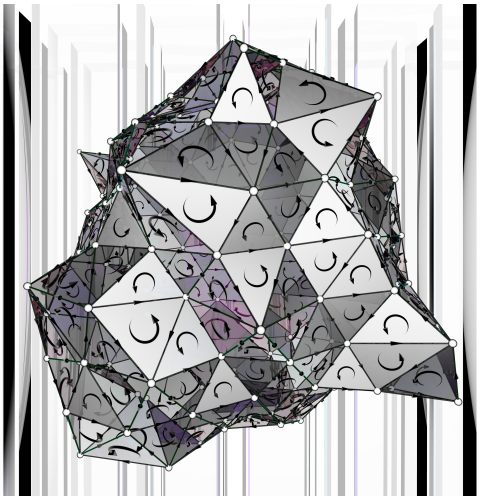


Doctoral Thesis, PhD Program in Mathematics

Topological Stability and Preconditioning of Higher-Order Laplacian Operators on Simplicial Complexes

Anton Savostianov[1]

[1]Gran Sasso Science Institute, viale F.Crispi 7, L'Aquila, Italy , email: anton.savostianov@gssi.it

Supervisors: Francesco Tudisco, francesco.tudisco@gssi.it
Nicola Guglielmi, nicola.guglielmi@gssi.it

Abstract: Relational data exhibiting the underlying structure of interactions of agents in the system is present in virtually every area of research such as biology, neurology, chemistry, transportation and social networks, etc, and can be described by classical graph models. Such models allow the injection of the interactions' structure into the system dynamics, govern its controllability and provide uniquely useful structural descriptions of the system through degree distribution and centrality measures and thus can be used for community detection, the study of synchronization, node importance, label spreading, and so on. At the same time, classical graph models are restricted to pair-wise linear interactions between the agents whilst various natural systems are governed through the multi-agent interactionsand lack the ability to describe higher-order topological features of the data. As a result, various higher-order network models have been introduced such as hypergraphs, motifs, cell and simplicial complexes. Whilst each model above remains useful, one needs to balance its capability to contain higher-order interactions and the complexity of the arising mathematical description. Simplicial complexes follow such a trade-off by assuming each interaction to be a simplex of the system's agents and describing higher-order topological features (connected components, one-dimensional holes, void, etc) through homology groups and corresponding higher-order Laplacian operators L_k .

In the current thesis, we posit questions of the stability of homology groups through the numerical lens of the stability of higher-order Laplacian operators for weighted simplicial complexes. In the first part of the work, we formulate and study the topological aspect of the stability in the sense of minimal perturbation sufficient to increase the homology group. This problem can be reformulated as a spectral matrix nearness problem for the corresponding Laplacian operator which is prone to exhibit the phenomenon of homological pollution. We propose a bi-level optimization framework based on the gradient flow and matrix differential equations which allows us to obtain the minimal perturbation. The developed approach is then tested on the synthetic and real-life transportation datasets. The second part of the work, conversely, considers the numerical stability of the linear system for higher-order Laplacian operator, $L_k\mathbf{x} = \mathbf{f}$. We demonstrate that the system can be reduced to a sparser one and develop a preconditioning scheme based on the notion of a weakly collapsible simplicial complex which we refer to as preconditioning by a heavy collapsible subcomplex (HeCS-preconditioner). The performance of the preconditioning routine is then tested for the computationally demanding cases on the synthetic triangulation-based datasets.

Keywords: simplicial complex, homology group, Hodge Laplacian, collapsibility, preconditioning, stability

Acknowledgements

As with virtually any work, this thesis would not have been written if not for the efforts, help and support of various people whom I would like to take a moment to mention here.

I would like to extend my immense gratitude to my scientific advisors, professors Nicola Guglielmi and Francesco Tudisco, for their help, guidance, and direction in my research throughout these four years. As a PhD student, one typically struggles to find a balance between independent work and supervision, and it is hard to underestimate how successful Nicola and Francesco have been in finding such a balance for me. Additionally, I thank my fellow students and professors from Gran Sasso Science Institute for various discussions that have helped me understand how to present and describe this work in a somewhat comprehensible way.

On a separate note, I would like to thank Professors Sasha Shapoval and Mikhail G. Shnirman for reinstating my faith that exciting and interesting research, which is not buried under all the bureaucracy, still exists, and for their support and encouragement through one of the darkest periods of my life prior to starting my PhD.

Finally, it would be impossible for me not to thank my family — my parents, Lena and Sergey, my sister Dasha, and my wife, Dayana, — for more things than can fit on a sheet of paper.

The art on the title page is a beautiful work by Prof. Robert Ghrist. We also invite you to check out a fantastic set of videos on the foundations of Topological Data Analysis, done by the same author.

Contents

List of Figures

List of Tables

List of Algorithms

I Introduction

Multi-agent systems with structured interactions between the agents are ubiquitous and omnipresent throughout various areas of research and are used to model a significant amount of natural systems via **networks of interactions**. In such systems, the relational data can be induced by the geometry of the system or spatial reasoning (for instance, in the case of transportation networks, intersections/stops of public transport are connected through the preexisting physical driving routes; alternatively, in the case of opinion spreading, the geometry of the system is facilitated through the structure of human interactions), or by the functionality of the interactions (for instance, in gene regulatory networks or networks of chemical reactions only certain agents, or substances, can possibly interact with each other). Even systems with a perceived absence of the underlying structure frequently exist in the confinement of the trivial, uniform, or regular governing network: as a result, the dynamics of the system of particles on a simple regular lattice may be reduced into a mean-field model in the thermodynamic limit. At the same time, accounting for a less regular structure of interactions may greatly affect the system's evolution and principle statistics.

Graph models containing the set of system's agents and their pairwise interactions are a natural way to describe such networks of interactions. As a consequence of a high level of abstraction, graph models over the years have been introduced in virtually every area of research and have provided a uniquely useful machinery allowing the injection of the interacting structure into the system dynamics and into the abstract mechanisms of machine learning(thus developing Graph Neural Networks, GNNs), as well as providing the insight on the system solely based on the topology of interactions, e.g. via node importance, centrality measures, label spreading, etc. Understanding structural features, e.g. degree distribution, allows one to classify systems through their structure (scale-free, preferential attachment, hyperbolic, small-world networks, etc.) and define and assess the effect of the regularity of the structure (homophily), whilst random graphs models allow to separate meaningful structural phenomena from randomly occurring ones.

However, graph models by design fail to capture non-dyadic (higher-order) patterns of interactions present in various natural systems. For instance, a typical chemical reaction would involve one or more catalyzers besides two reacting substances; similarly, genes are typically regulated by several factors (enzymes) in gene regulatory networks; in the same manner, the

majority of social interactions are non-pair-wise (e.g. coauthorships in a coauthorship graph are naturally higher-order interactions since a scientific paper as an "interaction" may encompass any number of researchers from one to several hundred). Moreover, in the scope of topological data analysis (TDA), restriction to the case of only pairwise interaction severely limits the ability to establish the system's topological features and more intrinsic topology in general.

This reasoning has motivated the development of various methods and models incorporating higher-order (non-linear) structures into classical (linear) graph models: motifs, hypergraphs, line graphs, cell and simplicial complexes, etc. In that sense, motifs incorporate higher-order structures through the specific repeating subgraphs or neighborhoods in the linear structure that can additionally affect system dynamics, e.g. promote synchronization or facilitate a faster label spreading; additionally, assuming chosen motifs are functionally significant, the rate of its occurrence can be used as a precursor of a feature for network classification tasks. Hypergraphs, however, provide a more general model for higher-order interactions where each interaction as a subset of the set of nodes (agents) is denoted as hyperedge; by design, such models are larger and typically sparse and generalize a sufficient number of graph-related properties and features. Specifically, the transition from a linear graph model to a hypergraph can fasten the synchronization between nodes and improve label spreading. Nevertheless, by its definition, every matrix-based property of the classical graph models (adjacency matrix, various graph Laplacians, etc.) would now require a much less comprehensible tensor model in the case of hypergraphs. Alternatively, one can try to avoid tensor models by introducing the line graph of a hypergraph, where each hyperedge is assigned a node, and the connection corresponds to the adjacency. Still, such graphs were shown to struggle to convey the same topological insight, not to mention their explosively growing size. Although one should not discard all the advances made with such models, one may also try to obtain a more tractable higher-order generalization that does not require tensor or line graph extensions.

As a result, more structured or restricted types of hypergraphs, such as cell and simplicial complexes, have been introduced. In the case of a simplicial complex $\mathcal{K}$, each interaction in the system is a simplex with an inclusion rule: every face of the simplex from the complex should also remain in the complex; besides the immediate geometric interpretation, one can also refer to the case of social interaction or co-authorship graph where the interaction between a group of people necessarily implies the interaction between any subset of the individuals. Simplicial complex models are clearly a restricted case of a general hypergraph and, in that, are more topologically sound: due

to the inclusion principle, one can naturally relate simplices to their boundaries remaining in the complex through the boundary maps B_k, reminiscent of differential operators on manifolds. As a result, one can define a discrete counterpart of a homology group $\mathcal{H}_k$ which, similarly to the continuous case in differential geometry, formally describes topological features of the simplicial complex such as k-dimensional holes through higher-order Laplacian operators L_k, sometimes known as Helmholtz operators. For instance, in the case $k = 0$, the simplicial complex would coincide with a classical graph, and the homology group $\mathcal{H}_0$ would define its connected components through the graph Laplacian L_0 via Fiedler number and Fiedler vectors. As a result, L_k operators govern higher-order dynamics on the simplicial complex and describe random walk transitions in tasks like simplicial PageRank, and the corresponding homology groups outline the topology of the processes inside the simplicial complex (e.g. in brain activity or chemical reactions).

This work focuses on stability issues associated with higher-order topologies of simplicial complexes defined through homology groups. In the first part of the current work, we pose a question of the stability of the weighted homology group $\mathcal{H}_k$: specifically, what is the minimal perturbation (e.g. in terms of edge elimination) sufficient for changing the dimensionality of the homology group H_k. Whilst the specific case of topological stability has been studied before in the isolated case of the persistent homology (where the simplicial complex is produced from a point cloud by filtering out overly distant connections), this consideration remains uniquely suitable only for the filtration case in which it is still computationally demanding. Instead, we pose a more general question of the proximity of another simplicial complex $\mathcal{K}'$ with a larger homology group ($\dim \mathcal{H}_k(\mathcal{K}') > \dim \mathcal{H}_k(\mathcal{K})$) where $\mathcal{K}'$ is obtained from the original complex $\mathcal{K}$ via edge elimination. Whilst by design such a question is combinatorial, one can move it to the continuous setting by associating edge elimination to vanishing weights of the edges after the perturbation of the weights, reducing the problem to a continuous optimization task with a spectral target functional motivated by a spectral matrix nearness problem. However, while the topology of the simplicial complex is related to the spectrum of Laplacians L_k, the appropriate target spectral functional should be carefully tailored to account for the phenomenon of homological pollution. Note that due to the homology-induced decomposition of the space, the spectrum of higher-order Laplacian operators L_k typically inherits part of the spectrum of the previous Laplacian L_{k-1}. As a result, it may exhibit instability of the spectrum, which is not associated with the instability of the homology group $\mathcal{H}_k$ (but, instead, $\mathcal{H}_{k-1}$). We refer to the underlying mechanism as principal spectral inheritance and to the effect of spectral instability, which does not lead to topological one, as homological

pollution. Additionally, one should, in general, take into account the effects of the dimensionality reduction of the tasks since, for example, in the case of $k = 1$, edge elimination may cause the reduction of the dimension of the underlying matrix L_1, which can create additional faux zeros in the spectrum. Finally, amongst the variety of matrix nearness methods, we will adopt the already established idea of a bi-level norm-constrained/unconstrained gradient flow integration to optimize the corresponding spectral functional, avoiding homological pollution. The performance of developed methods is then demonstrated on illustrative examples, synthetic datasets based on the Delaunay triangulation, and real-life data on transport networks.

Topological instability in our definition asks for the graph theoretical entity (e.g. "how many edges does one need to eliminate to obtain another k-dimensional hole") obtained through the computational framework via spectral optimization. At the same time, one can ask the opposite (in spirit): assuming a linear system $L_k\mathbf{x} = \mathbf{f}$ associated with a simplicial complex is unstable (poorly conditioned), how can one exploit the underlying structure of the simplicial complex in order to stabilize it and develop a more efficient solver? In terms of the homology of the simplicial complex, the solution of such linear system is asked for during the computation of the lower part of the spectrum of Laplacian L_k, as well as in many other applications, such as computing stationary point of the system dynamics on the simplicial complex (e.g. $\dot{\mathbf{x}} = L_k\mathbf{x} - \mathbf{f}$) or implicit simplicial complex convolution neural networks. We demonstrate that due to the homological decomposition of the space and principal spectral inheritance, it is sufficient to propose efficient solver only for a part of the Laplacian L_k describing the action of the border operator on simplices of the highest possible order (e.g. on triangles in the case of L_1), known as up-Laplacian $L_k^{\uparrow}\mathbf{x} = \mathbf{f}$. Then, the question of stabilization is the question of optimizing the condition number $\kappa(L_k^{\uparrow})$ governing the convergence rate of various iterative solvers such as GMRES, CG, CGLS, and so on, which can be achieved via fast and efficient symmetric preconditioning close to Cholesky or incomplete Cholesky methods. One can show that the computation of an exact Cholesky multiplier in the general case of $L_k^{\uparrow}$, $k > 0$, is unfeasible and can not be easily approximated since, unlike the case of $k = 0$, the corresponding Schur complements no longer belong to the set of k-th order up-Laplacians; nevertheless, it is still possible in the case when the corresponding simplicial complex is collapsible in the topological sense. More precisely, we introduce the concept of weak collapsibility as a sufficient topological property for an efficient Cholesky multiplier. We demonstrate that the question of weak collapsibility is polynomially solvable and can be consistently checked by a greedy algorithm; then we show that one can efficiently precondition the original simplicial complex through the

fast Cholesky decomposition of its collapsible subcomplex; for this, we prove a general result of the preconditioning quality over the set of subcomplexes and tie it to the cumulative weight of the subcomplex, thus motivating a search for a "heaviest possible" collapsible subcomplex. Finally, we propose an algorithm that efficiently constructs a heavy weakly collapsible subcomplex $\mathcal{L}$ and uses its Cholesky multiplier $C(\mathcal{L})$ as a preconditioner for the original up-Laplacian $L_k^{\uparrow}(\mathcal{K})$; the developed algorithm is then tested on the synthetic triangulation dataset vis-a-vis improvement of the condition number and the number of iterations for one of the iterative linear solvers (CGLS).

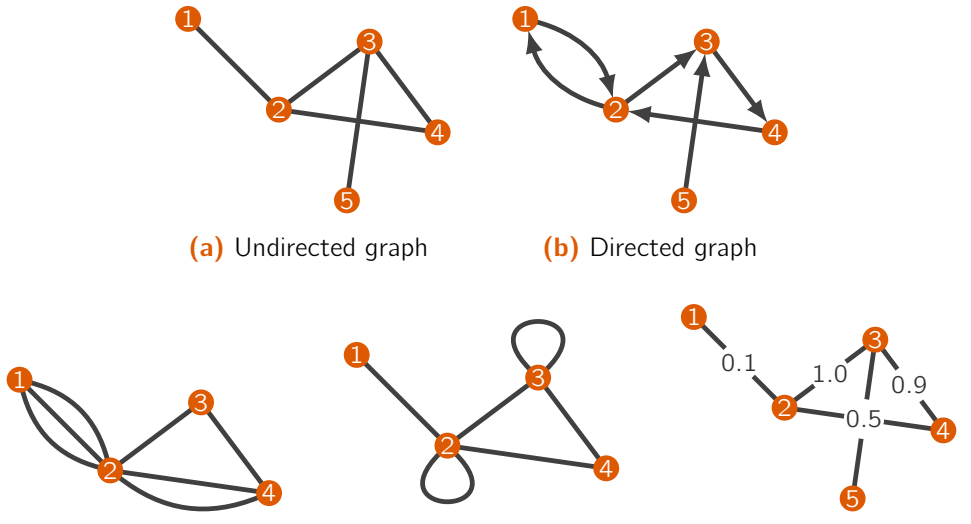
The remainder of the thesis is organized as follows: next Chapter II discusses graph models and motivations for the higher-order structures and in-depth introduces all the machinery and notions associated with weighted simplicial complexes and their homology groups. Chapter III formulates the question of topological stability, describes the framework of spectral matrix nearness problems in the general case, and discusses the principal spectral inheritance and its effect on the spectral matrix nearness problem; then, it formulates and demonstrates the performance of the developed bi-level optimization method which avoids homological pollution. In Chapter IV, the effects of the computational stability and overall idea and convergence of the iterative solvers are introduced. Building upon the already defined notion of collapsible simplicial complexes, we introduce weakly collapsibility motivated by the structure of Schur complements for higher-order Laplacians and describe the heavy weakly collapsible subcomplex (`HeCS`) preconditioning method. The last chapter concludes and discusses several possible future prospects of research.

II Simplicial complex as Higher-order Topology Descrip-

I. From graph to higher-order models

Graph $\mathcal{G}$ is a pair $\mathcal{G} = (\mathcal{V}_0, \mathcal{V}_1)$ with $\mathcal{V}_1 \subseteq \mathcal{V}_0 \times \mathcal{V}_0$ where $\mathcal{V}_0 = \{v_1, v_2, \dots v_{m_0}\}$ is the set of nodes (or vertices) corresponding to agents in the system and $\mathcal{V}_1$ is the collection of pairs of nodes describing the relational data (i.e. the structure of interactions) between the nodes; we denote the overall number of edges in the graph by m_1.

One frequently considers two natural sets of graphs: undirected (such that if $[v_i, v_j] \in \mathcal{V}_1$, then $[v_j, v_i] \in \mathcal{V}_1$, or, in other words, the order of vertices in the edge does not matter) and directed (where the order of vertices in the edge do matter), Figure I.1a and Figure I.1b. Additionally, one frequently requires $\mathcal{V}_1$ to be a set so that an edge can enter into the graph only one time; otherwise, such edges are known as **multiple edges**, Figure I.1c; edges $[v_i, v_i]$ are known as **loops** whose existence typically depends on the structure of interactions graph $\mathcal{G}$ describes, Figure I.1d. Finally, graphs can be generalized from the combinatorial to the weighted case where each node and edge is assigned (typically non-negative) weights, f.i. corresponding to the intensity of connection, resistance, etc., Figure I.1e, [W+01].



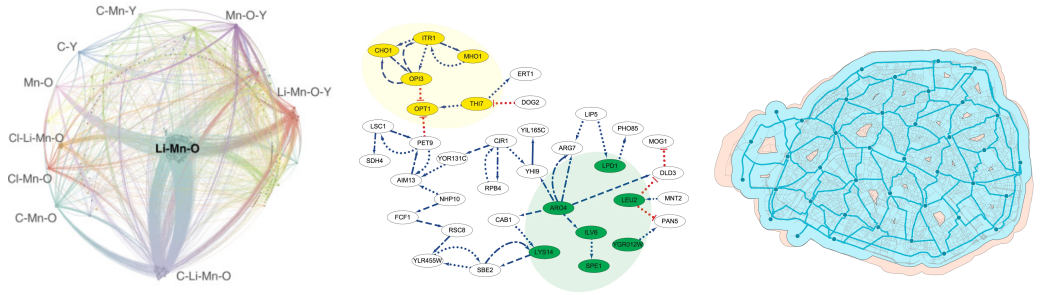
(a) Undirected graph **(b)** Directed graph

(c) Undirected graph with multiple edges **(d)** Undirected graph with loops **(e)** Undirected weighted graph

Figure I.1: Examples of basic graph structures

Graph models of multi-agent systems are ubiquitous throughout the sciences, including chemistry, where the network is typically built between a set

of reactants and products evolving in time inside a closed system, [TZB96, MDP21]; biology with concepts such as gene regulatory networks or by injecting community structure into the mean-field epidemiological models like SIR, [MV07, SB07]; to model traffic flows in the transportation network and social interactions, [New06, YL14, GLF+19], and as a general data abstraction in a variety of machine learning frameworks via graph neural networks, GNNs, [ZCH+20, ZJL+22].We provide several examples of real-life networks in Figure I.2.



(a) Chemical reaction network for C–Cl–Li–Mn–O–Y chemical system. Adapted from [MDP21]

(b) Gene regulation network for chemi-yeast. Adapted from [CZHZ19]

(c) Transportation network for bycicle reach in Paris. Adapted from [SMP+22]

Figure I.2: Examples of real-life graph networks

For each graph $\mathcal{G}$, one can define a vast number of matrices containing its structure. Here and after, we are going to consider the unweighted, undirected graph without loops and multiple edges for simplicity. Typically, the most common graph matrices are:

- ♢ **incidence matrix** $B \in \mathbb{R}^{m_0 \times m_1}$: the matrix maps vertices to edges directly, so $B_{ik} \neq 0$ if and only if node v_i belongs to the k-th edge; normally, if the k-th edge is $[v_i, v_j]$, then one of the elements, say B_{ik}, is assigned to be -1 (**tail**) and the other one, B_{jk}, is set to 1 (**head**). However, one can have $B_{ik} = B_{jk}$ when the ordering is not relevant depending on specific applications;

- ♢ **adjacency matrix** $A \in \mathbb{R}^{m_0 \times m_0}$ has 1s only in elements corresponding to existing edges: $A_{ij} = 1 \iff [v_i, v_j] \in \mathcal{V}_1$ and $A_{ij} = 0$ otherwise. Powers of adjacency matrix A^l contain the number of paths between each pair of nodes of length exactly l; by definition of directed and undirected graphs, the adjacency matrix A is symmetric ($A = A^\top$) if and only if the graph $\mathcal{G}$ is undirected;

- ♢ (classical) **Laplacian matrix** $L_0 = D - A = BB^\top$ where A is the adjacency matrix and D is the diagonal matrix of nodes' degrees: $D = \mathrm{diag}(A\mathbf{1})$. Here, the degree of a node $\deg v_i$ is defined as the number of adjacent edges from $\mathcal{V}_1$ and can be calculated for each

node as $\mathbf{d} = A\mathbf{1}$. Graph Laplacian L_0 is a symmetric positive definite matrix whose spectral properties describe various topological characteristics of the structured system, and the normalized Laplacian matrix $D^{-1/2}L_0D^{-1/2}$ governs random walks on graphs.

As a product of such vast matrix machinery, graph models remain uniquely useful and efficient for cluster and general structure detection in the systems, opinion spreading, and nodal importance, [GS14, FH16, Fri91, Vig16, EH10].

I.I Higher-order Graph Models

At the same time, various systems naturally require more than dyadic interaction between agents, [BGL16, BCI+20, BGHS23]. Examples include:

- ♢ the majority of biochemical reactions includes more than two reagents (or require a catalyst or an enzyme) and more than one product [KHT09];
- ♢ in the "science of science", e.g. in co-authorship and co-citation networks such as Cora, PubMed, and DBLP, a single interaction (coauthoring a paper) is not limited to a pairwise interaction;
- ♢ similarly, social interactions, e.g. friendship graph, frequently exhibit polyadic interactions ("a friend of a friend is a friend"), [AU18, NKL19], etc.

As a result of an abundance of more than dyadic interactions in real-life systems, a number of higher-order network models, such as graph motifs, hypergraphs, cell complexes, simplicial complexes, etc., have been proposed in recent years.

Assuming one aims to maintain the pairwise structure of the graph, one may look for repeating patterns (subgraphs) in the given network known as **graph motifs** (e.g. completely connected cliques inside social networks or chain-like structures in the gene regulatory graphs), assuming their frequency is statistically significant in comparison with random model baselines; considering motifs on their own implies the consideration of the higher-order structures and frequency signatures for a number of typical motifs can then be used to classify and characterize networks and their higher-order structure. Moreover, graph motifs can arise in various applications in social sciences, [NKL19, AU18] or biological regulatory networks, [SOMMA02, BKMZ11, MSOI+02].

A direct way to introduce higher-order interactions into graph models would be to add the sets $\mathcal{V}_2 \subseteq \mathcal{V}_0 \times \mathcal{V}_0 \times \mathcal{V}_0, \ldots, \mathcal{V}_k \subseteq \mathcal{V}_0^{k+1}$ of k-th order interactions to the graph. Such structures are known as **hypergraphs** with each interaction called **hyperedge**, where $\mathcal{V}_k$ is the set of the hyperedges of order k. By their definition, hypergraph models have the ability to describe various systems with higher-order interactions, assuming the interactions are

undirected. However, somewhat limited generalizations to the directed case exist with head and tail sets specified per each hyperedge, [AL17]. One frequently meets general hypergraph models that incorporate unstructured high-order relations and provide better performance for clustering, link prediction, and opinion-spreading tasks, [Ben19, TH21, TBP21].

Naturally, one aims to extend the matrix-based graph machinery to higher-order models, but the rising complexity of such models generally prevents direct generalizations. Indeed, the simple instance of the incidence matrix in the case of a hypergraph naturally (although not necessarily) extends into the incidence tensor, where each submatrix maps nodes to the hyperedges of a chosen cardinality, or into the aggregated incidence matrix (de facto a flattened incidence tensor), [BCI+20]. A similar argument applies to the adjacency tensor. While it is undeniable that tensor models allow an impressive amount of structure exploration on par with the pairwise case, they are still far less tractable and computationally more involved than the classical graph matrix models. Note that one may attempt to model the higher-order structures through the classical pairwise models (e.g. in dual structures of the line graph), but such attempts have yet to encompass topological features of the original hypergraph.

Simplicial complexes are a higher-order model, which is, in a sense, a restriction of a general hypergraph where some additional structure is required on hyperedges and, by which, allows a matrix-based description of higher-order structures, similar to the case of pairwise graph models. We argue that simplicial complexes and the corresponding higher-order Laplacian operators L_k are well suited for simultaneously retaining tractability and providing a way to incorporate higher-order structures. We dedicate the following section to the detailed definition of simplicial complexes.

II. Simplicial Complexes

Let $\mathcal{V}_0(\mathcal{K}) = \{v_1, v_2, \dots, v_n\}$ be a set of nodes. As discussed above, such a set may refer to various interacting entities and agents in the system, e.g. neurons, genes, traffic stops, online actors, publication authors, etc. Then:

Def. 1 **(Simplicial Complex)** The collection of subsets $\mathcal{K}$ of the nodal set $\mathcal{V}_0(\mathcal{K})$ is a (abstract) simplicial complex[1] if for each subset $\sigma \in \mathcal{K}$ all its subsets σ', $\sigma' \subseteq \sigma$, enter $\mathcal{K}$ as well, $\sigma' \in \mathcal{K}$. Elements $\sigma \in \mathcal{K}$ are referred to as **simplices** and subsets σ' of a given simplex σ are known as its **faces**.

We denote a simplex σ on the set of vertices $\{u_1, u_2, \dots u_{k+1}\} \in \mathcal{V}_0(\mathcal{K})$ as $\sigma = [u_1, u_2, \dots u_{k+1}]$. Then, a simplex $\sigma \in \mathcal{K}$ on $k+1$ vertices is said to be of order k, $\mathrm{ord}(\sigma) = k$; alternatively, we refer to it as a k-order simplex or k-simplex. Let $\mathcal{V}_k(\mathcal{K})$ be the set of all k-order simplices in $\mathcal{K}$ and m_k

[1] addition of the word "abstract" to the term is more common in the topological setting

the cardinality of $\mathcal{V}_k(\mathcal{K})$, $m_k = |\mathcal{V}_k(\mathcal{K})|$; then $\mathcal{V}_0(\mathcal{K})$ is the set of nodes in the simplicial complex $\mathcal{K}$, $\mathcal{V}_1(\mathcal{K})$ — the set of edges, $\mathcal{V}_2(\mathcal{K})$ — the set of triangles, or 3-cliques, and so on, with $\mathcal{K} = \{\mathcal{V}_0(\mathcal{K}), \mathcal{V}_1(\mathcal{K}), \mathcal{V}_2(\mathcal{K}) \ldots\}$. Note that due to the inclusion rule in Definition 1, the number of non-empty $\mathcal{V}_k(\mathcal{K})$ is finite and, moreover, uninterrupted in the sense of the order: if $\mathcal{V}_k(\mathcal{K}) = \varnothing$, then $\mathcal{V}_{k+1}(\mathcal{K})$ is also necessarily empty.

Def. 2

(k-skeleton) For a given simplicial complex $\mathcal{K}$, a k-skeleton is defined as a simplicial complex $\mathcal{K}^{(k)}$ containing all simplices of $\mathcal{K}$ of order at most k,

$$\mathcal{K}^{(k)} = \cup_{i=0}^{k} \mathcal{V}_i(\mathcal{K}) \qquad \text{(Eqn. 1)}$$

For instance, the 1-skeleton is the underlying graph, and the 2-skeleton of $\mathcal{K}$ consists of all nodes, edges and triangles of $\mathcal{K}$.

It is easy to note that the k-skeleton remains a simplicial complex: if $\sigma \in \mathcal{K}^{(k)}$, then all simplices τ from the original complex $\mathcal{K}$, $\text{ord}(\tau) \leq \text{ord}(\sigma)$, belong to $\mathcal{K}^{(k)}$ by definition; then, by inclusion principle, all faces σ' of σ belong to $\mathcal{K}$ and $\text{ord}(\sigma)' < \text{ord}(\sigma) \leq k$, so all faces of σ are necessarily included in the k-skeleton.

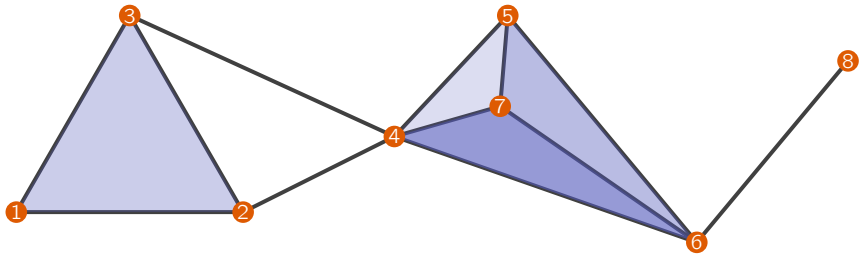


Figure II.1: Example of a simplicial complex on 8 nodes; nodes included in the complex are shown in orange, edges — in black, and triangles — in blue.

Example

Here, we provide the following example of a simplicial complex $\mathcal{K}$:

$$\begin{aligned}
\mathcal{V}_0(\mathcal{K}) &= \{[1], [2], [3], [4], [5], [6], [7], [8]\} \\
\mathcal{V}_1(\mathcal{K}) &= \{[1,2], [1,3], [2,3], [2,4], [3,4], [4,5], \\
&\qquad [4,6], [4,7], [5,6], [5,7], [6,7], [6,8]\} \\
\mathcal{V}_2(\mathcal{K}) &= \{[1,2,3], [4,5,7], [4,6,7], [5,6,7]\}
\end{aligned} \qquad \text{(Eqn. 2)}$$

Example of Simplicial Complex

In Figure II.1 we provide an illustration of $\mathcal{K}$ where we colour elements of $\mathcal{V}_0(\mathcal{K})$ (vertices) in orange, elements of $\mathcal{V}_1(\mathcal{K})$ (edges) in black, and elements of $\mathcal{V}_2(\mathcal{K})$ (triangles) in blue.
Note that $\mathcal{V}_3(\mathcal{K}) = \varnothing$, so the highest order of simplices in $\mathcal{K}$ is 2. Additionally, edges $[4,5]$, $[4,6]$ and $[5,6]$ are included in $\mathcal{K}$, but the triangle $[4,5,6]$ is not; this does not violate the inclusion rule. Instead, every edge

and every vertex of every triangle in $\mathcal{V}_2(\mathcal{K})$, as well as every vertex of every edge in $\mathcal{V}_1(\mathcal{K})$, are contained in $\mathcal{K}$, fulfilling the inclusion principle.

Example

Extension of the graph model to simplicial complexes is frequently used in studies of more intrinsic topological features of the system. For instance, one can extend a graph corresponding to the protein molecule (where nodes correspond to atoms and two nodes are connected with an edge if and only if the distance between them falls under 4Angstrom) to a simplicial complex with inclusion of triangles and tetrahedrons, Figure II.2.

Real Life Simplicial Complex

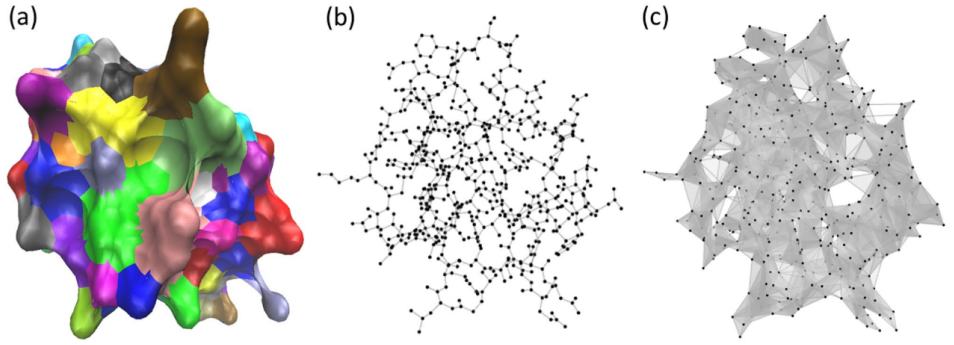


Figure II.2: Modeling molecule's structure of 2OFS protein via simplicial complex: (a) surface representation; (b) graph model; (c) simplicial complex (3-skeleton). Adapted from [WWLX22]

Compared to the definition of the hypergraph above, it is easy to see that a simplicial complex is a particular case of a hypergraph where every hyperedge is enclosed with respect to the inclusion (every subset of every hyperedge is a hyperedge). In other words, the simplicial complex contains additional structural rigidness, which allows one to formally describe the topology of $\mathcal{K}$; as a result, one is specifically interested in the formal description of the nested inclusion principle achieved through **boundary operators** defined in the subsections below.

Prior to discussing boundary mappings, we briefly cover the algebraic structure of such operators known as **Hodge's theory**.

III. Hodge's Theory

Two linear operators A and B are said to satisfy Hodge's theory if and only if their composition is a null operator,

$$AB = 0 \tag{Eqn. 3}$$

which is equivalent to $\text{im}\, B \subseteq \ker A$.

Def. 3

For a pair of operators A and B satisfying Hodge's theory, the **quotient**

space $\mathcal{H}$ is defined as follows:

$$\mathcal{H} = {}^{\ker A}\!/_{\operatorname{im} B} \tag{Eqn. 4}$$

where each element of $\mathcal{H}$ is an affine space $\mathbf{x} + \operatorname{im} B = \{\mathbf{x} + \mathbf{y} \mid \forall \mathbf{y} \in \operatorname{im} B\}$ for $\mathbf{x} \in \ker A$. It follows directly from the definition that $\mathcal{H}$ is an abelian group under addition.

By Definition 3, the quotient space $\mathcal{H}$ is a collection (in a general sense) of **equivalence classes** $\mathbf{x} + \operatorname{im} B$. Then, each class $\mathbf{x} + \operatorname{im} B = \mathbf{x}_H + \operatorname{im} B$ for some $\mathbf{x}_H \perp \operatorname{im} B$ (both $\mathbf{x}, \mathbf{x}_H \in \ker A$); indeed, since the orthogonal component $\mathbf{x}_H$ (referred as **harmonic representative**) of $\mathbf{x}$ with respect to $\operatorname{im} B$ is unique, the map $\mathbf{x}_H \leftrightarrow \mathbf{x} + \operatorname{im} B$ is a bijection.

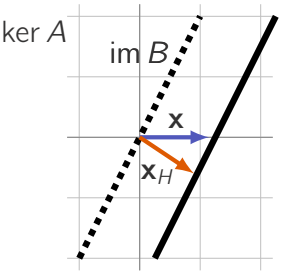


Figure III.1: Illustration of a harmonic representative for an equivalence class

Th II.III.1 **([Lim20, Thm 5.3])** Let A and B be linear operators, $AB = 0$. Then the homology group $\mathcal{H}$ satisfies:

$$\mathcal{H} = {}^{\ker A}\!/_{\operatorname{im} B} \cong \ker A \cap \ker B^\top, \tag{Eqn. 5}$$

where $\cong$ denotes the isomorphism[4].

Proof One builds the isomorphism through the harmonic representative, as discussed above. It is sufficient to note that $\mathbf{x}_H \perp \operatorname{im} B \Leftrightarrow \mathbf{x}_H \in \ker B^\top$ to complete the proof. ■

Lem III.1 **([Lim20, Thm 5.2])** Let A and B be linear operators, $AB = 0$. Then:

$$\ker A \cap \ker B^\top = \ker \left(A^\top A + B B^\top\right) \tag{Eqn. 6}$$

Proof Note that if $\mathbf{x} \in \ker A \cap \ker B^\top$, then $\mathbf{x} \in \ker A$ and $\mathbf{x} \ker B^\top$, so $\mathbf{x} \in \ker \left(A^\top A + B B^\top\right)$. As a result, $\ker A \cap \ker B^\top \subset \ker \left(A^\top A + B B^\top\right)$. On the other hand, let $\mathbf{x} \in \ker \left(A^\top A + B B^\top\right)$, then

$$A^\top A \mathbf{x} + B B^\top \mathbf{x} = 0 \tag{Eqn. 7}$$

Exploiting $AB = 0$ and multiplying the equation above by $B^\top$ and A one gets the following:

$$\begin{aligned} B^\top B B^\top \mathbf{x} &= 0 \\ A A^\top A \mathbf{x} &= 0 \end{aligned} \tag{Eqn. 8}$$

Note that $A A^\top A \mathbf{x} = 0 \Leftrightarrow A^\top A \mathbf{x} \in \ker A$, but $A^\top A \mathbf{x} \in \operatorname{im} A^\top$, so by Fredholm alternative, $A^\top A \mathbf{x} = 0$. Finally, for $A^\top A \mathbf{x} = 0$:

$$A^\top A \mathbf{x} = 0 \Longrightarrow \mathbf{x}^\top A^\top A \mathbf{x} = 0 \iff \|A\mathbf{x}\|^2 = 0 \Longrightarrow \mathbf{x} \in \ker A \tag{Eqn. 9}$$

[4] Two vector spaces V and W over the same field F are isomorphic if there is a bijection $T : V \mapsto W$ which preserves addition and scalar multiplication that is, for all vectors $\mathbf{u}, \mathbf{v} \in V$ and all $c \in F$

$$T(\mathbf{u}+\mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v}),$$

Similarly, for the second equation, $\mathbf{x} \in \ker B^\top$, which completes the proof. ■

By Theorem II.III.1 and Lemma III.1, the quotient space $\mathcal{H} = {}^{\ker A}\!/_{\operatorname{im} B} \cong \ker\left(A^\top A + BB^\top\right)$, so instead of the complicated structure of the equivalence classes $\mathcal{H}$ one can investigate a more manageable kernel of $A^\top A + BB^\top$ operator.

Additionally, the kernel above provides a natural decomposition of $\mathbb{R}^n$. Since $AB = 0$, $B^\top A^\top = 0$ or $\operatorname{im} A^\top \subset \ker B^\top$. Then, exploiting $\mathbb{R}^n = \ker A \oplus \operatorname{im} A^\top$:

$$\begin{aligned} \ker B^\top = \ker B^\top \cap \mathbb{R}^n = \ker B^\top \cap \left(\ker A \oplus \operatorname{im} A^\top\right) = \\ = \left(\ker A \cap \ker B^\top\right) \oplus \left(\operatorname{im} A^\top \cap \ker B^\top\right) \end{aligned} \tag{Eqn. 10}$$

Given Lemma III.1, $\ker A \cap \ker B^\top = \ker\left(A^\top A + BB^\top\right)$ and, since $\operatorname{im} A^\top \subset \ker B^\top$, $\operatorname{im} A^\top \cap \ker B^\top = \operatorname{im} A^\top$, yeilding the decomposition of the whole space:

Th II.III.2 **(Hodge Decomposition)** Let A and B be linear operators, $AB = 0$. Then:

$$\mathbb{R}^n = \overbrace{\operatorname{im} A^\top \oplus \underbrace{\ker\left(A^\top A + BB^\top\right)}_{\ker A}}^{\ker B^\top} \oplus \operatorname{im} B \tag{Eqn. 11}$$

IV. Boundary and Laplacian Operators

IV.I Boundary operators B_k

Each simplicial complex $\mathcal{K}$ has a nested structure of simplicies: indeed, if σ is a simplex of order k, $\sigma \in \mathcal{V}_k(\mathcal{K})$, then all of $(k-1)$-th order faces forming the boundary of σ also belong to $\mathcal{K}$: for instance, for the triangle $\{1,2,3\}$ all the border edges $\{1,2\}$, $\{1,3\}$ and $\{2,3\}$ are also in the simplicial complex, Figure II.1.

This nested property implies that one can build a formal map from a simplex to its boundary enclosed inside the simplicial complex.

Def. 4 **(Chain spaces)** Let $\mathcal{K}$ be a simplicial complex; then the space $\mathcal{C}_k$ of formal sums of simplices from $\mathcal{V}_k(\mathcal{K})$ over real numbers is called **a k-th chain space**.

Chain spaces on their own are naturally present in the majority of the network models: $\mathcal{C}_0$ is a space of states of vertices (e.g. in the dynamical system $\dot{\mathbf{x}} = A\mathbf{x}$, the evolving vector $\mathbf{x} \in \mathcal{C}_0$), $\mathcal{C}_1$ — is a space of (unrestricted) flows on graphs edges, and so on.

