_{[j,i]} - \sum_{i<k}\mathbf{f}_{[i,k]}, \tag{3}$$

which is the inflow minus the outflow at node i. In Fig. 1b, $[\mathbf{B}_1\mathbf{f}]_6 = \mathbf{f}_{[5,6]} + \mathbf{f}_{[3,6]} - \mathbf{f}_{[6,7]} = -0.38+0.72+0.86 = 1.2$

Gradient. By applying $\mathbf{B}_1^\top$ to a node signal $\mathbf{v}$, we induce a *gradient flow* $\mathbf{f}_{\mathrm{G}} = \mathbf{B}_1^\top\mathbf{v}$ via a gradient operation $[\mathbf{f}_{\mathrm{G}}]_{[i,j]} = \mathbf{v}_j - \mathbf{v}_i$, which is a differentiation operation along the edge.

Curl. By applying $\mathbf{B}_2^\top$ to an edge flow $\mathbf{f}$, we compute its netflow circulating in a triangle $t=[i,j,k]$ as

$$[\mathbf{B}_2^\top\mathbf{f}]_t = \mathbf{f}_{[i,j]} + \mathbf{f}_{[j,k]} - \mathbf{f}_{[i,k]}, \tag{4}$$

which is the *curl* operation. In Fig. 1b, $[\mathbf{B}_2^\top\mathbf{f}]_{[5,6,7]} = \mathbf{f}_{[5,6]} + \mathbf{f}_{[6,7]} - \mathbf{f}_{[5,7]} = -0.38-0.86-0.69 = -1.93$. By applying $\mathbf{B}_2$ to a triangle flow $\mathbf{t}$, we induce a *curl flow* $\mathbf{f}_{\mathrm{C}} = \mathbf{B}_2\mathbf{t}$.

The Hodge Laplacian admits a *Hodge decomposition*, which is, for $k=1$, $\mathbb{R}^{N_1} = \mathrm{im}(\mathbf{B}_1^\top) \oplus \mathrm{im}(\mathbf{B}_2) \oplus \ker(\mathbf{L}_1)$ [10], [13]. That is, any edge flow can be decomposed into three orthogonal components. The gradient space $\mathrm{im}(\mathbf{B}_1^\top)$ consists of all the flows induced by a node signal, which are *curl-free*, while the curl space $\mathrm{im}(\mathbf{B}_2)$ comprises the flows induced from a triangle flow, which are *divergence-free*. If an edge flow is both divergence- and curl-free, it is a *harmonic* flow in the harmonic space $\ker(\mathbf{L}_1)$.

In this paper, we study the reconstruction of a simplicial signal from its noisy and partial observation, i.e., simplicial signal denoising and interpolation. For the ease of exposition, we focus primarily on the edge flow case. We consider the settings where the underlying signal is approximately either divergence-free or curl-free. Thus, we propose a regularized filtering that penalizes the ℓ_1 norm of the divergence and the curl. We show how this filter relates to the conventional and graph trend filtering [8], [20], thus, we refer to it as a simplicial trend filtering (STF).

III. Simplicial Trend Filtering

Consider a noisy edge flow observation $\mathbf{y} = \mathbf{f}^* + \mathbf{n}$ with the additive noise $\mathbf{n}$. The 0-order SFT seeks for the estimate $\hat{\mathbf{f}}$ by solving

$$\hat{\mathbf{f}} = \underset{\mathbf{f}\in\mathbb{R}^{N_1}}{\arg\min} \|\mathbf{y} - \mathbf{f}\|_2^2 + \alpha\|\mathbf{B}_1\mathbf{f}\|_1 + \beta\|\mathbf{B}_2^\top\mathbf{f}\|_1, \tag{5}$$

where parameters $\alpha \geq 0$ and $\beta \geq 0$ control the trade-off between the data fidelity and the ℓ_1 norm of the divergence and the curl. When $\alpha = 0$ or $\beta = 0$, the SFT only accounts for the regularization on the curl or the divergence, respectively.

In optimization theory, the ℓ_1 norm is often regarded as a closet relaxation of the ℓ_0 norm which is the sparsity measure. Thus, the regularizer $\|\mathbf{B}_1\mathbf{f}\|_1$ seeks for a solution that has as many as possible zero divergence at nodes so to promote its sparsity; likewise for the regularizer $\|\mathbf{B}_2^\top\mathbf{f}\|_1$ aims to obtain an estimate that has as many as possible zero curls in trignales. To solve the 0-STF, which is convex, we can use off-the-shelf algorithms, e.g., ADMM or Newton method [20].

Moreover, the Hodge Laplacian acts as a shift operator for simplicial signals, also used in Fourier analysis and as building blocks for filtering [11], [14]–[16]. By applying $\mathbf{L}_{1,\ell}$ or $\mathbf{L}_{1,\mathrm{u}}$ to a flow $\mathbf{f}$, we obtain a shifted edge flow $\mathbf{L}_{1,\ell}\mathbf{f}$ or $\mathbf{L}_{1,\mathrm{u}}\mathbf{f}$, which accounts for the one-hop lower or upper neighbouring information, as in (1) and (2); likewise for a general shifting $\mathbf{L}_{1,\ell}^p\mathbf{f}$ or $\mathbf{L}_{1,\mathrm{u}}^q\mathbf{f}$, we account for p-hop lower neighbours and q-hop upper neighbours [14], [15]. Thus, we consider a general (p, q)-order SFT to account for the information from multi-hop neighbours as

$$\hat{\mathbf{f}} = \underset{\mathbf{f}\in\mathbb{R}^{N_1}}{\arg\min} \|\mathbf{y} - \mathbf{f}\|_2^2 + \alpha\|\Delta_\ell^{(p)}\mathbf{f}\|_1 + \beta\|\Delta_\mathrm{u}^{(q)}\mathbf{f}\|_1, \tag{6}$$

where the operators $\Delta_\ell^{(p)}$ and $\Delta_\mathrm{u}^{(q)}$ have the forms

$$\Delta_\ell^{(p)} = \begin{cases} \mathbf{B}_1^\top\Delta_\ell^{(p-1)} = \mathbf{L}_{1,l}^{\frac{p+1}{2}}, \text{ for odd } p, \\ \mathbf{B}_1\Delta_\ell^{(p-1)} = \mathbf{B}_1\mathbf{L}_{1,l}^{\frac{p}{2}}, \text{ for even } p, \end{cases} \tag{7}$$

with $\Delta_\ell^{(0)} = \mathbf{B}_1$ and

$$\Delta_\mathrm{u}^{(q)} = \begin{cases} \mathbf{B}_2\Delta_\mathrm{u}^{(q-1)} = \mathbf{L}_{1,u}^{\frac{q+1}{2}}, \text{ for odd } q, \\ \mathbf{B}_2^\top\Delta_\mathrm{u}^{(q-1)} = \mathbf{B}_2^\top\mathbf{L}_{1,u}^{\frac{q}{2}}, \text{ for even } q, \end{cases} \tag{8}$$

with $\Delta_\mathrm{u}^{(0)} = \mathbf{B}_2^\top$. When $p = 0$ and $q = 0$, this gives the 0-STF (5). When $p = 1$, we have the regularizer $\mathbf{L}_{1,\ell}\mathbf{f}$ which is a one-step lower shifting. At an edge i, this can be detailed as

$$[\mathbf{L}_{1,\ell}\mathbf{f}]_i = 2\mathbf{f}_i + \textstyle\sum_{j\in\mathcal{N}_{i,\ell}^{1+}}\mathbf{f}_j - \sum_{k\in\mathcal{N}_{i,\ell}^{1-}}\mathbf{f}_k, \tag{9}$$

where edges j and k are the positive and negative lower neighbours. In Fig. 1b, $[\mathbf{L}_{1,\ell}\mathbf{f}]_{[5,6]} = 2\mathbf{f}_{[5,6]} + (\mathbf{f}_{[3,6]} + \mathbf{f}_{[5,7]}) - (\mathbf{f}_{[4,5]} + \mathbf{f}_{[6,7]})$. When $q = 1$, the regularizer $\mathbf{L}_{1,\mathrm{u}}\mathbf{f}$ has its ith entry as