Example

We provide an example of chains from $\mathcal{C}_0$, $\mathcal{C}_1$ and $\mathcal{C}_2$ in Figure IV.1:

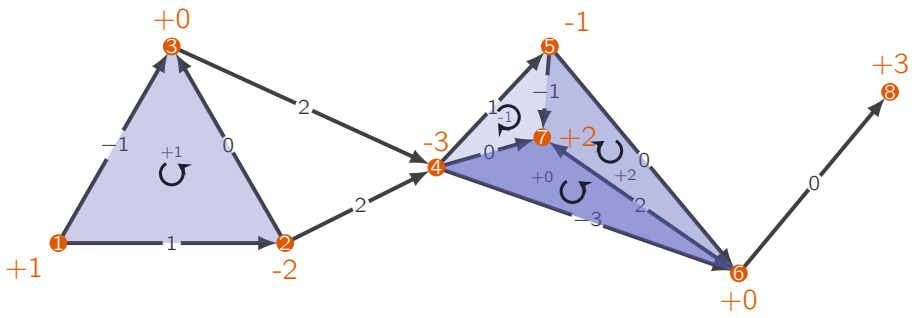


Figure IV.1: Example of chains on the simplicial complex

$$\mathbf{c}_0 = [1] - 2[2] - 3[4] - [5] + 2[7] + 3[8]$$
$$\mathbf{c}_1 = [1,2] - [1,3] + 2[2,4] + 2[3,4] + [4,5] - 3[4,6] - [5,7] + 2[6,7]$$
$$\mathbf{c}_2 = [1,2,3] - [4,5,7] + 2[5,6,7] \qquad \text{(Eqn. 12)}$$

Since $\mathcal{C}_k$ is a linear space, the elements of $\mathcal{V}_k(\mathcal{K})$ are a natural basis of $\mathcal{C}_k$ and $\mathcal{C}_k \cong \mathbb{R}^{m_k}$ with versor vectors forming the basis and corresponding to simplices in $\mathcal{V}_k(\mathcal{K})$. For instance, in Example 3:

$$\mathbf{c}_0 = \begin{pmatrix} 1 & -2 & 0 & -3 & -1 & 0 & 2 & 3 \end{pmatrix}^\top$$
$$\mathbf{c}_1 = \begin{pmatrix} 1 & -1 & 0 & 2 & 2 & 1 & -3 & 0 & -1 & 0 & 2 & 0 \end{pmatrix}^\top \qquad \text{(Eqn. 13)}$$
$$\mathbf{c}_2 = \begin{pmatrix} 1 & -1 & 0 & 2 \end{pmatrix}^\top$$

To obtain a canonical matrix representation of any operator acting on chain spaces $\mathcal{C}_k$, it is natural to order simplices in $\mathcal{V}_k(\mathcal{K})$ in some way. Additionally, one introduces the notion of **orientation** of each simplex in $\mathcal{C}_k$, e.g. for simplex $\sigma = [u_1, u_2, \ldots u_{k+1}]$ the orientation may be assigned as the permutation sign, $\mathrm{sgn}(u_1, u_2, \ldots u_{k+1})$. We provide examples of oriented simplices in Figure IV.1 in the case of the lexicographical orientation defined through the permutation sign above. Note that neither the ordering of simplices nor their orientation should be able to substantially alter the topological properties of the simplicial complex if defined correctly.

To form a boundary map, one aims to replicate the action of the operator on Figure IV.2: to map a simplex (f.i. $[1,2,3]$) to some combination of faces on its border (in case of Figure IV.2, $[1,2]$, $[1,3]$, $[2,3]$). This implies that a boundary operator B_k should map $\mathcal{C}_k$ onto $\mathcal{C}_{k-1}$. Formally,

Def. 5

Let $\mathcal{K}$ be a simplicial complex with the corresponding family of chain spaces $\mathcal{C}_k$. Then the action of a boundary map B_k, $B_k : \mathcal{C}_k \mapsto \mathcal{C}_{k-1}$, is defined

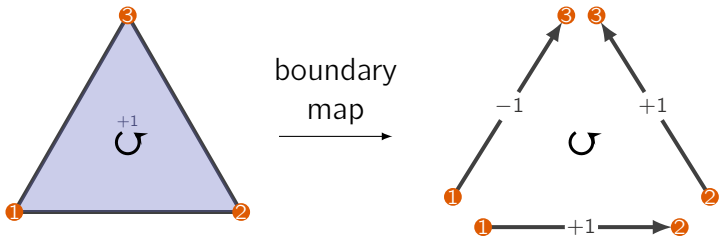


Figure IV.2: Sample action of the boundary operators

as an alternating sum:

$$B_k[u_1, u_2, \dots u_{k+1}] = \sum_{i=1}^{k+1} (-1)^i [u_1, u_2, \dots u_{i-1}, u_{i+1}, \dots u_{k+1}] \quad \text{(Eqn. 14)}$$

In the case of Figure IV.1,

$$B_2[1, 2, 3] = [1, 2] - [1, 3] + [2, 3] \quad \text{(Eqn. 15)}$$

The alternating nature of the definition upholds so-called **fundamental lemma of homology** stating "the boundary of the boundary is zero". Indeed,

$$\begin{aligned} B_1 B_2[1, 2, 3] &= B_1([1, 2] - [1, 3] + [2, 3]) = \\ &= [1] - [2] - [1] + [3] + [2] - [3] = 0 \end{aligned} \quad \text{(Eqn. 16)}$$

Lem IV.2 **(Fundamental Lemma of Homology, FLoH)** Let $\mathcal{K}$ be a simplicial complex with corresponding boundary operators B_k. Then:

$$B_k B_{k+1} = 0 \quad \text{(Eqn. 17)}$$

Proof It is sufficient to directly calculate the action of the composition of B_k and B_{k+1} on $\sigma = [u_1, u_2, \dots u_{k+2}]$:

$$B_k B_{k+1}[u_1, u_2, \dots u_{k+2}] = B_k \left(\sum_{i=1}^{k+2} (-1)^i [u_1, u_2, \dots u_{i-1}, u_{i+1}, \dots u_{k+2}] \right) =$$

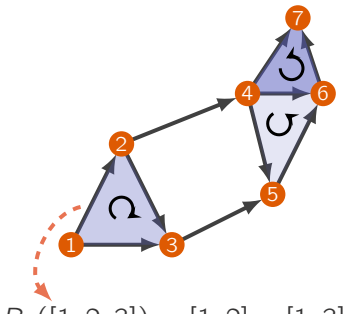
$$= \sum_{i=1}^{k+2} (-1)^i B_k[u_1, u_2, \dots u_{i-1}, u_{i+1}, \dots u_{k+2}] =$$

$$= \sum_{i=1}^{k+2} (-1)^i \left(\sum_{j=1}^{i-1} (-1)^j [u_1, u_2, \dots u_{j-1}, u_{j+1}, \dots u_{i-1}, u_{i+1}, \dots u_{k+2}] + \right.$$
$$\left. + \sum_{j=i+1}^{k+2} (-1)^{j-1} [u_1, u_2, \dots u_{i-1}, u_{i+1}, \dots u_{j-1}, u_{j+1}, \dots u_{k+2}] \right) =$$
$$= \sum_{i=1}^{k+2} \sum_{j=1}^{i-1} (-1)^{i+j} [u_1, u_2, \dots u_{j-1}, u_{j+1}, \dots u_{i-1}, u_{i+1}, \dots u_{k+2}] +$$
$$- \sum_{i=1}^{k+2} \sum_{j=i+1}^{k+2} (-1)^{i+j} [u_1, u_2, \dots u_{i-1}, u_{i+1}, \dots u_{j-1}, u_{j+1}, \dots u_{k+2}] =$$
$$= \sum_{\substack{i,j=1 \\ j<i}}^{k+2} (-1)^{i+j} [u_1, u_2, \dots u_{j-1}, u_{j+1}, \dots u_{i-1}, u_{i+1}, \dots u_{k+2}] +$$
$$- \sum_{\substack{i,j=1 \\ j>i}}^{k+2} (-1)^{i+j} [u_1, u_2, \dots u_{i-1}, u_{i+1}, \dots u_{j-1}, u_{j+1}, \dots u_{k+2}] = 0$$

(Eqn. 18)

For the final nullification, it is sufficient to notice that two terms coincide upon the interchange $i \leftrightarrow j$. ■

Since we already established basis in $\mathcal{C}_k$ and $\mathcal{C}_{k-1}$ via elements of $\mathcal{V}_k(\mathcal{K})$ and $\mathcal{V}_{k-1}(\mathcal{K})$ respectively, for the rest of the work we assume boundary operators B_k in the matrix form, $B_k \in \mathbb{R}^{m_{k-1} \times m_k}$, see an example in Figure IV.3. Matrices B_k are naturally sparse and are de facto oriented incidence matrices for higher-order structures; specifically, as seen on Figure IV.3, B_1 is known in the classical graph models as **incidence matrix**.



$$B_1 = \begin{array}{c|ccccccccccc} & \begin{bmatrix}1\\2\end{bmatrix} & \begin{bmatrix}1\\3\end{bmatrix} & \begin{bmatrix}2\\3\end{bmatrix} & \begin{bmatrix}2\\4\end{bmatrix} & \begin{bmatrix}3\\5\end{bmatrix} & \begin{bmatrix}4\\5\end{bmatrix} & \begin{bmatrix}4\\6\end{bmatrix} & \begin{bmatrix}4\\7\end{bmatrix} & \begin{bmatrix}5\\6\end{bmatrix} & \begin{bmatrix}6\\7\end{bmatrix} \\ \hline [1] & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ [2] & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ [3] & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ [4] & 0 & 0 & 0 & 1 & 0 & -1 & -1 & -1 & 0 & 0 \\ [5] & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ [6] & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \\ [7] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$$B_2 = \begin{array}{c|ccc} & \begin{bmatrix}1\\2\\3\end{bmatrix} & \begin{bmatrix}4\\5\\6\end{bmatrix} & \begin{bmatrix}4\\6\\7\end{bmatrix} \\ \hline [1,2] & 1 & 0 & 0 \\ [1,3] & -1 & 0 & 0 \\ [2,3] & 1 & 0 & 0 \\ [2,4] & 0 & 0 & 0 \\ [3,5] & 0 & 0 & 0 \\ [4,5] & 0 & 1 & 0 \\ [4,6] & 0 & -1 & 1 \\ [4,7] & 0 & 0 & -1 \\ [5,6] & 0 & 1 & 0 \\ [6,7] & 0 & 0 & 1 \end{array}$$

Figure IV.3: Left-hand side panel: example of simplicial complex $\mathcal{K}$ on 7 nodes and of the action of B_2 on the 2-simplex $[1,2,3]$; 2-simplices included in the complex are shown in red, arrows correspond to the orientation. Panels on the right: matrix forms B_1 and B_2 of boundary operators, respectively.

IV.II Homology group and Hodge Laplacians L_k

IV.II.I Homology group as Quontinent space

For a given simplicial complex $\mathcal{K}$, each pair of boundary maps B_k and B_{k+1} satisfy Hodge's theory due to the Fundamental Lemma of Homology, Lemma IV.2, which means that this special case of quotient space $\ker B_k / \operatorname{im} B_{k+1}$ is correctly defined:

Def. 6 **(Homology group)** Let $\mathcal{K}$ be a simplicial complex with corresponding boundary maps B_k. Then, the quotient space:

$$\mathcal{H}_k = \ker B_k / \operatorname{im} B_{k+1} \tag{Eqn. 19}$$

is referred as k-th **homology group** of the simplicial complex $\mathcal{K}$.

The homology group, on its own, is an object of quite a high level of abstraction that can be met in various areas of algebra. Instead, since $\mathcal{H}_k$ connects simplices and their borders by definition, we exploit the very first definition of the homology group in the algebraic topology as a way to define and categorize k-dimensional holes in $\mathcal{K}$, [Lim20].

Rem IV.1 The actual connection between the homology group and the corresponding k-dimensional holes is quite challenging to describe: the elements of $\ker B_k$ are k-dimensional cycles (e.g. elements in $\ker B_1$ correspond to cycles on graphs) whereas quotient by $\operatorname{im} B_{k+1}$ distinguishes between cycles encircling k-dimensional holes, [Hat05]. Typically, in the continuous case, Figure IV.4A, this notion is given by "fillability" or contractivity of the cycles: indeed, blue and red cycles on Figure IV.4A cannot be filled or contracted to a single point since they are encircling two different holes in the manifold; instead, the green cycles can be filled and contracted to a point (so it corresponds to $\mathbf{0}$ harmonic representative).

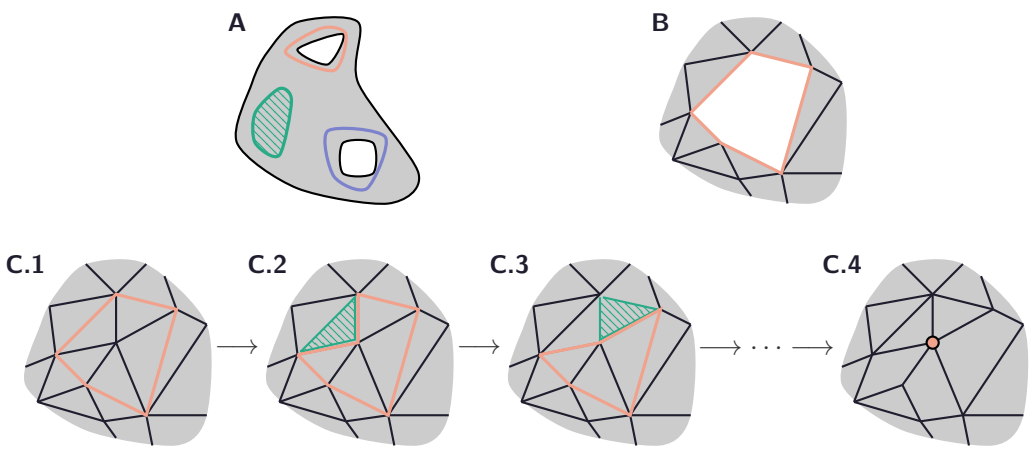


Figure IV.4: Connection between the homology group and k-dimensional holes: contractivity ("fillability") and non-contractivity of cycles in continuous (A) and discrete cases (B and C): gradual contraction of a cycle not containing a hole is given on (C1)–(C4).

This mechanism is much more tractable in the case of a simplicial complex Figure IV.4B,C: the space im B_{k+1} is spanned by circulation on the boundaries of simplices from $\mathcal{V}_{k+1}(\mathcal{K})$ (shown in green triangles on Figure IV.4C). As a result, each equivalence class in $\mathcal{H}_k$ starts from some cycle (in red) and then adds and subtracts circulations around simplices from $\mathcal{V}_{k+1}(\mathcal{K})$, Figure IV.4C1–C4; consequently, each cycle may contract only into a single point (Figure IV.4C4) or k-dimensional holes (Figure IV.4B) which are non-contractable.

Finally, although a simplicial complex $\mathcal{K}$ is not a manifold, it may be seen as a discretization of a manifold, Figure IV.5. In particular, one can show the convergence of the discrete homology group $\mathcal{H}_k$ to its continuous counterpart in case of $k = 1$, in the thermodynamic limit, [CZHZ19, CM21]; analogous results holds for the classical graph Laplacian L_0 in the case of $k = 0$, [GTGHS20].

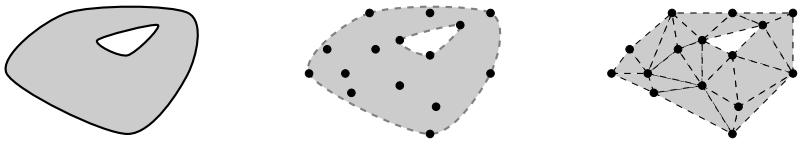


Figure IV.5: Continuous and analogous discrete manifolds with one 1-dimensional hole ($\dim \overline{\mathcal{H}}_1 = 1$). Left pane: the continuous manifold; center pane: the discretization with mesh vertices; right pane: a simplicial complex built upon the mesh. Triangles in the simplicial complex $\mathcal{K}$ are colored gray (right).

IV.II.II Elements of the Hodge decomposition as harmonic/vorticity/potential flow

Since the pair of boundary operators B_k and B_{k+1} satisfy Hodge's theory, the simplicial complex-specific case of Hodge's decomposition, Theorem II.III.2,

holds:

$$\mathbb{R}^{m_k} = \overbrace{\operatorname{im} B_k^\top \oplus \underbrace{\ker\left(B_k^\top B_k + B_{k+1}B_{k+1}^\top\right)}_{\ker B_k} \oplus \operatorname{im} B_{k+1}}^{\ker B_{k+1}^\top} \quad \text{(Eqn. 20)}$$

Def. 7 **(Hodge Laplacian Operator)** Let $\mathcal{K}$ be a simplicial complex with corresponding boundary maps B_k. Then due to Theorem II.III.1 and Lemma III.1,

$$\mathcal{H}_k \cong \ker\left(B_k^\top B_k + B_{k+1}B_{k+1}^\top\right) \quad \text{(Eqn. 21)}$$

Operator $L_k = B_k^\top B_k + B_{k+1}B_{k+1}^\top$ is known as k-th **Hodge Laplacian** (or **higher-order Laplacian**) operator. Two terms $L_k^\downarrow = B_k^\top B_k$ and $L_k^\uparrow = B_{k+1}B_{k+1}^\top$ are known as the k-th **down**- and **up**-Laplacians, respectively.

As established above, the homology group $\mathcal{H}_k \cong \ker L_k$ consists of **harmonic representative** or **harmonic chains**. Elements of the remaining components of the decomposition can be described through the analogy with differential operator on simplicial complexes. For instance,

1. the conjugate boundary map $B_1^\top$ is a discrete gradient on the graph: $B_1^\top[u_1, u_2] = [u_2] - [u_1]$. Hence $B_1^\top = \operatorname{grad}$ and $B_1 = -\operatorname{div}$ is a divergence;

2. the conjugate boundary map $B_2^\top$ is a discrete curl operator: $B_2^\top[u_1, u_2, u_3] = [u_1, u_2] - [u_1, u_3] + [u_2, u_3] = [u_1, u_2] + [u_2, u_3] + [u_3, u_1]$; note that the fundamental lemma of homology de facto restates the widely known fact $\operatorname{curl}\operatorname{grad} = 0$;

3. the 1-st order Hodge Laplacian operator then can be rewritten as a composition of the differential operators:

$$L_1 = -\operatorname{grad}\operatorname{div} + \operatorname{curl}^*\operatorname{curl} \quad \text{(Eqn. 22)}$$

   The operator $-\operatorname{grad}\operatorname{div} + \operatorname{curl}^*\operatorname{curl}$ is known as a **Helmholz operator**, an analog of the continuous differential operator on manifolds, [Han02].

Following this, the elements of $\operatorname{im} B_1^\top = \operatorname{im}(\operatorname{grad})$ are referred to as **potential flows** (since each element y_i of a vector $\mathbf{y} = B_1^\top \mathbf{x}$ is a difference of potentials between some pair of nodes α and β forming the i-th edge) or, frequently, **gradient flows**; similarly, elements of $\operatorname{im} B_2 = \operatorname{im}(\operatorname{curl}^*)$ are **vector potentials**, **vorticities** or simply **curl flows**, and $\ker B_1$ and $\ker B_2^\top$

are divergence- and curl-free flows respectively (a more low-level discussion of these two subspaces is provided further).

Rem IV.2 Considerations above provide a solid intuition but lack clear formality: indeed, to properly define the graph's gradient, divergence, and curl operators, one would need to discuss alternating functions on a graph (cochains) and corresponding coboundary operator and cohomology, [Lim20], and show a direct connection with discrete differential forms on manifolds. We choose to refrain from introducing another quite abstract entity since it should not affect the clarity of the numerical analysis of the Laplacian operators conducted further.

IV.II.III Laplacian operators L_k

The k-th order Hodge Laplacian operator $L_k = B_k^\top B_k + B_{k+1} B_{k+1}^\top$ naturally joins the boundary relational information about simplices of different orders in $\mathcal{K}$ and describes the topological structure of the complex.

In its matrix form, L_k is a symmetric ($L_k^\top = L_k$) and semi-positive definite operator; indeed, ($\mathbf{x}^\top L_k \mathbf{x} = \mathbf{x}^\top B_k^\top B_k \mathbf{x} + \mathbf{x}^\top B_{k+1} B_{k+1}^\top \mathbf{x} = \|B_k \mathbf{x}\|^2 + \|B_{k+1}^\top \mathbf{x}\|^2 \geq 0$. Moreover, individual entries of down- and up-Laplacians $L_k^\downarrow$ and $L_k^\uparrow$ describe oriented adjacencies of simplices in $\mathcal{V}_k(\mathcal{K})$. Namely, two simplices $\sigma, \sigma' \in \mathcal{V}_k(\mathcal{K})$ are upper-adjacent if they belong to the same higher-order simplex $\tau \in \mathcal{V}_{k+1}(\mathcal{K})$. Similarly, two simplices $\sigma, \sigma' \in \mathcal{V}_k(\mathcal{K})$ are down-adjacent if they have a common face $\tau \in \mathcal{V}_{k-1}(\mathcal{K})$, $\tau \subset \sigma, \sigma'$. In either case, simplices σ, σ' can be adjacent with similar or dissimilar orientations: the common face in the down-adjacency can agree (e.g., edge $[1,2]$ in the triangle $[1,2,3]$ in Figure IV.1) or disagree (e.g., edge $[1,3]$ in the triangle $[1,2,3]$) with the orientation of σ and σ'. If the common face simultaneously agrees or simultaneously disagrees with both orientations σ and σ', it is referred to as a similarly oriented case and dissimilar otherwise. The given definition immediately extends to the case of upper-adjacency. Then

$$[L_k^\downarrow]_{ij} = \begin{cases} k+1, & \text{if } i = j \\ 1, & \text{if } i \neq j \text{ and } \sigma_i, \sigma_j \text{ are upper-adjacent with similar orientation} \\ -1, & \text{if } i \neq j \text{ and } \sigma_i, \sigma_j \text{ are upper-adjacent with dissimilar orientation} \\ 0 & \text{otherwise} \end{cases} \tag{Eqn. 23}$$

$$[L_k^{\uparrow}]_{ij} = \begin{cases} \deg(\sigma_i), & \text{if } i = j \\ 1, & \text{if } i \neq j \text{ and } \sigma_i, \sigma_j \text{ are down-adjacent with similar orientation} \\ -1, & \text{if } i \neq j \text{ and } \sigma_i, \sigma_j \text{ are down-adjacent with dissimilar orientation} \\ 0 & \text{otherwise} \end{cases}$$

(Eqn. 24)

where $\deg(\sigma_i)$ is the number of simplices in $\mathcal{V}_{k+1}(\mathcal{K})$ having σ_i as a face, [LCK+19].

Conj. As follows from Definition 7, the homology group $\mathcal{H}_k \cong \ker L_k$; hence, elements of $\ker L_k$ describe the k-dimensional holes in the simplicial complex $\mathcal{K}$. The dimensionality of the kernel of the Hodge Laplacian coincides with the number of the k-dimensional holes and is frequently referred to as k-th Betti number,

$$\beta_k = \dim \ker L_k \qquad \text{(Eqn. 25)}$$

IV.II.IV Classical Laplacian and its kernel elements

Homology groups described above are not necessarily devoted to the case of higher-order interaction (or, equally, high-order simplicial complexes $\mathcal{K}$). Indeed, the 0-order homology group is defined correctly for 0-skeleton of any simplicial complex, or, in other words, the classical graph.

Assuming the absent boundary of a node, $B_0 = 0$, the 0-order Hodge Laplacian or classical graph Laplacian L_0 is defined as

$$L_0 = B_1 B_1^{\top} \qquad \text{and} \qquad \mathcal{H}_0 = \ker B_1^{\top} \qquad \text{(Eqn. 26)}$$

Alternatively, the graph Laplacian can be defined as $L_0 = D - A$ where A is the adjacency matrix of the graph ($a_{ij} = 1 \iff [v_i, v_j] \in \mathcal{V}_1(\mathcal{K})$) and D is the diagonal matrix of nodes' degrees ($D_{ii} = \deg v_i = \#\{v_j \mid [v_i, v_j] \in \mathcal{V}_1(\mathcal{K})\}$ or $D = \operatorname{diag}(A\mathbf{1})$). The classical Laplacian matrix is well-known and has numerous applications in network science, such as spectral clustering, random walks, graph partitioning, etc.

Elements of the homology group $\mathcal{H}_0$ are relatively simple to describe: they correspond to **connected components**. A subset of nodes $X \subseteq \mathcal{V}_0(\mathcal{K})$ is called a connected component if every pair of nodes $v_i, v_j \in X$ is connected through a path on the graph's edges from $\mathcal{V}_1(\mathcal{K})$ and no other vertex can be added to X. Then the indicator vector of the connected component $\mathbb{1}_X$ falls in the kernel of $B_1^{\top}$. Indeed, note that if $\mathbf{x} \in \ker B_1^{\top}$, then $\mathbf{x}_i = \mathbf{x}_j$ for every $[v_i, v_j] \in \mathcal{V}_1(\mathcal{K})$ and $\mathbf{x}_i = \mathbf{x}_j$ for every pair of vertices v_i and v_j connected by a path, hence $\mathbf{x}_i = \mathbf{x}_j$ for all vertices in the same connected component. It is immediate to notice that indicators of connected components are linearly independent and, thus, span $\ker B_1^{\top}$, so $\beta_0 = \ker L_0 =$ number of connected

components.

Rem IV.3 Note that if the graph is connected, then it has only one connected component with $\mathbf{1} \in \ker L_0$ as the only indicator of the component. Then the second smallest eigenvalue λ_2 of L_0 is positive. Eigenvalue λ_2 is known as Fiedler number or **algebraic connectivity** of the graph and can be used as an indicator of the graph being connected: the graph is connected if and only if the second smallest eigenvalue λ_2 of its classical Laplacian L_0 is strictly positive, [Fie89, Che15].

IV.II.V Kernel elements of L_1

As described above, harmonic representatives from $\ker L_0$ explicitly describe the 0-dimensional holes (connected components) in the simplicial complex by being indicator vectors of corresponding connected components. This notion suggests a similar possible connection between elements of $\ker L_k$ and k-dimensional holes they correspond to, although such connection is far less trivial and tractable, as we demonstrate in the case of $k = 1$.[5]

Firstly, one needs to acknowledge the mismatch between the intuitive definition of the hole in the simplicial complex and elements of the homology group $\mathcal{H}_1$. One could attempt to search for a hole as a simple cycle minimal by inclusion such that no subset of the vertices in the cycle form another, shorter cycle; we call such structures "naive holes". Then, one arrives at the following contradiction:

[5] we assert that similar considerations can be easily done in a general case

Example Assume that $\mathcal{K}$ is a simplicial complex containing a completely connected graph of m_0 nodes and $\mathcal{V}_2(\mathcal{K}) = \varnothing$; then $L_1 = B_1^\top B_1$. Since in that case $m_1 = \frac{m_0(m_0-1)}{2}$, $\dim\ker L_1 \leq \frac{m_0(m_0-1)}{2} - 1$. In this setup, every face (=every triangle) is a "naive hole"; as a result, the number of triangles is $\binom{m_0}{3}$, which is asymptotically much larger than the simplest upper bound on the dimensionality of the kernel.

Example 4 implies that the **first Betti number** β_1 defines the number of **different** kind of holes; naturally since we use the dimensionality, some type of "linear independence" should be implied.

Let us remind that $\ker L_1 = \ker B_1 \cap \ker B_2^\top$, so each flow harmonic flow $\mathbf{x}$ simultaneously belongs to $\ker B_1$ and $\ker B_2^\top$. These subspaces can be characterized as follows:

- ♢ we refer to the elements of $\ker B_1$ as **balanced circulations** since this subspace contains flows with equal inflows and outflow per vertex. Indeed, let v_i be an arbitrary vertex; then, for $\mathbf{x} \in \ker B_1$, it holds that

$$\sum_{\substack{v_j \in \mathcal{V}_0(\mathcal{K}),\, v_i \leftarrow v_j \\ [v_j, v_i] \in \mathcal{V}_1(\mathcal{K})}} x_{[v_i, v_j]} = \sum_{\substack{v_j \in \mathcal{V}_0(\mathcal{K}),\, v_i \rightarrow v_j \\ [v_i, v_j] \in \mathcal{V}_1(\mathcal{K})}} x_{[v_i, v_j]} \qquad \text{(Eqn. 27)}$$

where $v_i \leftarrow v_j$ and $v_i \rightarrow v_j$ denote the orientation of the edges $[v_i, v_j]$. Namely, all edges on the left-hand side are influx one for the i-th vertex, and all the edges on the right-hand side are outflux; thus, the relation states equal total inflows and outflows at each v_i;

♢ at the same time, elements of $\ker B_2^\top$ correspond to the zero circulations around each triangle in $\mathcal{V}_2(\mathcal{K})$. In other words, for each triangle $[v_i, v_j, v_k]$, it holds that $x_{[v_i,v_j]} + x_{[v_j,v_k]} + x_{[v_i,v_k]} = 0$ (assuming $v_i \rightarrow v_j$, $v_j \rightarrow v_k$, $v_i \leftarrow v_k$; otherwise, the flow circulates around the triangle through the edges with contradicting orientation reflected in the "$-$" sign before $x_{[v_i,v_j]}$).

Def. 8 The flow $\mathbf{x} \in \mathcal{C}_1$ is referred as **indicator flow** for each simple cycle $(v_1, \ldots, v_p)$ where each $[v_i, v_{i+1}] \in \mathcal{V}_1(\mathcal{K})$ (following the definition of the "naive holes" above), if $\mathbf{x} \in \{0, 1, -1\}^{m_1}$ and

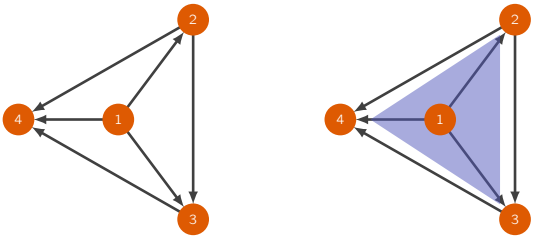
$$x_{[v_i,v_{i+1}]} = \begin{cases} 1, & v_i \rightarrow v_{i+1} \\ -1, & v_i \leftarrow v_{i+1} \end{cases}$$

for each i and $v_{p+1} = v_1$; otherwise $x_j = 0$. Basically, this vector describes the path through the cycle with respect to the edge orientation.

Rem IV.4 It is easy to see that if $B_2 \equiv 0$, then any indicator flow $\mathbf{x}$ for any simple cycle lies in $\ker B_1$. This does not necessarily hold for $\ker B_2^\top$; instead, one may find the corresponding harmonic representative by the following alternating procedure: initialize with the indicator flow $\mathbf{x}$; then, update $\mathbf{x} \rightarrow \mathbf{x}'$ such that $\mathbf{x}' \in \ker B_2^\top$ (balance the circulation around each triangle); then, fix $\mathbf{x}'$ such that accumulated flow per each vertex is still 0, and repeat until convergence.
This process is equivalent to considering the homology generator in $\mathbb{Z}/3\mathbb{Z}$ (which is the indicator flow), lifting it to $\mathbb{R}$ and then projecting it onto $\mathcal{H}_1$.

In the setup of Example 4 and Remark IV.4, it is clear that every naive hole indeed lies in $\ker L_1$ and is represented by its **indicator flow**. Thus, while we do not find the number of naive holes with the dimensionality of the Hodge Laplacian's kernel, we find the number of "essential" ones, such that other naive holes are linearly dependent on essentials in the sense of the indicator flows.

Example If we consider simplicial complex $\mathcal{K}$ consisting of the completely connected graph on 4 vertices, $\mathcal{C}_4$, with $\mathcal{V}_2(\mathcal{K}) = \varnothing$, we get the spectrum $\sigma(L_1) = \{0, 0, 0, 4, 4, 4\}$, so the kernel has dimension 3 (Figure IV.6a). Let us



(a) No faces in $\mathcal{V}_2(\mathcal{K})$

(b) One (blue) face in $\mathcal{V}_2(\mathcal{K})$

Figure IV.6: Example for the complete graph of 4 vertices; orientation is shown by arrow, and each edge has the same weight.

order the edges lexicographically[6] and consider three indicator flows:

$$\mathbf{x}_{123} = \begin{pmatrix} 1 & -1 & 0 & 1 & 0 & 0 \end{pmatrix}^T$$
$$\mathbf{x}_{124} = \begin{pmatrix} 1 & 0 & -1 & 0 & 1 & 0 \end{pmatrix}^T$$
$$\mathbf{x}_{134} = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 1 \end{pmatrix}^T$$

Then, the indicator flow $\mathbf{x}_{234} = \begin{pmatrix} 0 & 0 & 0 & 1 & -1 & 1 \end{pmatrix}^T$ can be written as $\mathbf{x}_{234} = \mathbf{x}_{123} - \mathbf{x}_{124} + \mathbf{x}_{134}$.
Now what happens if $[2, 3, 4] \in \mathcal{V}_2(\mathcal{K})$, as in Figure IV.6b? Then $\sigma(L_1) = \{0, 0, 3, 4, 4, 4\}$. This is counter-intuitive since 3 naive holes from the previous case are still present ($[1, 2, 3]$, $[1, 2, 4]$ and $[1, 3, 4]$), but now they are **not independent** in a sense that the 0-circulation through $[2, 3, 4]$ cycle in $\mathcal{V}_2(\mathcal{K})$ binds them together.

[6] e.g. $[v_1, v_2]$, $[v_1, v_3]$, $[v_1, v_4]$, $[v_2, v_3]$, $[v_2, v_4]$, $[v_3, v_4]$

One can interpret the latter in the following way: every "naive" hole has a degree of freedom represented by its accumulated flow; generally, each simplex in $\mathcal{V}_2(\mathcal{K})$ eliminates one degree of freedom. At the same time, in some configurations, the accumulated flow through the "naive" hole is fully determined by other accumulated flows of adjacent "naive" holes (see Figure IV.6a), thus stripping the cycle of its own degree of freedom.

We summarize all the information and terms around the subspaces induced by the homology group $\mathcal{H}_1$ in Table 1.

Subspace	Continuous counterpart	Elements' Names
$\ker L_1$	$\ker(-\operatorname{grad}\operatorname{div} + \operatorname{curl}^* \operatorname{curl})$	harmonic flows / chains
$\operatorname{im} B_1^\top$	$\operatorname{im}\operatorname{grad}$	potential / gradient flows
$\operatorname{im} B_2$	$\operatorname{im}\operatorname{curl}^*$	vorticities / curl flows
$\ker B_1$	$\ker\operatorname{div}$	solenoidal / balanced circulations
$\ker B_2^\top$	$\ker\operatorname{curl}$	curl-free / irrotational flows

Table 1: Naming conventions

V. Weighted and Normalized Boundary Operators

In the case of classical graph models, one can frequently find a generalization to the case of weighted graphs where nodes and edges are assigned some, normally non-negative, weight. Thus, it is natural to consider the weighted case for simplicial complexes:

Def. 9 **(Weight functions)** The family of functions $w_k : \mathcal{V}_k(\mathcal{K}) \mapsto \mathbb{R}_+$ mapping each simplex from $\mathcal{V}_k(\mathcal{K})$ into strictly positive weights is called the family of weight functions. Additionally, W_k is called a weight matrix for k-th order simplices from $\mathcal{V}_k(\mathcal{K})$ iff W_k is diagonal and $[W_k]_{ii} = w_k(\sigma_i)$ for each $\sigma_i \in \mathcal{V}_k(\mathcal{K})$.

Upon introducing the weights W_k, one needs to extend the definition of the boundary operators B_k to the weighted case $\overline{B}_k$. Since each boundary map B_k acts from the chain space C_k to C_{k-1}, f.i. edges to nodes or triangles to edges, it is natural to weigh rows and columns of each B_k by the corresponding weight matrices, $\overline{B}_k = g(W_{k-1})B_k h(W_k)$. More importantly, the corresponding weighted homology group $\overline{\mathcal{H}}_k$ should be properly defined, so that $\overline{B}_k\overline{B}_{k+1} = 0$, e.g. with $g \equiv h^{-1}$. In our work, we propose the following weighting scheme,

$$B_k \longrightarrow \overline{B}_k = W_{k-1}^{-1} B_k W_k,$$

which contains combinatorial and various particular weight choices throughout the literature, [LCK+19, CZHZ19, SBH+20, BGB22].

Note that, from the definition $\overline{B}_k = W_{k-1}^{-1}B_kW_k$ and the Fundamental Lemma of Homology, Lemma IV.2, we immediately have that $\overline{B}_k\overline{B}_{k+1} = 0$. Thus, the group $\overline{\mathcal{H}}_k = \ker \overline{B}_k / \operatorname{im} \overline{B}_{k+1}$ is well defined for any choice of positive weights w_k and is isomorphic to $\ker \overline{L}_k$. Note that, in general, one could aim for a more comprehensive non-typical weighting scheme provided the fundamental lemma of homology holds and the homology group is correctly defined.

While the homology group may depend on the weights, we observe below that its dimension does not. Precisely, we have:

Prop. The dimension of the homology groups of $\mathcal{K}$ is not affected by the weights of its k-simplicies. Precisely, if W_k are positive diagonal matrices, we have

$$\dim \ker \overline{B}_k = \dim \ker B_k, \quad \dim \ker \overline{B}_k^\top = \dim \ker B_k^\top, \quad \dim \overline{\mathcal{H}}_k = \dim \mathcal{H}_k \,. \tag{Eqn. 28}$$

Moreover, $\ker B_k = W_k \ker \overline{B}_k$ and $\ker B_k^\top = W_{k-1}^{-1} \ker \overline{B}_k^\top$.

Proof Since W_k is an invertible diagonal matrix,

$$\overline{B}_k\mathbf{x} = 0 \iff W_{k-1}^{-1}B_kW_k\mathbf{x} = 0 \iff B_kW_k\mathbf{x} = 0.$$

Hence, if $\mathbf{x} \in \ker \overline{B}_k$, then $W_k\mathbf{x} \in \ker B_k$, and, since W_k is bijective, $\dim\ker \overline{B}_k = \dim\ker B_k$. Similarly, one observes that $\dim\ker \overline{B}_k^\top = \dim\ker B_k^\top$.
Moreover, since $\overline{B}_k\overline{B}_{k+1} = 0$, then $\operatorname{im}\overline{B}_{k+1} \subseteq \ker\overline{B}_k$ and $\operatorname{im}\overline{B}_k^\top \subseteq \ker\overline{B}_{k+1}^\top$. This yields $\ker\overline{B}_k \cup \ker\overline{B}_{k+1}^\top = \mathbb{R}^{\mathcal{V}_k} = \ker B_k \cup \ker B_{k+1}^\top$.
Thus, for the homology group, it holds:

$$\begin{aligned}\dim\overline{\mathcal{H}}_k &= \dim\left(\ker\overline{B}_k \cap \ker\overline{B}_{k+1}^\top\right) = \\ &= \dim\ker\overline{B}_k + \dim\ker\overline{B}_{k+1}^\top - \dim\left(\ker\overline{B}_k \cup \ker\overline{B}_{k+1}^\top\right) = \\ &= \dim\ker B_k + \dim\ker B_{k+1}^\top - \dim\left(\ker B_k \cup \ker B_{k+1}^\top\right) = \dim\mathcal{H}_k\end{aligned}$$

■

One should also note that rescaling operators B_k does not preserve the elements of the kernel spaces themselves, so $\dim\ker\overline{B}_k = \dim\ker B_k$ but $\ker\overline{B}_k \neq \ker B_k$. Nevertheless, all the definitions above are immediately extendable to the weighted case. For instance, one can rewrite the characterizations for flows in $\ker\overline{B}_1$ and $\ker\overline{B}_2^\top$ as done above:

$$\mathbf{x} \in \ker\overline{B}_1 \iff \sum_{\substack{v_j\in\mathcal{V}_0(\mathcal{K}),\, v_i\leftarrow v_j\\ [v_j,v_i]\in\mathcal{V}_1(\mathcal{K})}} w_1([v_i,v_j])x_{[v_i,v_j]} = \sum_{\substack{v_j\in\mathcal{V}_0(\mathcal{K}),\, v_i\rightarrow v_j\\ [v_i,v_j]\in\mathcal{V}_1(\mathcal{K})}} w_1([v_i,v_j])x_{[v_i,v_j]}$$

$$\mathbf{x} \in \ker\overline{B}_2^\top \iff \frac{x_{[v_i,v_j]}}{w_1([v_i,v_j])} + \frac{x_{[v_j,v_k]}}{w_1([v_j,v_k])} + \frac{x_{[v_i,v_k]}}{w_1([v_i,v_k])} = 0 \qquad \text{(Eqn. 29)}$$

with all the homology group machinery remaining intact. As a result, the weighted k-th order Laplacian operator has the form:

$$\overline{L}_k = \underbrace{W_kB_k^\top W_{k-1}^{-2}B_kW_k}_{\overline{L}_k^\downarrow} + \underbrace{W_k^{-1}B_{k+1}W_{k+1}^2B_{k+1}^2W_k^{-1}}_{\overline{L}_k^\uparrow} \qquad \text{(Eqn. 30)}$$

Rem V.5 **(Normalisation)** Frequently, an extension of any model to the weighted case requires some form of normalization. For instance, one can find weighted graph Laplacians $\overline{L}_0$ in the form $\overline{L}_0 = B_1W_1^2B_1^\top$ (so $W_0 = I$) with the whole spectrum and various other matrix entities being scaled by W_1. Instead, assuming $W_0 = f(W_1)$ allows to guarantee $\sigma(L_0) \subset [0;\lambda]$ for some fixed dimensionless λ: typically, assuming the weight of the ver-

tex is the sum of all adjacent edges, $w_0(v_i) = \sum_{(v_i,v_j)\in\mathcal{V}_1(\mathcal{K})} w_1([v_i, v_j])$, one obtains $\sigma(L_0) \subset [0; 2]$.
Similarly, W_k and W_{k-1}^{-1} in the weighted $\overline{B}_k$ can be used as a normalizing tool depending on the relation between W_k and W_{k-1} for the adjacent simplices. Normalization is, however, not guaranteed for any choice of weight functions $w_k(\cdot)$: indeed, the unweighted combinatorial case $w_k(\sigma) = 1$ satisfies the proposed weighting scheme without providing any normalization for $\sigma(L_k)$. Note that one can guarantee normalization for $L_k^{\uparrow}$ for similar weight choices $w_k(\sigma_i) = \sum_{[\sigma_i,v_j]\in\mathcal{V}_{k+1}(\mathcal{K})} w_{k+1}([\sigma_i, v_j])$ (where by $[\sigma_i, v_j]$ we mean a simplicial complex that includes σ_i as a face with an additional node v_j), but as we discuss further, such choice of weights maybe counter-intuitive for a lot of tasks.

Lem V.3 Let $\overline{L}_k^{\uparrow} = \overline{B}_{k+1}\overline{B}_{k+1}^{\top} = \left(W_k^{-1}B_{k+1}W_{k+1}\right)\cdot\left(W_k^{-1}B_{k+1}W_{k+1}\right)^{\top}$ with $w_k^2(\sigma_i) = \sum_{[\sigma_i,v_j]\in\mathcal{V}_{k+1}(\mathcal{K})} w_{k+1}^2([\sigma_i, v_j])$. Then $\sigma(\overline{L}_k^{\uparrow}) \subseteq [0; 2]$.