$$[\mathbf{L}_{1,\mathrm{u}}\mathbf{f}]_i = d_{i,\mathrm{u}}\mathbf{f}_i + \textstyle\sum_{j\in\mathcal{N}_{i,\mathrm{u}}^{1+}}\mathbf{f}_j - \sum_{k\in\mathcal{N}_{i,\mathrm{u}}^{1-}}\mathbf{f}_k \tag{10}$$

which is a linear operation with in the upper neighbourhood. In Fig. 1b, $[\mathbf{L}_{1,\mathrm{u}}\mathbf{f}]_{[5,6]} = \mathbf{f}_{[5,6]} + \mathbf{f}_{[6,7]} - \mathbf{f}_{[5,7]}$. The regularizers $\|\Delta_\ell\mathbf{f}\|_1$ and $\|\Delta_\mathrm{u}\mathbf{f}\|_1$ can also promote the divergence- and curl-free property because if $\mathbf{L}_{1,\ell}\mathbf{f} = \mathbf{0}$, then $\mathbf{B}_1\mathbf{f} = \mathbf{0}$ from the definition of $\mathbf{L}_{1,\ell}$; likewise for $\mathbf{L}_{1,\mathrm{u}}$. For larger orders p or q, the STF seeks for the edge flows which are divergence- or curl-free after shifting.

The proposed STF can be easily adapted to the interpolation task by replacing the data fitting term $\|\mathbf{y} - \mathbf{f}\|_2^2$ by $\|\mathbf{y} - \mathbf{C}\mathbf{f}\|_2^2$ where $\mathbf{y} \in \mathbb{R}^M$ and $\mathbf{C} \in \{0, 1\}^{M\times N_1}$ is the selection matrix. In the case of node signals, operator Δ_ℓ does not exist and the STF gives the form of graph trend filtering [8].

IV. Trend filtering versus ℓ_2 regression

In this section, we study the ℓ_2 regularizer from the frequency domain, showing that it aims to generate a globally smooth edge flow. The ℓ_2 regularization has the form [11], [18]

$$\min_{\mathbf{f}\in\mathbb{R}^{N_1}} \|\mathbf{y} - \mathbf{f}\|_2^2 + \alpha\|\mathbf{B}_1\mathbf{f}\|_2^2 + \beta\|\mathbf{B}_2^\top\mathbf{f}\|_2^2, \tag{11}$$

where the regularizers $\|\mathbf{B}_1\mathbf{f}\|_2^2$ and $\|\mathbf{B}_2^\top\mathbf{f}\|_2^2$ are the edge flow variation measures [14], [15]. It has the closed-form solution

$$\hat{\mathbf{f}} = (\mathbf{I} + \alpha\mathbf{B}_1^\top\mathbf{B}_1 + \beta\mathbf{B}_2\mathbf{B}_2^\top)^{-1}\mathbf{y}. \tag{12}$$

The operator $\mathbf{H} := (\mathbf{I} + \alpha\mathbf{B}_1^\top\mathbf{B}_1 + \beta\mathbf{B}_2\mathbf{B}_2^\top)^{-1}$ is a *low-pass* simplicial filter.

By performing the eigendecomposition $\mathbf{L}_1 = \mathbf{U}_1\mathbf{\Lambda}\mathbf{U}_1^\top$, one could define the simplicial Fourier bases as the eigenvectors $\mathbf{U}_1 = [\mathbf{u}_1, \dots, \mathbf{u}_{N_1}]$ and the simplicial frequency as the eigenvalues $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_{N_1})$ [10]. Furthermore, it is shown that there exist three set of Fourier bases in $\mathbf{U}_1$ which span the three orthogonal subspaces given by the Hodge decomposition. More specifically, we have $\mathbf{U}_1 = [\mathbf{U}_\mathrm{H}\ \mathbf{U}_\mathrm{G}\ \mathbf{U}_\mathrm{C}]$ where (i) the eigenvectors $\mathbf{U}_\mathrm{H}$ of $\mathbf{L}_1$ associated to zero eigenvalues (collected in set $\mathcal{Q}_\mathrm{H}$) span the harmonic space $\ker(\mathbf{L}_1)$, (ii) the eigenvectors $\mathbf{U}_\mathrm{G}$ of $\mathbf{L}_{1,\ell}$ associated to nonzero eigenvalues (collected in set $\mathcal{Q}_\mathrm{G}$) span the gradient space $\mathrm{im}(\mathbf{B}_1^\top)$, and (iii) the eigenvectors $\mathbf{U}_\mathrm{C}$ of $\mathbf{L}_{1,\mathrm{u}}$ associated to nonzero eigenvalues (collected in set $\mathcal{Q}_\mathrm{C}$) span the curl space $\mathrm{im}(\mathbf{B}_2)$. The frequency in $\mathcal{Q}_\mathrm{G}$ and $\mathcal{Q}_\mathrm{C}$ measure the total ℓ_2 norm of the divergence and the curl, i.e., the edge flow variations [14], [15].

Based on this finer simplicial Fourier transform, at frequency λ, we have the frequency response $\tilde{H}(\lambda)$ of $\mathbf{H}$ as

$$\tilde{H}(\lambda) = \begin{cases} 1, & \text{for } \lambda = 0 \in \mathcal{Q}_\mathrm{H}, \\ (1 + \alpha\lambda)^{-1}, & \text{for } \lambda \in \mathcal{Q}_\mathrm{G}, \\ (1 + \beta\lambda)^{-1}, & \text{for } \lambda \in \mathcal{Q}_\mathrm{C}, \end{cases} \tag{13}$$

where $\mathcal{Q}_\mathrm{H}, \mathcal{Q}_\mathrm{G}, \mathcal{Q}_\mathrm{C}$ are the harmonic, gradient and curl frequency sets. From (13), we see that the ℓ_2 regularizer smoothens the gradient and curl components of the input flow by multiplying factors $(1+\alpha\lambda)^{-1}$ and $(1+\beta\lambda)^{-1}$ respectively in the frequency domain, leading to a suppressed divergence and curl of the output. Different from the STF that promotes an output with zero divergence and curl, it seeks for a globally

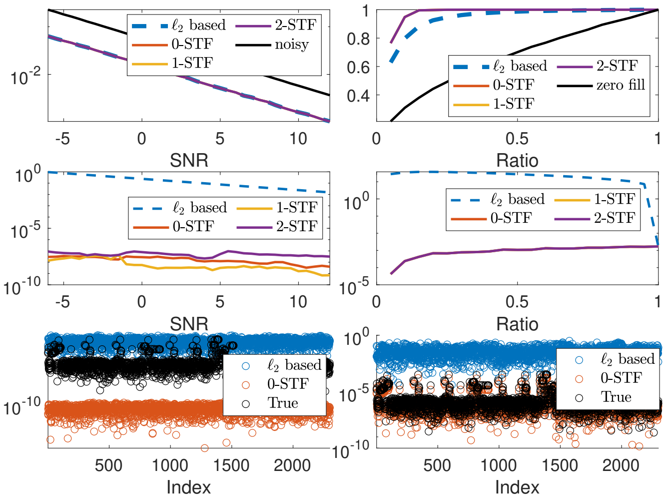


Fig. 2. Reconstruction performance for forex. Top left: denoising NRMSE performance. Middle left: total curl measure, $\|\mathbf{B}_2^\top\hat{\mathbf{f}}\|_2$. Bottom left: the curl vector $\mathbf{B}_2^\top\hat{\mathbf{f}}$ when SNR is 3dB. Top right: interpolation correlation performance. Middle right: total curl measure. Bottom right: the curl vector $\mathbf{B}_2^\top\hat{\mathbf{f}}$ when ratio is 50%.