Proof It is immediate to see that $\overline{L}_k^{\uparrow}$ remains a semi-positive definite operator, so each $\lambda(\overline{L}_k^{\uparrow}) \geq 0$. Then, in the chosen weighting scheme, all diagonal elements of $\overline{L}_k^{\uparrow}$ are exactly 1:

$$\left(\overline{L}_k^{\uparrow}\right)_{ii} = \left\langle \left(\overline{B}_{k+1}\right)_{i,\cdot}, \left(\overline{B}_{k+1}\right)_{i,\cdot} \right\rangle = \| \left(\overline{B}_{k+1}\right)_{i,\cdot} \|^2 = \frac{1}{w_k^2(\sigma_i)} \| \left(B_{k+1}W_{k+1}\right)_{i,\cdot} \|^2 =$$
$$= \frac{\sum_{[\sigma_i,v_j]\in\mathcal{V}_{k+1}(\mathcal{K})} w_{k+1}^2([\sigma_i, v_j])}{w_k^2(\sigma_i)} = 1 \qquad \text{(Eqn. 31)}$$

Similarly,

$$\sum_j \left(\overline{L}_k^{\uparrow}\right)_{ij} = \sum_j \left\langle \left(\overline{B}_{k+1}\right)_{i,\cdot}, \left(\overline{B}_{k+1}\right)_{j,\cdot} \right\rangle = \sum_j \mathbb{1}_{[\sigma_i,v_j]\in\mathcal{V}_{k+1}(\mathcal{K})} \frac{w_{k+1}^2([\sigma_i, v_j])}{w_k^2(\sigma_i)} = 1 \qquad \text{(Eqn. 32)}$$

so by Gershgorin circle theorem, [Ger31], $\sigma(\overline{L}_k^{\uparrow}) \subseteq [0; 2]$. ■

Rem V.6 Symmetric normalization of the classical graph Laplacian $L_0 = L_0^{\uparrow}$ by $D^{-1/2}$ where D is the diagonal matrix of nodes' degrees (or weighted degrees) is coherent with Lemma V.3 and yields the same bounds on the spectrum.

Rem V.7 On the contrary to the normalization idea, where the weight of low-order simplex $w_k(\sigma_i)$ is inferred from the weights of (adjacent) higher-order simplices $w_{k+1}([\sigma_i, v_j])$, one frequently needs or chooses to infer the weights of higher-order simplices from the low-order ones (e.g. weights of triangles from the weights of edges, etc.); most frequent choices are

♢ **min-rule**: the weight of the triangle is the minimal weight of adjacent

edges, $w_2(\tau) = \min\{w_1(\sigma_1), w_1(\sigma_2), w_1(\sigma_3)\}$ where triangle τ is formed by edges σ_1, σ_2, σ_3, [LCK+19];

♢ **product**: the weight of the triangle is the product of weights of adjacent edges, $w_2(\tau) = \sqrt[3]{w_1(\sigma_1)w_1(\sigma_2)w_1(\sigma_3)}$, [CM21].

III Topological Stability as MNP

I. General idea of the topological stability

The concept of the homology group $\overline{\mathcal{H}_k}$ provides a new, more detailed look into the topology of the simplicial complex and, thus, the underlying system/data. Moreover, curl/gradient/harmonic decomposition in the Hodge decomposition, Theorem II.III.2, allows separating any simplicial trajectory (e.g. random walk on $\mathcal{V}_k(\mathcal{K})$) into the meaningful topological components. At the same time, one can also imagine that not all structural data is perfectly given without mistakes, random noise on the weights, or may be targeted by attacks. For instance, simplicial complexes and graphs created by sensor networks tend to have extra erroneous or missing connections, affecting the structure and certainly affecting the topology. Thus, it raises a more general question:

How stable is the homology group $\overline{\mathcal{H}_k}$?

Posing such a question, one needs to determine what we mean by stability: indeed, as stated in Proposition 1, a perturbation of weights $\{W_k\}$ that does not create vanishing weights does not change the dimensionality of the homology group $\overline{\mathcal{H}_k}$ but clearly alters the space of harmonic representatives $\ker \overline{L}_k$. As a result, one may be interested in a more drastic perturbation of structure, such that it changes the dimensionality of $\overline{\mathcal{H}_k}$; specifically, we focus on finding the smallest perturbation that creates a new principal k-dimensional hole in the complex.

Formally, suppose we are given a simplicial complex $\mathcal{K} = \{\mathcal{V}_0(\mathcal{K}), \mathcal{V}_1(\mathcal{K}), \mathcal{V}_2(\mathcal{K}), \ldots\}$ with weight functions $w_0, w_1, \ldots$, and let $\beta_k = \dim \mathcal{H}_k = \dim \overline{\mathcal{H}}_k$ be the dimension of its k-homology. We aim to find the closest simplex on the same vertex set $\mathcal{V}_0(\mathcal{K})$, with a strictly larger number of holes.

Problem For a given weighted $\mathcal{K}$ and Betti number $\beta_k(\mathcal{K})$ find the closest $\mathcal{K}'$ such that $\mathcal{V}_0(\mathcal{K}') = \mathcal{V}_0(\mathcal{K})$ and

$$\beta_k(\mathcal{K}') \geq \beta_k(\mathcal{K}) + 1 \qquad \text{(Eqn. 33)}$$

In this formulation, we assume that $\mathcal{K}'$ is obtained only through the elimination of simplices; indeed, by allowing the introduction of new weighted edges, the question of creating a new hole becomes meaningless (since it

is sufficient to add an edge to compete a 4-cycle with an arbitrary small weight).

Rem I.8 Note that Problem 1 has a natural counterpart which asks to eliminate edges to decrease the dimensionality of $\overline{\mathcal{H}}_k$; we do not discuss this problem in the current work, but agree that improving the complexity of the direct combinatorial approach for it would be a worthy contribution.

While following considerations and developed approach hold in the general case of $\overline{\mathcal{H}}_k$, we assume the case of the 1st homology $\overline{\mathcal{H}}_1$ for 2-skeleton $\mathcal{K} = \{\mathcal{V}_0(\mathcal{K}), \mathcal{V}_1(\mathcal{K}), \mathcal{V}_2(\mathcal{K})\}$ and the corresponding Hodge Laplacian $\overline{L}_1$. Then, as stated above, the topological stability of $\mathcal{K}$ is defined in terms of edge elimination sufficient to increase the dimensionality of the homology group. The concept of edge elimination is, by definition, discrete and combinatorial; instead, we assume that the elimination of edges is modeled by the vanishing weights $w_1(\sigma_i) = 0$. Then perturbation of the simplicial complex $\mathcal{K} \to \mathcal{K}'$ is achieved through the perturbation of weight of the edges W_1. Note that zero weights are not allowed in Proposition 1, so one loses the conservation of the dimensionality as required by Problem 1.

Problem Let $\mathcal{K}$ be a simplicial complex of order at least 2 with associated edge weight function w_1 and corresponding diagonal weight matrix W_1, and let $\beta_1(W_1)$ be the dimension of the homology group corresponding to the weights in W_1. For $\varepsilon > 0$, let

$$\Omega(\varepsilon) = \Big\{\text{diagonal matrices } W \text{ such that } \|W\| = \varepsilon\Big\},$$
$$\Pi(W_1) = \Big\{\text{diagonal matrices } W \text{ such that } W_1 + W \geq 0\Big\}.$$

In other words, $\Omega(\varepsilon)$ is an ε-sphere and $\Pi(W_1)$ allows only non-negative simplex weights. We look for the smallest perturbation ε such that there exists a weight modification $\delta W_1 \in \Omega(\varepsilon) \cap \Pi(W_1)$ such that $\beta_1(W_1) < \beta_1(W_1 + \delta W_1)$.

Note that since $\beta_1 = \dim\ker L_1$, Problem 2 poses a question of the closest matrix with specific spectral properties (in that case, a bigger kernel), which is known as spectral matrix nearness problem. In the following subsections, we provide a short overview of the gradient flow approach for the spectral matrix nearness problem and outline several considerations necessary to modify the classical approach for the case of Problem 2.

I.1 Persistent homology as a facet of topological stability

One should also mention a preexisting, albeit somewhat restricted, way to characterize the topological stability of a graph or simplicial complex induced by a point cloud — the **persistent homology**, [OPT+17, GS23b, LLO+21].

The concept is defined as follows: let $\{\mathbf{x}_i\}_{i=1}^N$ be a point cloud in $\mathbb{R}^d$, so $\mathbf{x}_i \in \mathbb{R}^d$. Then, for a fixed filtration parameter ε, an edge $[\mathbf{x}_i, \mathbf{x}_j]$ is included in the graph if and only if nodes are at most ε-close, $\|\mathbf{x}_i - \mathbf{x}_j\| < \varepsilon$, for some chosen norm $\|\cdot\|$. Similarly, in case of the simplicial complex, the simplex $[\mathbf{x}_1, \ldots \mathbf{x}_l]$ is included in the complex if and only if $\|\mathbf{x}_i - \mathbf{x}_j\| < \varepsilon$ for all $1 \leq i, j, \leq l$ (it is immediate to see that such definition upholds the inclusion principle in the definition of simplicial complex). Note that the actual filtration mechanisms may vary, but the overall principle should remain as described above.

One can study the topology of the generated simplicial complex through the associated Hodge Laplacian $L_k(\varepsilon)$, which is, as we stress in the notation, heavily dependent on the filtration parameter ε. As a result, one can try to disturb ε hoping for a change in the homology group $\mathcal{H}_k$, which is by the very list reminiscent of the notion of topological stability: in that case, one asks for the value of the filtration parameter such that the topology of the complex changes significantly (or, in more broad terms, the value of ε where one can get a phase transition on the spectral diagram over a set of filtration parameters), Figure I.1. Similarly to Figure I.1, one can track the evolution of the spectrum of L_k between harmonic, curl, and gradient parts through filtration, [GS23a].

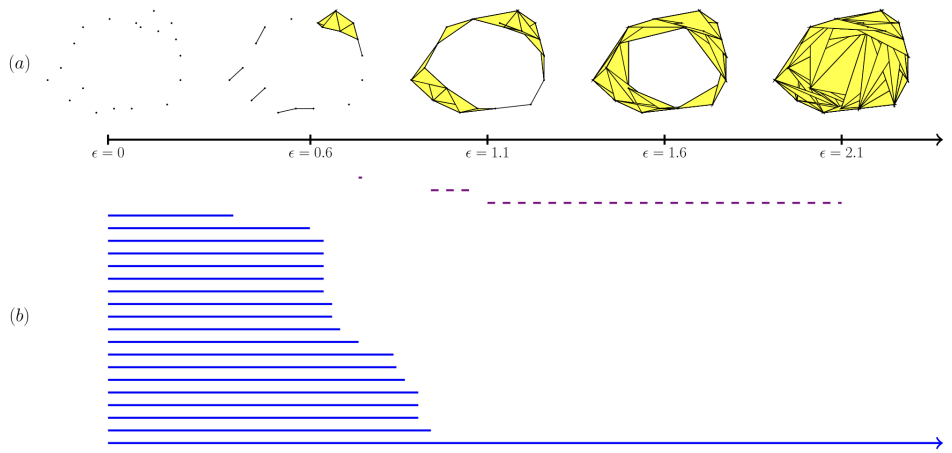


Figure I.1: Persistent homology for the graph case. (a) Examples of growing complexes along the increasing filtration parameter ε; (b) bar code of the existence of connected components (solid) and holes (dashed). Adapted from [OPT+17].

It is clear, however, that persistent homology and the stability of the persistence diagrams are primarily dependent on the setting (f.i. the existence of the point cloud), which means that at the very least one would need an embedding of a given simplicial complex in $\mathbb{R}^d$ and an appropriate choice of the norm $\|\cdot\|$ and overall filtration mechanism to study the topological stability of the complex in such a way. Additionally, the stability in terms of the filtration parameter ε would need to be interpreted in terms of the

original complex; instead, we aim to provide a computational answer for a general case of simplicial complexes without additional callbacks to the computationally heavy case of the persistent homology via spectral matrix nearness problems (MNPs).

We next proceed with a general description of the bi-level optimization framework for spectral MNPs and outline the structural problem with the direct application of the pre-existing method for $\overline{\mathcal{H}}_k$; instead, we outline the spectral relation, Theorem III.III.5, which infers complex-aware choice of the optimization problem which requires a more sensitive approach to optimization.

II. Spectral Matrix Nearness Problems: overview

In the scope of Problem 2, we aim to change the topology of the simplicial complex through the minimal perturbation δW_1 of the weight matrix W_1 that increases the dimensionality of the kernel of the Hodge Laplacian β_1. In a broad sense, this implies obtaining the nearest (in terms of δW_1) matrix $\overline{L}_1$ with the desired spectral properties, yielding a spectral matrix nearness problem. We proceed with a brief description of a general framework for such tasks.

Generally speaking, for a given matrix A, a **spectral matrix nearness problem** consists of finding the closest possible matrix X among the admissible set with a number of desired properties. For instance, one may search for the nearest (in some metric) symmetric positive/negative definite matrix, unitary matrix, or the closest graph Laplacian.

Motivated by the topological meaning of **kernels** of Hodge Laplacians L_k, we assume the specific case of **spectral** MNPs: here one aims for the target matrix X to have a particular spectrum $\sigma(X)$. For instance in the stability study of the dynamical system $\dot{\mathbf{x}} = A\mathbf{x}$ one can search for the closest Hurwitz matrix such that $\mathrm{Re}\,[\lambda_i] < 0$ for all $\lambda_i \in \sigma(X)$; similarly, assuming given matrix A is a graph Laplacian, one can search for the closest disconnected graph (so the algebraic connectivity $\lambda_2 = 0$).

Here we recite the optimization framework developed by [GL22, AEGL19, GLS23] for the class of the spectral matrix nearness problems; one should note, however, that this is by far not the only approach to the task, [GS17, DT08].

II.I Target functional for optimization

Let us assume that A is a given matrix with X being a target matrix, $X = A+\Delta$, and, instead of searching for X, we search for the perturbation matrix Δ; additionally, we assume that Ω is the admissible set containing all possible perturbations Δ such that $A+\Delta$ exhibits desired spectral properties. In broad

terms, one attempts to solve the following problem

$$\min_{\Delta\in\Omega} \|\Delta\|, \qquad \text{such that } A+\Delta \text{ satisfies certain spectral properties} \tag{Eqn. 34}$$

For the purposes of this chapter, we search for the minimal perturbation in terms of the Frobenius norm, $\|\Delta\| = \|\Delta\|_F = \left(\mathrm{Tr}(\Delta^\top\Delta)\right)^{1/2}$, although in general one can search for the minimal perturbation in other norms.

One may transform the optimization task above into a minimization problem for a target functional $F(\Delta, A)$ such that its constrained minimum corresponds to $A+\Delta$ exhibiting desired properties. In the case of the spectral MNP, one normally has the target functional directly dependent on the spectrum of $A+\Delta$, $F(A+\Delta) = F\left(\lambda(A+\Delta), \overline{\lambda}(A+\Delta)\right)$ where $\lambda(A+\Delta)$ is an eigenvalue of the perturbed matrix $A+\Delta$ and $\overline{\lambda}$ is a complex conjugate. To properly define a continuous optimization problem, we assume a smooth-enough target functional $F(\lambda,\overline{\lambda})\colon \mathbb{C}\times\mathbb{C}\mapsto\mathbb{R}$ such that $F(\lambda,\overline{\lambda}) = F(\overline{\lambda},\lambda)$, e.g. $F(\lambda,\overline{\lambda}) = \mathrm{Re}\frac{\lambda+\overline{\lambda}}{2}$ or $F(\lambda,\overline{\lambda}) = |\lambda|^2 = \lambda\overline{\lambda}$. In the scope of Hodge Laplacian operators, all considered matrices are symmetric semi-positive definite, so one can omit the dependence on the conjugate $\overline{\lambda}$ since $\sigma(L_k)\in\mathbb{R}$. Additionally, depending on the task at hand, the target functional F may depend on a number of eigenvalues of several matrices or their eigenvectors.

Rem II.9 Note that even if the original general matrix nearness problem by itself may be combinatorial, we build a continuous facet for it that is suitable for a continuous optimization routine. For instance, the search for the closest disconnected graph for a given graph Laplacian L_0 is combinatorial by definition (since we look for the set of edges to eliminate, or, in other words, a set of matrix entries to eliminate). At the same time, it can be reformulated as a continuous optimization task by bringing the weights of the edges to 0 as a way to model their elimination.

Example (Contractivity of the dynamical system) Let us assume a dynamical system $\dot{\mathbf{x}} = A\mathbf{x}$; then the contractivity of the solution is guaranteed by the one-sided Lipshitz condition:

$$\frac{1}{2}\frac{d}{dt}\|\mathbf{x}-\mathbf{y}\|^2 = \langle A(\mathbf{x}-\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle \le C\|\mathbf{x}-\mathbf{y}\|^2 \tag{Eqn. 35}$$

with $C = \lambda_{\max}\left(\frac{A+A^*}{2}\right) < 0$ being the negative logarithmic norm of the matrix. Thus, if one aims to find the nearest matrix governing a contractive dynamical system, the utmost right eigenvalue $\lambda_{\max}$ of the symmetrized operator $\frac{A+A^*}{2}$ needs to be moved to the negative hyperplane (so $\lambda_{\max}\left(\frac{A+A^*}{2}\right) < 0$). To achieve this, one may opt to use it as the

target eigenvalue in the functional, $F(\lambda, \overline{\lambda}) = \lambda_{\max}\left(\frac{A+A^*}{2}\right)$.

For the purposes of this work, we assume that $F(\lambda, \overline{\lambda}) = 0$ if and only if perturbed $A + \Delta$ lands in the set with desired spectral properties (e.g. every time the graph become disconnected, target functional vanishes) and $F(\lambda, \overline{\lambda}) > 0$ otherwise.

II.II Formulation as a bi-level optimization task

Given the target functional of the target eigenvalue $F(\lambda(A+\Delta), \overline{\lambda}(A+\Delta))$, it is convenient to write the perturbation $\Delta = \varepsilon E$ with $\|E\| = 1$, so that ε directly models the norm of the perturbation Δ, which can be interpreted as the perturbation budget, while the matrix E models the shape of the perturbation. Then, MNP given by Equation (34) can be rewritten as:

$$\min_{\varepsilon \geq 0} \varepsilon \qquad \text{such that } \exists E: \quad \begin{array}{c} \|E\| = 1 \\ \varepsilon E \in \Omega \\ F_\varepsilon(E) = F(\lambda(A+\varepsilon E), \overline{\lambda}(A+\varepsilon E)) = 0 \end{array} \qquad \text{(Eqn. 36)}$$

where Ω is a constraints set. Clearly, the set $\{E \mid F_\varepsilon(E) = 0\}$ is, in general, not trivial to obtain, and running an optimization routine on it is even less trivial. Instead, as a way to approach problem 36, one can introduce the following bi-level optimization procedure:

1. **inner level**: for a fixed perturbation norm ε find the optimal shape of the perturbation $E(\varepsilon)$ such that $F_\varepsilon(E(\varepsilon))$ is minimal:

$$E(\varepsilon) = \arg\min_{\substack{E,\ \|E\|=1 \\ \varepsilon E \in \Omega}} F_\varepsilon(E) \qquad \text{(Eqn. 37)}$$

2. **outer level**: find the smallest possible perturbation norm $\varepsilon^* > 0$ such that $F_{\varepsilon^*}(E(\varepsilon^*)) = 0$ assuming existence of solution:

$$\varepsilon^* = \arg\min_{\varepsilon > 0} F_\varepsilon(E(\varepsilon)) \qquad \text{(Eqn. 38)}$$

In the proposed scheme, the outer level is a root search/optimization problem for a single-variable function $F_\varepsilon(E(\varepsilon))$, albeit not easily obtainable, with a wide variety of solution methods. Given the complex nature of $E(\varepsilon)$, one should aim to minimize the number of calls to $F_\varepsilon(E(\varepsilon))$ in the outer level. At the same time, the inner level is a **constrained matrix optimization problem**. Additionally, it is worth noting that the eigenvalue optimization is generally non-convex and non-smooth, and the norm-constrained gradient

flow on the inner level may help to deter the overall optimization procedure away from local minima towards better optimizers.

Generally speaking, optimization on both inner and outer levels can only guarantee the computation of an approximate solution as a local minimum. Moreover, a global solution may be theoretically unobtainable: for instance, if one searches for the nearest disconnected graph through the MNP on graph Laplacian L_0, the problem is famously polynomial, [DJP+94]. However, after the introduction of cardinality or membership constraints (requiring a certain number of vertices or certain vertices in each component), the task is known to become NP-hard, [DJP+94, AEGL19], while the continuous optimization on both inner and outer levels of MNP is considered to be polynomial. In that case, the bi-level procedure would provide an ε-approximation of the optimal combinatorial solution in the polynomial time.

Transition to the gradient flow. We solve the resulting matrix optimization problem 37 on the inner level $\min\{F_\varepsilon(E)\colon \|E\| = 1 \text{ and } \varepsilon E \in \Omega\}$ by integrating the associated constrained gradient system:

$$\begin{aligned} \dot{E}(t) &= -\mathbb{P}\left(\nabla_{E(t)} F_\varepsilon(E(t)) + \kappa E(t)\right) \\ \text{s.t. } \|E(t)\| &= 1 \quad \text{and} \quad \varepsilon E(t) \in \Omega \end{aligned} \tag{Eqn. 39}$$

where $\mathbb{P}$ is the projector on the tangent plane to the admissible set $\{E\colon \varepsilon E \in \Omega\}$ and the term $\kappa E(t)$ guarantees the norm conservation $E(t) = 1$. We refer to the unconstrained part of the system above, $G(\varepsilon, E(t)) = \nabla_{E(t)} F_\varepsilon(E(t))$, as a free gradient part.

Then, the optimizer $E(\varepsilon)$ is obtained in the limit, $E(\varepsilon) = \lim_{t\to\infty} E(t)$, since we integrate the system in the direction of the anti-gradient $-\nabla_{E(t)} F_\varepsilon(E(t))$ on the admissible set. Moving from 37 to 39 is beneficial for a number of reasons, such as the possibility of using higher-order numerical integrators (assuming one is able to control higher-order derivatives of eigenvalues in $F_\varepsilon(E(t))$), but also the principal monotonicity of the functional $F_\varepsilon(E(t))$ along the flow allowing larger steps in the numerical integration, as we will demonstrate later.

II.III Inner level

Here, we describe the optimization procedure used on the inner level, which is naturally divided into the computation of the free gradient and the projection onto the appropriate set to achieve the constrained flow.

II.III.I Free gradient calculation

Let us assume that one aims to compute a free gradient of the functional

$$G(\varepsilon, E(t) = \nabla_{E(t)} F_\varepsilon(E(t))\,. \tag{Eqn. 40}$$

By the chain rule we have

$$\frac{d}{dt}F_\varepsilon(E(t)) = \left\langle G(\varepsilon, E(t)), \frac{d}{dt}E(t) \right\rangle \qquad \text{(Eqn. 41)}$$

so if $\dot{E}(t) = -G(\varepsilon, E(t)) = -\nabla_{E(t)}F_\varepsilon(E(t))$, then $\frac{d}{dt}F_\varepsilon(E(t)) = -\|G(\varepsilon, E(t))\| < 0$ and the target functional $F_\varepsilon(E(t))$ is non-increasing along the flow; similar monotonicity estimation can be shown for the constrained case, Lemma IV.9. Moreover, it is sufficient to compute the time derivative $\frac{d}{dt}F_\varepsilon(E(t))$ to isolate the free gradient part $G(\varepsilon, E(t))$.

As a result,

$$\begin{aligned}\frac{d}{dt}F_\varepsilon(E(t)) &= \frac{d}{dt}F(\lambda(A+\varepsilon E(t)), \overline{\lambda}(A+\varepsilon E(t))) = \\ &= \frac{\partial}{\partial\lambda}F(\lambda,\overline{\lambda})\cdot\frac{d}{dt}\lambda(A+\varepsilon E(t)) + \frac{\partial}{\partial\overline{\lambda}}F(\lambda,\overline{\lambda})\cdot\frac{d}{dt}\overline{\lambda}(A+\varepsilon E(t))\end{aligned} \qquad \text{(Eqn. 42)}$$

which requires a way to compute the time derivative of the eigenvalue $\lambda(A+\varepsilon E(t))$ following the classical result described below.

Let $A_0 \in \mathbb{C}^{n\times n}$ have a simple eigenvalue λ_0 corresponding to the right eigenvector $\mathbf{x}_0$ (so $A\mathbf{x}_0 = \lambda_0\mathbf{x}_0$ with $\mathbf{x}_0 \neq 0$) and the left eigenvector $\mathbf{y}_0$ (so $\mathbf{y}_0^* A_0 = \lambda_0\mathbf{y}_0^*$ with $\mathbf{y}_0 \neq 0$), normalized so that $\|\mathbf{x}_0\| = \|\mathbf{y}_0\| = 1$ and $\mathbf{y}_0^*\mathbf{x}_0 > 0$. Let $\tau_0 \in \mathbb{C}$ and let $A(\tau)$ be a complex-valued matrix function of a complex parameter τ that is analytic in a neighborhood of τ_0, satisfying $A(\tau_0) = A_0$.

Th III.II.3 **(Derivative of the eigenvalue)** $A(\tau)$ has a unique eigenvalue $\lambda(\tau)$ that is analytic in a neighborhood of τ_0, with $\lambda(\tau_0) = \lambda_0$

$$\dot{\lambda}(\tau_0) = \frac{1}{\mathbf{y}_0^*\mathbf{x}_0}\mathbf{y}_0^*\dot{A}(\tau_0)\mathbf{x}_0 \qquad \text{(Eqn. 43)}$$

We provide here a self-contained proof in the assumption of all functions $A(\tau), \mathbf{x}(\tau), \mathbf{y}(\tau)$ being analytic in the neighborhood of τ_0. One can relax this requirement, maintaining the main result, [GLO20, HJ12].

Proof Consider the equation $A(\tau)\mathbf{x}(\tau) = \lambda(\tau)\mathbf{x}(\tau)$ and its derivative at $\tau = \tau_0$:

$$\dot{A}(\tau_0)\mathbf{x}(\tau_0) + A(\tau_0)\dot{\mathbf{x}}(\tau_0) = \dot{\lambda}(\tau_0)\mathbf{x}(\tau_0) + \lambda(\tau_0)\dot{\mathbf{x}}(\tau_0) \qquad \text{(Eqn. 44)}$$

Multiplying this equation on both sides with $\mathbf{y}^*(\tau_0)$, we obtain:

$$\begin{aligned}\mathbf{y}^*(\tau_0)\dot{A}(\tau_0)\mathbf{x}(\tau_0)+\mathbf{y}^*(\tau_0)A(\tau_0)\dot{\mathbf{x}}(\tau_0)&=\dot{\lambda}(\tau_0)\mathbf{y}^*(\tau_0)\mathbf{x}(\tau_0)+\lambda(\tau_0)\mathbf{y}^*(\tau_0)\dot{\mathbf{x}}(\tau_0)\\ \mathbf{y}^*(\tau_0)\dot{A}(\tau_0)\mathbf{x}(\tau_0)+\lambda(\tau_0)\mathbf{y}^*(\tau_0)\dot{\mathbf{x}}(\tau_0)&=\dot{\lambda}(\tau_0)\mathbf{y}^*(\tau_0)\mathbf{x}(\tau_0)+\lambda(\tau_0)\mathbf{y}^*(\tau_0)\dot{\mathbf{x}}(\tau_0)\\ \mathbf{y}^*(\tau_0)\dot{A}(\tau_0)\mathbf{x}(\tau_0)&=\dot{\lambda}(\tau_0)\mathbf{y}^*(\tau_0)\mathbf{x}(\tau_0)\end{aligned} \tag{Eqn. 45}$$

Noting $\mathbf{x}(\tau_0)=\mathbf{x}_0$ and $\mathbf{y}(\tau_0)=\mathbf{y}_0$, we get $\dot{\lambda}(\tau_0)=\frac{\mathbf{y}_0^*\dot{A}(\tau_0)\mathbf{x}_0}{\mathbf{y}_0^*\mathbf{x}_0}$. ■

Combining the results above, one can compute the free gradient of the functional:

Th III.II.4 **(Free time gradient)** Let $E(t)\in\mathbb{C}^{n\times n}$ be a continuously differentiable matrix function in the neighborhood of t_0 with the derivative $\dot{E}(t)$ and let $\lambda(t)$ be the target eigenvalue of $A+\varepsilon E(t)$ which is simple and has left and right unit eigenvectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ satisfying conditions of Theorem III.II.3. Then

1. $F_\varepsilon(E(t))$ is continuously differentiable with respect to time
2. λ continuously differentiable on t in the neighborhood of t_0, λ in C^1
3. let $\kappa(t)=\frac{1}{\mathbf{x}^*(t)\mathbf{y}(t)}$ be a condition number of $\lambda(t)$, then

$$\frac{1}{\varepsilon}\kappa(t)\frac{d}{dt}F_\varepsilon(E(t))=\operatorname{Re}\left\langle G(\varepsilon,E(t)),\dot{E}(t)\right\rangle \tag{Eqn. 46}$$

4. free gradient is given by

$$G(\varepsilon,E(t))=2\frac{\partial}{\partial\overline{\lambda}}F(\lambda(A+\varepsilon E(t)),\overline{\lambda}(A+\varepsilon E(t)))\mathbf{x}(t)\mathbf{y}^*(t) \tag{Eqn. 47}$$

Proof Note that for $\frac{d}{dt}F_\varepsilon(E(t))$ it is sufficient to input $\frac{d}{dt}\lambda$ into Equation (42). As a result

$$\begin{aligned}\frac{d}{dt}F_\varepsilon(E(t))&=\frac{\varepsilon}{\mathbf{x}^*(t)\mathbf{y}(t)}\left(\frac{\partial}{\partial\lambda}F(\lambda,\overline{\lambda})\cdot\mathbf{x}^*(t)\dot{E}(t)\mathbf{y}(t)+\frac{\partial}{\partial\overline{\lambda}}F(\lambda,\overline{\lambda})\cdot\overline{\mathbf{x}^*(t)\dot{E}(t)\mathbf{y}(t)}\right)=\\&=\frac{2\varepsilon}{\mathbf{x}^*(t)\mathbf{y}(t)}\operatorname{Re}\left(\frac{\partial}{\partial\lambda}F(\lambda,\overline{\lambda})\mathbf{x}^*(t)\dot{E}(t)\mathbf{y}(t)\right)\end{aligned} \tag{Eqn. 48}$$

Now, the only thing left is the trace trick:

$$\operatorname{Re}\left(\frac{\partial}{\partial\lambda}F(\lambda,\overline{\lambda})\mathbf{x}^*(t)\dot{E}(t)\mathbf{y}(t)\right)=\operatorname{Re}\left\langle\frac{\partial}{\partial\overline{\lambda}}F(\lambda,\overline{\lambda})\mathbf{x}(t)\mathbf{y}^*(t),\dot{E}(t)\right\rangle \tag{Eqn. 49}$$

yielding the theorem. ■

II.III.II Constrained gradient flow, stationary points, and rank-1 optimizers

In the integration of the inner level, Equation (39), one needs to uphold $\|E(t)\| = 1$ (so the integration remains on the unit sphere) and $\varepsilon E(t) \in \Omega$. While the admissible set Ω is highly dependent on the task and may be quite complicated to project onto, one can always easily obtain a norm-preserving constrained flow. Indeed, if $\|E(t)\|_F = 1$, then $\|E(t)\|_F^2 = 1$ and

$$\frac{d}{dt}\|E(t)\|_F^2 = 1 \quad \Longleftrightarrow \quad 2\mathrm{Re}\left\langle E(t), \dot{E}(t)\right\rangle = 0 \qquad \text{(Eqn. 50)}$$

Hence, one can obtain the norm preserving flow from the free gradient system $\dot{E}(t) = -G(\varepsilon, E(t))$ by introducing the modified flow $\dot{E}(t) = -G(\varepsilon, E(t)) + \kappa E(t)$.

Lem II.4 **(Steepest norm-preserving descent direction)** The optimal norm-constrained descent direction for $E(t)$ given by the system Equation (39) can be written as:

$$\dot{E}(t) = -G(\varepsilon, E(t)) + \mathrm{Re}\left\langle G(\varepsilon, E(t)), E(t)\right\rangle E(t) \qquad \text{(Eqn. 51)}$$

Proof For this flow, we observe:

$$\begin{aligned} \dot{E}(t) &= -G(\varepsilon, E(t)) + \kappa E(t) \\ 0 = \left\langle E(t), \dot{E}(t)\right\rangle &= -\left\langle E(t), G(\varepsilon, E(t))\right\rangle + \left\langle E(t), \kappa E(t)\right\rangle \qquad \text{(Eqn. 52)} \\ 0 = \left\langle \dot{E}(t), E(t)\right\rangle &= -\left\langle G(\varepsilon, E(t), E(t)\right\rangle + \left\langle \kappa E(t), E(t)\right\rangle \end{aligned}$$

so $\kappa = \mathrm{Re}\left\langle G(\varepsilon, E(t)), E(t)\right\rangle$, which is precisely the projection of the flow on the unit sphere in the Frobenious norm. ■

Note that in the case of non-trivial admissible set Ω, one can directly generalize the steepest descent flow in Equation (51) given the projector $\mathbb{P}$ and the fact that $\varepsilon E(t) \in \Omega$ along the flow:

$$\dot{E}(t) = -\mathbb{P}G(\varepsilon, E(t)) + \mathrm{Re}\left\langle \mathbb{P}G(\varepsilon, E(t)), E(t)\right\rangle E(t) \qquad \text{(Eqn. 53)}$$

Lem II.5 **(Stationary points)** Assuming $\|G(\varepsilon, E(t))\| \neq 0$, the following statements about the stationary point of the norm-preserving gradient flow 51 are equivalent:

1. $\frac{d}{dt}F_\varepsilon(E(t)) = 0$

2. $\dot{E}(t) = 0$

3. $E(t)$ is a real multiple of $G(\varepsilon, E(t))$

Moreover, in a stationary point, the optimizer $E(t) = E(\varepsilon)$ is necessarily of rank-1.

Proof Since $F_\varepsilon(E(t)) = \frac{\varepsilon}{\mathbf{x}^*(t)\mathbf{y}(t)} \mathrm{Re}\left\langle G(\varepsilon, E(t)), \dot{E}(t)\right\rangle$ and $\|G(\varepsilon, E(t))\| > 0$, $\frac{d}{dt}F_\varepsilon(E(t)) = 0 \iff \dot{E}(t) = 0$ is immediate. Now, given $\dot{E}(t) = 0$, one obtains $0 = -G(\varepsilon, E(t)) + \kappa E(t)$ or $E(t) = \frac{1}{\kappa}G(\varepsilon, E(t))$. Finally, by Theorem III.II.4 $G(\varepsilon, E(t)) = 2\frac{\partial}{\partial\overline{\lambda}}F(\lambda(A + \varepsilon E(t)), \overline{\lambda}(A + \varepsilon E(t)))\mathbf{x}(t)\mathbf{y}^*(t) \propto \mathbf{x}(t)\mathbf{y}^*(t)$ which is a rank-1 matrix, so $E(t) \propto \mathbf{x}(t)\mathbf{y}^*(t)$ is also rank-1. ■

The admissible set Ω is highly dependent on the task and may affect the structure of the optimizer $E(\varepsilon)$ since the projector $\mathbb{P}$ on its tangent space enters the constrained flow, see Equation (39) and Equation (53); typical examples of such constraints are sparsity pattern (where by the nature of the task $A + \varepsilon E(t)$ is required to maintain certain structure), non-negativity of some entries (e.g. the weights of the simplices as we demonstrate below) or trivial admissible set with $\Omega = \mathbb{R}^{n\times n}$ and $\mathbb{P} = I$. Note that rank-1 optimizers for the inner-level (see Lemma II.5) are only obtainable for $\mathbb{P} = I$. In the case of non-trivial projections (e.g. Ω as a sparsity pattern), one cannot guarantee rank-1 since one no longer controls the part of the optimizer belonging to $\ker P$.

II.IV Outer level and overall optimization scheme

Assuming the integration of the flow on the inner level provides an optimizer $E(\varepsilon)$ per each ε, one needs to develop an optimization routine or a root-finding routine for the outer level in order to find the smallest possible perturbation norm ε such that $F_\varepsilon(E(\varepsilon)) = 0$ (or, alternatively, that $F_\varepsilon(E(\varepsilon))$ is the smallest possible). In a lot of cases, it is sufficient to apply preexisting methods like bisection or the Newton method:

$$\varepsilon_{k+1} = \varepsilon_k - \frac{F_{\varepsilon_k}(E(\varepsilon_k))}{\frac{d}{d\varepsilon}F_{\varepsilon_k}(E(\varepsilon_k)} = \varepsilon_k + \frac{\mathbf{x}(\varepsilon_k)^*\mathbf{y}(\varepsilon_k)}{\|G(\varepsilon_k, E(\varepsilon_k))\|^2}F_{\varepsilon_k}(E(\varepsilon_k)) \quad \text{(Eqn. 54)}$$

However, all those methods require an efficiently and correctly computed optimizer $E(\varepsilon)$ on the inner level; however, this is not guaranteed.

Note that in most cases of interest, the non-convex and non-smooth nature of the spectral functionals $F_\varepsilon(E(t))$ depend on the initialization $E(0)$ in the optimization routine, which makes the ability to inherit nearby optimizer $E(\varepsilon - \Delta\varepsilon)$ for small $\Delta\varepsilon$ as $E(0)$ for $F_\varepsilon(E(t))$ almost essential. Large jumps in ε_k in the bisectional or Newton methods may prevent such inheritance. As

a result, one may need to adjust the outer level to leverage the inheritance of the optimizer instead of using fast convergent routines like bisection or Newton method.

Amid formulating the bi-level optimization routine for a general spectral MNP, one poses a question: can such an approach be directly applied to the weighted homology group $\overline{\mathcal{H}}_k$ and its corresponding higher-order Laplacian L_k? In the next section, we demonstrate a number of problems arising that require careful consideration.

III. Direct approach: failure and discontinuity problems

In order to compute the topological stability of the weighted simplicial complex $\mathcal{K}$, one aims to find the minimal perturbation δW_1 of the weights of edges that increases the homology group, $\beta_1(W_1) < \beta_1(W_1 + \delta W_1)$, Problem 2.

To apply the developed gradient flow optimization approach for the spectral matrix nearness problem, one needs to (a) reformulate Problem 2 in terms of the spectral properties of Laplacian operators; (b) compose an appropriate target functional $F(\varepsilon, E)$ and (c) check the coherency of combinatorial-to-continuous transition along the gradient flow.

Let us start from the last point: the idea of mimicking edge elimination via $w_1(\sigma_i) + \delta w_1(\sigma_i) = 0$ requires consistent updates of weights of nodes and triangles: indeed, if for edge σ_i the perturbed weight vanishes ($w_1(\sigma_i) + \delta w_1(\sigma_i) = 0$), then every triangle τ such that $\sigma_i \subset \tau$ should also vanish, $w_2(\tau) = 0$ to adhere to the inclusion principle in the definition of the simplicial complex. As a result, if $\tilde{w}_1(\sigma) = w_1(\sigma) + \delta w_1(\sigma)$ is the new edge weight function, we require the weight function of the 2-simplices to change into $\tilde{w}_2$, defined as

$$\tilde{w}_2(i_1 i_2 i_3) = f\left(\frac{\delta w_1(i_1 i_2)}{w_1(i_1 i_2)}, \frac{\delta w_1(i_2 i_3)}{w_1(i_2 i_3)}, \frac{\delta w_1(i_1 i_3)}{w_1(i_1 i_3)}\right) \cdot w_2(i_1 i_2 i_3) \tag{Eqn. 55}$$

where $f(u_1, u_2, u_3)$ is a function such that $f(0, 0, 0) = 1$ and that monotonically decreases to zero as $u_i \to -1$, for any $i = 1, 2, 3$. An example of such f is

$$f(u_1, u_2, u_3) = 1 - \min\{u_1, u_2, u_3\}\,. \tag{Eqn. 56}$$

Rem III.10 Note that the definition above does not restrict our simplicial complex: we assume that initial weights are given without any requirements, and then we introduce the weight dependency entirely for the purposes of the optimization procedure.

Rem III.11 The fundamental difference between the combinatorial problem (=eliminating the edges) and its continuous facet of vanishing weights is that edge elimination may reduce the problem's dimensionality. Specifically, in the case of Hodge Laplacian $L_1(\mathcal{K})$, up-Laplacian $L_1(\mathcal{K}')$ is a smaller after the elimination. This, however, can not happen in the case of the vanishing weights; instead, some additional zeros may be introduced into the spectrum of $L_1(\delta W_1)$ that correspond to the reduced dimensions and not to the extended kernel. This phenomenon should be carefully checked for during the integration of the gradient flow.

Target functional in the direct approach. Aiming to add another dimension to $\overline{\mathcal{H}}_1(W_1)$, one seeks to bring another 0 into the kernel of $L_1(W_1)$; in other words, if the initial dimensionality of the homology group is $\beta_1 = \dim \overline{\mathcal{H}}_1(W_1)$, one may attempt to push the first non-zero eigenvalue λ_{β_1+1} in $\sigma(L_1)$ to 0 with a target functional:

$$F(\delta W_1) = \frac{1}{2}\lambda^2_{\beta_1+1}(L_1(W_1 + \delta W_1)) \qquad \text{(Eqn. 57)}$$

with an appropriate δW_1. Unfortunately, the direct approach via the simple functional $F(\delta W_1)$ is ultimately unsuccessful due to the spectral relations between consecutive Laplacians L_{k-1} and L_k leading to homological pollution which we describe in the following section.

III.I Principal spectral inheritance

Here, we recall a relatively direct but substantial spectral property that connects the spectra of the k-th and $(k+1)$-th order Laplacians.

Th III.III.5 **(HOL's spectral inheritance)** Let L_k and L_{k+1} be higher-order Laplacians for the same simplicial complex $\mathcal{K}$. Let $\overline{L}_k = \overline{L}^{\downarrow}{}_k + \overline{L}^{\uparrow}{}_k$, where $\overline{L}^{\downarrow}{}_k = \overline{B}_k^\top \overline{B}_k$ and $\overline{L}^{\uparrow}{}_k = \overline{B}_{k+1}\overline{B}_{k+1}^\top$. Then:

1. $\sigma_+(\overline{L}^{\uparrow}{}_k) = \sigma_+(\overline{L}^{\downarrow}{}_{k+1})$, where $\sigma_+(\cdot)$ denotes the positive part of the spectrum;
2. if $0 \neq \mu \in \sigma_+(\overline{L}^{\uparrow}{}_k) = \sigma_+(\overline{L}^{\downarrow}{}_{k+1})$, then the eigenvectors are related as follows:
   (a) if $\mathbf{x}$ is and eigenvector for $\overline{L}^{\uparrow}{}_k$ with the eigenvalue μ, then $\mathbf{y} = \frac{1}{\sqrt{\mu}}\overline{B}_{k+1}^\top \mathbf{x}$ is an eigenvector for $\overline{L}^{\downarrow}{}_{k+1}$ with the same eigenvalue;
   (b) if $\mathbf{u}$ is and eigenvector for $\overline{L}^{\downarrow}{}_{k+1}$ with the eigenvalue μ and $\mathbf{u} \notin \ker \overline{B}_{k+1}$, then $\mathbf{v} = \frac{1}{\sqrt{\mu}}\overline{B}_{k+1}\mathbf{u}$ is an eigenvector for $\overline{L}^{\uparrow}{}_k$ with the same eigenvalue;

3. for each Laplacian $\overline{L}_k$: if $\mathbf{v} \notin \ker \overline{L}^{\downarrow}{}_k$ is the eigenvector for $\overline{L}^{\downarrow}{}_k$, then $\mathbf{v} \in \ker \overline{L}^{\uparrow}{}_k$; vice versa, if $\mathbf{u} \notin \ker \overline{L}^{\uparrow}{}_k$ is the eigenvector for $\overline{L}^{\uparrow}{}_k$, then $\mathbf{v} \in \ker \overline{L}^{\downarrow}{}_k$;

4. consequently, there exist $\mu \in \sigma_+(\overline{L}_k)$ with an eigenvector $\mathbf{u} \in \ker \overline{L}^{\uparrow}{}_k$, and $\nu \in \sigma_+(\overline{L}_{k+1})$ with an eigenvector $\mathbf{u} \in \ker \overline{L}^{\downarrow}{}_{k+1}$, such that:

$$\overline{B}_k^\top \overline{B}_k \mathbf{v} = \mu \mathbf{v}, \qquad \overline{B}_{k+2} \overline{B}_{k+2}^\top \mathbf{u} = \nu \mathbf{u}\,.$$

Proof Note that if $\mathbf{x}$ is an eigenvector of $\overline{L}^{\uparrow}{}_k$, then for $\mathbf{y} = \frac{1}{\sqrt{\mu}} \overline{B}_{k+1}^\top \mathbf{x}$ one obtains

$$\overline{L}^{\downarrow}{}_{k+1} \mathbf{y} = \overline{B}_{k+1}^\top \overline{B}_{k+1} \frac{1}{\sqrt{\mu}} \overline{B}_{k+1}^\top \mathbf{x} = \frac{1}{\sqrt{\mu}} \overline{B}_{k+1}^\top \overline{L}^{\uparrow}{}_k \mathbf{x} = \sqrt{\mu} \overline{B}_{k+1}^\top \mathbf{x} = \mu \mathbf{y}, \tag{Eqn. 58}$$

so $\mathbf{y}$ is an eigenvector $\overline{L}^{\downarrow}{}_{k+1}$ giving $(2a)$.
Similarly, for $(2b)$: if $\mathbf{u}$ is an eigenvector $\overline{L}^{\downarrow}{}_{k+1}$ and $\mathbf{v} = \frac{1}{\sqrt{\mu}} \overline{B}_{k+1} \mathbf{u}$, then

$$\overline{L}^{\uparrow}{}_k \mathbf{v} = \overline{B}_{k+1} \overline{B}_{k+1}^\top \frac{1}{\sqrt{\mu}} \overline{B}_{k+1} \mathbf{u} = \frac{1}{\sqrt{\mu}} \overline{B}_{k+1} \overline{L}^{\downarrow}{}_{k+1} \mathbf{u} = \mu \mathbf{v}; \tag{Eqn. 59}$$

since in both $(2a)$ and $(2b)$ we ask for $\mu \neq 0$, joint $2(a)$ and $2(b)$ yield (1).
Hodge decomposition, Theorem II.III.2, immediately yields the strict separation of eigenvectors between $\overline{L}^{\uparrow}{}_k$ and $\overline{L}^{\downarrow}{}_k$, (3); given (3), all the inherited eigenvectors from $(2a)$ fall into the $\ker \overline{L}^{\downarrow}{}_{k+1}$, thus resulting into (4). ■

In other words, the variation of the spectrum of the k-th Laplacian when moving from one order to the subsequent one works as follows: the down-term $\overline{L}^{\downarrow}{}_{k+1}$ inherits the positive part of the spectrum from the up-term of $\overline{L}^{\uparrow}{}_k$; the eigenvectors corresponding to the inherited positive part of the spectrum lie in the kernel of $\overline{L}^{\uparrow}{}_{k+1}$; at the same time, the "new" up-term $\overline{L}^{\uparrow}{}_{k+1}$ has a new, non-inherited, part of the positive spectrum (which, in turn, lies in the kernel of the $(k+2)$-th down-term).

In particular, we notice that for $k = 0$, since $B_0 = 0$ and $\overline{L}_0 = \overline{L}^{\uparrow}{}_0$, the theorem yields $\sigma_+(\overline{L}_0) = \sigma_+(\overline{L}^{\downarrow}{}_1) \subseteq \sigma_+(\overline{L}_1)$. In other terms, the positive spectrum of the $\overline{L}_0$ is inherited by the spectrum of $\overline{L}_1$, and the remaining (non-inherited) part of $\sigma_+(\overline{L}_1)$ coincides with $\sigma_+(\overline{L}^{\uparrow}{}_1)$. Figure III.1 provides an illustration of the statement of Theorem III.III.5 for $k = 0$.

Theorem III.III.5 is built as a natural extension of Hodge decomposition, Theorem II.III.2, and the structure of $\overline{L}^{\downarrow}{}_k$ and $\overline{L}^{\uparrow}{}_k$; however, it provides a

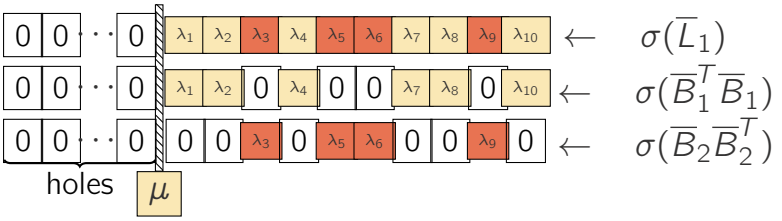


Figure III.1: Illustration for the principal spectrum inheritance (Theorem III.III.5) in case $k = 0$: spectra of $\overline{L}_1$, $\overline{L_1^{\downarrow}}$ and $\overline{L_1^{\downarrow}}$ are shown. Colors signify the splitting of the spectrum, $\lambda_i > 0 \in \sigma(\overline{L}_1)$; all yellow eigenvalues are inherited from $\sigma_+(\overline{L}_0)$; red eigenvalues belong to the non-inherited part. Dashed barrier μ signifies the penalization threshold (see the target functional in Subsection IV.I) preventing homological pollution (see Subsection III.II).

valuable description of the evolution of spectrum in terms of $\overline{L}_k$ operators. Specifically, in the case of MNPs, one needs to take into account that small and close-to-zero entries in $\sigma_+(L_k)$ may be inherited from $\sigma_+(L_{k-1})$ and, as such, may refer to a close $(k-1)$-dimensional hole instead of k-dimensional; we describe this phenomenon in details next.

III.II Homological pollution: inherited almost disconnectedness

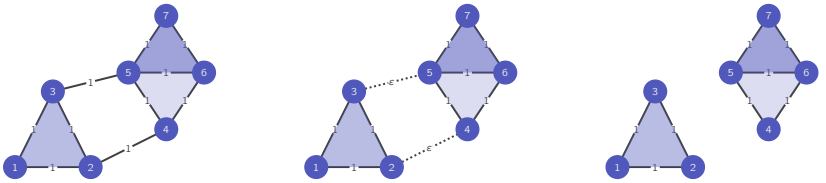
As the dimension of Hodge homology β_1 corresponds to the number of zero eigenvalues in $\overline{L}_1$, the intuition suggests that if $\overline{L}_1$ has some eigenvalue that is close to zero, then the simplicial complex is "close to" having at least one more 1-dimensional hole. There are a number of problems with this intuitive consideration.

By Theorem III.III.5 for $k = 0$, $\sigma_+(\overline{L}_1)$ inherits $\sigma_+(\overline{L}_0)$. Hence, if the weights in W_1 vary continuously so that a positive eigenvalue in $\sigma_+(\overline{L}_0)$ approaches 0, the same happens to $\sigma_+(\overline{L}_1)$. Assuming the initial graph $\mathcal{G}_\mathcal{K}$ is connected, an eigenvalue that approaches zero in $\sigma(\overline{L}_0)$ would imply that $\mathcal{G}_\mathcal{K}$ is approaching disconnectedness. This leads to a sort of **pollution** of the kernel of $\overline{L}_1$ as an almost-zero eigenvalue which corresponds to an "almost" 0-dimensional hole (disconnected component) from $\overline{L}_0$ is inherited into the spectrum of $\overline{L}_1$, but this small eigenvalue of $\overline{L}_1$ does not correspond to the creation of a new 1-dimensional hole in the reduced complex.

To better explain the problem of homological pollution, let us consider the following illustrative example.

Example

Consider the simplicial complex of order 2 depicted in Figure III.2a. In this example, we have $\mathcal{V}_0 = \{1, \dots, 7\}$, $\mathcal{V}_1 = \{[1,2],[1,3],[2,3],[2,4],[3,5],[4,5],\ [4,6],\ [5,6],[5,7],[6,7]\}$ and $\mathcal{V}_2 = \{[1,2,3],[4,5,6],[5,6,7]\}$, all with weight equal to one: $w_k \equiv 1$ for $k = 0, 1, 2$. The only existing 1-dimensional hole is shown in red, and thus the corresponding Hodge homology is $\beta = 1$. Now, consider perturbing the weight of edges $[2,4]$ and $[3,5]$ by setting their weights to



(a) Connected **(b)** Close to disconnected **(c)** Disconnected

Figure III.2: Example of the homological pollution, Example 7, for the simplicial complex $\mathcal{K}$ on 7 vertices; the existing hole is $[2,3,4,5]$ (left and center pane), all 3 cliques are included in the simplicial complex and shown in blue. The left pane demonstrates the initial setup with 1 hole; the center pane retains the hole exhibiting spectral pollution; the continuous transition to the eliminated edges with $\beta_1 = 0$ (no holes) is shown on the right pane.

$\varepsilon > 0$ Figure III.2b. For small ε, this perturbation implies that the smallest nonzero eigenvalue μ_2 in $\sigma_+(\overline{L}_0)$ is scaled by ε. As $\sigma_+(\overline{L}_0) \subseteq \sigma_+(\overline{L}_1)$, we have that $\dim\ker \overline{L}_1 = 1$ and $\sigma_+(\overline{L}_1)$ has an arbitrary small eigenvalue, approaching 0 with $\varepsilon \to 0$.
At the same time, when $\varepsilon \to 0$, the reduced complex obtained by removing the zero edges as in Figure III.2c does not have any 1-dimensional hole, i.e. $\beta_1 = 0$. Thus, in this case, the presence of a very small eigenvalue $\mu_2 \in \sigma_+(\overline{L}_1)$ does not imply that the simplicial complex is close to a simplicial complex with a larger dimension of the Hodge homology.

To mitigate the phenomenon of homological pollution, in our spectral-based functional for Problem 2, we include a term that penalizes the spectrum of $\overline{L}_0$ from approaching zero. To this end, we observe below that a careful choice of the vertex weights is required.

The smallest non-zero eigenvalue of the Laplacian $\mu_2 \in \sigma(\overline{L}_0)$ is directly related to the connectedness of the graph $\mathcal{G}_\mathcal{K}$. This relation is well-known and dates back to the pioneering work of Fiedler [Fie89]. In particular, as μ_2 is a function of node and edge weights, the following generalized version of the Cheeger inequality holds (see e.g. [TH18])

$$\frac{1}{2}\mu_2 \leq h(\mathcal{G}_\mathcal{K}) \leq \left(2\,\mu_2 \max_{i\in\mathcal{V}_0} \frac{\deg(i)}{w_0(i)}\right)^{1/2}, \qquad \text{(Eqn. 60)}$$

where $h(\mathcal{G}_\mathcal{K})$ is the Cheeger constant of the graph $\mathcal{G}_\mathcal{K}$, defined as

$$h(\mathcal{G}_\mathcal{K}) = \min_{S\subset\mathcal{V}_0} \frac{w_1(S, \mathcal{V}_0\backslash S)}{\min\{w_0(S), w_0(\mathcal{V}_0\backslash S)\}},$$

with $w_1(S, \mathcal{V}_0\backslash S) = \sum_{ij\in\mathcal{V}_1 : i\in S, j\notin S} w_1(ij)$, $\deg(i) = \sum_{j:ij\in\mathcal{V}_1} w_1(ij)$, and $w_0(S) = \sum_{i\in S} w_0(i)$.

We immediately see from (60) that when the graph $\mathcal{G}_\mathcal{K}$ is disconnected,

then $h(\mathcal{G}_\mathcal{K}) = 0$ and $\mu_2 = 0$ as well. Vice-versa, if μ_2 goes to zero, then $h(\mathcal{G}_\mathcal{K})$ decreases to zero too. While this happens independently of w_0 and w_1, if w_0 is a function of w_1, then it might fail to capture the presence of edges whose weight is decreasing and is about to disconnect the graph.

To see this, consider the example choice $w_0(i) = \deg(i)$, the degree of node i in $\mathcal{G}_\mathcal{K}$. Note that this is a very common choice in the graph literature, with several useful properties, including that no other graph-dependent constant appears in the Cheeger inequality (60) other than μ_2. For this weight choice, consider the case of a leaf node, a node $i \in \mathcal{V}_0$ that has only one edge $ij_0 \in \mathcal{V}_1$ connecting i to the rest of the (connected) graph $\mathcal{G}_\mathcal{K}$ via the node j_0. If we set $w_1(ij_0) = \varepsilon$ and we let ε decrease to zero, the graph $\mathcal{G}_\mathcal{K}$ is approaching disconnectedness and we would expect $h(\mathcal{G}_\mathcal{K})$ and μ_2 to decrease as well. However, one easily verifies that both μ_2 and $h(\mathcal{G}_\mathcal{K})$ are constant with respect to ε in this case, as long as $\varepsilon \neq 0$.

In order to avoid such a discontinuity, in our weight perturbation strategy for the simplex $\mathcal{K}$, if w_0 is a function of w_1, we perturb it by a constant shift. Precisely, if w_0 is the initial vertex weight of $\mathcal{K}$, we set $\tilde{w}_0(i) = w_0(i) + \varrho$, with $\varrho > 0$. So, for example, if $w_0 = \deg$ and the new edge weight function $\tilde{w}_1(\sigma) = w_1(\sigma) + \delta w_1(\sigma)$ is formed after the addition of δW_1, we set $\tilde{w}_0(i) = \sum_j [w_1(ij) + \delta w_1(ij)] + \varrho$.

III.III Dimensionality reduction: faux edges

Another source of spectral pollution for $\overline{L}_1$ stems from the discrepancy between the original and the "reduced" complexes and from the presence of edges that are not adjacent to any 2-simplex. This is primarily driven by the underlying reduction of the dimensionality.

When the weight of an edge σ is moved to zero, we are formally reducing the initial complex $\mathcal{K}$ to a smaller $\tilde{\mathcal{K}}$ with $\mathcal{V}_1(\tilde{\mathcal{K}}) = \mathcal{V}_1 \setminus \{\sigma\}$. The Hodge Laplacian of $\tilde{\mathcal{K}}$ has dimension $|\mathcal{V}_1| - 1$, and the dimension of its kernel $\tilde{\beta}_1$ is the dimension of the corresponding first homology. However, in our perturbative approach, we want to maintain the dimension of $\overline{L}_1$ unchanged to be able to explore the set of possible perturbations $\Omega(\varepsilon) \cap \Pi(W_1)$ in a continuous way. When the weight of σ decreases to zero, we create a "faux" edge which corresponds to a zero row in $\overline{B}_1$ and thus a zero row and zero column in $\overline{L}^{\downarrow}{}_1 = \overline{B}_1^\top \overline{B}_1$. If σ is not adjacent to any 2-simplex in $\mathcal{K}$, then the same row and column are also zero in $\overline{L}^{\uparrow}{}_1 = \overline{B}_2 \overline{B}_2^\top$ and in the whole Hodge Laplacian $\overline{L}_1$. Therefore, a faux edge creates a 0 row and column in $\overline{L}_1$, creating an additional 0 entry in the spectrum of $\overline{L}_1$, which does not correspond to a different homology for the reduced complex $\tilde{\mathcal{K}}$. In other words, when quantifying $\tilde{\beta}_1$ from the kernel of the perturbed $\overline{L}_1$, we need to rule out the number of 0-rows in $\overline{L}_1$.

Rem III.12 Note that in the general case, "faux" edges can create additional zero rows and columns also in $\overline{L}^{\uparrow}{}_1$ even if the edge σ has some adjacent triangles from $\mathcal{V}_2(\mathcal{K})$. Specifically, this could happen if the weight update $\tilde{w}_2$ is not properly defined, e.g. does not vanish when the edge vanishes or vanishes asymptotically faster (for instances, the min-rule scheme, i.e. Equation (56) has neither of these problems).

As a result, one needs to modify the target functional $F_\varepsilon(E)$ in such a way that it avoids homological pollution and faux edges in accordance with the spectral inheritance principle, Theorem III.III.5. We propose such functional and perform all the necessary calculations for the optimization routine next.

IV. Functional, Derivative, and Alternating Scheme for Topological Stability

IV.I Target Functional and Main Problem for $\overline{L}_k$

We are now in the position to formulate our proposed spectral-based functional, whose minimization leads to the desired small perturbation that changes the first homology of $\mathcal{K}$. In the notation of Problem 2, we are interested in the smallest perturbation ε and the corresponding modification $\delta W_1 \in \Omega(\varepsilon) \cap \Pi(W_1)$ that increases β_1, where $\Omega(\varepsilon)$ is the set of diagonal matrices W such that $\|W\| = \varepsilon$ and $\Pi(W_1)$ contains diagonal matrices W avoiding negative weights, $W_1 + W \geq 0$.

As $\|\delta W_1\| = \varepsilon$, for convenience we indicate $\delta W_1 = \varepsilon E$ with $\|E\| = 1$ so $E \in \Omega(1) \cap \Pi_\varepsilon(W_1)$, where $\Pi_\varepsilon(W_1) = \{W \mid \varepsilon E \in \Pi(W_1)\}$. For the sake of simplicity, we will omit the dependencies and write Ω and Π_ε when there is no danger of ambiguity. Finally, let us denote by $\beta_1(\varepsilon, E)$ the first Betti number corresponding to the simplicial complex perturbed via the edge modification εE. With this notation, we can reformulate Problem 2 as follows:

Problem Find the smallest $\varepsilon > 0$, such that there exists an admissible perturbation $E \in \Omega \cap \Pi_\varepsilon$ with an increased number of holes, i.e.

$$\min \left\{ \varepsilon > 0 \,:\, \beta_1(\varepsilon, E) \geq \beta_1 + 1 \text{ for some } E \in \Omega \cap \Pi_\varepsilon \right\} \quad \text{(Eqn. 61)}$$

where $\beta_1 = \beta_1(0, \cdot)$ is the first Betti number of the original simplicial complex.

Finally, in order to approach Problem 3 and complete the framework of spectral matrix nearness problems established in Section II, one needs the objective functional $F(\varepsilon, E)$, its free gradient $G(\varepsilon, E) = \nabla_E F(\varepsilon, E)$ and the projector $\mathbb{P}$ on the tangent plane to the admissible set Π_ε to integrate the constrained flow:

$$\dot{E}(t) = -\mathbb{P}G(\varepsilon, E(t)) + \operatorname{Re}\langle \mathbb{P}G(\varepsilon, E(t)), E(t) \rangle E(t) \quad \text{(Eqn. 62)}$$

We introduce a target functional $F_\varepsilon(E)$, based on the spectrum of the 1-Laplacian $\overline{L}_1(\varepsilon, E)$ and the 0-Laplacian $\overline{L}_0(\varepsilon, E)$, where the dependence on ε and E is to emphasize the corresponding weight perturbation is of the form $W_1 \mapsto W_1 + \varepsilon E$.

We aim to move a positive entry from $\sigma_+(\overline{L}_1(\varepsilon, E))$ to the kernel. At the same time, assuming the initial graph $\mathcal{G}_\mathcal{K}$ is connected, one should avoid the inherited almost disconnectedness with small positive entries of $\sigma_+(\overline{L}_0(\varepsilon, E))$. As, by Theorem III.III.5 for $k = 0$, $\sigma_+(\overline{L}_0(\varepsilon, E)) = \sigma_+(\overline{L}^{\downarrow}{}_1(\varepsilon, E))$, the only eigenvalue of $\overline{L}_1(\varepsilon, E)$ that can be continuously driven to 0 comes from $\overline{L}^{\uparrow}{}_1(\varepsilon, E)$. For this reason, let us denote the **first non-zero eigenvalue** of the up-Laplacian $\overline{L}^{\uparrow}{}_1(\varepsilon, E)$ by $\lambda_+(\varepsilon, E)$. The proposed target functional is defined as:

$$F_\varepsilon(E) = \frac{\lambda_+(\varepsilon, E)^2}{2} + \frac{\alpha}{2} \max\left(0, 1 - \frac{\mu_2(\varepsilon, E)}{\mu}\right)^2 \qquad \text{(Eqn. 63)}$$

where α and μ are positive parameters, and $\mu_2(\varepsilon, E)$ is the first nonzero eigenvalue of $\overline{L}_0(\varepsilon, E)$. As recalled in Subsection III.II, $\mu_2(\varepsilon, E)$ is an algebraic measure of the connectedness of the perturbed graph, thus the whole second term in (63) "activates" when such algebraic connectedness falls below the threshold μ.

By design, $F_\varepsilon(E)$ is non-negative and is equal to 0 iff $\lambda_+(\varepsilon, E)$ reaches 0, increasing the dimension of $\mathcal{H}_1$. Using this functional, we recast the Problem 3 as

$$\min\{\varepsilon > 0 : F_\varepsilon(E) = 0 \text{ for some } E \in \Omega_\varepsilon\} \qquad \text{(Eqn. 64)}$$

Rem IV.13 When $\mathcal{G}_\mathcal{K}$ is connected, $\dim\ker \overline{L}_0 = 1$ and, by the Theorem III.III.5, $\dim\ker \overline{L}^{\uparrow}{}_1 = \dim\ker \overline{L}_1 + (n - \dim\ker \overline{L}_0) = n + \beta_1 - 1$, so the first nonzero eigenvalue of $\overline{L}^{\uparrow}{}_1$ is the $(n + \beta_1)$-th. While $(n + \beta_1)$ can be a large number in practice, we will discuss in Section V.I an efficient method that allows us to compute $\lambda_+(\varepsilon, E)$ without computing any of the previous $(n + \beta_1 - 1)$ eigenvalues.

Rem IV.14 Note that since we moved to consider $\lambda_+ \in \overline{L}^{\uparrow}{}_1$, the absence of faux edges in the flow is guaranteed to the weighing scheme, Equation (56), and edges not adjacent to any triangle do not affect the spectral profile anymore.

IV.II Free gradient calculation

Let us denote the perturbed weight matrix by $\tilde{W}_1(t) = W_1 + \varepsilon E(t)$, and the corresponding $\tilde{W}_0(t) = W_0(\tilde{W}_1(t))$ and $\tilde{W}_2(t) = W_2(\tilde{W}_1(t))$, defined accordingly as discussed in Subsection IV.I and Section III. From now on we omit the time dependence for the perturbed matrices to simplify the notation. Since $\tilde{W}_0$, $\tilde{W}_1$ and $\tilde{W}_2$ are necessarily diagonal, by the chain rule we have $\dot{\tilde{W}}_i(t) = \varepsilon \operatorname{diag}\left(J_1^i \dot{E}\mathbf{1}\right)$, where $\mathbf{1}$ is the vector of all ones, $\operatorname{diag}(\mathbf{v})$ is the diagonal matrix with diagonal entries the vector $\mathbf{v}$, and J_1^i is the Jacobian matrix of the i-th weight matrix with respect to $\tilde{W}_1$, which for any $u_1 \in \mathcal{V}_1$ and $u_2 \in \mathcal{V}_i$, has entries $[J_1^i]_{u_1,u_2} = \frac{\partial}{\partial \tilde{w}_1(u_1)} \tilde{w}_i(u_2)$.

Next, in the following two lemmas, we express the time derivative of the Laplacian $\overline{L}_0$ and $\overline{L}_1^{up}$ as functions of $E(t)$. The proofs of these results are straightforward and omitted for brevity. In what follows, $\operatorname{Sym}[A]$ denotes the symmetric part of the matrix A, namely $\operatorname{Sym}[A] = (A + A^\top)/2$.

Lem IV.6 **(Derivative of $\overline{L}_0$)** For the simplicial complex $\mathcal{K}$ with the initial edges' weight matrix W_1 and fixed perturbation norm ε, let $E(t)$ be a smooth path and $\tilde{W}_0, \tilde{W}_1, \tilde{W}_2$ be corresponding perturbed weight matrices. Then,

$$\frac{1}{2\varepsilon}\frac{d}{dt}\overline{L}_0(t) = \tilde{W}_0^{-1} B_1 \tilde{W}_1 \dot{E} B_1^\top \tilde{W}_0^{-1} - \operatorname{Sym}\left[\tilde{W}_0^{-1} \operatorname{diag}\left(J_1^0 \dot{E}\mathbf{1}\right) \overline{L}_0\right] \tag{Eqn. 65}$$

Proof By definition, $\overline{L}_0(t) = \tilde{W}_0^{-1} B_1 \tilde{W}_1^2 B_1^\top \tilde{W}_0^{-1}$ where $\tilde{W}_1 = W_1 + \varepsilon E(t)$. Then

$$\begin{aligned}\frac{d}{dt}L_0(t) = &\left(\frac{d}{dt}\tilde{W}_0^{-1}\right) B_1 \tilde{W}_1^2 B_1^\top \tilde{W}_0^{-1} + \tilde{W}_0^{-1} B_1 \frac{d}{dt}\left(\tilde{W}_1^2\right) B_1^\top \tilde{W}_0^{-1} + \\ &+ \tilde{W}_0^{-1} B_1 \tilde{W}_1^2 B_1^\top \left(\frac{d}{dt}\tilde{W}_0^{-1}\right)\end{aligned} \tag{Eqn. 66}$$

Then it is sufficient to note for a diagonal matrix $\tilde{W}_0$ the derivative is given by $\frac{d}{dt}\tilde{W}_0^{-1} = -\tilde{W}_0^{-2}\frac{d}{dt}\tilde{W}_0 = -\tilde{W}_0^{-1}\frac{d}{dt}\tilde{W}_0\tilde{W}_0^{-1}$; finally, $\frac{d}{dt}\tilde{W}_1^2 = 2\varepsilon\tilde{W}_1\dot{E}$ and $\frac{d}{dt}\tilde{W}_0 = \operatorname{diag}\left(J_1^0\dot{E}\mathbf{1}\right)$ by the definition of matrix J_1^0. ■

Lem IV.7 **(Derivative of $\overline{L^\uparrow}_1$)** For the simplicial complex $\mathcal{K}$ with the initial edges' weight matrix W_1 and fixed perturbation norm ε, let $E(t)$ be a smooth path and $\tilde{W}_0, \tilde{W}_1, \tilde{W}_2$ be corresponding perturbed weight matrices. Then,

$$\frac{1}{2\varepsilon}\frac{d}{dt}\overline{L^\uparrow}_1(t) = -\operatorname{Sym}\left[\tilde{W}_1^{-1} B_2 \tilde{W}_2^2 B_2^\top \tilde{W}_1^{-1} \dot{E} \tilde{W}_1^{-1}\right] + \tilde{W}_1^{-1} B_2 \tilde{W}_2 \operatorname{diag}\left(J_1^2 \dot{E}\mathbf{1}\right) B_2^\top \tilde{W}_1^{-1}$$

Proof Similarly to Lemma IV.6, $\overline{L}^{\uparrow}{}_1 = \tilde{W}_1^{-1} B_2 \tilde{W}_2^2 B_2^\top \tilde{W}_1^{-1}$ and

$$\frac{d}{dt}\overline{L}^{\uparrow}{}_1 = \left(\frac{d}{dt}\tilde{W}_1^{-1}\right) B_2 \tilde{W}_2^2 B_2^\top \tilde{W}_1^{-1} + \tilde{W}_1^{-1} B_2 \left(\frac{d}{dt}\tilde{W}_2^2\right) B_2^\top \tilde{W}_1^{-1} + \\ + \tilde{W}_1^{-1} B_2 \tilde{W}_2^2 B_2^\top \left(\frac{d}{dt}\tilde{W}_1^{-1}\right)$$

with the same formula for the derivative of inverse $\frac{d}{dt}\tilde{W}_1^{-1} = -\varepsilon \tilde{W}_1^{-1}\dot{E}\tilde{W}_1^{-1}$ (Eqn. 67) and $\frac{d}{dt}\tilde{W}_2^2 = 2\tilde{W}_2 \operatorname{diag}\left(J_1^2 \dot{E}\mathbf{1}\right)$. ■

Combining Theorem III.II.3 with Lemma IV.6 and Lemma IV.7, we obtain the following expression for the free gradient of the functional.

Th III.IV.6 **(The free gradient of $F_\varepsilon(E)$)** Assume the initial weight matrices W_0, W_1 and W_2, as well as the parameters $\varepsilon > 0$, $\alpha > 0$ and $\mu > 0$, are given. Additionally, assume that $E(t)$ is a differentiable matrix-valued function such that the first non-zero eigenvalue $\lambda_+(\varepsilon, E)$ of $\overline{L}^{\uparrow}{}_1(\varepsilon, E)$ and the second smallest eigenvalue $\mu_2(\varepsilon, E)$ of $\overline{L}_0(\varepsilon, E)$ are simple. Let $\tilde{W}_0, \tilde{W}_1, \tilde{W}_2$ be corresponding perturbed weight matrices; then:

$$\frac{1}{\varepsilon}\nabla_E F_\varepsilon(E)(t) = \lambda_+(\varepsilon, E)\cdot \\ \cdot \left[\operatorname{Sym}\left[-\tilde{W}_1^{-1} B_2 \tilde{W}_2^2 B_2^\top \tilde{W}_1^{-1}\mathbf{x}_+\mathbf{x}_+^\top \tilde{W}_1^{-1}\right] + \right. \\ \left. + \operatorname{diag}\left(J_1^{2\top} \operatorname{diagvec}\left(B_2^\top \tilde{W}_1^{-1}\mathbf{x}_+\mathbf{x}_+^\top \tilde{W}_1^{-1} B_2 \tilde{W}_2\right)\right)\right] - \\ - \frac{\alpha}{\mu}\max\left\{0, 1 - \frac{\mu_2(\varepsilon, E)}{\mu}\right\} \cdot \left[B_1^\top \tilde{W}_0^{-1}\mathbf{y}_2\mathbf{y}_2^\top \tilde{W}_0^{-1} B_1 \tilde{W}_1 - \right. \\ \left. - \operatorname{diag}\left(J_1^{0\top} \operatorname{diagvec}\left(\operatorname{Sym}[\tilde{W}_0^{-1}\mathbf{y}_2\mathbf{y}_2^\top \overline{L}_0]\right)\right)\right]$$

where $\mathbf{x}_+$ is a unit eigenvector of $\overline{L}_1^{up}$ corresponding to λ_+, $\mathbf{y}_2$ is a unit eigenvector of $\overline{L}_0$ corresponding to μ_2, and the operator $\operatorname{diagvec}(X)$ returns the main diagonal of X as a vector.

Proof To derive the expression for the gradient $\nabla_E F$, we exploit the chain rule for the time derivative: $\dot{\lambda} = \langle \frac{d}{dt} A(E(t)), \mathbf{x}\mathbf{x}^\top \rangle = \langle \nabla_E \lambda, \dot{E} \rangle$. Then, it is sufficient to apply the cyclic perturbation for the scalar products of Lemma IV.6 and Lemma IV.7 with $\mathbf{x}_+\mathbf{x}_+^\top$ and $\mathbf{y}_2\mathbf{y}_2^\top$ respectively. The final transition requires the formula:

$$\langle A, \operatorname{diag}(BE\mathbf{1})\rangle = \left\langle \operatorname{diag}\left(B^\top(\operatorname{diagvec} A)\right), E\right\rangle.$$

Indeed,

$$\langle A, \operatorname{diag}(BE\mathbf{1})\rangle = \operatorname{Tr}\left(A^\top \operatorname{diag}(BE\mathbf{1})\right) = \sum_i \left(A^\top \operatorname{diag}(BE\mathbf{1})\right)_{ii}$$

and

$$\begin{aligned}\langle A, \operatorname{diag}(BE\mathbf{1})\rangle &= \sum_i \sum_j A^\top_{ij} \operatorname{diag}(BE\mathbf{1})_{ji} = \sum_i A^\top_{ii}(BE\mathbf{1})_i = \\ &= \sum_i A_{ii} \sum_j (BE)_{ij} = \sum_{i,j} A_{ii} B_{ij} E_{jj} = \\ &= \sum_{i,j} B_{ij} (\operatorname{diagvec} A)_i E_{jj} = \left\langle \operatorname{diag}\left(B^\top (\operatorname{diagvec} A)\right), E\right\rangle.\end{aligned}$$

■

Rem IV.15 The derivation above assumes the simplicity of both $\mu_2(\varepsilon, E)$ and $\lambda_+(\varepsilon, E)$. This assumption is not restrictive as simplicity for these extremal eigenvalues is a generic property. Indeed, we observe simplicity in all our numerical tests.

IV.III The constrained gradient system and its stationary points

In this section, we are deriving from the free gradient determined in Theorem III.IV.6 the constrained gradient of the considered functional, that is, the projected gradient (with respect to the Frobenius inner product) onto the manifold $\Omega \cap \Pi_\varepsilon$, which consists of perturbations E of the unit norm, which preserve the structure of W.

To obtain the constrained gradient system, we need to project the unconstrained gradient given by Theorem III.IV.6 onto the feasible set and also to normalize E to preserve its unit norm. Using the Karush-Kuhn-Tucker conditions on a time interval where the set of 0-weight edges remain unchanged, the projection is done via the mapping $\mathbb{P}_+ G(\varepsilon, E)$, where

$$[\mathbb{P}_+ X]_{ij} = \begin{cases} X_{ij}, & [W_1 + \varepsilon E]_{ij} > 0 \\ 0, & \text{otherwise} \end{cases}.$$

Note that $\mathbb{P}_+$ is the projector on the tangent space of the admissible set Π_ε as in Equation (39) and Equation (53); we use $\mathbb{P}_+$ notation to emphasize the non-negativity constraints. As a result, we are ready to reformulate a steepest descent direction result (Lemma II.4) for the particular case of Problem 3:

Lem IV.8 **(Direction of steepest admissible descent)** Let $E, G \in \mathbb{R}^{m_1 \times m_1}$ with G given by Theorem III.IV.6, and $\|E\| = 1$. On a time interval where the

set of 0-weight edges remains unchanged, the gradient system reads

$$\dot{E}(t) = -\mathbb{P}_+ G(\varepsilon, E(t)) + \kappa \mathbb{P}_+ E(t), \quad \text{where} \quad \kappa = \frac{\langle \varepsilon, G(E(t)),\, \mathbb{P}_+ E(t)\rangle}{\|\mathbb{P}_+ E(t)\|^2}. \tag{Eqn. 68}$$

Equation (68) suggests that the system goes "primarily" in the direction of the antigradient $-G(\varepsilon, E)$; thus the functional is expected to decrease along it.

Lem IV.9 **(Monotonicity)** Let $E(t)$ of unit Frobenius norm satisfy the differential equation (68), with G given by Theorem III.IV.6. Then, $F_\varepsilon(E(t))$ decreases monotonically with t.

Proof We consider first the simpler case where the non-negativity projection does not apply so that $G = G(\varepsilon, E)$ (without $\mathbb{P}_+$). Then

$$\begin{aligned}\frac{d}{dt}F_\varepsilon(E)(t) &= \left\langle \nabla_E F_\varepsilon(E),\, \dot{E}\right\rangle = \langle \varepsilon G(\varepsilon, E(t)),\, -G(\varepsilon, E(t)) + \kappa E(t)\rangle \\ &= -\varepsilon \|G(\varepsilon, E)\|^2 + \varepsilon \frac{\langle G(\varepsilon, E),\, E\rangle}{\langle E,\, E\rangle} \langle G(\varepsilon, E),\, E\rangle \\ &= \varepsilon \left(-\|G(\varepsilon, E)\|^2 + \frac{|\langle G(\varepsilon, E),\, E\rangle|^2}{\|E\|^2} \right) \leq 0\end{aligned} \tag{Eqn. 69}$$

where the final estimate is given by the Cauchy-Bunyakovsky-Schwarz inequality. The derived inequality holds on the time interval without the change in the support of $\mathbb{P}_+$ (so that no new edges are prohibited by the non-negativity projection). ■

IV.IV Free Gradient Transition in the Outer Level

The optimization problem in the **inner level** is non-convex due to the non-convexity of the functional $F_\varepsilon(E)$; we denote $F_\varepsilon(E) = F(\varepsilon, E)$ for the clarity in this section. Hence, for a given ε, the computed minimizer $E(\varepsilon)$ may depend on the initial guess $E_0 = E_0(\varepsilon)$.

The effects of the initial choice are particularly important in the transition $\widehat{\varepsilon} \to \varepsilon = \widehat{\varepsilon} + \Delta\varepsilon$ between constrained inner levels: given reasonably small $\Delta\varepsilon$, one should expect relatively close optimizers $E(\widehat{\varepsilon})$ and $E(\varepsilon)$, and, hence, the initial guess $E_0(\varepsilon)$ being close to and dependent on $E(\varepsilon)$.

This choice, which seems very natural, determines, however, a discontinuity

$$F(\widehat{\varepsilon}, E(\widehat{\varepsilon})) \neq F(\varepsilon, E(\widehat{\varepsilon})),$$

which may prevent the expected monotonicity property with respect to ε in the (likely unusual case) where $F(\widehat{\varepsilon}, E(\widehat{\varepsilon})) < F(\varepsilon, E(\widehat{\varepsilon}))$. This may happen in particular when $\Delta\varepsilon$ is not taken small; since a Newton-like iteration drives

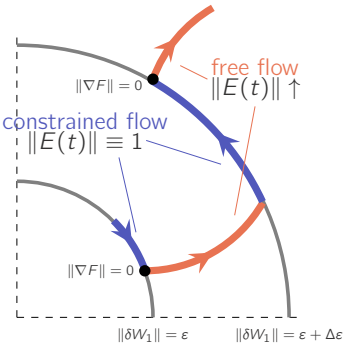


Figure IV.1: The scheme of alternating constrained (blue, $\|E(t)\| \equiv 1$) and free gradient (red) flows. Each stage inherits the final iteration of the previous stage as initial $E_0(\varepsilon_i)$ or $\tilde{E}_0(\varepsilon_i)$ respectively; constrained gradient is integrated till the stationary point ($\|\nabla F(E)\| = 0$), free gradient is integrated until $\|\delta W_1\| = \varepsilon_i + \Delta\varepsilon$. The scheme alternates until the target functional vanishes ($F(\varepsilon, E) = 0$).

the choice of $\Delta\varepsilon$, we are interested in finding a way to prevent this situation and making the whole iterative method more robust. The goal of that is to guarantee monotonicity of the functional both with respect to time and with respect to ε.

When in the outer iteration we increase ε from a previous value $\widehat{\varepsilon} < \varepsilon$, we have the problem of choosing a suitable initial value for the constrained gradient system (68), such that at the stationary point $E(\widehat{\varepsilon})$ we have $F(\widehat{\varepsilon}, E(\widehat{\varepsilon})) < F(\varepsilon, E(\varepsilon))$ (which we assume both positive, that is on the left of the value $\varepsilon^\star$ which identifies the closest zero of the functional).

To maintain monotonicity with respect to time and also with respect to ε, it is convenient to start to look at the optimization problem with value ε, with the initial datum $\delta W_1 = \widehat{\varepsilon} E(\widehat{\varepsilon})$ of norm $\widehat{\varepsilon} < \varepsilon$.

This means we have essentially to optimize with respect to the inequality constraint $\|\delta W_1\| \leq \varepsilon$, or equivalently solve the problem (now with inequality constrain on $\|E\|_F$):

$$E(\varepsilon) = \operatorname*{arg\,min}_{E \in \Omega, \|E\| \leq 1} F(\varepsilon, E)$$

The situation changes only slightly from the one discussed above. If $\|E\| < 1$, every direction is admissible, and the direction of the steepest descent is given by the negative gradient. So, we choose the free gradient flow (in the direction of the free gradient on the admissible set without the norm conservation)

$$\dot{E} = -\mathbb{P}_+ G(\varepsilon, E(t)) \qquad \text{as long as} \quad \|E(t)\| < 1. \qquad \text{(Eqn. 70)}$$

When $\|E(t)\| = 1$, then there are two possible cases. If $\langle \mathbb{P}_+ G(\varepsilon, E), E \rangle \geq$

0, then the solution of (70) has

$$\frac{d}{dt}\|E(t)\|^2 = 2\langle \dot{E}, E\rangle = -2\langle \mathbb{P}_+ G(\varepsilon, E(t)), E\rangle \leq 0,$$

and hence the solution of (70) remains of Frobenius norm at most 1.

Otherwise, if $\langle \mathbb{P}_+ G(\varepsilon, E), E\rangle < 0$, the admissible direction of steepest descent is given by the right-hand side of (68), and so we choose this ODE to evolve E. The situation can be summarized as taking, if $\|E(t)\| = 1$,

$$\dot{E} = -\mathbb{P}_+ G(\varepsilon, E) + \mu E \quad \text{with} \quad \mu = \min\big(0, \kappa\big) \qquad \text{(Eqn. 71)}$$

with $\kappa = \langle G(\varepsilon, E), \mathbb{P}_+ E\rangle / \|\mathbb{P}_+ E\|^2$. Along the solutions of (71), the functional F decays monotonically, and stationary points of (71) (i.e. points such that $\dot{E} = 0$) with $\mathbb{P}_+ G(\varepsilon, E(t)) \neq 0$ are characterized by

$$E \text{ is a negative real multiple of } \mathbb{P}_+ G(\varepsilon, E(t)). \qquad \text{(Eqn. 72)}$$

If it can be excluded that the gradient $\mathbb{P}_+ G(\varepsilon, E(t))$ vanishes at an optimizer, it can thus be concluded that the optimizer of the problem with inequality constraints is a stationary point of the gradient flow (68) for the problem with equality constraints.

Rem IV.16 As a result, $F(\varepsilon, E(t)) = F_\varepsilon(E(t))$ monotonically decreases both with respect to time t and to the value of the norm ε, when $\varepsilon \leq \varepsilon^\star$.

V. Algorithm details

In Algorithm 1, we provide the pseudo-code of the whole bi-level procedure.

The initial "α-phase" is used to choose an appropriate value for the regularization parameter α. In order to avoid the case in which the penalizing term on the right-hand side of (63) dominates the loss $F_\varepsilon(E(t))$ in the early stages of the descent flow, we select α by first running such an initial phase, prior to the main alternated constrained/free gradient loop. In this phase, we fix a small $\varepsilon = \varepsilon_0$ and run the constrained gradient integration for an initial $\alpha = \alpha_*$. After the computation of a local optimum E_*, we then increase α and rerun for the same ε_0 with E_* as the starting point. We iterate until no change in E_* is observed or until α reaches an upper bound α^*.