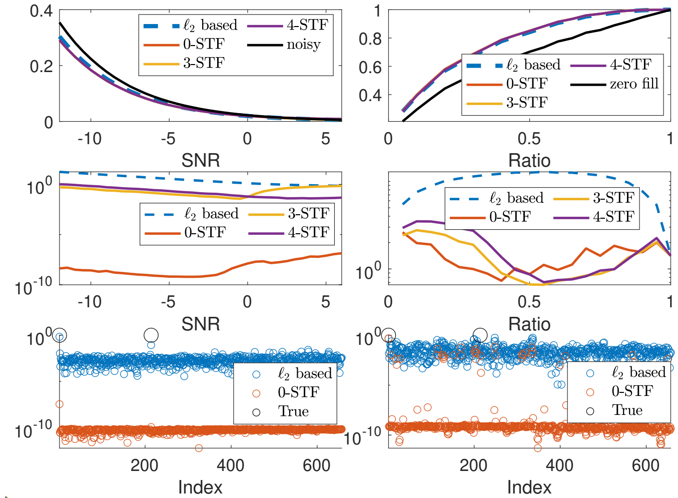


Fig. 3. Reconstruction performance for forex. Top left: Denoising NRMSE performance. Top right: total divergence measure, $\mathbf{B}_2^\top\hat{\mathbf{f}}$. Bottom left: Interpolation correlation performance. Bottom right: total curl measure.

smooth output but not necessarily with zero divergences and curls. This in turn may be problematic in applications where the underlying edge flow is required to be divergence- or curl-free, e.g., in a water network it is expected for flows to have zero netflow at junction points, and in forex problem the arbitrage-free condition is necessary for a fair market.

V. Experiments

In this section, we compare the performance of the STF and the ℓ_2 regularization (11) in edge flow denoising and interpolation tasks. Specifically, for the ℓ_2 based interpolation, we compare with the least-squares solution in [19, eq. 7]. To average the performance, 100 realizations were conducted. For evaluation, we used the NRMSE for denoising task and the Pearson correlation for interpolation. We also computed the ℓ_2 norm of the divergence ($\|\mathbf{B}_1\hat{\mathbf{f}}\|_2$) or the curl ($\|\mathbf{B}_1\hat{\mathbf{f}}\|_2$) of the reconstructed edge flow as the real-world flows are intrinsically divergence-free or curl-free, and showed instances of the divergence and the curl.

A. Foreign currency exchange

We first considered a forex problem consisting of 25 different currencies collected from [21] and every two currencies can be exchanged. Thus, it forms a currency SC with 25 nodes, 300 edges and 2300 triangles. For any currencies $i, j, k \in \mathcal{V}$, the arbitrary-free condition requires that $r^{i/j}r^{j/k} = r^{i/k}$ with the exchange rate $r^{i/j}$ between i and j. This implies that it provides no gain or loss with a successive trading path $i \to j \to k$ over a direct trading $i \to k$. If we represent the exchange rates as edge flows $\mathbf{f}_{[i,j]} = \log(r^{i/j})$, the arbitrage condition can be translated into that $\mathbf{f}$ is curl-free [12]. The curl of the collected data is shown in Fig. 2 (bottom left, black circles) with an ℓ_2 norm 0.0017. It is not exactly curl-free and we expect to remove these nonzero curls. We added artificial noise with SNRs ranging from $[-6\text{dB}, 12\text{dB}]$ in the denoising task and randomly sampled the edge flows with ratios of $[0.05, 1]$. For both approaches, we set $\alpha = 0$ so to only penalize the curl and chose $\beta = 2$ for the best denoising performance of the ℓ_2 regularizer when SNR is 3dB. We considered STF of orders from 0 to 4.

Since STF of different orders perform very similar for this dataset, we only plotted the first three cases. The results are reported in Fig. 2. For denoising, we observe that both regularizers perform very similarly in reducing the noise in terms of the NRMSE. However, the STF (of different orders) reconstructed exchange rates have a much smaller total curl than the ℓ_2 based one (middle and bottom left). That is, the former is able to reconstruct the exchange rates following the arbitrage-free condition while the latter fails so. Moreover, the STF can remove the intrinsic curls from the underlying data which is desired in forex for the arbitrage-free condition.

Similar observations can be made in the interpolation task. The STF performs much better in both interpolating the missing exchange rates and preserving the arbitrage-free property, though the intrinsic curls are not fully removed by the STF.

B. Lastfm dataset

We then considered the transitions among music artists in a user's play recordings on Lastfm[1]. We represented each unique artist as a node and any adjacent artists in the recordings as edges. Following [19], we constructed the edge flow as follows: every time the user listened to artist A followed by artist B, add one unit to the flow from A to B. The constructed flow is everywhere divergence-free as the user always transitioned to another artist after one, except two nodes have a nonzero divergence where the user started and stopped as shown in the bottom of Fig. 3. The ℓ_2 norm of the underlying divergence is 1.14.