The resulting value of α and E_* are then used in the main loop where we first increase ε by the chosen step size, we rescale E_i by $0 < \varepsilon/(\varepsilon+\Delta\varepsilon) < 1$, and then we perform the free gradient integration described in Section IV.IV until we reach a new point E_i on the unit sphere $\|E_i\| = 1$. Then, we perform the inner constrained gradient step by integrating Equation (68),

Algorithm 1 Pseudo-code of the complete constrained- and free-gradient flow.

Require: initial edge perturbation guess E_0; initial $\varepsilon_0 > 0$; ε-stepsize $\Delta\varepsilon > 0$; bounds α_*, α^* for the α-phase;

$\alpha, E \leftarrow \texttt{AlphaPhase}(E_0, \varepsilon_0, \alpha_*, \alpha^*)$ ▷ for details see Subsection V.I
while $|F_\varepsilon(E)| < 10^{-6}$ **do**
  $\varepsilon \leftarrow \varepsilon + \Delta\varepsilon$
  $E \leftarrow \frac{\varepsilon}{\varepsilon+\Delta\varepsilon} E$ ▷ before the free gradient $\|E\| < 1$
  $E_i \leftarrow \texttt{FreeGradientFlow}(E, \Delta\varepsilon, \varepsilon)$ ▷ see Section IV.IV
  $E \leftarrow \texttt{ConstrainedGradientFlow}(E, \varepsilon)$ ▷ see Section V
end while

iterating the following two-step norm-corrected Euler scheme:

$$\begin{cases} E_{i+1/2} = E_i - h_i \left(\mathbb{P}_+ G(E_i, \varepsilon) - \kappa_i \mathbb{P}_+ E_i\right) . \\ E_{i+1} = \mathbb{P}_{\Pi_\varepsilon} E_{i+1/2} / \|\mathbb{P}_{\Pi_\varepsilon} E_{i+1/2}\| \end{cases} \tag{Eqn. 73}$$

where the second step is necessary to numerically guarantee the Euler integration remains in the set of admissible flows since the discretization does not conserve the norm and larger steps h_i may violate the non-negativity of the weights.

In both the free and constrained integration phases, since we aim to obtain the solution at $t \to \infty$ instead of the exact trajectory, we favor larger steps h_i given that the established monotonicity is conserved. Specifically, if $F_\varepsilon(E_{i+1}) < F_\varepsilon(E_i)$, then the step is **accepted** and we set $h_{i+1} = \beta h_i$ with $\beta > 1$; otherwise, the step is **rejected** and repeated with a smaller step $h_i \leftarrow h_i/\beta$; we provide detailed look in Algorithm 2.

Rem V.17 The step acceleration strategy described above, where βh_i is immediately increased after one accepted step, may lead to "oscillations" between accepted and rejected steps in the event the method would prefer to maintain the current step size h_i. To avoid this potential issue, we increase the step length in our experiments after two consecutive accepted steps. Alternative step-length selection strategies are also possible, for example, based on Armijo's rule or non-monotone stabilization techniques [GLL91].

V.I Computational costs

Each step of either the free or the constrained flows requires one step of explicit Euler integration along the anti-gradient $-\nabla_E F_\varepsilon(E(t))$. As discussed above, the construction of such a gradient requires several sparse and diagonal matrix-vector multiplications as well as the computation of the smallest nonzero eigenvalue of both $\overline{L}^{\uparrow}{}_1(\varepsilon, E)$ and $\overline{L}_0(\varepsilon, E)$. The latter two represent the major computational requirements of the numerical procedure. Fortunately, as both matrices are of the form $A^\top A$, with A being either of

Algorithm 2 Single Run of The Constrained Gradient Flow

Require: initial perturbation E_0, fixed perturbation ε

$t \leftarrow 0$, $h \leftarrow h_0$, $i \leftarrow 1$, $\texttt{oldVal} \leftarrow F(\varepsilon, E)$ ▷ initialization
while oldVal $> 10^{-5}$ and $i \leq 1000$ **do**
  $G(\varepsilon, E_{i-1}) \leftarrow \frac{1}{\varepsilon}\nabla_E F_\varepsilon(E_{i-1})$ ▷ compute gradient, Theorem III.IV.6
  if $\|G(\varepsilon, E_{i-1})\| < 10^{-4}$ **then**
    break
  end if ▷ stop at local minimum
  while true do
    $E_i \leftarrow \texttt{normCorrectedEuler}(E_{i-1}, h, G(\varepsilon, E_{i-1}))$ ▷ see (**??**
    $\texttt{newVal} \leftarrow F(\varepsilon, E_i)$ ▷ updated functional
    if $\texttt{newVal} < \texttt{oldVal}$ **then** ▷ check monotonicity
      $h \leftarrow \beta \cdot h$ ▷ step accepted and increased
      break
    else
      $h \leftarrow h/\beta$ ▷ switch to the more precise step and recalculate E_i
    end if
    $t \leftarrow t + h$
    $i \leftarrow i + 1$
  end while
end while

the two boundary or co-boundary operators $\overline{B}_2$ and $\overline{B}_1^\top$, we have that both the two eigenvalue problems boil down to a problem of the form

$$\min_{\mathbf{x} \perp \ker A} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}$$

i.e., the computation of the smallest singular value of the sparse matrix A. This problem can be approached by a sparse singular value solver based on a Krylov subspace scheme for the pseudo inverse of $A^\top A$. In practice, we implement the pseudo inversion by solving the corresponding least squares problems

$$\min_{\mathbf{x}} \|\overline{L}_1^{up}(\varepsilon, E)\mathbf{x} - \mathbf{b}\|, \qquad \min_{\mathbf{x}} \|\overline{L}_0(\varepsilon, E)\mathbf{x} - \mathbf{b}\|,$$

which, in our experiments, we solved using the least square minimal-residual method (LSMR) from [FS11]. This approach allows us to use a preconditioner for the normal equation corresponding to the least square problem. For simplicity, in our tests, we used a constant preconditioner computed by means of an incomplete Cholesky factorization of the initial unperturbed $\overline{L}^\uparrow{}_1$, or $\overline{L}_0$. Possibly, much better performance can be achieved with a tailored preconditioner that is efficiently updated throughout the matrix flow. We explore the idea of efficient preconditioning for $\overline{L}^\uparrow{}_1$ in the second part of the current work, Subsection VI.III; additionally, we also note that the eigenvalue

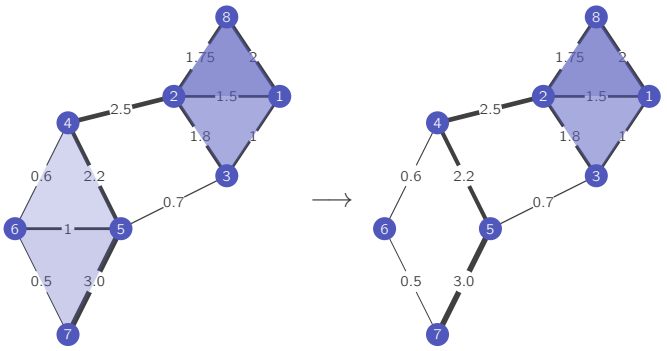


Figure VI.1: Simplicial complex $\mathcal{K}$ on 8 vertices for the illustrative run (on the left): all 2-simplices from $\mathcal{V}_2$ are shown in blue, the weight of each edge $w_1(e_i)$ is given on the figure. On the right: perturbed simplicial complex $\mathcal{K}$ through the elimination of the edge $[5, 6]$ creating additional hole $[5, 6, 7, 8]$.

problem for the graph Laplacian $\overline{L}_0(\varepsilon, E)$ may be alternatively approached by a combinatorial multigrid strategy [ST14] or stochastic Cholesky preconditioner, [KS16, Tro19].

VI. Benchmarking

VI.I Illustrative Example

We consider here a small example of a simplicial complex $\mathcal{K}$ of order 2 consisting of eight 0-simplices (vertices), twelve 1-simplices (edges), four 2-simplices $\mathcal{V}_2 = \{[1, 2, 3], [1, 2, 8], [4, 5, 6], [5, 6, 7]\}$ and one corresponding hole $[2, 3, 4, 5]$, hence, $\beta_1 = 1$. By design, the dimensionality of the homology group $\overline{\mathcal{H}}_1$ can be increased only by eliminating edges $[1, 2]$ or $[5, 6]$; for the chosen weight profile $w_1([1, 2]) > w_1([5, 6])$, hence, the method should converge to the minimal perturbation norm $\varepsilon = w_1([5, 6])$ by eliminating the edge $[5, 6]$, Figure VI.1.

The exemplary run of the optimization framework in time is shown in Figure VI.2. The top panel of Figure VI.2 provides the continued flow of the target functional $F_\varepsilon(E(t))$ consisting of the initial α-phase (in green) and alternated constrained (in blue) and free gradient (in orange) stages. As stated above, $F_\varepsilon(E(t))$ is strictly monotonic along the flow since the support of $\mathbb{P}_+$ does not change. Since the initial setup is not pathological with respect to the connectivity, the initial α-phase essentially reduces to a single constrained gradient flow and terminates after one run with $\alpha = \alpha_*$. The constrained gradient stages are characterized by a slow-changing $E(t)$, which is essentially due to the flow performing minor adjustments to find the correct rotation on the unit sphere. In contrast, the free gradient stage quickly decreases the target functional.

The second panel shows the behavior of first non-zero eigenvalue $\lambda_+(\varepsilon, E(t))$ (solid line) of $\overline{L}^{\uparrow}{}_1(\varepsilon, E(t))$ dropping through the ranks of $\sigma(\overline{L}_1(\varepsilon, E(t)))$ (semi-transparent); similar to the case of the target functional $F_\varepsilon(E(t))$,

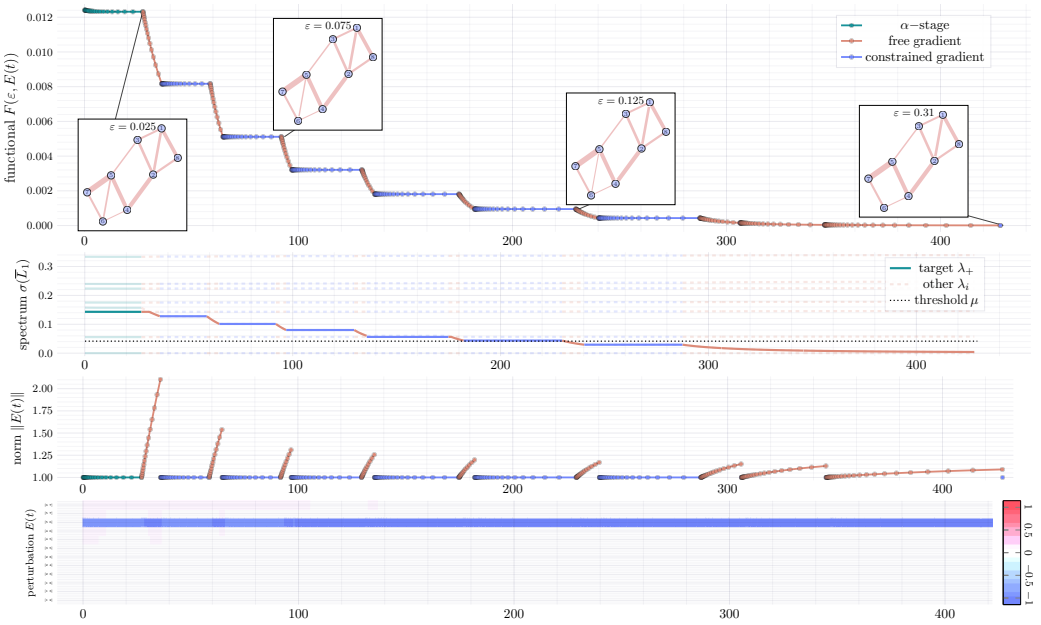


Figure VI.2: Illustrative run of the framework determining the topological stability: the top pane — the flow of the functional $F_\varepsilon(E(t))$; the second pane — the flow of $\sigma(\overline{L}_1)$, λ_+ is highlighted; third pane — the change of the perturbation norm $\|E(t)\|$; the bottom pane — the heatmap of the perturbation profile $E(t)$.

$\lambda_+(\varepsilon, E(t))$ monotonically decreases. The rest of the eigenvalues exhibit only minor changes, and the rapidly changing λ_+ successfully passes through the connectivity threshold μ (dotted line).

The third and the fourth panels show the evolution of the norm of the perturbation $\|E(t)\|$ and the perturbation $E(t)$ itself, respectively. The norm $\|E(t)\|$ is conserved during the constrained-gradient and the α- stages; these stages correspond to the optimization of the perturbation shape, as shown by the small positive values at the beginning of the bottom panel which eventually vanish. During the free gradient integration, the norm $\|E(t)\|$ increases, but the relative change of the norm declines with the growth of ε_i to avoid jumping over the smallest possible ε. Finally, due to the simplicity of the complex, the edge we want to eliminate, 56, dominates the flow from the very beginning (see bottom panel); such a clear pattern persists only in small examples, whereas for large networks, the perturbation profile is initially spread out among all the edges.

VI.II Triangulation Benchmark

To provide more insight into the computational behavior of the method, we synthesize here an almost planar graph dataset. Namely, we assume N uniformly sampled vertices on the unit square with a network built by the Delaunay triangulation; then, edges are randomly added or erased to obtain the sparsity ν (so that the graph has $\frac{1}{2}\nu N(N-1)$ edges overall). An order-2 simplicial complex $\mathcal{K} = (\mathcal{V}_0, \mathcal{V}_1, \mathcal{V}_2)$ is then formed by letting

$\mathcal{V}_0$ be the generated vertices, $\mathcal{V}_1$ the edges, and $\mathcal{V}_2$ every 3-clique of the graph; edges' weights are sampled uniformly between $1/4$ and $3/4$, namely $w_1(e_i) \sim U[\frac{1}{4}, \frac{3}{4}]$.

An example of such triangulation is shown in Figure VI.3; here, $N = 8$ and edges $[6, 8]$ and $[2, 7]$ were eliminated to achieve the desired sparsity.

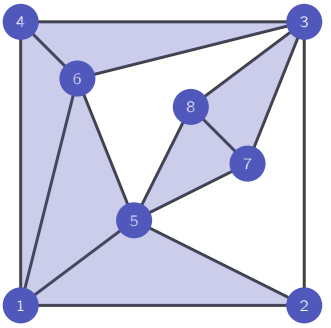
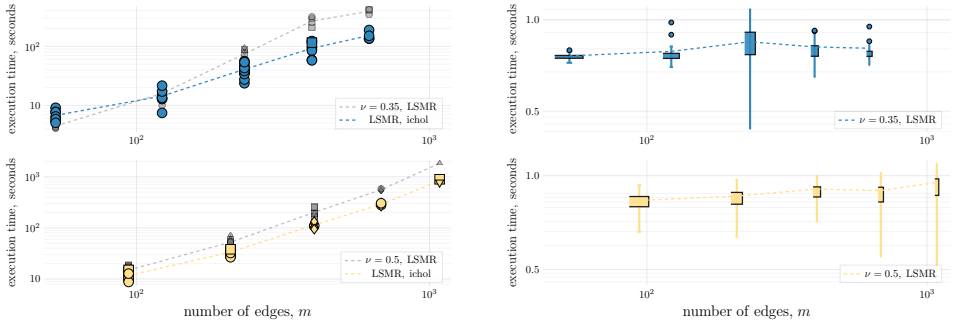


Figure VI.3: Example of Triangulation and Holes



(a) Time (in seconds) **(b)** Perturbation norm, ε

Figure VI.4: Benchmarking Results on the Synthetic Triangulation Dataset: varying sparsities $\nu = 0.35, 0.5$ and $N = 16, 22, 28, 34, 40$; each network is sampled 10 times. Shapes correspond to the number of eliminated edges in the final perturbation: $1 : \bigcirc$, $2 : \square$, $3 : \pentagon$, $4 : \triangle$. For each pair (ν, N), the un-preconditioned and Cholesky-preconditioned execution times are shown.

We sample networks with a varying number of vertices $N = 10, 16, 22, 28$ and varying sparsity pattern $\nu = 0.35, 0.5$ which determine the number of edges in the output as $m = \nu \frac{N(N-1)}{2}$. Due to the highly randomized procedure, topological structures of a sampled graph with a fixed pair of parameters may differ substantially, so 10 networks with the same (N, ν) pair are generated. For each network, the working time (without considering the sampling itself) and the resulting perturbation norm ε are reported in Figure VI.4a and Figure VI.4b, respectively. As anticipated in Section V.I, we show the performance of two implementations of the method, one based on LSMR and one based on LSMR preconditioned by using the incomplete Cholesky factorization of the initial matrices. We observe that,

- ♢ the computational cost of the whole procedure lies between $\mathcal{O}(m^2)$ and $\mathcal{O}(m^3)$
- ♢ denser structures, with a higher number of vertices, result in a higher number of edges being eliminated; at the same time, even most dense cases still can exhibit structures requiring the elimination of a single edge, showing that the flow does not necessarily favor multi-edge optima;
- ♢ the required perturbation norm ε is growing with the size of the graph, Figure VI.4b, but not too fast: it is expected that denser networks would require larger ε to create a new hole; at the same time if the

perturbation were to grow drastically with the sparsity ν, it would imply that the method tries to eliminate sufficiently more edges, a behavior that resembles convergence to a sub-optimal perturbation;

♢ preconditioning with a constant incomplete Cholesky multiplier, computed for the initial Laplacians, provides a visible execution time gain for medium and large networks. Since the quality of the preconditioning deteriorates as the flow approaches the minimizer (as a non-zero eigenvalue becomes 0), it is worth investigating the design of a preconditioner for the up-Laplacian that can be efficiently updated.

VI.III Transportation Networks

Finally, we provide an application to real-world examples based on city transportation networks. We consider networks for Bologna, Anaheim, Berlin Mitte, and Berlin Tiergarten; each network consists of nodes — intersections/public transport stops — connected by edges (roads) and subdivided into zones; for each road, the free flow time, length, speed limit are known; moreover, the travel demand for each pair of nodes is provided through the dataset of recorded trips. All the datasets used here are publicly available at `https://github.com/bstabler/TransportationNetworks`; Bologna network is provided by the Physic Department of the University of Bologna (enriched through the Google Maps API `https://developers.google.com/maps`).

The regularity of city maps naturally lacks 3-cliques, hence forming the simplicial complex based on triangulations as done before frequently leads to trivial outcomes. Instead, here we "lift" the network to city zones, thus more effectively grouping the nodes in the graph. Specifically:

1. we consider the completely connected graph where the nodes are zones in the city/region;

2. the free flow time between two zones is temporarily assigned as a weight of each edge: the time is as the shortest path between the zones (by the classic Dijkstra algorithm) on the initial graph;

3. similarly to what is done in the filtration used for persistent homology, we filter out excessively distant nodes; additionally, we exclude the longest edges in each triangle in case it is equal to the sum of two other edges (so the triangle is degenerate and the trip by the longest edge is always performed through to others);

4. finally, we use the travel demand as an actual weight of the edges in the final network; travel demands are scaled **logarithmically** via the

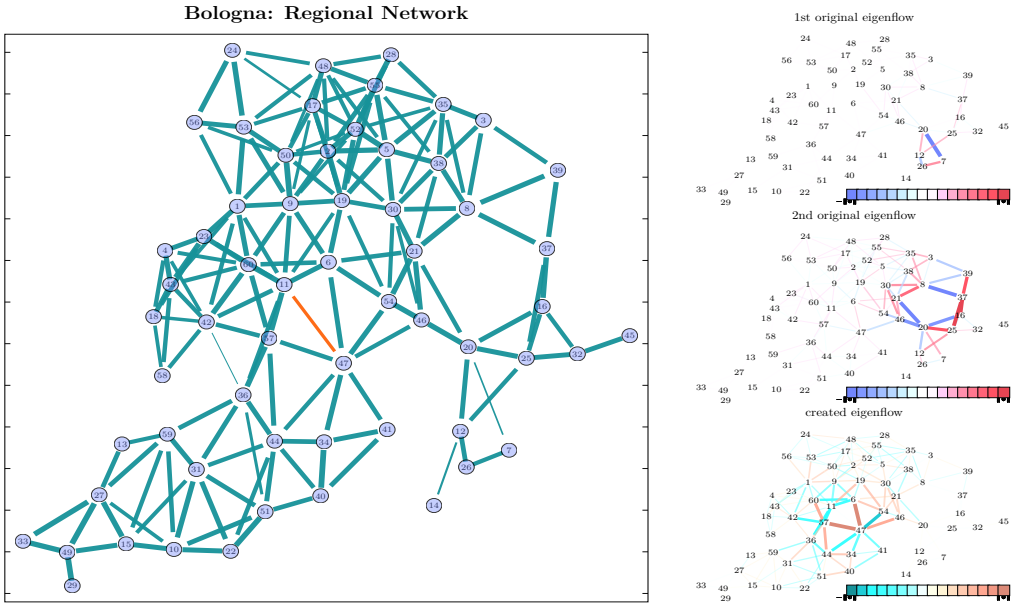


Figure VI.5: Example of the Transportation Network for Bologna. Left pane: original zone graph where the width of edges corresponds to the weight; to-be-eliminated edge is colored in red. Right pane: eigenflows, original and created; color and width correspond to the magnitude of entries.

transformation $w_i \mapsto \log_{10}\left(\frac{w_i}{0.95 \min w_i}\right)$; see the example on the left panel of Figure VI.5.

Given the definition of weights in the network, high instability (corresponding to small perturbation norm ε) implies structural phenomena around the "almost-hole", where the faster and shorter route is sufficiently less demanded.

In the case of Bologna, Figure VI.5, the algorithm eliminates the edge $[11, 47]$ (Casalecchio di Reno – Pianoro), creating a new hole $6 - 11 - 57 - 47$. We also provide examples of the eigenflows in the kernel of the Hodge Laplacian (original and additional perturbed): original eigenvectors correspond to the circulations around holes $7 - 26 - 12 - 20$ and $8 - 21 - 20 - 16 - 37$ non-locally spread in the neighborhood, [SBH+20].

The results for four different networks are summarized in Table 2; p mimics the percentile, $\varepsilon / \sum_{e \in \mathcal{V}_1} w_i(e)$, showing the overall small perturbation norm contextually. At the same time, we emphasize that except for Bologna (which is influenced by the geographical topology of the land), the algorithm does not choose the smallest weight possible; indeed, given our interpretation of the topological instability, the complex for Berlin-Tiergarten is stable, and the transportation network is effectively constructed.

Cities	network			β_1	logarithmic weights		
	m_0	m_1	m_2		time	ε	p
Bologna	60	175	171	2	2.43s	0.65	0.003
					[11, 47] (4^{th} smallest)		
Anaheim	38	159	221	1	5.39s	0.57	0.003
					[10, 29] (11^{th} smallest)		
Berlin-Tiergarten	26	63	55	0	2.46s	1.18	0.015
					[6, 16] (20^{th} smallest)		
Berlin-Mitte	98	456	900	1	127s	0.887	0.0016
					[57, 87] (6^{th}), **[58, 87]**, (17^{th})		

Table 2: Topological instability of the transportation networks: filtered zone networks with the corresponding perturbation norm ε and its percentile among $w_1(\cdot)$ profile. For each simplicial complex, the number of nodes, edges, and triangles in $\mathcal{V}_2(\mathcal{K})$ are provided alongside the initial number of holes β_1. The results of the algorithm consist of the perturbation norm, ε, computation time, and approximate percentile p.

IV Preconditioning for efficient solver of Laplacian LS

Linear systems are ubiquitous in science and engineering, and the study and the design of efficient solvers is of utmost importance as a consequence. In the case of higher-order Laplacians L_k, the system

$$L_k \mathbf{x} = \mathbf{f} \tag{Eqn. 74}$$

appears in various problems (generally, in the least square sense). Examples include:

- ♢ in the computation of the lower part of the spectrum of L_k (e.g. for the first non-zero eigenvalue λ_+ from the target functional in Chapter III or in the evolution of spectrum in the persistent homology, [GS23a]) where one typically needs an iterative solution $L_k \mathbf{x}_l = \mathbf{x}_{l-1}$;
- ♢ for dynamical systems governed by the Laplacian operator L_k (e.g. random walks):

  $$\dot{\mathbf{x}} = L_k \mathbf{x} - \mathbf{f} \tag{Eqn. 75}$$

  inside implicit numerical integrators or when studying the stationary point, both based on Equation (74);
- ♢ in the case of implicit graph neural networks, [GCZ+20], on simplices where the implicit equation determines the output feature vector of the layer

  $$\mathbf{x} = \phi\left(W \mathbf{x} L_k + B\right) \tag{Eqn. 76}$$

- ♢ finally, given the Hodge Decomposition in Theorem II.III.2, one is frequently interested in the projection of a flow $\mathbf{x}$ onto harmonic, curl or gradient subspaces which requires efficient solvers for $L_k^{\downarrow}\mathbf{x} = \mathbf{f}$ and $L_k^{\uparrow}\mathbf{x} = \mathbf{f}$ in the least square sense, [SBH+20].

The efficiency of the pre-existing solvers for Equation (74) is frequently determined by what we call computational stability of the matrix L_k (i.e. sensitivity of solutions of the linear system to small perturbations of the right-hand side $\mathbf{f}$), especially for a vast array of iterative solvers. In that scope, we associate computational or numerical stability with a condition number of the matrix, $\kappa(L_k)$, so unstable systems are typically poorly conditioned. Moreover, as we will demonstrate later, the convergence of iterative methods is frequently governed by the condition number of the matrix.

As a result, in this chapter, we aim to develop an efficient solver for the system in Equation (74) (in the least square sense) by reducing it to a smaller sparser system and providing a preconditioning scheme for such a system. Whilst in the case of the topological stability above, we explored a graph theoretical notion of instability (how many edges one needs to eliminate to create another hole in the homology group $\overline{\mathcal{H}}_k$) through a computational lens, here we attempt to mitigate a form of numerical instability by exploiting the underlying topology of the simplicial complex.

I. Reduction to a least-square problem for up-Laplacian

We have outlined above the motivation for the efficient solver for linear system $L_k\mathbf{x} = \mathbf{f}$ in the least square sense for determining topological feature through the spectrum $\sigma(L_k)$ or describing system's dynamics and stationary points. At the same time, higher-order Laplacians L_k exhibit two fundamental properties further implying the existence of efficient solvers: (1) they are naturally sparse and (2) $\ker L_k$ induces the Hodge Decomposition of the space, Theorem II.III.2, which can be leveraged to obtain a simpler solver.

Indeed, note that k-th order Laplacians L_k, $k > 0$, are matrices with a large number of zero entries since $m_k \ll m_{k-1}^2$ asymptotically (e.g. the number of triangles m_2 is always bounded by $m_1^{3/2}$) and, thus, are never dense. Consequently, L_k is necessarily a sparse matrix and, therefore, its corresponding `matvec` operation is cheap; as a result, any solver that is limited to exclusively calling `matvecs` with L_k would be relatively inexpensive.

Due to this intrinsic form of sparsity for the matrix L_k, in the following, we shall use the term **sparse** to indicate a structural property of the simplex $\mathcal{K}$ rather than its Laplacian matrices. In particular, in analogy with the classical graph case, we say that the simplicial complex $\mathcal{K}$ is k-sparse if $m_k = \mathcal{O}(m_{k-1} \log m_{k-1})$, that is the number of simplices of higher order is comparable up to a constant times the number of simplices of lower order times its logarithm. This is, for example, the case for structured simplicies such as trees for $k = 0$ or triangulations for $k = 1$. Moreover, one can show that for each dense simplicial complex $\mathcal{K}$ and for any given order k, there is a k-sparse $\mathcal{K}'$ such that the corresponding up-Laplacians $L_k^{\uparrow}(\mathcal{K})$ and $L_k^{\uparrow}(\mathcal{K}')$ are spectrally close, [OPW22], which we comment on later in Theorem IV.III.9.

Finally, we note that the existence of the Hodge Decomposition and Principle Spectral Inheritance, Theorem III.III.5, allows us to reduce solving $L_k\mathbf{x} = \mathbf{f}$ to the solution of $L_k^{\uparrow}\mathbf{x} = \mathbf{f}$ via the following direct result:

Th IV.I.7 **(Joint k-Laplacian solver)** The least-square problem $L_k\mathbf{x} = \mathbf{f}$ can be reduced to a sequence of consecutive least-square problems for isolated up-Laplacians. Precisely, $\mathbf{x}$ is a solution of IV.I.7,

$$L_k\mathbf{x} = \mathbf{f} \qquad \text{s. t.} \qquad \mathbf{x}, \mathbf{f} \perp \ker L_k$$

if and only if it can be written as $\mathbf{x} = B_k^\top \mathbf{u} + \mathbf{x}_2$, where:

$$\widehat{\mathbf{u}} = \arg\min_{\mathbf{z}} \left\| L_{k-1}^\uparrow \mathbf{z} - B_k \mathbf{f}_1 \right\|, \quad \mathbf{u} = \arg\min_{\mathbf{z}} \left\| L_{k-1}^\uparrow \mathbf{z} - \widehat{\mathbf{u}} \right\|,$$
$$\mathbf{x}_2 = \arg\min_{\mathbf{y}} \left\| L_k^\uparrow \mathbf{y} - \mathbf{f}_2 \right\|$$

and $\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2$ with $\mathbf{f}_1 = B_k^\top \mathbf{z}_1$, $\mathbf{z}_1 = \arg\min_{\mathbf{z}} \left\| L_{k-1}^\uparrow \mathbf{z} - B_k \mathbf{f} \right\|$.

Proof Given Hodge decomposition, Theorem II.III.2, the solution $\mathbf{x}$ can be decomposed into the gradient and curl parts, $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 = B_k^\top \mathbf{u} + B_{k+1}\mathbf{v}$ for some $\mathbf{u}$, $\mathbf{v}$ (since $\mathbf{x} \perp \ker L_k$). Then, the system $L_k^\uparrow \mathbf{x} = \mathbf{f}$ is equivalent to:

$$\begin{cases} L_k^\downarrow B_k^\top \mathbf{u} = B_k^\top B_k B_k^\top \mathbf{u} = \mathbf{f}_1 \\ L_k^\uparrow B_{k+1}\mathbf{v} = B_{k+1} B_{k+1}^\top B_{k+1}\mathbf{v} = \mathbf{f}_2 \\ \mathbf{f}_1 + \mathbf{f}_2 = \mathbf{f} \end{cases}$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ is the similar Hodge decomposition of the right hand side $\mathbf{f}$, with $\mathbf{f}_1 \in \operatorname{im} B_k^\top$, $\mathbf{f}_2 \in \operatorname{im} B_{k+1}$ and $\mathbf{f} \perp \ker L_k$. Then $\mathbf{f} = B_k^\top \mathbf{z}_1 + B_{k+1}\mathbf{z}_2$ and, after multiplication by B_k, $B_k B_k^\top \mathbf{z}_1 = L_{k-1}^\uparrow \mathbf{z}_1 = B_k \mathbf{f} \iff \min_{\mathbf{z}_1} \left\| L_{k-1}^\uparrow \mathbf{z}_1 - B_k \mathbf{f} \right\|$ and $\mathbf{f}_2 = \mathbf{f} - B_k^\top \mathbf{z}_1$.
Finally, we note that equation $B_k^\top B_k B_k^\top \mathbf{u} = \mathbf{f}_1 \iff (L_{k-1}^\uparrow)^2 \mathbf{u} = B_k \mathbf{f}_1$ can be solved by two consecutive least-square problems:

$$\begin{cases} \min_{\widehat{\mathbf{u}}} \left\| L_{k-1}^\uparrow \widehat{\mathbf{u}} - B_k \mathbf{f}_1 \right\| \\ \min_{\mathbf{u}} \left\| L_{k-1}^\uparrow \mathbf{u} - \widehat{\mathbf{u}} \right\| \end{cases}$$

which corresponds to the solution of the down part of the original system. ■

Rem I.18 Note that Theorem IV.I.7 can be immediately generalized to the case of weighted operators $\overline{B}_k$ and $\overline{L}_k$ without any alterations. Moreover, in order to solve $L_k^\uparrow \mathbf{x} = \mathbf{f}$ we assume $W_{k-1} = I$ from now and on; indeed, since W_{k-1} is diagonal and non-singular, the transition between $L_k^\uparrow \mathbf{x} = W_{k-1}^{-1} B_k W_k^2 B_k^\top W_{k-1}^{-1} \mathbf{x} = \mathbf{f}$ and $B_k W_k^2 B_k^\top \widehat{\mathbf{x}} = \widehat{\mathbf{f}}$ is immediate and cheap. Finally, to simplify the notation, we further refer to the weighted operator $\overline{B}_k = B_k W_k$ as B_k unless the unweighted case is specifically stated.

To summarize:

♢ L_k is a necessarily sparse matrix with cheap `matvec` operations (and maybe even cheaper in the case of sparse simplicial complexes) and benefits from a solver limited to such operations;

♢ due to Hodge's theory and Theorem IV.I.7, it is sufficient to develop efficient solvers only for the least-square problem for the up-Laplacian, $\arg\min_{\mathbf{x}\perp\ker L_k^{\uparrow}} \left\| L_k^{\uparrow}\mathbf{x} - \mathbf{f} \right\|$, instead of the whole operator L_k.

In the next section, we recall the main ideas of standard iterative methods for linear systems and least-square problems that rely exclusively on `matvec` operations and show the connection between the convergence rate of such methods and the condition number of $L_k^{\uparrow}$.We, then, will introduce a preconditioning strategy for $L_k^{\uparrow}$ named **heavy collapsible subcomplex** preconditioning (HeCS-preconditioning), which aims at reducing the condition number $\kappa(L_k^{\uparrow})$ by leveraging topological properties of $\mathcal{K}$.

II. Iterative Methods and Preconditioning: an overview

Before addressing the question of the higher-order up-Laplacian $L_k^{\uparrow}$, let us briefly recite the main ideas of iterative methods for linear systems and least-square problems and their connection to the matrix's condition number.

II.I Iterative methods

An iterative method for solving a linear system $A\mathbf{x} = \mathbf{b}$ consists of computing a sequence $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots\}$ converging to the exact solution, $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^* = A^{-1}\mathbf{b}$. Typically, the following classes of iterative methods are distinguished:

1. **stationary iterative methods**: for the system $A\mathbf{x} = \mathbf{b}$, one searches for the solution as a stationary point of a suitable fixed point problem $\mathbf{x} = B\mathbf{x} + \mathbf{c}$. The sequence $\{\mathbf{x}_k\}$ generated by the method converges to the unique fixed point $\mathbf{x}^*$, coinciding with a correct solution of the linear system.

   For instance, setting $A = M - N$ with M is invertible, we get:

$$\begin{aligned} A\mathbf{x} = \mathbf{b} \quad &\Leftrightarrow \quad (M-N)\,\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad M\mathbf{x} = N\mathbf{x} + \mathbf{b} \\ &\Leftrightarrow \quad \mathbf{x} = \underbrace{\left(M^{-1}N\right)}_{B}\mathbf{x} + \underbrace{M^{-1}\mathbf{b}}_{\mathbf{c}} \end{aligned} \qquad \text{(Eqn. 77)}$$

   Specific choices and forms of matrices M and N define Jacobi, Gauss-Seidel, and relaxation methods, [Saa03]. Moreover, one immediately sees that the spectral radius $\rho(M^{-1}N)$ characterizes the convergence

of the sequence $\mathbf{x}_{k+1} = M^{-1}N\mathbf{x}_k + M^{-1}\mathbf{b}$. Additionally, if A is Hermitian, one can guarantee $\rho(M^{-1}N) < 1$ for the specific choice of M providing a convergent fixed point method;

2. **Krylov subspace methods**: for each vector $\mathbf{v}$ one defines a **Krylov subspace** $\mathcal{K}_l(A, \mathbf{v}) = \text{span}\left\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots A^{l-1}\mathbf{v}\right\}$ and each next approximation step $\mathbf{x}_l$ is searched for in the affine space $\mathbf{x}_0 + \mathcal{K}_l(A, \mathbf{b} - A\mathbf{x}_0)$. Naturally, one searches for the best possible approximation in such space minimizing the L^2-residual, $\min_{\mathbf{v}\in\mathbf{x}_0+\mathcal{K}_l} \|A\mathbf{v} - \mathbf{b}\|_2^2$. In a similar fashion, one can move from solving the linear system to the optimization of the function $\phi(\mathbf{x})$ with the global optimum $\mathbf{x}^* = A^{-1}\mathbf{b}$ using Krylov subspace methods and other iterative optimization routines to obtain the solution; we provide an example of such a function below.

For the sake of completeness, in the following subsection, we demonstrate the connection between the Krylov subspace optimization method known as **conjugate gradient method** and the operator's condition number $\kappa(A)$.

II.II Conjugate gradient method and its convergence

Note that all discussed higher-order Laplacian operators L_k, including up- and down-terms $L_k^{\uparrow}$ and $L_k^{\downarrow}$, are symmetric semi-positive definite. Let us define the **conjugate gradient** (CG) method for a strictly positive definite operator A. This is a particular Krylov-based method motivated by the optimization of the objective function $\phi(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} - \mathbf{x}^\top\mathbf{b}$. An extension of the method is known as **conjugate gradient method for least-square problems** (CGLS) and exhibits virtually the same convergence estimate.

The idea of the CG method is to perform iterative updates of $\mathbf{x}_k$ in the direction $\mathbf{d}_k$ such that directions $\{\mathbf{d}_k\}$ are A-orthogonal (or A-conjugate). In other words, one aims to move $\mathbf{x}_k$ orthogonally to the direction of the previous step to avoid inefficient "zigzagging" of the classical gradient descent method, [HS+52]. Below, we formulate this idea in terms of Krylov subspaces and discuss its convergence rate.

Def. 10 **(Krylov subspace)** Let $\mathbf{x}_0$ be the initial guess with the initial residual $\mathbf{r}_0 = \mathbf{b}_0 - A\mathbf{x}_0$ and initial error $\mathbf{e}_0 = \mathbf{x}^* - \mathbf{x}_0$. Consider the l-th **Krylov subspace**:

$$\mathcal{K}_l = \mathcal{K}_l(A, \mathbf{r}_0) = \text{span}\left\{\mathbf{r}_0, A\mathbf{r}_0, \dots A^{l-1}\mathbf{r}_0\right\}. \quad \text{(Eqn. 78)}$$

Let P_m be a space of m-degree polynomials, we have

$$\mathcal{K}_l = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots A\mathbf{r}_{l-1}\} = \{p(A)\mathbf{r}_0 \mid p \in P_{l-1}\}. \quad \text{(Eqn. 79)}$$

In other words, the Krylov subspace $\mathcal{K}_l$ is the space of actions of all polynomials of the matrix A of degree $\leq l-1$ on the initial residual vector $\mathbf{r}_0$.

Note that, by definition, the initial error $\mathbf{e}_0$ and the initial residual $\mathbf{r}_0$ are connected through the equation:

$$A\mathbf{e}_0 = A(\mathbf{x}_* - \mathbf{x}_0) = \mathbf{b} - A\mathbf{x}_0 = \mathbf{r}_0 \qquad \text{(Eqn. 80)}$$

Let us assume that one has the initial guess, residual, and error $\mathbf{x}_0, \mathbf{r}_0, \mathbf{e}_0$ respectively; now let us formulate the problem of finding the best approximation $\mathbf{x}_l$ in the shifted Krylov subspace as we stated earlier:

Problem Find $\mathbf{e}_l \in \mathcal{K}_l$ such that it is the best approximation of $\mathbf{e}_0$ in $\mathcal{K}_l$ in terms of the operator norm $\|\mathbf{v}\|_A = \sqrt{\langle \mathbf{v}, \mathbf{v}\rangle_A} = \sqrt{\langle \mathbf{v}, A\mathbf{v}\rangle}$:

$$\text{find } \mathbf{e}_l \in \mathcal{K}_l: \quad \|\mathbf{e}_l - \mathbf{e}_0\|_A \leq \|\mathbf{e}_0 - \mathbf{e}\|_A \quad \text{for } \forall \mathbf{e} \in \mathcal{K}_l \qquad \text{(Eqn. 81)}$$

Note that this problem coincides with:

$$\text{find } \mathbf{x}_l \in \mathbf{x}_0 + \mathcal{K}_l: \quad \|\mathbf{x}_l - \mathbf{x}^*\|_A \leq \|\mathbf{x}^* - \mathbf{x}\|_A \quad \text{for } \forall \mathbf{x} \in x_0 + \mathcal{K}_l \qquad \text{(Eqn. 82)}$$

Expectably, the concept of the closest approximation in the sense of the A-norm involves the idea of A-orthogonality, as we review next.

Note that the residuals also span each Krylov subspace:

$$\mathcal{K}_{l+1}(A, \mathbf{r}_0) = \text{span}\left\{\mathbf{r}_0, A\mathbf{r}_0 \ldots A^l\mathbf{r}_0\right\} = \text{span}\{\mathbf{r}_0, \mathbf{r}_1 \ldots \mathbf{r}_l\} \qquad \text{(Eqn. 83)}$$

Indeed, by definition Krylov subspaces are nested, $\mathcal{K}_l(A, \mathbf{r}_0) \subseteq \mathcal{K}_{l+1}(A, \mathbf{r}_0)$, and, if $\mathbf{x}_l - \mathbf{x}_0 \in \mathcal{K}_l(A, \mathbf{r}_0)$, then

$$\mathbf{r}_l = \mathbf{b} - A\mathbf{x}_l = \underbrace{\underbrace{\mathbf{b} - A\mathbf{x}_0}_{=\mathbf{r}_0 \in \mathcal{K}_0 \subseteq \mathcal{K}_{l+1}} - A\underbrace{(\mathbf{x}_l - \mathbf{x}_0)}_{\in \mathcal{K}_l}}_{\in \mathcal{K}_{l+1}} \in \mathcal{K}_{l+1} \qquad \text{(Eqn. 84)}$$

so $\mathbf{r}_l \in \mathcal{K}_{l+1}$. At the same time, $\mathbf{r}_l \perp \mathcal{K}_l(A, \mathbf{r}_0)$: indeed, for the best approximation $\mathbf{x}_l \in \mathbf{x}_0 + \mathcal{K}_l(A, \mathbf{r}_0)$, the A-distance between $\mathbf{x}^*$ and $\mathbf{x}_l$ is minimal. As a result, if one adds an arbitrary vector $\alpha\mathbf{v}$ with $\mathbf{v} \in \mathcal{K}_l$, then the norm $\|\mathbf{x}^* - \mathbf{x}_l + \alpha\mathbf{v}\|_A^2 = \|\mathbf{x}^* - \mathbf{x}_l\|_A^2 + 2\alpha\langle \mathbf{x}^* - \mathbf{x}_l, \mathbf{v}\rangle_A + \alpha^2\|v\|_A^2$ has the minimum at $\alpha = 0$, so $0 = \langle \mathbf{x}^* - \mathbf{x}_l, \mathbf{v}\rangle_A = \langle A(\mathbf{x}^* - \mathbf{x}_l), \mathbf{v}\rangle = \langle \mathbf{b} - A\mathbf{x}_l, \mathbf{v}\rangle = \langle \mathbf{r}_l, \mathbf{v}\rangle$ for any $\mathbf{v} \in \mathcal{K}_l(A, \mathbf{r}_0)$. So, each residue vector $\mathbf{r}_l$ lies in the next Krylov subspace $\mathcal{K}_{l+1}$ and is orthogonal to the previous subspace $\mathcal{K}_l$, thus forming a basis.

Instead of $\mathbf{r}_l$, one can define an orthogonal (with respect to A) basis of $\mathcal{K}_{l+1}(A, \mathbf{r}_0) = \text{span}\{\mathbf{d}_0, \mathbf{d}_1 \dots \mathbf{d}_l\}$ which can be done iteratively via Gram-Schmidt orthogonalization:

$$\mathbf{d}_l = \mathbf{r}_l - \sum_{i=0}^{l-1} \beta_i \mathbf{d}_i, \text{ where } \beta_i = \frac{\langle \mathbf{r}_l, \mathbf{d}_i \rangle_A}{\|\mathbf{d}_i\|_A^2} \quad \text{(Eqn. 85)}$$

Note that $\mathbf{r}_l \perp \mathcal{K}_l$, so $\langle \mathbf{r}_l, \mathbf{d}_i \rangle_A = \langle \mathbf{r}_l, \underbrace{A\mathbf{d}_i}_{\in \mathcal{K}_{i+2}} \rangle = 0$ if $i \leq l-2$, so

$$\mathbf{d}_l = \mathbf{r}_l - \beta_{l-1}\mathbf{d}_{l-1}, \quad \beta_{l-1} = \frac{\langle \mathbf{r}_l, \mathbf{d}_{l-1} \rangle_A}{\|\mathbf{d}_{l-1}\|_A^2} \quad \text{(Eqn. 86)}$$

Since $\mathbf{x}_l - \mathbf{x}_{l-1} \in \mathcal{K}_l(A, \mathbf{r}_0)$ and $\mathbf{x}_l - \mathbf{x}_{l-1} = (\mathbf{x}_l - \mathbf{x}^*) + (\mathbf{x}^* - \mathbf{x}_{l-1}) \perp K_{l-1}(A, \mathbf{r}_0)$ by the similar arguments, the difference between approximation is colinear to the direction vector $\mathbf{d}_{l-1}$, $\mathbf{x}_l - \mathbf{x}_{l-1} = \alpha_l \mathbf{d}_{l-1}$. To obtain the coefficient α_l, it is sufficient to note that:

$$\alpha_l A\mathbf{d}_{l-1} = A(\mathbf{x}_l - \mathbf{x}_{l-1}) = (A\mathbf{x}_l - \mathbf{b}) - (A\mathbf{x}_{l-1} - \mathbf{b}) = \mathbf{r}_{l-1} - \mathbf{r}_l \quad \text{(Eqn. 87)}$$

so

$$\alpha_l \langle A\mathbf{d}_{l-1}, \mathbf{r}_{l-1} \rangle = \langle \mathbf{r}_{l-1} - \mathbf{r}_l, \mathbf{r}_{l-1} \rangle = \|\mathbf{r}_{l-1}\|^2 \quad \text{(Eqn. 88)}$$

As a result, we obtain the following iterative scheme which defines the CG method (more details in Algorithm 3):

$$\begin{aligned} \mathbf{d}_l &= \mathbf{r}_l - \beta_{l-1}\mathbf{d}_{l-1} \quad \text{with } \beta_{l-1} = \frac{\langle \mathbf{r}_l, \mathbf{d}_{l-1} \rangle_A}{\|\mathbf{d}_{l-1}\|_A^2} \\ \mathbf{r}_l &= \mathbf{r}_{l-1} - \alpha_l A\mathbf{d}_{l-1} \quad \text{with } \alpha_l = \frac{\|\mathbf{r}_{l-1}\|^2}{\langle \mathbf{d}_{l-1}, \mathbf{r}_{l-1} \rangle_A} \\ \mathbf{x}_l &= \mathbf{x}_{l-1} + \alpha_l \mathbf{d}_{l-1} \end{aligned} \quad \text{(Eqn. 89)}$$

II.III Condition number and convergence rate of CG

Let us briefly discuss the condition number $\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2$ which, for a positive definite matrix A, can be written as $\kappa = \lambda_n / \lambda_1$ where λ_1 and λ_n are the smallest and largest eigenvalue of A, respectively. Typically, the condition number is used to characterize the stability of the matrix in terms of the solution of the corresponding linear system. Specifically, assume one has the perturbed linear system:

$$A(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b} \quad \text{(Eqn. 90)}$$

Algorithm 3 Conjugate Gradient Method [HS+52, BES98]

Require: positive definite A, RHS $\mathbf{b}$, initial guess $\mathbf{x}_0$

$\mathbf{r}_0 \leftarrow \mathbf{b} A\mathbf{x}_0$ ▷ compute initial residual

$\mathbf{d}_0 \leftarrow \mathbf{r}_0$ ▷ compute initial direction

for $l = 1, \ldots$ and until `stoppingCriterion` **do**

  $\alpha_l \leftarrow \frac{\|\mathbf{r}_{l-1}\|^2}{\|\mathbf{d}_{l-1}\|_A^2}$

  $\mathbf{r}_l \leftarrow \mathbf{r}_{l-1} - \alpha_l A\mathbf{d}_{l-1}$

  $\mathbf{x}_l \leftarrow \mathbf{x}_{l-1} + \alpha_l \mathbf{d}_{l-1}$

  $\beta_{l-1} \leftarrow -\frac{\|\mathbf{r}_l\|^2}{\|\mathbf{r}_{l-1}\|^2}$

  $\mathbf{d}_l \leftarrow \mathbf{r}_l - \beta_{l-1}\mathbf{d}_{l-1}$

end for

where $\delta\mathbf{x}$ is the error of the solution given by the perturbation of the input $\mathbf{b}$. One aims to characterize the relative error $\|\delta\mathbf{x}\|\backslash\|\mathbf{x}\|$ in terms of A and $\mathbf{b}$. Note that $A\mathbf{x} = \mathbf{b}$ and $A\delta\mathbf{x} = \delta\mathbf{b}$, then by the definition of the operator's norm:

$$\begin{matrix} \|A\|_2\|\mathbf{x}\| \geq \|\mathbf{b}\| \\ \|A^{-1}\|_2\|\delta\mathbf{x}\| \leq \|\delta\mathbf{b}\| \end{matrix} \quad \Longrightarrow \quad \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \kappa(A)\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \qquad \text{(Eqn. 91)}$$

So the condition number $\kappa(A)$ characterizes the size of the error with respect to the size of the perturbation on the data, or, in other words, how the solution of the linear system with matrix A blows up the perturbation $\delta\mathbf{b}$. Similarly, the condition number $\kappa(A)$ governs the quality of the approximation $\mathbf{x}_l$ in the CG method, as we demonstrate below.

Let us reiterate that by definition, a vector from the Krylov subspace $\mathcal{K}_l(A, \mathbf{r}_0)$ is the action of the polynomial of matrix A on the initial residue vector $\mathbf{r}_0$. As a result:

$$\|\mathbf{x}^* - \mathbf{x}_l\|_A = \min_{\mathbf{x}\in\mathbf{x}_0+\mathcal{K}_l} \|\mathbf{x}^* - \mathbf{x}\|_A = \min_{\mathbf{x}\in\mathbf{x}_0+\mathcal{K}_l} \|\mathbf{x}^* - \mathbf{x}_0 - (\underbrace{\mathbf{x} - \mathbf{x}_0}_{\mathbf{z}\in\mathcal{K}_l})\|_A = \min_{\mathbf{z}\in\mathcal{K}_l} \|\mathbf{e}_0 - \mathbf{z}\|_A =$$

$$= \min_{p\in P_{l-1}} \|\mathbf{e}_0 - p(A)\underbrace{\mathbf{r}_0}_{=A\mathbf{e}_0}\|_A = \min_{p\in P_l : p(0)=0} \|\mathbf{e}_0 - p(A)\mathbf{e}_0\|_A = \min_{p\in P_l : p(0)=1} \|p(A)\mathbf{e}_0\|_A$$

(Eqn. 92)

The remaining term $\|p(A)\mathbf{e}_0\|_A$ can be computed explicitly:

$$p(A)\mathbf{e}_0 = \sum_j p_j A^j \mathbf{e}_0 = \sum_j p_j A^j \sum_i x_i \xi_i = \sum_{i,j} p_j x_i \lambda_i^j \xi_i = \sum_i x_i p(\lambda_i)\xi_i$$

$$\|p(A)\mathbf{e}_0\|_A^2 = \langle p(A)\mathbf{e}_0, Ap(A)\mathbf{e}_0\rangle = \sum_{ij} x_i x_j p(\lambda_i)p(\lambda_j)\lambda_j\langle\xi_i, \xi_j\rangle = \sum_i |x_i|^2\lambda_i|p(\lambda_i)|^2$$

(Eqn. 93)

Then

$$\|\mathbf{x}^* - \mathbf{x}_l\|_A^2 = \min_{p\in P_l: p(0)=1} \|p(A)\mathbf{e}_0\|_A = \sum_i |x_i|^2 \lambda_i |p(\lambda_i)|^2 \leq \max_{\lambda\in\sigma(A)} p^2(\lambda) \sum_i \lambda_i |x_i|^2 =$$
$$= \max_{\lambda\in\sigma(A)} p^2(\lambda) \|\mathbf{e}_0\|_A^2 \tag{Eqn. 94}$$

As a result we have

$$\|\mathbf{x}^* - \mathbf{x}_l\|_A \leq \min_{p\in P_l: p(0)=1} \max_{\lambda\in\sigma(A)} |p(\lambda)| \|\mathbf{e}_0\|_A \tag{Eqn. 95}$$

Note that the estimation above bounds the approximation error $\|\mathbf{x}^* - \mathbf{x}\|_A$ from above by the minimum over the set of polynomials. Then, such estimation holds for every polynomial $p(x)$ of degree l and $p(0) = 1$. In order to obtain the final bound on the approximation error, one chooses the following family of polynomials.

Let $T_l(x)$ be a Chebyshev polynomial of the first kind defined on $x \in [-1, 1]$, such that

$$T_0(x) = 1, \quad , T_1(x) = x, \quad T_l = 2xT_{l-1}(x) - T_l(x), \quad \text{or}$$
$$T_l(x) = \frac{1}{2}\left[\left(x + \sqrt{x^2-1}\right)^l + \left(x - \sqrt{x^2-1}\right)^l\right] \tag{Eqn. 96}$$

It is well known that Chebyshev polynomials $T_l(x)$ has the minimal L_∞ norm (minimal largest absolute value, $\max_{x\in[-1,1]} |p(x)|$) over $[-1, 1]$ among monic polynomials of the same order. Then, in order to construct $p(x)$, we rescale the corresponding Chebyshev polynomial to take values from $x \in [\lambda_1, \lambda_n]$:

$$p(x) = \frac{1}{T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right)} T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1} - \frac{2}{\lambda_n-\lambda_1}x\right) \tag{Eqn. 97}$$

It is easy to see that $p(0) = T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right) / T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right) = 1$, moreover, $\|T_l(x)\|_\infty = 1$ and, due to the optimality of the L_∞ norm for Chebyshev polynomials, the estimation 95 is tight for the chosen $p(x)$. As a result,

$$\max_{\lambda\in\sigma(A)} |p(\lambda)| \leq \max_{x\in[\lambda_1,\lambda_n]} |p(x)| = \frac{1}{T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right)} \tag{Eqn. 98}$$

As a result

Th IV.II.8 **(Convergence rate of Conjugate Gradient method)** Let A be a symmetric positive definite matrix, $A \in \mathbb{R}^{n\times n}$. Then, the convergence of the

CG method follows the estimate:

$$\|\mathbf{x}_l - \mathbf{x}^*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^l \|\mathbf{e}_0\|_A \qquad \text{(Eqn. 99)}$$

where $\mathbf{e}_0$ is the initial error.

Proof As we already showed,

$$\|\mathbf{x}^* - \mathbf{x}_l\|_A \leq \min_{p\in P_l : p(0)=1} \max_{\lambda\in\sigma(A)} |p(\lambda)| \|\mathbf{e}_0\|_A \qquad \text{(Eqn. 100)}$$

and, for the rescaled Chebyshev polynomial,

$$\min_{p\in P_l : p(0)=1} \max_{\lambda\in\sigma(A)} |p(\lambda)| \leq \frac{1}{T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right)} \qquad \text{(Eqn. 101)}$$

Then it is sufficient to estimate $T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right)$:

$$\begin{aligned} 2T_l\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right) &= \left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1} + \sqrt{\frac{(\lambda_1+\lambda_n)^2}{(\lambda_n-\lambda_1)^2}-1}\right)^l + \left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1} - \sqrt{\frac{(\lambda_1+\lambda_n)^2}{(\lambda_n-\lambda_1)^2}-1}\right)^l = \\ &= \left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1} + \frac{2\sqrt{\lambda_1\lambda_n}}{\lambda_n-\lambda_1}\right)^l + \left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1} - \frac{2\sqrt{\lambda_1\lambda_n}}{\lambda_n-\lambda_1}\right)^l = \\ &= \left(\frac{\sqrt{\lambda_n}+\sqrt{\lambda_1}}{\sqrt{\lambda_n}-\sqrt{\lambda_1}}\right)^l + \left(\frac{\sqrt{\lambda_n}-\sqrt{\lambda_1}}{\sqrt{\lambda_n}+\sqrt{\lambda_1}}\right)^l \geq \left(\frac{\sqrt{\kappa(A)}+1}{\sqrt{\kappa(A)}-1}\right)^l \end{aligned} \qquad \text{(Eqn. 102)}$$

■

By Theorem IV.II.8, poorly conditioned (large $\kappa(A)$) operators A have a convergence exponent $\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1} \approx 1$ and, thus, CG converges slowly; on the contrary, the closer $\kappa(A)$ is to 1, the smaller is the exponent and the convergency is faster. Moreover, similar convergence bounds hold for all Krylov-type methods. As a result, one aims to transform the operator A in such a way that $\kappa(A)$ is as close to 1 as possible by means of a so-called **preconditioner**.

II.IV Zoo of preconditioners

Owing to what is discussed above, given a Krylov-type iterative method for the solution of the system $A\mathbf{x} = \mathbf{b}$, one aims to improve the condition number of the matrix A. The common idea here is to move to a new operator via simple matrix multiplication:

$$A\mathbf{x} = \mathbf{b} \qquad \longleftrightarrow \qquad (MA)\mathbf{x} = M\mathbf{b} \qquad \text{(Eqn. 103)}$$

where MA is the new operator for the iterative method. If A was poorly conditioned, then one aims for $\kappa(MA) \approx 1$, so M should be a easily computable approximation of A^{-1}. Then, the new system with MA operator is called a preconditioned system where M is referred to as a (left) preconditioner of the operator A.

Example

Assuming the operator A has no zero entries on the main diagonal, let D be a diagonal matrix with the same main diagonal as A, so $A_{ii} = D_{ii}$. Then, one can use $M = D^{-1}$, which is clearly always cheap to compute, as a left preconditioner for matrix A. Such a preconditioner is known to be effective for unnormalized or diagonally dominant matrices.

Rem II.19

Note that in practical implementations, the actual computation of M or the whole new operator MA is not necessary. Instead, one needs efficient `matvec` operations $(MA)\mathbf{x}_k = M(A\mathbf{x}_k)$, and, as a result, one searches for the preconditioner M with a fast `matvec` operation, simultaneously with $M \approx A^{-1}$ (note that matvecs of the inverse of the original operator A^{-1} is typically expensive).

In other cases, one can search for left and right preconditioning,

$$A\mathbf{x} = \mathbf{b} \qquad \longleftrightarrow \qquad (M^{-1}AN^{-1})(N\mathbf{x}) = M^{-1}\mathbf{b}\,, \quad \text{(Eqn. 104)}$$

where $A \approx MN$, or its symmetric version for symmetric positive definite A,

$$A\mathbf{x} = \mathbf{b} \qquad \longleftrightarrow \qquad (C^{-1}AC^{-\top})(C^{\top}\mathbf{x}) = C^{-1}\mathbf{b}\,, \quad \text{(Eqn. 105)}$$

where $A \approx CC^{\top}$. In both cases, one naturally needs to be able to perform matvec operations with the inverse operators M^{-1}, N^{-1} and C^{-1}. One of the most standard cases for the symmetric preconditioner C is searched for amongst lower-triangular matrices such that matvec operations with C^{-1} can be done efficiently through the forward substitution. The decomposition $A = CC^{\top}$, where C is lower-triangular and A is a positive definite, is known as **Cholesky decomposition** and C is known as **Cholesky multiplier**, [Hig90]. Direct calculations of the Cholesky multiplier typically require a full pass of Gauss elimination; moreover, sparse operators A tend to have dense Cholesky multiplier C, so `matvec` of the initial operator A may be ultimately cheaper than `matvec` with C^{-1}, despite the triangular structure. As much more efficient alternative, one can try to maintain the sparsity pattern or obtain some form of approximation of the Cholesky multiplier, known as **incomplete Cholesky preconditioner**, [Man80, GVL13]; similarly, one can obtain incomplete LU-decomposition as the left and right preconditioners for non-symmetric systems, [HW92].

Other examples of popular and efficient preconditioning schemes include

Algebraic Multigrid preconditioners (AGM), combinatorial preconditioners, and polynomial preconditioners. In these cases, the application and efficiency are primarily dependent on the nature of operator A, [Stü01, Y$^+$02, NN16, LB12, Saa85, SS08].

Often, preconditioners used in the literature do not provide theoretical guarantees on the preconditioning quality and typically are efficient in specific settings. Moreover, higher-order Laplacian operators L_k and, even more so, up- and down- terms $L_k^\uparrow$ and $L_k^\downarrow$ are singular and only symmetric semi-positive definite, which renders the majority of the existing preconditioners either expensive to compute, inefficient or inapplicable. In the rest of the chapter, we develop a new Cholesky-like preconditioning scheme based on the underlying simplicial complex that avoids all the abovementioned problems.

III. Preconditioning of the up-Laplacian

In the previous sections, we have reduced the task of solving the LS-problem for L_k to the corresponding sparse LS-problem for $L_k^\uparrow$ in the form $\min_{\mathbf{x}} \|L_k^\uparrow \mathbf{x} - \mathbf{f}\|_2^2$. To leverage the sparsity of $L_k^\uparrow$, one aims to apply one of the iterative solvers (like CGLS, [BES98, HS$^+$52]) whose convergence is highly dependent on the condition number. To be more precise, one needs to generalize the condition number $\kappa(\cdot)$ for the case of singular matrices since any up-Laplacian $L_k^\uparrow$ is necessarily singular. This is done by considering the "least-square" or "positive' condition number:

$$\kappa_+(L_k^\uparrow) = \frac{\sigma^+_{\max}(L_k^\uparrow)}{\sigma^+_{\min}(L_k^\uparrow)}, \qquad \text{(Eqn. 106)}$$

where $\sigma^+_{\max}(L_k^\uparrow)$ and $\sigma^+_{\min}(L_k^\uparrow)$ are the maximal and minimal positive singular values of $L_k^\uparrow$, respectively. It is immediate to note that such definition is coherent with the introduction of $\mathbf{x} \perp \ker L_k^\uparrow$ in the definition of the operator norm, i.e. $\| \left(L_k^\uparrow\right)^{-1} \|_2 = \sup_{\mathbf{x} \perp \ker L_k^\uparrow} \frac{\|\mathbf{x}\|_2}{\|L_k^\uparrow \mathbf{x}\|_2}$, similar to the least-square problem for $L_k^\uparrow$.

In order to reduce $\kappa_+(L_k^\uparrow)$, we move from $\min_{\mathbf{x}} \|L_k^\uparrow \mathbf{x} - \mathbf{f}\|$ to the symmetrically preconditioned system

$$\min_{\mathbf{x}} \| \left(C^\dagger L_k^\uparrow C^{\top\dagger}\right) \left(C^\top \mathbf{x}\right) - C^\dagger \mathbf{f}\| \qquad \text{(Eqn. 107)}$$

where $C^\dagger$ denotes the Moore-Penrose pseudoinverse of C, and C is chosen so that (a) the transition between unconditioned and preconditioned systems are bijective, (b) the matrix $C^\dagger L_k^\uparrow C^{\top\dagger}$ is better conditioned than the initial $L_k^\uparrow$, and (c) the pseudo-inverse of the preconditioner C can be efficiently im-

plemented. In particular, a cheap `matvec` of the pseudo-inverse $C^\dagger \mathbf{f}$ is guaranteed in the case of C being a lower-triangular matrix (e.g. for the **Cholesky** and the **incomplete Cholesky** preconditioners [Hig90, Man80, GVL13]). When C in Equation (107) is lower-triangular, we say that we are looking for a **Cholesky-like preconditioner**.

Proposed approach To obtain such preconditioner C for $L_k^\uparrow$, we aim to leverage the underlying structure of the simplicial complex $\mathcal{K}$. Our approach will follow similar graph-based preconditioning ideas from the literature as a starting point. Precisely,

- ♢ in the case of standard graphs, it is known that computing graph-induced preconditioners for the classical graph Laplacian L_0, [KS16, Tro19, LB12], is far more efficient when the graph is sparse. Moreover, the preconditioning of a dense graph is typically done by constructing a sparser approximating graph first and then by computing a preconditioner for the sparse approximant. For these reasons, here we assume the given simplicial complex $\mathcal{K}$ to be sparse. If this is not the case, then one can use sparsification ideas that we describe below to compute an initial sparse approximant;
- ♢ graph-based Cholesky or approximate Cholesky preconditioners for the case of classical graphs, $k = 0$, rely on the particular structures of the exact Cholesky multipliers of the graph Laplacian. We briefly describe the procedure for $k = 0$ in Subsection III.III to highlight the difference with the high-order case $k > 0$;
- ♢ we observe that computation of the exact Cholesky preconditioner for up-Laplacians $L_k^\uparrow$ is cheap and efficient for a specific topological class of simplicial complexes; as a result, we propose a preconditioning scheme based on finding a subcomplex of such topological class and exploiting its Cholesky multiplier to construct an efficient preconditioner.

III.I Sparsification of simplicial complexes

Frequently, one is interested in finding a sparser approximation M of a given operator A in order to use $M^\dagger$ as a left preconditioner of A; since M is sparser than A, the `matvec` operation for $M^\dagger$ is cheaper. Note that the usage of **any** existing sparsification of A does not guarantee better conditioning of $M^\dagger A$. Moreover, obtaining M may be computationally expensive. We review here a general result for sparsification from [OPW22], which generalizes to up-Laplacians of general simplicial complexes the graph Laplacian sparsification theorem from [SS08], which in practice exhibits both of aforementioned problems. Given an initial simplicial complex $\mathcal{K}$, the goal is to sample a number of simplices from $\mathcal{V}_{k+1}(\mathcal{K})$ in such a way that the obtained

sub-simplicial complex $\mathcal{L}$ is arbitrarily close to the original $\mathcal{K}$, in terms of the spectrum of the up-Laplacians. To state the precise result, we first introduce the definition of 'spectral proximity' for Hermitian matrices:

Def. 11 **(Spectral Approximation)** The Hermitian matrix A is called **spectrally ε-close** to the Hermitian matrix B, $A \underset{\varepsilon}{\approx} B$, if $(1-\varepsilon)B \preceq A \preceq (1+\varepsilon)B$, where $\succeq$ is the partial ordering induced by the positive definite cone, i.e. $A \succeq B$ if $A - B$ is positive semi-definite.

Rem III.20 Note that if $A \underset{\varepsilon}{\approx} B$, then one can directly bound the distance $\|\mathbf{x}_A - \mathbf{x}_B\|$, where $A\mathbf{x}_A = \mathbf{f}$ and $B\mathbf{x}_B = \mathbf{f}$. Indeed, $\mathbf{x}_B = B^{-1}A\mathbf{x}_A$ and $\|\mathbf{x}_A - \mathbf{x}_B\| = \|\left(I - B^{-1}A\right)\mathbf{x}_A\| \leq \|I - B^{-1}A\|\|\mathbf{x}_A\| \leq \|B^{-1}\| \cdot \|B - A\| \cdot \|\mathbf{x}_A\| \leq \varepsilon\|B^{-1}\| \cdot \|B\| \cdot \|\mathbf{x}_A\| = \varepsilon\kappa(B)\|\mathbf{x}_A\|$, with $\kappa(B)$ being the condition number of B. Thus, the relative error is controlled by the quality of the approximation ε and the condition number $\kappa(B)$. This does not necessarily mean that one can use an approximation B to obtain a solution of $A\mathbf{x} = \mathbf{f}$ (since obtaining a high-quality approximation for small ε may be expensive), but implies that assuming one has a preconditioner for one of the matrices, one can use it to precondition the other.

Th IV.III.9 **(Simplicial Sparsification, [OPW22])** Let $\mathcal{K}$ be a simplicial complex restricted to its p-skeleton, $\mathcal{K} = \bigcup_{i=0}^{p} \mathcal{V}_i(\mathcal{K})$. Let $L_k^{\uparrow}(\mathcal{K})$ be its k-th up-Laplacian and let $m_k = |\mathcal{V}_k(\mathcal{K})|$. For any $\varepsilon > 0$, a sparse simplicial complex $\mathcal{L}$ can be sampled as follows:

(1) compute the probability measure $\mathbf{p}$ on $\mathcal{V}_{k+1}(\mathcal{K})$ proportional to the generalized resistance vector $\mathbf{r} = \operatorname{diag}\left(B_{k+1}^{\top}(L_k^{\uparrow})^{\dagger}B_{k+1}\right)$, where $\operatorname{diag}(A)$ denotes the vector of the diagonal entries of A;

(2) sample q simplices τ_i from $\mathcal{V}_{k+1}(\mathcal{K})$ according to the probability measure $\mathbf{p}$, where q is chosen so that $q(m_k) \geq 9C^2 m_k \log(m_k/\varepsilon)$, for some absolute constant $C > 0$;

(3) form a sparse simplicial complex $\mathcal{L}$ with all the sampled simplexes of order k and all its faces with the weight $\frac{w_{k+1}(\tau_i)}{q(m_k)\mathbf{p}(\tau_i)}$; weights of repeated simplices are accumulated.

Then, with probability at least $1/2$, the up-Laplacian of the sparsifier $\mathcal{L}$ is ε-close to the original one, i.e. it holds $L_k^{\uparrow}(\mathcal{L}) \underset{\varepsilon}{\approx} L_k^{\uparrow}(\mathcal{K})$.

In other words, the generalized resistance $\mathbf{r}$ provides an estimation of the contribution of each simplex from $\mathcal{V}_{k+1}(\mathcal{K})$ into the spectral profile of $L_k^{\uparrow}$. It is sufficient to sample $q(m_k)$ $\left(\sim \frac{1}{\varepsilon}\right)$ simplices from the measure $\mathbf{p}$ to obtain an ε-approximation; note that in term of our definition above, that

would make $\mathcal{L}$ k-sparse. Additionally, note that the exact computation of the probability measure $\mathbf{p}$ is computationally expensive since it requires $(L_k^{\uparrow})^{\dagger}$; however, several approaches have been **proposed** to approximate it efficiently, [SS08]. While the efficiency of such algorithms for the computation of the measure is yet to be shown in the general case of $L_k^{\uparrow}$, such a study falls outside the scope of the current work.

Supposing the effective resistance $\mathbf{r}$ can be cheaply obtained and assuming one has a dense simplicial complex $\mathcal{K}$, Theorem IV.III.9 implies that it is sufficient to obtain a preconditioner C for its sparsifier $\mathcal{L}$ which then can be efficiently used to precondition the original complex $\mathcal{K}$. As a result, the task of finding an efficient preconditioner for any up-Laplacian in that scenario would be reduced to the case of k-sparse simplicial complexes. For this reason, as said before, we focus on the case of k-sparse simplicial complexes in the rest of the work.

Note that even in the case of a sparse simplicial complex, attempts to obtain the exact Cholesky multiplier directly may be computationally unfeasible; this, however, does not hold for the case of the classical graph Laplacian, $k = 0$, where one can indeed obtain approximate Cholesky factor corresponding to a sparsifier $\mathcal{L}$ (albeit not necessarily coinciding with Theorem IV.III.9 which is fundamentally an existence result), which we discuss next.

III.II Schur complements and Cholesky preconditioner

We start with a useful observation about the structure of the Gaussian elimination process to form the Cholesky factor. Let $A \in \mathbb{R}^{n\times n}$ be a real symmetric positive definite matrix. Then its **Schur complements** S_i obtained in the process of Cholesky factorization via Gaussian elimination can be defined recursively as follows: let δ_i be the i-th canonical vector, $(\delta_i)_i = 1$ and $(\delta_i)_j = 0$, $i \neq j$, then set $S_0 = A$ and for $i = 1, 2, 3, \ldots$

$$\begin{aligned} S_i &= S_{i-1} - \frac{1}{\alpha_i} S_{i-1}\delta_i\delta_i^{\top} S_{i-1}^{\top} \\ \mathbf{c}_i &= \frac{1}{\sqrt{\alpha_i}} S_{i-1}\delta_i \\ \alpha_i &= \delta_i^{\top} S_{i-1}\delta_i \end{aligned} \qquad \text{(Eqn. 108)}$$

With these definitions, the Cholesky factor C such that $A = CC^{\top}$ is formed by the columns $\mathbf{c}_i$, namely

$$C = [\mathbf{c}_1\, \mathbf{c}_2\, \ldots\, \mathbf{c}_n].$$

Since $\mathbf{c}_i$ is the ith column of the i-th step in Gauss elimination, matrix C defined above is indeed lower triangular. Moreover, note that

$$\frac{1}{\alpha_i} S_{i-1}\delta_i\delta_i^\top S_{i-1}^\top = \frac{1}{\sqrt{\alpha_i}} S_{i-1}\delta_i \left(\frac{1}{\sqrt{\alpha_i}} S_{i-1}\delta_i\right)^\top = \mathbf{c}_i\mathbf{c}_i^\top \quad \text{(Eqn. 109)}$$

so $S_i - S_{i-1} = \mathbf{c}_i\mathbf{c}_i^\top$ and $S_n = 0$. Thus, we have $A = S_0 = S_0 - S_n = \sum(S_i - S_{i-1}) = \sum \mathbf{c}_i\mathbf{c}_i^\top = CC^\top$. So, to obtain the exact Cholesky multiplier, one needs to collect the corresponding columns of Schur complements S_i during Gauss elimination; consequently, the algorithmic complexity of obtaining C is $\mathcal{O}\left(\frac{n^3}{6}\right)$. However, in the case of classical Laplacian L_0, the procedure can be sped up by exploiting the underlying graph structures, as we demonstrate next.

III.III Cholesky preconditioner for $k = 0$

In the case of the operator A being the classical graph Laplacian $L_0 = L_0^\uparrow$, there is a particularly useful way to interpret the Gaussian elimination process. Let $\mathcal{V}_0(\mathcal{K}) = \{[1], [2], \ldots, [m_0]\}$; then, structurally, each Schur complement S_k of the classical Laplacian L_0 remains a Laplacian of a smaller graph without vertices $[1], \ldots [k]$ and denser edge structure. Indeed, each step of Gauss elimination $S_k \to S_{k+1}$ eliminates the vertex $[k]$ from the graph and substitutes it with a completely connected clique on the vertices previously adjacent to $[k]$.

To formally show this, we introduce one of the key underlying structural features of L_0. Let L_0 be the Laplacian of the graph $\mathcal{G} = \mathcal{V}_0(\mathcal{K}), \mathcal{V}_1(\mathcal{K})$. Then, the rank-1 decomposition $L_0 = \sum_{\sigma\in\mathcal{V}_1(\mathcal{K})} L_0(\sigma)$ holds, where $L_0(\sigma)$ is the graph Laplacian corresponding to edge σ and has the form $L_0(\sigma) = w_1(\sigma)\mathbf{e}_\sigma\mathbf{e}_\sigma^\top$ with $\mathbf{e}_\sigma$ being incidence column from operator $W_0^{-1}B_1$ corresponding to the edge $\sigma \in \mathcal{V}_1(\mathcal{K})$. Then, the first Schur complement S_1 for L_0 is computed as follows: the first column $\mathbf{c}_1 = \frac{1}{\sqrt{\alpha_1}} L_0\delta_1$ in graph terms is provided by:

$$\mathbf{c}_1 = \frac{1}{\sqrt{\alpha_1}} \sum_\sigma w_1(\sigma)\mathbf{e}_\sigma\mathbf{e}_\sigma^\top\delta_1 = \frac{1}{\sqrt{\alpha_1}} \sum_{1\in\sigma} w_1(\sigma)\mathbf{e} \quad \text{(Eqn. 110)}$$

with $\alpha_1 = \delta_1^\top L_0\delta_1 = (L_0)_{11} = \deg[1]$ and the entire Schur complement is given by, [KS16, Tro19]:

$$S_1 = \underbrace{\sum_{\sigma|1\notin\sigma} w(\sigma)\mathbf{e}_\sigma\mathbf{e}_\sigma^\top}_{\text{star-like term}} + \underbrace{\frac{1}{2\deg[1]} \sum_{\substack{\sigma_1|1\in\sigma_1\\ \sigma_2|1\in\sigma_2}} w(\sigma_1)w(\sigma_2)\left[\mathbf{e}_{\sigma_1} - \mathbf{e}_{\sigma_2}\right]\left[\mathbf{e}_{\sigma_1} - \mathbf{e}_{\sigma_2}\right]^\top}_{\text{clique term}}. \quad \text{(Eqn. 111)}$$

Notice that the first star-like term of S_1 is the rank-1 decomposition of a graph Laplacian without edges adjacent to vertex 1. As for the second term, each vector $\mathbf{e}_{\sigma_1} - \mathbf{e}_{\sigma_2}$ corresponds to an edge. In fact, since $\sigma_1 \cap \sigma_2 = [1]$, then $\mathbf{e}_{\sigma_1} - \mathbf{e}_{\sigma_2}$ corresponds to a column of an incidence matrix describing the edge between two vertices that are not shared by the edges σ_1 and σ_2. As a result, the second term (clique-term) in Equation (111) describes a graph Laplacian for a completely connected graph on all the vertices adjacent to vertex $[1]$. Thus, overall, S_1 remains a graph Laplacian. Moreover, as a result of the first step of Gauss elimination, one substituted a sparse star-like structure of edges (everything adjacent to the vertex $[1]$) with a dense, completely connected subgraph (the clique term). This also supports the fact that sparse matrices, in general, have a dense Cholesky decomposition.

This observation is at the basis of a successful line of work on sparse Laplacian approximations, whose key results are summarized in the following:

Th IV.III.10 **(Approximate stochastic Cholesky decomposition of $L_0^{\uparrow}$, [Tro19, KS16])** For a classical graph Laplacian $L_0 = L_0^{\uparrow}$ of a graph $\mathcal{K} = (\mathcal{V}_0(\mathcal{K}), \mathcal{V}_1(\mathcal{K}))$ with $|\mathcal{V}_0(\mathcal{K})| = m_0$ vertices and $|\mathcal{V}_1(\mathcal{K})| = m_1$ edges, by Theorem IV.III.9, there exist sparse simplicial complexes $\{\mathcal{L}_i\}_i$ such that $L_0^{\uparrow}(\mathcal{K}) \underset{\varepsilon}{\approx} L_0^{\uparrow}(\mathcal{L}_i)$. In this set, one can construct a specific sparsifier $\mathcal{L} \in \{\mathcal{L}_i\}_i$ such that

(1) the Cholesky decomposition $L_0^{\uparrow}(\mathcal{L}) = CC^{\top}$ is inexpensive and computed alongside with the sparsification;

(2) the Cholesky decomposition is ε-close to the original complex, i.e. $CC^{\top} \underset{\varepsilon}{\approx} L_0^{\uparrow}(\mathcal{K})$, with probability at least $1 - {}^{2}\!/_{m_1}$;

(3) C is sparse with $\mathcal{O}(\varepsilon^{-2} m_1 \log^2 m_0)$ non-zero entries (in comparison with $\mathcal{O}(m_1)$ non-zero entries in the original $L_0^{\uparrow}(\mathcal{K})$);

(4) the computational complexity of constructing C is $\mathcal{O}(\varepsilon^{-2} m_1 \log^3 m_0)$.

The fundamental idea of Theorem IV.III.10 is that one can avoid the introduction of the clique-term at each step of Gauss elimination, Equation (111). Instead, one can subsample a sparser subgraph around the eliminated vertex spectrally close to the original full clique (in expectation), thus building a sparsifier akin Theorem IV.III.9. One should note that the rigorous proof of the spectral approximation in the case of Theorem IV.III.10 requires substantial work with matrix concentration inequalities or bag-of-dice martingale and presents a very intriguing research venue.

Unfortunately, the cornerstone property at the basis of the theorem above,

i.e., the fact that all the S_i are themselves up-Laplacians of 0-order, does not carry over in general to the higher-order setting $k > 0$. In the following subsection, we show why this is the case for the case $k = 1$ by analyzing the structure of the Schur complements arising in the Gaussian elimination of $L_1^{\uparrow}$.

III.IV The structure of the Schur complements S_i for $k = 1$

It is immediate to show that the following decomposition into rank-1 terms holds for any up-Laplacian:

Lem III.10 **(Rank-1 decomposition of $L_k^{\uparrow}$)** For a simplicial complex $\mathcal{K}$, the following decomposition into structured rank-1 terms holds for any up-Laplacian $L_k^{\uparrow}$:

$$L_k^{\uparrow} = \sum_{t \in \mathcal{V}_k(\mathcal{K})} L_k^{\uparrow}(t) = \sum_{t \in \mathcal{V}_k(\mathcal{K})} w(t) \mathbf{e}_t \mathbf{e}_t^{\top}$$

where, for each $t \in \mathcal{V}_k(\mathcal{K})$, $L_k^{\uparrow}(t) = w(t)\mathbf{e}_t\mathbf{e}_t^{\top}$ is the rank-1 matrix such that (a) $w(t) = [W_k^2]_{tt}$ is the weight of the simplex t, and (b) $\mathbf{e}_t$ is the vector obtained as the action of the boundary operator $W_{k-1}^{-1}B_k$ on t, i.e. the t-th column of the matrix $W_{k-1}^{-1}B_k$.

Proof Let δ_i be a versor on the i-th position. Then, $\mathbf{e}_t = W_{k-1}^{-1}B_k\delta_t$ and $\sum_t w(t)\mathbf{e}_t\mathbf{e}_t^{T} = W_{k-1}^{-1}B_k\left(\sum_t w(t)\delta_t\delta_t^{\top}\right)B_k^{\top}W_{k-1}^{-1}$; since $\delta_t\delta_t^{\top} = \operatorname{diag}\delta_t$ and $w(t) = w_2^k(t)$, the central matrix is exactly $\sum_t w(t)\delta_t\delta_t^{\top} = W_k^2$, which gives the original up-Laplacian. ■

Using Lemma III.10, we obtain the following characterization for the first Schur complement S_1 for $L_1^{\uparrow}$ similar to the case of the classical Laplacian:

Lem III.11 **(First Schur complement S_1 for $L_1^{\uparrow}$)** For the up-Laplacian $L_1^{\uparrow}$, the following derivation of the first Schur complement holds:

$$S_1 = \sum_{t|1\notin t} w(t)\mathbf{e}_t\mathbf{e}_t^{\top} + \frac{1}{2\Omega_{\{1\}|\varnothing}} \sum_{\substack{t_1|1\in t_1\\ t_2|1\in t_2}} w(t_1)w(t_2)\left[\mathbf{e}_{t_1}-\mathbf{e}_{t_2}\right]\left[\mathbf{e}_{t_1}-\mathbf{e}_{t_2}\right]^{\top}$$

where $\Omega_{\{1\}|\varnothing}$ is the total weight of all triangles adjacent to the first edge, $\Omega_{\{1\}|\varnothing} = \sum_{t|1\in t} w(t)$.

Proof Following Equation (108), one needs to compute the constant $\delta_1^{\top}S_0\delta_1$ and the rank-1 matrix $S_0\delta_1\delta_1^{\top}S_0$ where $S_0 = L_1^{\uparrow}$. Note that $\mathbf{e}_t^{\top}\delta_1 = \frac{1}{\sqrt{w_1(1)}}\mathbb{1}_{1\in t}$, the indicator of the triangle t having the edge 1, since 1 is necessarily the first (in the chosen order) edge in the triangle; hence,

$\sum_t w_2(t)\mathbf{e}_t^\top \delta_1 = \frac{1}{\sqrt{w_1(1)}} \sum_{t|1\in t} w_2(t)$:

$$\delta_1^\top S_0 \delta_1 = \delta_1^\top \left(\sum_t w_2(t)\mathbf{e}_t\mathbf{e}_t^T \right) \delta_1 = \sum_t w_2(t) \left(\delta_1^\top \mathbf{e}_t \right)^2 = \frac{1}{w_1(1)} \sum_{t|1\in t} w_2(t) = \frac{\Omega_{\{1\}|\varnothing}}{w_1(1)}$$

$$S_0 \delta_1 \delta_1^\top S_0 = \sum_{t_1,t_2} w_2(t_1)w_2(t_2)\mathbf{e}_{t_1}\mathbf{e}_{t_1}^\top \delta_1 \delta_1^\top \mathbf{e}_{t_2}\mathbf{e}_{t_2}^\top = \frac{1}{w_1(1)} \sum_{\substack{t_1|1\in t_1 \\ t_2|1\in t_2}} w_2(t_1)w_2(t_2)\mathbf{e}_{t_1}\mathbf{e}_{t_2}^\top .$$

By symmetry,

$$\sum_{\substack{t_1|1\in t_1 \\ t_2|1\in t_2}} w(t_1)w(t_2)\mathbf{e}_{t_1}\mathbf{e}_{t_2}^\top = \frac{1}{2} \sum_{\substack{t_1|1\in t_1 \\ t_2|1\in t_2}} w(t_1)w(t_2) \left(\mathbf{e}_{t_1}\mathbf{e}_{t_2}^\top + \mathbf{e}_{t_2}\mathbf{e}_{t_1}^\top \right),$$

one can note by a straightforward arithmetic transformation that

$$\mathbf{e}_{t_1}\mathbf{e}_{t_2}^\top + \mathbf{e}_{t_2}\mathbf{e}_{t_1}^\top = -\left[\mathbf{e}_{t_1} - \mathbf{e}_{t_2}\right]\left[\mathbf{e}_{t_1} - \mathbf{e}_{t_2}\right]^\top + \mathbf{e}_{t_1}\mathbf{e}_{t_1}^\top + \mathbf{e}_{t_2}\mathbf{e}_{t_2}^\top$$

Finally, $\sum_{\substack{t_1|1\in t_1 \\ t_2|1\in t_2}} w(t_1)w(t_2)\mathbf{e}_{t_1}\mathbf{e}_{t_1}^\top = \Omega_1 \sum_{t|1\in t} w(t)\mathbf{e}_t\mathbf{e}_t^\top$. ■

The structure of the first Schur complement S_1 is reminiscent of the classical graph Laplacian case as S_1 is formed of two terms:

$$H_1 = \sum_{t|1\notin t} w(t)\mathbf{e}_t\mathbf{e}_t^\top, \qquad K_{1|\varnothing} = \frac{1}{2\Omega_{\{1\}|\varnothing}} \sum_{\substack{t_1|1\in t_1 \\ t_2|1\in t_2}} w(t_1)w(t_2)\left[\mathbf{e}_{t_1} - \mathbf{e}_{t_2}\right]\left[\mathbf{e}_{t_1} - \mathbf{e}_{t_2}\right]^\top. \tag{Eqn. 112}$$

The first term H_1 corresponds to the portion of the original $L_1^\uparrow$ not adjacent to the edge being eliminated (edge 1), which is kept intact. Instead, the "star-like" term $\sum_{t|1\in t} w(t)\mathbf{e}_t\mathbf{e}_t^\top$, consisting of the sum of up-Laplacians adjacent to edge 1, is substituted here by the matrix $K_{\{1\}|\varnothing}$, which we refer to as the **cyclic term** (on the contrary to the clique term in the case of L_0) which reveals the fundamental difference between the classical and higher-order cases. Unfortunately, unlike the 0-Laplacian case, it is easy to realize that the cyclic term $K_{\{1\}|\varnothing}$ is generally not an up-Laplacian $L_1^\uparrow(\tilde{\mathcal{K}})$. We show this with a simple illustrative example below.