For both approaches, we set $\beta = 0$ to only penalize the divergence and $\alpha = 0.5$ for a best denoising performance based on ℓ_2 regularizer when SNR is 3dB. We considered the STF of orders from 0 to 4, but only made plots of orders 0, 3

[1]This dataset is from https://www.last.fm/.

and 4 for illustration. For denoising with SNRs ranging from $[-12\text{dB}, 6\text{dB}]$, we observe that the STF performs comparably with than the ℓ_2 regularizer in terms of the NRMSE. To preserve a small total divergence, the 0-STF performs the best with $\alpha = 0.5$ while the ℓ_2 regularizer output a much larger total divergence, as except at two nodes, the 0-STF basically fully removed the nonzero divergence (bottom left). Both are similar in the interpolation task in terms of the correlation, and, again, the STF preserves the divergence much better than the ℓ_2 regularizer.

VI. Conclusion

Edge flows arising from the real-world tend to be either divergence- or curl-free, e.g.,water flows are expected to have zero netflow at junction points, in a fair forex market, it is required to not have circulating flows in any three pairs of exchanging currencies. Reconstructing such edge flows from noisy and partial observations is paramount in applications where we cannot have access to all information. In these cases, regularization-based techniques are of interest to impose particular prior about the underlying signal and its coupling with the topology. The typical ℓ_2 based regularization reconstructs a globally smooth flow, as shown via its frequency analysis, not necessarily exactly zero divergence or curl. This paper proposed the simplicial trend filtering by leveraging an ℓ_1 norm regularization on the divergence and the curl of the reconstructed signal, ultimately, promoting sparsity. Experimental results support the necessity of the STF especially when the simplicial signals are divergence- or curl-free. All in all, the STF contributes to the emerging signal processing on simplicial complexes as an alternative for reconstruction.

References

[1] H. Qi and S. Hughes, "Using the kernel trick in compressive sensing: Accurate signal recovery from fewer measurements," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 3940–3943.

[2] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[3] 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 Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[4] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[5] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning theory and kernel machines*. Springer, 2003, pp. 144–158.

[6] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 764–778, 2016.

[7] M. O. Jackson, "Social and economic networks," in *Social and Economic Networks*. Princeton university press, 2010.

[8] Y.-X. Wang, J. Sharpnack, A. Smola, and R. Tibshirani, "Trend filtering on graphs," in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 1042–1050.

[9] R. Varma, H. Lee, J. Kovačević, and Y. Chi, "Vector-valued graph trend filtering with non-convex penalties," *IEEE transactions on signal and information processing over networks*, vol. 6, pp. 48–62, 2019.

[10] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992–3007, 2020.

[11] 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," *arXiv preprint arXiv:2101.05510*, 2021.

[12] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, "Statistical ranking and combinatorial hodge theory," *Mathematical Programming*, vol. 127, no. 1, pp. 203–244, 2011.

[13] L.-H. Lim, "Hodge laplacians on graphs," *SIAM Review*, vol. 62, no. 3, pp. 685–715, 2020.

[14] M. Yang, E. Isufi, M. T. Schaub, and G. Leus, "Finite impulse response filters for simplicial complexes," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 2005–2009.

[15] ——, "Simplicial convolutional filters," *arXiv preprint arXiv:2201.11720*, 2022.

[16] E. Isufi and M. Yang, "Convolutional filtering in simplicial complexes," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 5578–5582.

[17] M. Yang, E. Isufi, and G. Leus, "Simplicial convolutional neural networks," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8847–8851.

[18] M. T. Schaub and S. Segarra, "Flow smoothing and denoising: Graph signal processing in the edge-space," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 735–739.

[19] J. Jia, M. T. Schaub, S. Segarra, and A. R. Benson, "Graph-based semi-supervised & active learning for edge flows," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 761–771.

[20] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "\ell_1 trend filtering," *SIAM review*, vol. 51, no. 2, pp. 339–360, 2009.

[21] O. Corporation. (2018) Foreign exchange data. [Online]. Available: https://www.oanda.com/