Example

Assume the simplicial complex formed by two triangles, $\mathcal{V}_0(\mathcal{K}) = \{1,2,3,4\}$, $\mathcal{V}_1(\mathcal{K}) = \{12,13,14,23,24\}$ and $\mathcal{V}_2(\mathcal{K}) = \{123,124\}$, adjacent by the first edge 12 with $W_2 = I$. Then, $\mathbf{e}_{t_1} = \begin{pmatrix}1 & -1 & 0 & 1 & 0\end{pmatrix}^\top$, $\mathbf{e}_{t_2} = \begin{pmatrix}1 & 0 & -1 & 0 & 1\end{pmatrix}^\top$ and $K_{\{1\}|\varnothing} = \begin{pmatrix}0 & -1 & 1 & 1 & -1\end{pmatrix} \cdot \begin{pmatrix}0 & -1 & 1 & 1 & -1\end{pmatrix}^\top$. Note that $K_{\{1\}|\varnothing}$ is denser

than $L_1^{\uparrow}$ and has lost the structural balance of $L_1^{\uparrow}$: $\operatorname{diag} K_{\{1\}|\varnothing} = K_{\{1\}|\varnothing}\mathbf{1} = \mathbf{0}$ where $\operatorname{diag} L_1^{\uparrow} = L_1^{\uparrow}\mathbf{1} = \left(\Omega_{\{i\}|\varnothing}\right)_{i=1}^{m_2}$.

Indeed, cyclic term $K_{\{1\}|\varnothing}$ has a denser structure similar to the clique term in Equation (111), but each vector $\mathbf{e}_{t_1} - \mathbf{e}_{t_2}$ is no longer structurally equivalent to columns of B_2 (at the very least, it has a bigger number of non-zero elements). This problem stems from a relatively trivial change of paradigm: in the case of the classical graph, every two vertices can potentially form an edge, but for a simplicial complex, not every pair of edges may form a triangle. Moreover, further computations of Schur complements $S_2, S_3, S_4 \ldots$ inevitably introduce a fastly growing number of similar cyclic terms. Thus, to build an efficient preconditioner for $L_1^{\uparrow}$, one needs to deal with the cyclic terms arising on each step S_i.

The obstacle of the cyclic terms $K_{\{1\}|\varnothing}$ in Lemma III.11 brings up a natural question: when does one avoid getting a cyclic term? Note that edge 1 has at least one adjacent triangle; otherwise, it corresponds to a zero row and column in $L_k^{\uparrow}$ and does not contribute to the up-Laplacia. Then $K_{\{1\}|\varnothing} = 0$ if and only if edge 1 has a unique adjacent triangle. Moreover, assuming this holds for edge 1, one may ask the same of edge 2 in the computation of S_2; note that for S_2, edge 2 should have a unique adjacent triangle in the simplicial complex **without consideration of edge 1 and its corresponding triangle**; and so on. This concept of iteratively obtaining edges with a unique adjacent triangle is connected to the topological concept of **collapsibility** of a simplicial complex [Whi39a] and, to be precise, a less demanding version of it which we call **weak collapsibility** and which we introduce specifically for this purpose in this work. While not working in the general case, the Schur decomposition approach works for the special family of weakly collapsible simplices; we describe the concepts of collapsibility and weak collapsibility next.

IV. Collapsibility of a simplicial complex

In this section, we borrow the terminology from [Whi39a] to introduce the concept of collapsibility of a simplicial complex. The simplex $\sigma \in \mathcal{K}$ is **free** if it is a face of exactly one simplex $\tau = \tau(\sigma) \in \mathcal{K}$ of higher order (maximal face). The **collapse** $\mathcal{K}\backslash\{\sigma\}$ of $\mathcal{K}$ at a free simplex σ is the operation of reducing $\mathcal{K}$ to $\mathcal{K}'$, where $\mathcal{K}' = \mathcal{K} - \sigma - \tau$; namely, this is the operation of removing a simplex τ having an accessible (not included in another simplex) face σ. A sequence of collapses done at the simplicies $\Sigma = \{\sigma_1, \sigma_2, \ldots\}$ is called a **collapsing sequence**; formally:

Def. 12 **(Collapsing sequence)** Let $\mathcal{K}$ be a simplicial complex. $\Sigma = \{\sigma_1, \sigma_2, \ldots\}$ is a **collapsing sequence** for $\mathcal{K}$ if σ_1 is free in $\mathcal{K} = \mathcal{K}^{(1)}$ and each σ_i, $i > 1$, is free at $\mathcal{K}^{(i)} = \mathcal{K}^{(i-1)} \backslash \{\sigma_i\}$. The resulting complex $\mathcal{L}$ obtained collapsing $\mathcal{K}$ at Σ is denoted by $\mathcal{L} = \mathcal{K} \backslash \Sigma$.

Note that, by definition, every collapsing sequence Σ has a corresponding sequence $\mathbb{T} = \{\tau(\sigma_1), \tau(\sigma_2), \ldots\}$ of maximal faces being collapsed at every step. The notion of **collapsible simplicial complex** is defined in [Whi39a] as follows

Def. 13 **(Collapsible simplicial complex)** The simplicial complex $\mathcal{K}$ is collapsible if there exists a collapsing sequence Σ such that $\mathcal{K}$ collapses to a single vertex, i.e. $\mathcal{K} \backslash \Sigma = \{v\}$ for some $v \in \mathcal{V}_0(\mathcal{K})$.

While least-square problems with collapsible simplicial complexes can be solved directly in an efficient way, [CFM+14a], collapsibility is a strong requirement for a simplicial complex. In fact, determining whether the complex is collapsible is in general **NP-complete**, [Tan16a], even though it can be almost linear for a specific set of families of $\mathcal{K}$, [CFM+14a]. Moreover, simplicial complexes are rarely collapsible, as we discuss in the following.

Next, we recall the concept of a d-Core, [Tan10]:

Def. 14 **(d-Core)** A d-Core is a subcomplex of $\mathcal{K}$ such that every simplex of dimension $d-1$ belongs to at least 2 d-simplices.

So, for example, a 2-Core of a 2-skeleton $\mathcal{K}$, is a subcomplex of the original simplex $\mathcal{K}$ such that every edge from $\mathcal{V}_1(\mathcal{K})$ belongs to at least 2 triangles from $\mathcal{V}_2(\mathcal{K})$. Finally, we say that $\mathcal{K}$ is d-collapsible if it can be collapsed only by collapses at simplices σ_i of order smaller than d, $\dim \sigma_i \leq d-1$. Then we have the following criteria:

Lem IV.12 **([LN21a])** $\mathcal{K}$ is d-collapsible if and only if it does not contain a d-core.

Proof The proof of the lemma above naturally follows from the definition of the core: if the d-collapsing sequence is stuck, then the simplex collapsed up to d-Core; conversely, if a d-Core exists in the complex, the collapsing sequence necessarily includes its $(d-1)$-faces which are not collapsible. ■

The d-Core is the generalization of the cycle for the case of 1-collapsibility of a classical graph, and finding a d-Core inside a complex $\mathcal{K}$ is neither trivial nor computationally cheap. Note that a d-Core is generally dense due to its definition and does not have a prescribed structure. We illustrate simple exemplary cores in the case of $d = 2$ in Figure IV.1, hinting at the combinatorial many possible configurations for a general d-Core, for $d \geq 2$.

In Figure IV.2, we show that an arbitrary simplicial complex $\mathcal{K}$ tends to

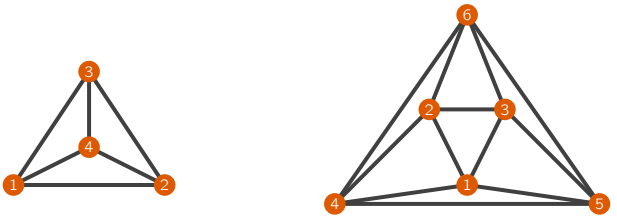


Figure IV.1: 2-Core, examples: all 3-cliques in graphs are included in corresponding $\mathcal{V}_2(\mathcal{K})$.

contain 2-Cores as long as $\mathcal{K}$ is denser than the trivially collapsible case. In the left panel of Figure IV.2, we consider the complex formed by the triangulation of m_0 random points on the unit square with a sparsity pattern ν (such that the complex has ν share of all possible edges, $\nu = m_1/\binom{m_0}{2}$), and we show that there is a threshold ν_Δ such that the simplex generated this way is collapsible when $\nu \leq \nu_\Delta$, but quickly forms a 2-Core when $\nu > \nu_\Delta$. A similar effect is observed in the case of sampled sensor networks, where the complex is formed looking at the Euclidean distance of the sensors $\exists \sigma \in \mathcal{V}_1(\mathcal{K}) : \sigma = [v_1, v_2] \iff \|v_1 - v_2\|_2 < \varepsilon$, for a chosen percolation parameter $\varepsilon > 0$. When the percolation parameter grows, the complex immediately forms a core, c.f. Figure IV.2 right panel.

While collapsibility is a strong requirement, in the next section, we show that a weaker condition is enough to design a preconditioner efficiently for any "sparse enough" simplicial complex.

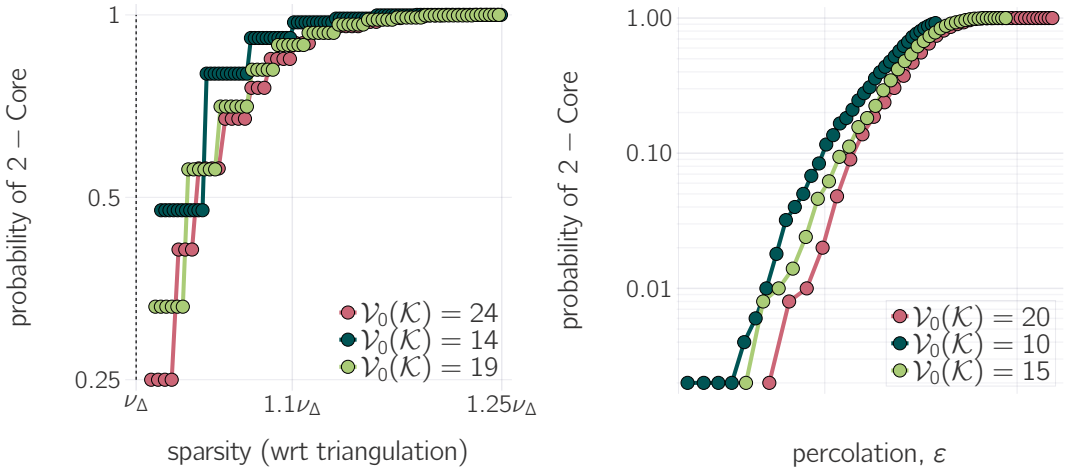


Figure IV.2: The probability of the 2-Core in richer-than-triangulation simplicial complexes: triangulation of random points modified to have $\left\lceil \nu \frac{m_0 \cdot (m_0 - 1)}{2} \right\rceil$ edges on the left; random sensor networks with ε-percolation on the right. ν_Δ defines the initial sparsity of the triangulated network; $\varepsilon_{\min} = \mathbb{E} \min_{x,y \in [0,1]^2} \|x - y\|_2$ is the minimal possible percolation parameter.

IV.I Weak collapsibility

Let the complex $\mathcal{K}$ be restricted up to its 2-skeleton, $\mathcal{K} = \mathcal{V}_0(\mathcal{K}) \cup \mathcal{V}_1(\mathcal{K}) \cup \mathcal{V}_2(\mathcal{K})$, and assume $\mathcal{K}$ is collapsible. Then a collapsing sequence Σ necessarily involves collapses at simplices σ_i of different orders: at edges (eliminating **edges** and **triangles**) and at vertices (eliminating **vertices** and **edges**). One can show that for a given collapsing sequence Σ there is a reordering $\tilde{\Sigma}$ such that $\dim \tilde{\sigma}_i$ in the reordered sequence are non-increasing, [CFM+14a, Lemma 2.5]. Namely, if such a complex is collapsible, then there is a collapsible sequence $\Sigma = \{\Sigma_1, \Sigma_0\}$ where Σ_1 contains all the collapses at edges first and Σ_0 is composed of collapses at vertices. Note that the partial collapse $\mathcal{K} \backslash \Sigma_1 = \mathcal{L}$ eliminates all the triangles in the complex, $\mathcal{V}_2(\mathcal{L}) = \varnothing$; otherwise, the whole sequence Σ is not collapsing $\mathcal{K}$ to a single vertex. Since $\mathcal{V}_2(\mathcal{L}) = \varnothing$, the associated up-Laplacian $L_1^{\uparrow}(\mathcal{L}) = 0$.

Def. 15 **(Weakly collapsible complex)** A simplicial complex $\mathcal{K}$ restricted to its 2-skeleton is called **weakly collapsible**, if there exists a collapsing sequence Σ_1 such that the simplicial complex $\mathcal{L} = \mathcal{K} \backslash \Sigma_1$ has no simplices of order 2, i.e. $\mathcal{V}_2(\mathcal{L}) = \varnothing$ and $L_1^{\uparrow}(\mathcal{L}) = 0$.

Note that while a collapsible complex is necessarily weakly collapsible, the opposite does not hold. Consider the following example in Figure IV.3: the initial complex is weakly collapsible either by a collapse at $[3, 4]$ or at $[2, 4]$. After this, the only available collapse is at the vertex $[4]$, leaving the uncollapsible 3-vertex structure.

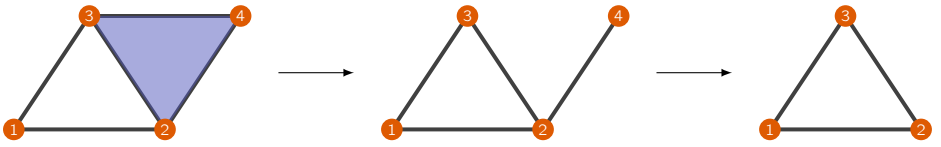


Figure IV.3: Example of weakly collapsible but not collapsible simplicial complex

For a given simplicial complex $\mathcal{K}$, one can use the following greedy algorithm to find a collapsing sequence Σ and test for collapsibility: at each iteration perform any of possible collapses; in the absence of free edges, the complex should be considered not collapsible. This greedy procedure is illustrated in Algorithm 4.

Specifically, let Δ_σ be a set of triangles of $\mathcal{K}$ containing the edge σ, $\Delta_\sigma = \{t \mid t \in \mathcal{V}_2(\mathcal{K}) \text{ and } \sigma \in t\}$. Then the edge σ is free and has only one adjacent triangle $\tau = \tau(\sigma)$ iff $|\Delta_\sigma| = 1$; let $F = \{\sigma \mid |\Delta_e| = 1\}$ is a set of all free edges. Note that $|\Delta_e| \leq m_0 - 2 = \mathcal{O}(m_0)$.

Algorithm 4 recursively picks up a free edge σ from the set F, performs a collapse $\mathcal{K} \leftarrow \mathcal{K} \backslash \{\sigma\}$ and updates F for the collapsed subcomplex. Then, the greedy approach may fail to find the collapsing sequence only if it gets

Algorithm 4 GREEDY_COLLAPSE($\mathcal{K}$): greedy algorithm for the weak collapsibility

Require: initial set of free edges F, adjacency sets $\{\Delta_{\sigma_i}\}_{i=1}^{m_1}$

1: $\Sigma = [\,],\ \mathbb{T} = [\,]$ ▷ initialize the collapsing sequence
2: **while** $F \neq \varnothing$ **and** $\mathcal{V}_2(\mathcal{K}) \neq \varnothing$ **do**
3: $\sigma \leftarrow \mathrm{pop}(F),\ \tau \leftarrow \tau(\sigma)$ ▷ pick a free edge σ
4: $\mathcal{K} \leftarrow \mathcal{K}\backslash\{\sigma\},\ \Sigma \leftarrow [\Sigma\ \sigma],\ \mathbb{T} \leftarrow [\mathbb{T}\ \tau]$ ▷ τ is a triangle being collapsed; $\tau = [\sigma, \sigma_1, \sigma_2]$
5: $\Delta_{\sigma_1} \leftarrow \Delta_{\sigma_1}\backslash\tau,\ \Delta_{\sigma_2} \leftarrow \Delta_{\sigma_2}\backslash\tau$ ▷ remove τ from adjacency lists
6: $F \leftarrow F \cup \{\sigma_i \,|\, i = 1, 2 \text{ and } |\Delta_{\sigma_i}| = 1\}$ ▷ update F if any of σ_1 or σ_2 has become free
7: **end while**
8: **return** $\mathcal{K}$, Σ, $\mathbb{T}$

stuck on the collapsible complex, so the order of collapses matters. We demonstrate the validity of the greedy Algorithm 4 in the following theorem:

Th IV.IV.11 Algorithm 4 finds a weakly collapsible sequence of 2-skeleton simplicies in polynomial time. Additionally, it finds a collapsing sequence if the simplicial complex is collapsible.

Proof Clearly, Algorithm 4 runs polynomially with respect to the number of simplexes in $\mathcal{K}$, so its consistency automatically yields polynomiality.

The failure of the greedy algorithm would indicate the existence of a weakly collapsible complex $\mathcal{K}$ such that the greedy algorithm gets stuck at a 2-Core, which is avoidable for another possible order of collapses. Among all such complexes, let $\mathcal{K}$ be any minimal one with respect to the number of triangles m_2. Then there exiss a free edge $\sigma \in \mathcal{V}_1(\mathcal{K})$ such that $\mathcal{K}\backslash\{\sigma\}$ is **collapsible** and another free edge $\sigma' \in \mathcal{V}_2(\mathcal{K})$ such that $\mathcal{K}\backslash\{\sigma'\}$ is **not collapsible**.

Note that if $\mathcal{K}$ is minimal then any pair of free edges σ_1 and σ_2 belong to the same triangle: $\tau(\sigma_1) = \tau(\sigma_2)$.

Indeed, for any $\tau(\sigma_1) \neq \tau(\sigma_2)$, $\mathcal{K}\backslash\{\sigma_1, \sigma_2\} = \mathcal{K}\backslash\{\sigma_2, \sigma_1\}$. Let $\tau(\sigma_1) \neq \tau(\sigma_2)$ for at least one pair of σ_1 and σ_2; in our assumption, either (1) both $\mathcal{K}\backslash\{\sigma_1\}$ and $\mathcal{K}\backslash\{\sigma_2\}$, (2) only $\mathcal{K}\backslash\{\sigma_1\}$ or (3) none are collapsible.

In the first case, either $\mathcal{K}\backslash\{\sigma_1\}$ or $\mathcal{K}\backslash\{\sigma_2\}$ is a smaller example of the complex satisfying the assumption above (since a bad edge σ' can not be either σ_1 or σ_2 and belongs to collapsed complexes), hence, violating the minimality. If only $\mathcal{K}\backslash\{\sigma_1\}$ is collapsible, then $\mathcal{K}\backslash\{\sigma_2, \sigma_1\}$ is not collapsible (since $\mathcal{K}\backslash\{\sigma_2\}$ is not collapsible and σ_1 is free in $\mathcal{K}$ and in $\mathcal{K}\backslash\{\sigma_2\}$ assuming $\tau(\sigma_1) \neq \tau(\sigma_2)$); hence, $\mathcal{K}\backslash\{\sigma_1, \sigma_2\}$ is not collapsible (since $\mathcal{K}\backslash\{\sigma_1, \sigma_2\} = \mathcal{K}\backslash\{\sigma_2, \sigma_1\}$ as we stated above), so $\mathcal{K}\backslash\{\sigma_1\}$ is a smaller example of a complex satisfying the assumption.

Finally, if both $\mathcal{K}\backslash\{\sigma_1\}$ and $\mathcal{K}\backslash\{\sigma_2\}$ are collapsible, then for known σ' such that $\mathcal{K}\backslash\{\sigma'\}$ is not collapsible, $\tau(\sigma') \neq \tau(\sigma_1)$ or $\tau(\sigma') \neq \tau(\sigma_2)$, which can be treated as the previous point.
As a result, for σ ($\mathcal{K}\backslash\{\sigma\}$ is collapsible) and for σ' ($\mathcal{K}\backslash\{\sigma'\}$ is not collapsible) it holds that $\tau(\sigma) = \tau(\sigma') \Rightarrow \sigma \cap \sigma' = \{v\}$, so after collapses $\mathcal{K}\{\sigma\}$ and $\mathcal{K}\backslash\{\sigma'\}$ we arrive at two identical simplicial complexes besides the hanging edge (σ' or σ) irrelevant for the weak collapsibility. A simplicial complex can not be simultaneously collapsible and not collapsible, so the question of weak collapsibility can always be resolved by the greedy algorithm, which has polynomial complexity. ■

Computational cost of the greedy algorithm The complexity of Algorithm 4 rests upon the precomputed $\sigma \mapsto \Delta_\sigma$ structure that de-facto coincides with the boundary operator B_2 (assuming B_2 is stored as a sparse matrix, the adjacency structure describes its non-zero entries). Similarly, the initial F set can be computed alongside the construction of B_2 matrix. Another concession is needed for the complexity of the removal of elements from Δ_{σ_i} and F, which may vary from $\mathcal{O}(1)$ on average up to guaranteed $\log(|\Delta_{\sigma_i}|)$. As a result, given a pre-existing B_2 operator, Algorithm 4 runs linearly, $\mathcal{O}(m_1)$, or almost linearly depending on the realisation, $\mathcal{O}(m_1 \log m_1)$. We demonstrate this asymptotic behavior later in the experiments section.

V. Preconditioning through the subsampling of the 2-Core

Above, we have shown that the efficient computation of the Cholesky multiplier for $L_1^{\uparrow}$ is complicated by arising cyclic terms. At the same time, the absence of the cyclic terms in the Schur complements is a characteristic of weakly collapsible complexes. Using this observation, in this section, we develop a Cholesky-like preconditioning scheme based on an efficient Cholesky multiplier for weakly collapsible complexes.

First, we demonstrate that a weakly collapsible simplicial complex $\mathcal{K}$ immediately yields an exact Cholesky decomposition for its up-Laplacian:

Lem V.13 Assume the 2-skeleton simplicial complex $\mathcal{K}$ is weakly collapsible through the collapsing sequence Σ. Let $\mathbb{T}$ be the corresponding sequence of maximal faces and let P_Σ and $P_\mathbb{T}$ be the permutation matrices of the two sequences, i.e. such that $[P_\Sigma]_{ij} = 1 \iff j = \sigma_i$, and similarly for $P_\mathbb{T}$. Then $C = P_\Sigma B_2 P_\mathbb{T}$ is an exact Cholesky mulitplier for $P_\Sigma L_1^{\uparrow}(\mathcal{K}) P_\Sigma^\top$, i.e. $P_\Sigma L_1^{\uparrow}(\mathcal{K}) P_\Sigma^\top = CC^\top$.

Proof Note that the sequences Σ and $\mathbb{T}$ (and the multiplication by the corresponding permutation matrices) impose only the reordering of $\mathcal{V}_1(\mathcal{K})$ and $\mathcal{V}_2(\mathcal{K})$, respectively; after such reordering the i-th edge collapses the i-

triangle. Hence, the first $(i-1)$ entries of the i-th column of the matrix B_2 ($[B_2]_{:,i} = \sqrt{w(t_i)}\mathbf{e}_{t_i}$) are 0, otherwise one of the previous edges is not free. As a result, C is lower-triangular and by a direct computation one has $CC^\top = P_\Sigma L_1^\uparrow(\mathcal{K}) P_\Sigma^\top$. ■

An arbitrary simplicial complex $\mathcal{K}$ is generally not weakly collapsible (see Figure IV.1). Specifically, weak collapsibility is a property of sparse simplicial complexes with the sparsity being measured by the number of triangles m_2 (in the weakly collapsible case $m_2 < m_1$ since each collapse at the edge eliminates exactly one triangle); hence, the removal of triangles from $\mathcal{V}_2(\mathcal{K})$ can potentially destroy the 2-Core structure inside $\mathcal{K}$ and make the complex weakly collapsible.

With this observation, in order to find a cheap and effective preconditioner for $L_1^\uparrow(\mathcal{K})$, one may search for a weakly collapsible subcomplex $\mathcal{L} \subseteq \mathcal{K}$ and use its exact Cholesky multiplier C as a preconditioner for $\mathcal{K}$.

Specifically, such subcomplex $\mathcal{L}$ should satisfy the following properties:

(1) it has the same set of nodes and edges, $\mathcal{V}_0(\mathcal{L}) = \mathcal{V}_0(\mathcal{K})$ and $\mathcal{V}_1(\mathcal{L}) = \mathcal{V}_1(\mathcal{K})$;

(2) triangles in $\mathcal{L}$ are subsampled, $\mathcal{V}_2(\mathcal{L}) \subseteq \mathcal{V}_2(\mathcal{K})$;

(3) $\mathcal{L}$ is weakly collapsible through some collapsing sequence Σ and corresponding sequence of maximal faces $\mathbb{T}$;

(4) $\mathcal{L}$ has the same 1-homology as $\mathcal{K}$, that is $\ker L_1(\mathcal{K}) = \ker L_1(\mathcal{L})$;

(5) the Cholesky multiplier $C = P_\Sigma B_2(\mathcal{L}) P_{\mathbb{T}}$ improves the condition number of $L_1^\uparrow(\mathcal{K})$, namely $\kappa_+(C^\dagger P_\Sigma L_1^\uparrow(\mathcal{K}) P_\Sigma^\top C^{\dagger\top}) \ll \kappa_+(L_1^\uparrow(\mathcal{K}))$.

Let us comment on the conditions above. Conditions (1) and (2) are automatically met when a subcomplex $\mathcal{L}$ is obtained from $\mathcal{K}$ through the elimination of triangles. Condition (3) is a structural requirement on $\mathcal{L}$ and can be guaranteed by the design of the subcomplex using the proposed Algorithm 5. Condition (4) guarantees that the preconditioning strategy is bijective, as we show in the next Lemma V.14. Finally, condition (5) asks for a better condition number and is checked numerically in Section VI. However, whilst one can not guarantee improvement in preconditioning quality, we can provide an explicit formula for the condition number $\kappa_+(C^\dagger P_\Sigma L_1^\uparrow(\mathcal{K}) P_\Sigma^\top C^{\dagger\top})$ assuming an arbitrary subcomplex forms the preconditioner C. We provide such a formula in the next Subsection V.I and discuss how to use it to construct a preconditioner via heavy collapsible subcomplex.

Lem V.14 **(On the conservation of the 1-homology red and condition (4))** For any subcomplex $\mathcal{L}$ above, the following statements about 1-homology hold:

(i) subcomplex $\mathcal{L}$ satisfying conditions (1) and (2) can only extend the

kernel, i.e. we have $\ker L_1(\mathcal{K}) \subseteq \ker L_1(\mathcal{L})$;

(ii) condition (4) guarantees a bijective preconditioning scheme in the sense that the solution to the original least-square problem and the preconditioned one coincide.

Proof For (i) it is sufficient to note that the elimination of the triangle $t \in \mathcal{V}_2(\mathcal{K})$ lifts the restriction $\mathbf{e}_t^\top \mathbf{x} = 0$ for $\mathbf{x} \in \ker L_1(\mathcal{K})$; hence, if $\mathbf{x} \in \ker L_1(\mathcal{K})$, then $\mathbf{x} \in \ker L_1(\mathcal{L})$. For (ii), note that bijection between the systems $L_k^\uparrow \mathbf{x} = \mathbf{f}$ and $\left(C^\dagger L_k^\uparrow C^{\dagger\top}\right) C^\top \mathbf{x} = C^\dagger \mathbf{f}$ is guaranteed by $\ker C^\top = \ker L_k^\uparrow = \ker B_{k+1}^\top$ (assuming $\mathbf{x} \perp \ker L_k^\uparrow$). Then, by the spectral inheritance principle, [GST23, Thm. 2.7], $\ker L_k^\uparrow(\mathcal{X}) = \ker L_k(\mathcal{X}) \oplus B_k^\top \cdot \operatorname{im} L_{k-1}^\uparrow$. The second part, $B_k^\top \cdot \operatorname{im} L_{k-1}^\uparrow$, consists of the action of $B_k^\top$ on non-zero related eigenvectors of $L_{k-1}^\uparrow$ and is not dependent on $\mathcal{V}_{k+1}(\mathcal{K})$ (triangles, in case $k = 1$), hence remains preserved in the subcomplex $\mathcal{L}$. Since by condition (4) $\ker L_1(\mathcal{K}) = \ker L_1(\mathcal{L})$, the same statement holds for up-Laplacians, $\ker L_1^\uparrow(\mathcal{K}) = \ker L_1^\uparrow(\mathcal{L})$. Since C is an exact Cholesky multiplier for $L_1^\uparrow(\mathcal{L})$, $\ker L_1^\uparrow(\mathcal{L}) = \ker C^\top$ and $\ker L_1^\uparrow(\mathcal{K}) = \ker B_2^\top$ yielding $\ker C^\top = \ker B_2^\top$ and bijectivity. ■

V.I Preconditioning quality by the subcomplex

Note that subcomplex $\mathcal{L}$ is fully defined by the subset $\mathbb{T}$ of subsampled triangles, $\mathbb{T} \subset \mathcal{V}_2(\mathcal{K})$, so $\mathcal{L} = \mathcal{V}_0(\mathcal{K}) \cup \mathcal{V}_1(\mathcal{K}) \cup \mathbb{T})$; let us assume, additionally, that subcomplex $\mathcal{L}$ satisfies condition (4) above. We introduce the following matrix notation corresponding to the subsampling:

Def. 16 **(Subsampling matrix)** Let $\mathbb{T}$ be a subset of triangles, $\mathbb{T} \subset \mathcal{V}_2(\mathcal{K})$, then Π is a subsampling matrix if

♢ Π is diagonal, $\Pi \in \mathbb{R}^{m_2 \times m_2}$;

♢ $(\Pi)_{ii} = 1 \iff i \in \mathbb{T}$; otherwise, $(\Pi)_{ii} = 0$.

Assuming subset $\mathbb{T}$ of subsampled triangles (or, equivalently, the subsampling matrix Π) is given, one needs only the triangle weight matrix $\widehat{W}$ in order to obtain $L_1^\uparrow(\mathcal{L})$ and the corresponding Cholesky multiplier C.

Generally speaking, (squared) weights $\widehat{W}_2^2$ of sampled triangles $\mathbb{T}$ may differ from the original weights W_2^2. Let $\widehat{W}_2^2 = W_2^2 + \Delta W_2$, where ΔW_2 is still diagonal, but entries are not necessarily positive. Then, one can formulate the question of the optimal weight redistribution as the optimization problem:

$$\min_{\Delta W_2} \left\| L_1^\uparrow(\mathcal{L}) - L_1^\uparrow(\mathcal{K}) \right\| = \min_{\Delta W_2} \left\| B_2 \left[\Pi(W_2^2 + \Delta W_2)\Pi - W_2^2 \right] B_2^\top \right\|$$

Here and only here, since we manipulate weights, we use the unweighted B_2 matrix so one can have explicit access to the weight matrix W_2.

Lem V.15 **(Optimal weight choice for the subcomplex)** Let $\mathcal{L}$ be subcomplex of $\mathcal{K}$ with fixed corresponding subsampling matrix Π. Then, the optimal weight perturbation for the subsampled triangles is

$$\Delta W_2 \equiv 0,$$

so the best choice of weights of subcomplex is $\widehat{W_2} = W_2\Pi$.

Proof Let $\Delta W_2 = \Delta W_2(t)$ where t is a time parametrization; we can compute the gradient $\nabla_{\Delta W_2}\sigma_1\left(L_1^{\uparrow}(\mathcal{L}) - L_1^{\uparrow}(\mathcal{K})\right)$ through the derivative $\frac{d}{dt}\sigma_1\left(L_1^{\uparrow}(\mathcal{L}) - L_1^{\uparrow}(\mathcal{K})\right)$:

$$\dot{\sigma}_1 = \mathbf{x}^\top B_2\Pi\Delta\dot{W}_2\Pi B_2^\top\mathbf{x} = \left\langle B_2\Pi\Delta\dot{W}_2\Pi B_2^\top,\, \mathbf{x}\mathbf{x}^\top\right\rangle = \operatorname{Tr}\left(B_2\Pi\Delta\dot{W}_2\Pi B_2^\top\mathbf{x}\mathbf{x}^\top\right) =$$
$$= \left\langle \Pi B_2^\top\mathbf{x}\mathbf{x}^\top B_2\Pi,\, \Delta\dot{W}_2\right\rangle = \left\langle \nabla_{\Delta W_2}\sigma_1, \Delta\dot{W}_2\right\rangle$$

By projecting onto the diagonal structure of the weight perturbation,

$$\nabla_{\operatorname{diag}\Delta W_2}\sigma_1 = \operatorname{diag}\left(\Pi B_2^\top\mathbf{x}\mathbf{x}^\top B_2\Pi\right).$$

Note that $\operatorname{diag}\left(\Pi B_2^\top\mathbf{x}\mathbf{x}^\top B_2\Pi\right)_{ii} = |\Pi B_2^\top\mathbf{x}|_i^2$; then the stationary point is characterized by $\Pi B_2^\top\mathbf{x} = 0 \iff \mathbf{x} \in \ker L_1^{\uparrow}(\mathcal{L}) = \ker L_1^{\uparrow}(\mathcal{K})$ (see Lemma V.14). The latter is impossible since $\mathbf{x}$ is the eigenvector corresponding to the largest eigenvalue; hence, since $\Pi(W_2^2 + \Delta W_2)\Pi \neq W_2^2$, the optimal perturbation is $\Delta W_2 \equiv 0$. ■

Rem V.21 Note that the results of Lemma V.15 do not contradict the sampling mechanism for the sparsified simplicial complex from Theorem IV.III.9. Indeed, in the construction (however computationally unobtainable in practice) of the sparsifier, simplices from $\mathcal{V}_{k+1}(\mathcal{K})$ are sampled with an altered weight. At the same time, due to the fact that sampling is performed with replacement, simplices in the final sparsified complex have, on average, the same weights (see the proof of Theorem IV.III.9 in [OPW22]), consistent with Lemma V.15.

We have established the optimal choice of the weights provided by the subsampling matrix Π. As a result, assuming B_2 is weighted, optimal $L_1^{\uparrow}(\mathcal{L}) = B_2\Pi B_2^\top$. Now we proceed to characterize the quality of the preconditioning in terms of the matrix Π in Theorem IV.V.12, starting with a necessary technical relation between $\operatorname{im} B_2^\top$ and $\ker\Pi$.

Rem V.22 Knowing the optimal weight for sampled triangles from Lemma V.15, one needs to preserve the kernel of subsampled Laplacian

$$\ker\left(B_2\Pi B_2^\top\right) = \ker\left(B_2 B_2^\top\right)$$

to form a correct preconditioner C. Since we have $\Pi = \Pi^2$, $\ker L_1^\uparrow = \ker B_2^\top$ and $\ker\left(B_2\Pi B_2^\top\right) = \ker\left(\Pi B_2^\top\right)$. Additionally, $\ker B_2^\top \subseteq \ker\left(\Pi B_2^\top\right)$, so $\ker\left(B_2\Pi B_2^\top\right) \neq \ker\left(B_2 B_2^\top\right) \iff$ there exists $\mathbf{y} \in \operatorname{im} B_2^\top$ such that $B_2^\top \mathbf{y} \neq 0$ and $B_2^\top \mathbf{y} \in \ker \Pi$. Then, in order to preserve the kernel, one needs $\operatorname{im} B_2^\top \cap \ker \Pi = \{0\}$.

Th IV.V.12 **(Conditioning by the Subcomplex)** Let $\mathcal{L}$ be a weakly collapsible subcomplex of $\mathcal{K}$ defined by the subsampling matrix Π and let C be a Cholesky multiplier of $L_1^\uparrow(\mathcal{L})$ defined as in Lemma V.13. Then the conditioning of the symmetrically preconditioned $L_1^\uparrow$ is given by:

$$\kappa_+\left(C^\dagger P_\Sigma L_1^\uparrow P_\Sigma^\top C^{\dagger\top}\right) = \left(\kappa_+\left(\left(S_1 V_1^\top \Pi\right)^\dagger S_1\right)\right)^2 = \left(\kappa_+(\Pi V_1)\right)^2,$$

where V_1 forms the orthonormal basis on $\operatorname{im} B_2^\top$.

Proof By Lemma V.15, $W_2(\mathcal{L}) = \Pi W_2$; then let us consider the lower-triangular preconditioner $C = P_\Sigma B_2 \Pi P_\mathbb{T}$ for $P_\Sigma L_1^\uparrow P_\Sigma^\top$; then the preconditioned matrix is given by:

$$C^\dagger\left(P_\Sigma L_1^\uparrow P_\Sigma^\top\right) C^{\top\dagger} = \left(P_\Sigma B_2 \Pi P_\mathbb{T}\right)^\dagger \left(P_\Sigma L_1^\uparrow P_\Sigma^\top\right) \left(P_\Sigma B_2 \Pi P_\mathbb{T}\right)^{\top\dagger} =$$
$$= P_\mathbb{T}^\top \left(B_2\Pi\right)^\dagger L_1^\uparrow \left(B_2\Pi\right)^{\top\dagger} P_\mathbb{T}$$

Note that $P_\mathbb{T}$ is unitary, so $\kappa_+(P_\mathbb{T} X P_\mathbb{T}^\top) = \kappa_+(X)$, hence the principle matrix is $\left(B_2\Pi\right)^\dagger L_1^\uparrow \left(B_2\Pi\right)^{\top\dagger} = \left(B_2\Pi\right)^\dagger B_2 B_2^\top \left(B_2\Pi\right)^{\top\dagger}$. Since we have that $\kappa_+(X^\top X) = \kappa_+^2(X)$, then in fact one needs to consider $\kappa_+\left(\left(B_2\Pi\right)^\dagger \left(B_2\right)\right)$. Let us consider the SVD-decomposition for $B_2 = USV^\top$; more precisely,

$$B_2 = USV^\top = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^\top \\ V_2^\top \end{pmatrix} = U_1 S_1 V_1^\top$$

where S_1 is a diagonal invertible matrix. Note that U and U_1 have orthonormal columns and S_1 is diagonal and invertible, so

$$\left(B_2\Pi\right)^\dagger B_2 = \left(SV^\top\Pi\right)^\dagger SV^\top = \left(S_1 V_1^\top \Pi\right)^\dagger S_1 V_1^\top$$

By the definition of the condition number κ_+, one needs to compute $\sigma^+_{\min}$ and $\sigma^+_{\max}$ where:

$$\sigma^+_{\min \setminus \max} = \min_{\mathbf{x} \perp \ker\left(\left(S_1 V_1^\top \Pi\right)^\dagger S_1 V_1^\top\right)} \setminus \max \frac{\left\|\left(S_1 V_1^\top \Pi\right)^\dagger S_1 V_1^\top \mathbf{x}\right\|}{\|\mathbf{x}\|}$$

Note that $\operatorname{im} B_2^\top = \operatorname{im} V_1 = \operatorname{im} V_1 S_1$, so by Remark V.22, $\ker \Pi \cap \operatorname{im} V_1 S_1 = \{0\}$, hence $\ker \Pi V_1 S_1 = \ker V_1 S_1$. Since $\ker V_1 S_1 \cap \operatorname{im} S_1 V_1^\top = \{0\}$, one gets $\ker \Pi V_1 S_1 \cap \operatorname{im} S_1 V_1^\top = \{0\}$. By the properties of the pseudo-inverse we have that $\ker \Pi V_1 S_1 = \ker \left(S_1 V_1^\top \Pi\right)^\top = \ker \left(S_1 V_1^\top \Pi\right)^\dagger$; as a result, $\ker\left(\left(S_1 V_1^\top \Pi\right)^\dagger S_1 V_1^\top\right) = \ker S_1 V_1^\top$. Since S_1 is invertible, $\ker\left(\left(S_1 V_1^\top \Pi\right)^\dagger S_1 V_1^\top\right) = \ker V_1^\top$.
For $\mathbf{x} \in \ker V_1^\top \Rightarrow \mathbf{x} \in \operatorname{im} V_1$, so $\mathbf{x} = V_1 \mathbf{y}$. Since $V_1^\top V_1 = I$, $\|\mathbf{x}\| = \|V_1 \mathbf{y}\|$ and:

$$\sigma^+_{\min \setminus \max} = \min_{\mathbf{y}} \setminus \max_{\mathbf{y}} \frac{\left\|\left(S_1 V_1^\top \Pi\right)^\dagger S_1 \mathbf{y}\right\|}{\|\mathbf{y}\|} \underset{\mathbf{z} = S_1 \mathbf{y}}{=} \min_{\mathbf{z}} \setminus \max_{\mathbf{z}} \frac{\left\|\left(S_1 V_1^\top \Pi\right)^\dagger \mathbf{z}\right\|}{\left\|S_1^{-1} \mathbf{z}\right\|}$$

Note that $\mathbf{v} = \left(S_1 V_1^\top \Pi\right)^\dagger \mathbf{z} \iff \begin{cases} S_1 V_1^\top \Pi \mathbf{v} = \mathbf{z} \\ \mathbf{v} \perp \ker S_1 V_1^\top \Pi \end{cases}$ and $\ker S_1 V_1^\top \Pi = \ker V_1^\top \Pi$, so:

$$\sigma^+_{\min \setminus \max} = \min_{\mathbf{v} \perp \ker V_1^\top \Pi} \setminus \max \frac{\|\mathbf{v}\|}{\|V_1^\top \Pi \mathbf{v}\|}$$

Hence $\kappa_+\left(C^\dagger P_\Sigma L_1^\uparrow P_\Sigma^\top C^{\dagger\top}\right) = \kappa_+^2(V_1^\top \Pi) = \kappa_+^2(\Pi V_1)$.

■

V.II Algorithm: Preconditioner via heavy collapsible subcomplex

Following Theorem IV.V.12, note that C is a perfect preconditioner for $P_\Sigma L_1^\uparrow P_\Sigma^\top$ if $\Pi = I$, since $\kappa_+(V_1) = 1$ and we would compute the exact Cholesky decomposition. However, this is computationally prohibitive. Thus, it is natural to try to find a sparser Π so that ΠV_1 is as close to V_1 as possible. Multiplication by Π cancels rows in V_1 corresponding to the eliminated triangles; at the same time, rows in V_1 are scaled by the weights of the triangles since $\operatorname{span} V_1 = W_2 \operatorname{im} B_2^\top$, so we expect to Π eliminating triangles with lowest weights to be a good choice.

Based on this observation and theorem IV.V.12, we provide an algorithm for preconditioning $L_1^\uparrow(\mathcal{K})$, which aims to eliminate triangles of the lowest

weight, thus constructing what we call a heavy weakly collapsible subcomplex $\mathcal{L}$ with largest possible total weight of triangles. The exact Cholesky multiplier C of $L_1^{\uparrow}(\mathcal{L})$ is cheap to compute and is used as a preconditioner for $L_1^{\uparrow}(\mathcal{K})$.

The proposed Algorithm 5 works as follows: start with an empty subcomplex $\mathcal{L}$; then, at each step, try to extend $\mathcal{L}$ with the heaviest unconsidered triangle t: $\mathcal{L} \to \mathcal{L} \cup \{t\}$ – here the extension includes the addition of the triangle t with all its vertices and edges to the complex $\mathcal{L}$. If the extension $\mathcal{L} \cup \{t\}$ is weakly collapsible, it is accepted as the new $\mathcal{L}$, otherwise t is rejected; in either case the triangle t is removed from the list of unconsidered triangles, i.e. t is not considered for a second time.

Algorithm 5 `HEAVY_SUBCOMPLEX`$(\mathcal{K}, W_2)$: construction a heavy collapsible subcomplex

Require: the original complex $\mathcal{K}$, weight matrix W_2

$\mathcal{L} \leftarrow \varnothing$, $\mathbb{T} \leftarrow \varnothing$ ▷ initial empty subcomplex
while there is unprocessed triangle in $\mathcal{V}_2(\mathcal{K})$ **do**
  $t \leftarrow$ `nextHeaviestTriangle`$(\mathcal{K}, W_2)$ ▷ e.g. iterate through $\mathcal{V}_2(\mathcal{K})$ sorted by weight
  if $\mathcal{L} \cup \{t\}$ is weakly collapsible **then** ▷ run `GREEDY_COLLAPSE`$(\mathcal{L} \cup \{t\})$ (Algorithm 4)
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{t\}$, $\mathbb{T} \leftarrow [\mathbb{T}\ t]$ ▷ extend $\mathcal{L}$ by t
  end if
end while
return $\mathcal{L}$, $\mathbb{T}$, Σ ▷ here Σ is the collapsing sequence of $\mathcal{L}$

Rem V.23

We show next that a subcomplex $\mathcal{L}$ sampled with Algorithm 5 satisfies properties (1)–(4) above: indeed, $\mathcal{V}_0(\mathcal{K}) = \mathcal{V}_0(\mathcal{L})$, $\mathcal{V}_1(\mathcal{K}) = \mathcal{V}_1(\mathcal{L})$ and $\mathcal{L}$ is weakly collapsible by construction. It is less trivial to show that the subsampling $\mathcal{L}$ does not increase the dimensionality of 1-homology.
Assuming the opposite, the subcomplex $\mathcal{L}$ cannot have any additional 1-dimensional holes in the form of smallest-by-inclusion cycles of more than 3 edges: since this cycle is not present in $\mathcal{K}$, it is "covered" by at least one triangle t which necessarily has a free edge, so $\mathcal{L}$ can be extended by t and remain weakly collapsible. Alternatively, if the only additional hole corresponds to the triangle t not present in $\mathcal{L}$, then, reminiscent of the proof for Theorem IV.IV.11, let us consider the minimal by inclusion simplicial complex $\mathcal{K}$ for which it happens. Then, the only free edges in $\mathcal{L}$ are the edges of t; otherwise, $\mathcal{K}$ is not minimal. At the same time, in such setups t is not registered as a hole since it is an outer boundary of the complex $\mathcal{L}$, e.g. consider the exclusion of exactly one triangle in the tetrahedron case, Figure IV.1 [7], which proves that $\mathcal{L}$ cannot extend the 1-homology of $\mathcal{K}$.

[7] algebraically, this fact is extremely dubious: due to the lack of free edges, there is a "path"

The complexity of Algorithm 5 is $\mathcal{O}(m_1 m_2)$ at worst, which could be considered comparatively slow: the algorithm passes through every triangle in $\mathcal{V}_2(\mathcal{K})$ and performs collapsibility check via Algorithm 4 on $\mathcal{L}$ which never has more than m_1 triangles since it is weakly collapsible. Note that Algorithm 5 and Theorem IV.V.12 do not depend on $\mathcal{K}$ being a 2-Core; moreover, the collapsible part of a generic $\mathcal{K}$ is necessarily included in the subcomplex $\mathcal{L}$ produced by Algorithm 5. Hence a prior pass of $\texttt{GREEDY\_COLLAPSE}(\mathcal{K})$ reduces the complex to a smaller 2-Core $\mathcal{K}'$ with faster $\texttt{HEAVY\_SUBCOMPLEX}(\mathcal{K}', W_2)$ since $\mathcal{V}_1(\mathcal{K}') \subset \mathcal{V}_1(\mathcal{K})$ and $\mathcal{V}_2(\mathcal{K}') \subset \mathcal{V}_2(\mathcal{K})$.

We summarise the whole procedure for computing the preconditioner next:

- ♢ reduce a generic simplicial complex $\mathcal{K}$ to a 2-Core $\mathcal{K}'$ through the collapsing sequence Σ_1 and the corresponding sequence of triangles $\mathbb{T}_1$ through the greedy Algorithm 4;
- ♢ form a heavy weakly connected subcomplex $\mathcal{L}$ from $\mathcal{K}'$ with the collapsing sequence Σ_2 and the corresponding sequence of triangles $\mathbb{T}_2$ using Algorithm 5;
- ♢ form the preconditioner C by permuting and subsampling B_2 using the subset of triangles $\mathbb{T} = \mathbb{T}_1 \cup \mathbb{T}_2$ (that determines the subsampling matrix Π) and the associated collapsing sequence $\Sigma = \Sigma_1 \cup \Sigma_2$, via $C = P_\Sigma B_2 \Pi P_\mathbb{T}$.

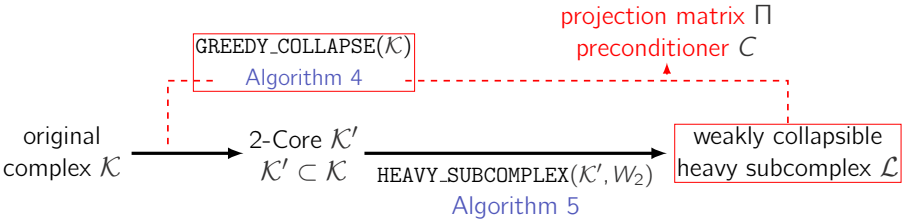


Figure V.1: The scheme of the simplicial complex transformation: from the original $\mathcal{K}$ to the heavy weakly collapsible subcomplex $\mathcal{L}$.

We refer to the preconditioner built in this way (see also Figure V.1) as a **heavy collapsible subcomplex** preconditioner (HeCS-preconditioner).

Rem V.24 **(Adjacency heuristic)** Algorithm 5 naturally starts with the accumulation phase where the number of triangles in $\mathcal{L}$ is too small for it to be non-collapsible. Indeed, one may avoid calling Algorithm 4 during that phase if the expansion of $\mathcal{L}$ by the next triangle t does not create triangles in $\mathcal{V}_2(\mathcal{L})$ adjacent to 3 or more already chosen triangles; with the preexisting B_2 structure such check costs $\mathcal{O}(1)$.

VI. Benchmarking: triangulation

We present here a number of numerical experiments to validate the performance of the proposed preconditioning strategy. All the experiments are run using `julia` on Apple M1 CPU and can be reproduced with the code available at `https://github.com/COMPiLELab/HeCS`.

VI.I Conjugate Gradient Least-Square method

The preconditioning performance for the least-square problem $\min_{\mathbf{x}\perp\ker L_1^{\uparrow}} \|L_1^{\uparrow}\mathbf{x}-\mathbf{f}\|$ is measured on the **conjugate gradient least-square method (CGLS)**, [HS+52, BES98], Algorithm 3. The method requires the ability to compute `matvec` operation for the matrix $L_1^{\uparrow}$ and its preconditioned alternatives; CGLS converges as $(\sqrt{\kappa_+(A)}-1)/(\sqrt{\kappa_+(A)}+1)$ and we run it until the infinity norm of the residual $\mathbf{r}_i = L_1^{\uparrow}\mathbf{x}_i - \mathbf{f}$ falls below a given threshold, i.e. $\|\mathbf{r}_i\|_\infty \leq \epsilon$.

VI.II Shifted incomplete Cholesky preconditioner

$L_1^{\uparrow}$ is a singular matrix. Assuming U is an orthogonal basis of $\ker L_1^{\uparrow}$, one can move to $L_1^{\uparrow} \to L_1^{\uparrow} + \alpha UU^\top$, which can be preconditioned by non-singular methods. Specifically, we use $C_\alpha = \texttt{ichol}(L_1^{\uparrow} + \alpha UU^\top)$ to compare with the `HeCS` preconditioner, Figure V.1.

Computing such a shift requires the ability to efficiently compute $\ker L_1^{\uparrow}$, which, in principle, has a complexity comparable to the original system. On the other hand, in our setting, given the spectral inheritance principal from [GST23, Thm. 2.7], an orthogonal basis U can be formed directly using the vectors $B_1^\top\mathbf{x}$, $\mathbf{x} \in (\mathbf{1})^\perp$, when $\mathcal{K}$ has trivial 0- and 1-homology (i.e. it is formed by one connected component, $\ker L_0 = \text{span}\{\mathbf{1}\}$, and has no 1-dimensional holes, $\ker L_1 = 0$).

Note that the `HeCS` preconditioner instead works without requiring any triviality of the topology of the complex.

VI.III Problem setting: Enriched triangulation as a simplicial complex

To illustrate the behaviour of the preconditioned system $C^\dagger P_\Sigma L_1^{\uparrow} P_\Sigma^\top C^{\dagger\top}$, we consider an sparse simplicial complex $\mathcal{K}$, i.e. we assume $m_2 = \mathcal{O}(m_1 \ln m_1)$. Note that the developed routine can be applied in denser cases, although one can expect a certain loss of efficiency. To generate problem settings within this range, $\mathcal{K}$ is synthesized as an enriched triangulation of N points on the unit square with a prescribed edge sparsity pattern ν as follows:

(1) $\mathcal{V}_0(\mathcal{K})$ is formed by the corners of the unit square and $(N-4)$ points sampled uniformly at random from $U\left([0,1]^2\right)$;

(2) the Delaunay triangulation of $\mathcal{V}_0(\mathcal{K})$ is computed; all edges and 3-cliques of the produced graph are included in $\mathcal{V}_1(\mathcal{K})$ and $\mathcal{V}_2(\mathcal{K})$ respectively;

(3) $d \geq 0$ edges (excluding the outer boundary) are chosen at random and eliminated from $\mathcal{V}_1(\mathcal{K})$; triangles adjacent to the chosen edges are eliminated from $\mathcal{V}_2(\mathcal{K})$. As result, produced complex $\mathcal{K}$ has a non-trivial 1-homology;

(4) the sparsity pattern is defined as $\nu = m_2/q(m_1)$ where parameter $q(m_1)$ is the highest density of triangles for the sparse case; additional edges on $\mathcal{V}_0(\mathcal{K})$ are added to $\mathcal{V}_1(\mathcal{K})$ alongside with new appearing 3-cliques to reach the target $m_2/q(m_1)$ value. The initial sparsity pattern of the triangulation is denoted by ν_Δ.

VI.IV Heavy subcomplex and triangle weight profile

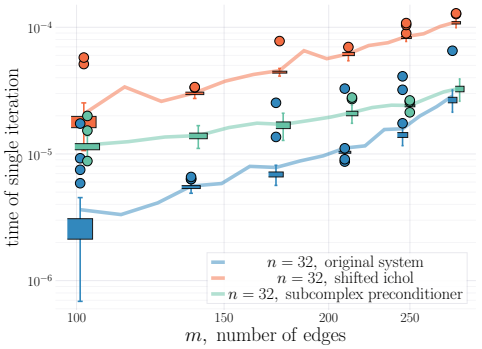
Algorithm 5, aims to build a heavy weakly collapsible subcomplex $\mathcal{L}$ such that the total weight of triangles in $\mathcal{L}$ is close to the total weight of triangles in the original complex $\mathcal{K}$. At the same time, the number of triangles in $\mathcal{L}$ is limited, $m_2(\mathcal{L}) < m_1(\mathcal{L}) = m_1(\mathcal{K})$, due to the weak collapsibility, while the number of triangles in $\mathcal{K}$ can go up to $q(m_1(\mathcal{K}))$. Hence, the quality of the preconditioner is determined by the triangle weight distribution $w(\cdot)$ on $\mathcal{V}_2(\mathcal{K})$: namely, if $w_2(t)$ are distributed uniformly and independently, the quality of the preconditioning falls rapidly after $\nu > \nu_\Delta$ and, at the same time, the original matrix $L_1^{\uparrow}$ remains well-conditioned in such configurations. Instead, in this section, we observe that unbalanced weight distributions lead to ill-conditioned $L_1^{\uparrow}$ and consider the following two situations:

♢ the weights of triangles are random variables that are **independent** of each other and distributed as heavy-tailed Cauchy distributions or bi-modal Gaussian distribution $\mathcal{N}(1, \sigma_1) + \mathcal{N}(1/3, \sigma_2)$. This way, we generate a sufficiently large number of heavy triangles and a cluster of reducible triangles with small $w_2(t)$;

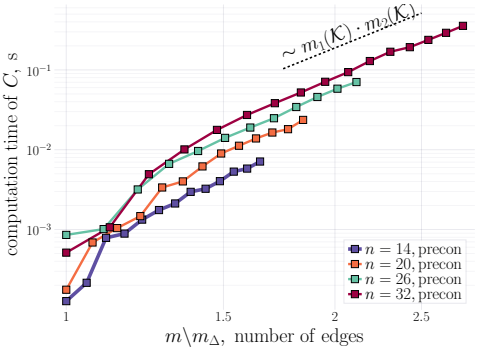
♢ the weights of triangles are **topologically dependent**, that is $w_2(t)$ is a function of the weights of the neighboring triangles. A way to implement this dependence is to set $w_2(t) = f(w_1(e_1), w_1(e_2), w_1(e_3))$ for the triangle $t = (e_1, e_2, e_3)$. Two well-known common choices of f are the min-rule $w_2(t) = \min(w_1(e_1), w_1(e_2), w_1(e_3))$, [GST23, LCK+19], and the product rule $w_2(t) = w_1(e_1)w_1(e_2)w_1(e_3)$, [CM21, CMK21]. In this way, the edge weight profile $w_1(\cdot)$ on $\mathcal{V}_1(\mathcal{K})$ is transformed to an unbalanced distribution $w_2(t)$.

VI.V Timings

One needs to separately discuss the time cost of the computation of the preconditioning operator (as a tuplet C and the permutation P_Σ) and the `matvec` computation for the preconditioned operator $C^\dagger P_\Sigma L_1^{\uparrow} P_\Sigma^\top C^{\dagger\top}$.



(a) Single iteration timing: the average time of `matvec` computation for the original system (blue), shifted `ichol` (orange) and `HeCS` preconditioner (green).



(b) Computation time for the heavy subcomplex preconditioner in case of enriched triangulations on m_0 vertices

Figure VI.1: Timings of `HeCS`-perconditioner

Note that `matvec` of $L_1^{\uparrow}$ has the complexity of the number of non-zero elements, $\mathcal{O}(m_2)$; the `HeCS` preconditioner C has $\leq 3m_1$ non-zero elments and lower triangular structure, so `matvecs` of either $C^{\dagger}$ and $C^{\dagger\top}$, as well as the permutation matrix P_{Σ}, have complexity $\mathcal{O}(m_1)$. Hence, the complexity of each preconditioned CGLS iteration is $\mathcal{O}(m_2 + m_1)$, as opposed to the original $\mathcal{O}(m_2)$; asymptotically one expects $m_2 \gg m_1$, so the preconditioning scheme is efficient. On the other hand, the shifted `ichol` preconditioner C_α loses the sparsity due to the shift; as a result, the application of $C_\alpha^{\dagger}$ costs $\mathcal{O}(m_0 m_1 + m_2)$ since $\operatorname{rank} U = \mathcal{O}(m_0)$ in all considered scenarios.

In Figure VI.1a we compare the performance of the two preconditioners for the enriched triangulation on $m_0 = 32$ vertices and a varying number of edges m_1: the cost of one CGLS iteration for `HeCS` preconditioner is higher than the original system but asymptotically approaches the `matvec` cost of the unpreconditioned system, whilst the `ichol`-preconditioner C_α is an order larger.

Additionally, we demonstrate the time complexity for `HeCS` preconditioner computation for enriched triangulation on $m_0 = 14, 20, 26, 32$ vertices and varying total edge number m_1, Figure VI.1b (here m_Δ denotes the number of edges in the initial triangulation); as $\mathcal{K}$ becomes denser simplicial complex, `HEAVY_COMPLEX`$(\mathcal{K})$ follows the expected complexity $\mathcal{O}(m_1 \cdot m_2)$. In comparison with the complexity of Cholesky decomposition which is $\mathcal{O}(m_1^3)$, by design $m_2 = \mathcal{O}(m_1 \ln m_1)$ and the overall cost of `HeCS` computation never breaches $\mathcal{O}(m_1^2 \ln(m_1))$.

VI.VI Performance of the preconditioner

We demonstrate the quality of `HeCS` preconditioner for enriched triangulations on m_0 vertices with $d = 2$ initially eliminated edges and for varying total number of edges m_1 such that the initial sparsity $m_2/q(m_1)$ is increased until the induced number of triangles m_2 reaches the density of the Spielman sparsifier $q(m_1) = 9C^2 m_1 \ln(4m_1)$, Theorem IV.III.9. For each pair of parameters $\left(m_0, m_2/q(m_1)\right)$, $k = 25$ simplicial complexes are generated; triangle weight profile $w(t)$ is given by the following two scenarios:

(1) indendent triangle weights with bi-modal imbalance, where $w(t) \sim \mathcal{N}(1, 1/3)$ for the original triangulation $t \in \mathcal{K}_\Delta$ and $w(t) \sim \mathcal{N}(1/2, 1/6)$ otherwise, Figure VI.2;

(2) dependent triangle weight through the min-rule: $w(t) = \min\{w_1(e_1), w_2(e_2), w_3(e_3)\}$ for $t = (e_1, e_2, e_3)$ and edge weights are folded normal variables, $w_1(e_i) \sim |\mathcal{N}|[0, 1]$, Figure VI.3.

For each weight profile we measure the conditionality ($\kappa_+(C^\dagger P_\Sigma L_1^\uparrow P_\Sigma^\top C^{\dagger\top})$ vs $\kappa_+(L_1^\uparrow)$), Figures VI.2 and VI.3, left, and the corresponding number of CGLS iteration, Figures VI.2 and VI.3, right. In the case of the min-rule, we provide a high-performance test for matrices up to 10^5 in size.

In the case of the independent triangle weights, Figure VI.2, `HeCS` preconditioning shows gains in κ_+ for the first, sparser part of the simplicial complexes; conversely, for the min-rule profile induced by the folded normal edges' weights, Figure VI.3, developed method outperforms the original system for all tested sparsity patterns $m_2/q(m_1)$. Noticeably, `HeCS` preconditioning performs better in terms of the actual CGLS iterations, Figures V.1 and VI.2, right, than in terms of κ_+, and, hence, significantly speeds up the iterative solver for $L_1^\uparrow$.

Finally, we demonstrate the comparative performance with the shifted incomplete Cholesky preconditioner, C_α, Figure VI.4; here we are forced to guarantee trivial 0- and 1-homologies, so no edges are eliminated in the triangulation, $d = 0$, and we check the kernel of L_1 for triviality after the generation of $\mathcal{K}$. Similarly to the previous results, preconditioning with the shifted `ichol` C_α is more efficient than `HeCS` preconditioning for the "denser half" of the considered simplicial complexes, which means that our developed method still performs better in case of the sparser $\mathcal{K}$. Moreover, the applicability of the shifted `ichol` is limited to the cases of trivial homologies, which is not the case for `HeCS` preconditioning.

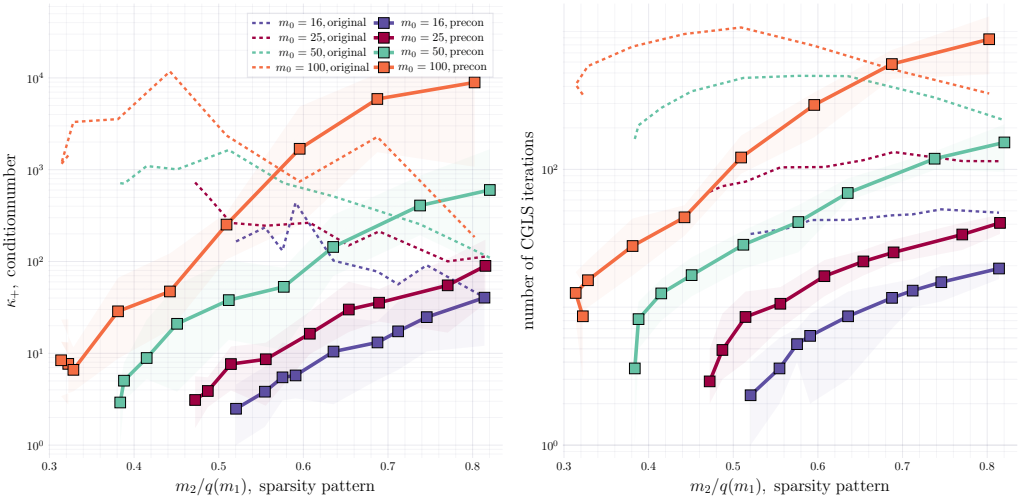


Figure VI.2: Preconditioning quality for enriched triangulations with a varying number of vertices $m_0 = 16, 25, 50, 100$ and sparsity patterns $m_2/q(m_1)$ and independent bi-modal weight profile: condition numbers κ_+ on the left and the number of CGLS iterations on the right. Average results among 25 generations are shown in solid (HeCS) and in the dash (original system); colored areas around the solid line show the dispersion among the generated complexes.

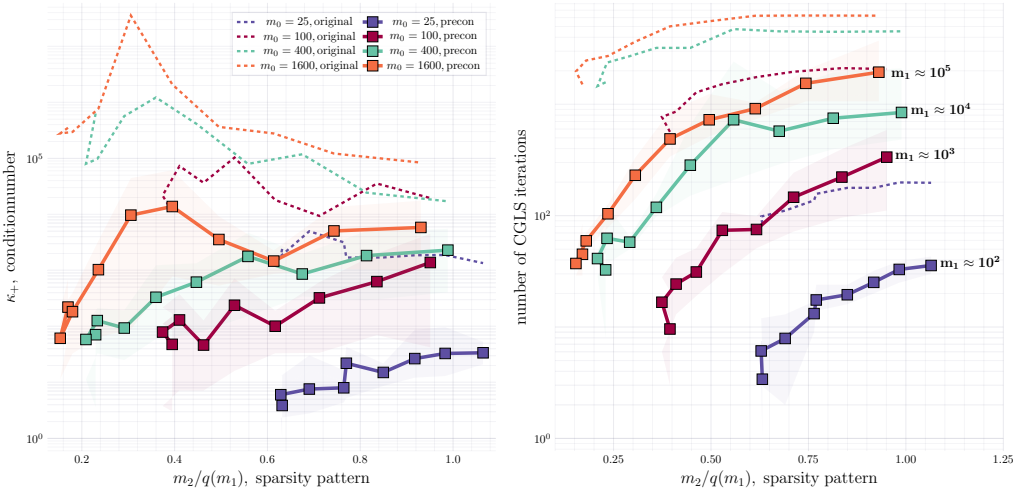


Figure VI.3: Preconditioning quality for enriched triangulations with a varying number of vertices $m_0 = 25, 100, 400, 1600$ and sparsity patterns $m_2/q(m_1)$ and dependent min-rule weight profile with folded normal edge weights: condition numbers κ_+ on the left and the number of CGLS iterations on the right. Average results among 25 generations are shown in solid (HeCS) and in the dash (original system); colored areas around the solid line show the dispersion among the generated complexes.

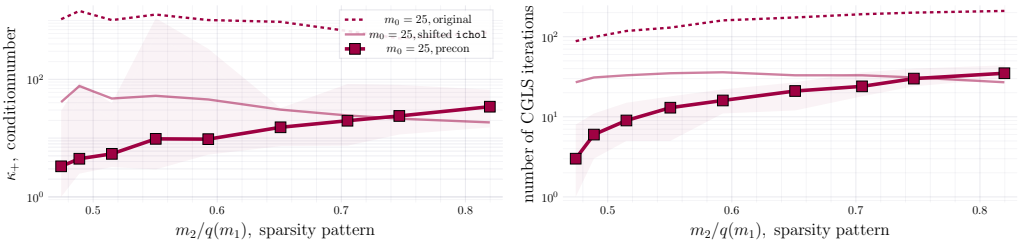


Figure VI.4: Comparison of the preconditioning quality between `HeCS`(solid), shifted `ichol` (semi-transparent) and original system (dashed) for the enriched triangulation on $m_0 = 25$ vertices and varying sparsity patterns ν and dependent min-rule weight profile with uniform edge weights: condition numbers κ_+ on the left and the number of CGLS iterations on the right. Average results among 25 generations are shown in solid (`HeCS` and `ichol`) and in the dash (original system); colored areas around the solid line show the dispersion among the generated complexes.

V Conclusion and future prospects

Among the wide variety of higher-order models for relational data, simplicial complexes strike a balance between an intrinsic description of the higher-order structure of the system and tractable matrix machinery allowing exploration of higher-order topological features via homology groups $\mathcal{H}_k$ and corresponding higher-order Laplacian operators L_k; such operators are direct discrete analogs of the continuous higher-order Helmholzian operators on manifolds and are shown to converge to them in the thermodynamic limit, [CM21, CMK21]. The elements of the kernel of the higher-order Laplacian operators correspond to k-dimensional holes in the complex (connected components, one-dimensional holes, voids, etc.); the corresponding eigenvectors and eigenvalues provide information about higher-order topological features [ES19, GS23a] and can be used to describe the underlying dynamics, e.g. in case of synchronization or higher-order diffusion, [GDPG+21, TB20], and simplicial random walks, [SBH+20], or identify functional higher-order structures in the data, [LCK+19].

I. Overview of main contributions

In this work we discussed the concepts of the topological stability of the weighted homology group $\overline{\mathcal{H}}_k$ and the corresponding k-th order Laplacian operator L_k through the intrinsic numerical procedure, and, vice versa, numerical stabilization of linear system associated with L_k (like $L_k\mathbf{x} = \mathbf{f}$) via topological properties of the underlying simplicial complex (heavy weakly collapsible subscomplex).

Specifically,

- ♢ we have proposed a generalized **weighting scheme** for the simplicial complex $\mathcal{K}$ coherent with Hodge's theory which allows the proper introduction of weighted homology groups $\overline{\mathcal{H}}_k$ and weighted Laplacian operators $\overline{L}_k$; additionally, for non-degenerate weights, we have demonstrated the conservation of dimensionality, Proposition 1, between all weighted and combinatorial cases for each of curl, gradient and harmonic subspaces as a conjecture of Hodge decomposition in Theorem II.III.2;

- ♢ we have shown **principle spectral inheritance** , Theorem III.III.5, describing the relationship between the spectra of up- and down-Laplacians $\sigma(L_k^{\uparrow})$ and $\sigma(L_k^{\downarrow})$ of consecutive orders as a foundation for topological

stability and developed solver for the linear system;

- ♢ we have addressed finding the smallest perturbation of weights of edges W_1 sufficient for creating a new k-dimensional hole in $\mathcal{K}$; based on the spectral inheritance principle, we have proposed a numerical method of minimizing target first non-zero eigenvalue λ_+ of $L_k^{\uparrow}$ (instead of L_k) thus adding additional dimension to $\ker \overline{L}_k$ and $\overline{\mathcal{H}}_k$ and avoiding **homological pollution**; the minimal perturbation is then obtained through the **alternating norm-constrained/unconstrained gradient flow** for the associated spectral matrix nearness problem;

- ♢ in the case of numerical stability, we have developed an efficient solver for linear system $L_k\mathbf{x} = \mathbf{f}$ in the least square sense; based on Hodge decomposition, Theorem II.III.2, and principle spectral inheritance, Theorem III.III.5, we have demonstrated the **reduction** of such system to the **system with up-Laplacian**, Theorem IV.I.7;

- ♢ we have shown that the attempt of building an exact **Cholesky preconditioner** for the up-Laplacian system $L_k^{\uparrow}\mathbf{x} = \mathbf{f}$ leads to the introduction of the concept of **weakly collapsible complex** for which Cholesky decomposition is immediate and cheap, Lemma V.13. Additionally, we have demonstrated that weak collapsibility is polynomially solvable via the greedy algorithm in Theorem IV.IV.11;

- ♢ we have built an efficient preconditioner for the up-Laplacian $L_k^{\uparrow}$ by finding a **heavy collapsible subcomplex** motivated by the demonstrated relation for preconditioning quality of any subcomplex in Theorem IV.V.12;

- ♢ Algorithm 1 for the topological instability and Algorithm 5 for preconditioning are successfully tested on the synthetic triangulation-based datasets; additionally, we provide results for several transportation networks in Table 2.

Contents of the thesis are primarily based on papers "Quantifying the Structural Stability of Simplicial Homology" (published in Journal of Scientific Computing in August 2023, [GST23]) and "Cholesky-like Preconditioner for Hodge Laplacians via Heavy Collapsible Subcomplex" (under review in SIMAX, arXiv:2401.15492).

II. Future projects

Results achieved in the current work suggest the following potential directions for further research:

1. in terms of future applications, the developed method of determining structural instabilities in the simplicial complexes may be further ex-

plored in various applications (e.g. one can provide a more detailed in-depth analysis of instability-related anomalies in transportation networks similar to Figure VI.5 and Table 2); specifically, as homology is frequently discussed in neuroscience, [LCK+19], we suggest further exploration of topological stability for brain connectomics (e.g. from ADNI dataset, `https://adni.loni.usc.edu/`, or Human Connectome Projects, `https://www.humanconnectome.org/`) for different stages of degenerative conditions (e.g. Alzheimer's disease). Any reasonable attempt of a meaningful analysis here would require a robust pipeline between fMRI/ECG images to the brain connectomics for varying levels of the correlation cutoffs (akin to persistent homology, [OPT+17]) with the following study of the instability distribution per each stage of the degenerative condition and their localizations on the statistically significant levels;

2. on the purely combinatorial side of the task, it would be a natural extension to posit a question of the minimal perturbation sufficient to **decrease** the homology group $\mathcal{H}_k$ (for instance, by eliminating the common edge between two adjacent holes). Note that since such a task would require a reduction of the dimension in the kernel of L_k, the generalization of the developed gradient flow routine is either not trivial or does not exist; instead, one may rely on a combinatorial approach which requires an efficient algorithm for detecting linearly independent holes in the complex (which is worth investigating on its own). It is worth noting that such an approach would be especially useful for determining and eliminating erroneous connections in sensor networks (similar to the buoy flows, see, for example, [SBH+20]);

3. in the case of `HeCS`-preconditioner, one should carefully examine the differences between weak collapsibility and d-collapsibility in terms of polynomiality in Theorem IV.IV.11 in order to generalize `HeCS`-preconditioner for $k > 1$. Moreover, developed Algorithm 5 does not aim to compute the heaviest collapsible subcomplex, instead opting for a heavy and easily obtainable one; this procedure could clearly be improved in quality and speed besides answering the question of whether the actually heaviest collapsible subcomplex is obtainable at all in polynomial time;

4. in the case of classical graph models, Laplacian operators were shown to have an efficient Algebraic Multigrid (AMG) preconditioner, [LB12], which one may attempt to generalize for the case of an arbitrary $L_k^{\uparrow}$; additionally, one may still attempt to find a stochastic sampled preconditioner C akin to the stochastic Cholesky preconditioner for the

classical Laplacian L_0, [KS16];

5. finally, concerning topological and numerical stabilities, their effects should be examined in the system where one injects a higher-order structure attempting to leverage it. For instance, in the case of simplicial complex convolutional graph neural networks (SCCGNNs), [EDS20], the output of each layer is defined as

$$\mathbf{x}_{k+1} = \sigma\left(\sum_{i=0}^{L} w_i L_k^i \mathbf{x}_k\right) \qquad \text{(Eqn. 113)}$$

As a result, the question of the effect of instabilities (and possible stabilization mechanisms) of the underlying simplicial complex on the trainability and overall network performance is worth carefully examining.

V References

[AEGL19] Eleonora Andreotti, Dominik Edelmann, Nicola Guglielmi, and Christian Lubich. Constrained graph partitioning via matrix differential equations. SIAM Journal on Matrix Analysis and Applications, 40(1):1–22, 2019.

[AL17] Giorgio Ausiello and Luigi Laura. Directed hypergraphs: Introduction and fundamental algorithms—a survey. Theoretical Computer Science, 658:293–306, 2017.

[AU18] Kristen M Altenburger and Johan Ugander. Monophily in social networks introduces similarity among friends-of-friends. Nature human behaviour, 2(4):284–290, 2018.

[BCI+20] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: structure and dynamics. Physics Reports, 874:1–92, 2020.

[BCKZ13] James Brannick, Yao Chen, Johannes Kraus, and Ludmil Zikatanov. An algebraic multigrid method based on matching in graphs. In Domain Decomposition Methods in Science and Engineering XX, pages 143–150. Springer, 2013.

[Ben19] Austin R Benson. Three hypergraph eigenvector centralities. SIAM Journal on Mathematics of Data Science, 1(2):293–312, 2019.

[BES98] Åke Björck, Tommy Elfving, and Zdenek Strakos. Stability of conjugate gradient and lanczos methods for linear least squares problems. SIAM Journal on Matrix Analysis and Applications, 19(3):720–736, 1998.

[BFF+11] Matthias Bolten, Stephanie Friedhoff, Andreas Frommer, Matthias Heming, and Karsten Kahl. Algebraic multigrid methods for laplacians of graphs. Linear Algebra and its Applications, 434(11):2225–2243, 2011.

[BGB22] Federica Baccini, Filippo Geraci, and Ginestra Bianconi. Weighted simplicial complexes and their representation power of higher-order network data and topology. Physical Review E, 106(3):034319, 2022.

[BGHS23] Christian Bick, Elizabeth Gross, Heather A Harrington, and Michael T Schaub. What are higher-order networks? SIAM Review, 65(3):686–731, 2023.

[BGL16] Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. Science, 353(6295):163–166, 2016.

[BKMZ11] Z. Burda, A. Krzywicki, O. C. Martin, and M. Zagorski. Motifs emerge from function in model gene regulatory networks. Proceedings of the National Academy of Sciences, 108(42):17263–17268, 2011.

[BMNW22] Mitchell Black, William Maxwell, Amir Nayyeri, and Eli Winkelman. Computational topology in a collapsing universe: Laplacians, homology, cohomology. In Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 226–251. SIAM, 2022.

[CFM+14a] Michael B Cohen, Brittany Terese Fasy, Gary L Miller, Amir Nayyeri, Richard Peng, and Noel Walkington. Solving 1-laplacians in nearly linear time: Collapsing and expanding a topological ball. In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 204–216. SIAM, 2014.

[CFM+14b] Michael B. Cohen, Brittany Terese Fasy, Gary L. Miller, Amir Nayyeri, Richard Peng, and Noel Walkington. Solving 1-laplacians in nearly linear time: Collapsing and expanding a topological ball. In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 204–216. SIAM, 2014.

[Che15] Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In Problems in analysis, pages 195–200. Princeton University Press, 2015.

[CM21] Yu-Chia Chen and Marina Meila. The decomposition of the higher-order homology embedding constructed from the k-laplacian. Advances in Neural Information Processing Systems, 34:15695–15709, 2021.

[CMK21] Yu-Chia Chen, Marina Meilă, and Ioannis G Kevrekidis. Helmholtzian eigenmap: Topological feature discovery &

edge flow learning from point cloud data. arXiv preprint arXiv:2103.07626, 2021.

[CZHZ19] Chen Chen, Dabao Zhang, Tony R Hazbun, and Min Zhang. Inferring gene regulatory networks from a population of yeast segregants. Scientific reports, 9(1):1197, 2019.

[Dem97] James W. Demmel. 7. Iterative Methods for Eigenvalue Problems, pages 361–387. SIAM, 1997.

[DJP+94] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. SIAM Journal on Computing, 23(4):864–894, 1994.

[DT08] Inderjit S Dhillon and Joel A Tropp. Matrix nearness problems with bregman divergences. SIAM Journal on Matrix Analysis and Applications, 29(4):1120–1146, 2008.

[EDS20] Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. arXiv preprint arXiv:2010.03633, 2020.

[EH10] Ernesto Estrada and Desmond J Higham. Network properties revealed through matrix functions. SIAM review, 52(4):696–714, 2010.

[ES19] Stefania Ebli and Gard Spreemann. A notion of harmonic clustering in simplicial complexes. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pages 1083–1090. IEEE, 2019.

[FH16] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. Physics reports, 659:1–44, 2016.

[Fie89] Miroslav Fiedler. Laplacian of graphs and algebraic connectivity. Banach Center Publications, 25(1):57–70, 1989.

[Fri91] Noah E Friedkin. Theoretical foundations for centrality measures. American journal of Sociology, 96(6):1478–1504, 1991.

[FS11] David Chin-Lung Fong and Michael Saunders. Lsmr: An iterative algorithm for sparse least-squares problems. SIAM Journal on Scientific Computing, 33(5):2950–2971, 2011.

[GCZ+20] Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. Advances

in Neural Information Processing Systems, 33:11984–11995, 2020.

[GDPG+21] Lucia Valentina Gambuzza, Francesca Di Patti, Luca Gallo, Stefano Lepri, Miguel Romance, Regino Criado, Mattia Frasca, Vito Latora, and Stefano Boccaletti. Stability of synchronization in simplicial complexes. Nature communications, 12(1):1255, 2021.

[Ger31] Semyon Aranovich Gershgorin. Uber die abgrenzung der eigenwerte einer matrix. Notes of Russian Academy of Science, (6):749–754, 1931.

[GL22] Nicola Guglielmi and Christian Lubich. Matrix nearness problems and eigenvalue optimization. Book in preparation, 2022.

[GLF+19] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 922–929, 2019.

[GLL91] L Grippo, F Lampariello, and S Lucidi. A class of nonmonotone stabilization methods in unconstrained optimization. Numerische Mathematik, 59(1):779–805, 1991.

[GLO20] Anne Greenbaum, Ren-cang Li, and Michael L Overton. First-order perturbation theory for eigenvalues and eigenvectors. SIAM review, 62(2):463–482, 2020.

[GLS23] Nicola Guglielmi, Christian Lubich, and Stefano Sicilia. Rank-1 matrix differential equations for structured eigenvalue optimization. SIAM Journal on Numerical Analysis, 61(4):1737–1762, 2023.

[GS14] Javad Ghaderi and Rayadurgam Srikant. Opinion dynamics in social networks with stubborn agents: Equilibrium and convergence rate. Automatica, 50(12):3209–3215, 2014.

[GS17] Nicolas Gillis and Punit Sharma. On computing the distance to stability for matrices using linear dissipative hamiltonian systems. Automatica, 85:113–121, 2017.

[GS23a] Vincent P Grande and Michael T Schaub. Disentangling the spectral properties of the hodge laplacian: Not all small eigenvalues are equal. arXiv:2311.14427, 2023.

[GS23b] Vincent P. Grande and Michael T. Schaub. Topological point cloud clustering. arXiv:2303.16716, 2023.

[GST23] Nicola Guglielmi, Anton Savostianov, and Francesco Tudisco. Quantifying the structural stability of simplicial homology. Journal of Scientific Computing, 97(2), 2023.

[GTGHS20] Nicolás García Trillos, Moritz Gerlach, Matthias Hein, and Dejan Slepčev. Error estimates for spectral convergence of the graph laplacian on random geometric graphs toward the laplace–beltrami operator. Foundations of Computational Mathematics, 20(4):827–887, 2020.

[GVL13] Gene H Golub and Charles F Van Loan. Matrix computations. JHU press, 2013.

[Han02] Phil Hanlon. The Laplacian Method. In Sergey Fomin, editor, Symmetric Functions 2001: Surveys of Developments and Perspectives, NATO Science Series, pages 65–91, Dordrecht, 2002. Springer Netherlands.

[Hat05] Allen Hatcher. Algebraic topology. Cambridge University Press, 2005.

[Hig90] Nicholas J Higham. Analysis of the cholesky decomposition of a semi-definite matrix. 1990.

[Hig08] Nicholas J. Higham. Functions of Matrices. Society for Industrial and Applied Mathematics, 2008.

[HJ12] Roger A Horn and Charles R Johnson. Matrix analysis. Cambridge University Press, 2012.

[HS$^+$52] Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards, 49(6):409–436, 1952.

[HW92] Wolfgang Hackbusch and Gabriel Wittum. Incomplete decomposition (ilu): Algorithms, theory, and applications. Notes Numer. Fluid Mech, 41, 1992.

[KHT09] Steffen Klamt, Utz-Uwe Haus, and Fabian Theis. Hypergraphs and cellular networks. PLoS computational biology, 5(5):e1000385, 2009.

[KS16] Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for Laplacians-fast, sparse, and simple. In 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pages 573–582. IEEE, 2016.

[LB12] Oren E Livne and Achi Brandt. Lean algebraic multigrid (lamg): Fast graph laplacian linear solver. SIAM Journal on Scientific Computing, 34(4):B499–B522, 2012.

[LCK+19] Hyekyoung Lee, Moo K Chung, Hyejin Kang, Hongyoon Choi, Seunggyun Ha, Youngmin Huh, Eunkyung Kim, and Dong Soo Lee. Coidentification of group-level hole structures in brain networks via hodge laplacian. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22, pages 674–682. Springer, 2019.

[Lim20] Lek-Heng Lim. Hodge Laplacians on Graphs. SIAM Review, 62(3):685–715, January 2020.

[LLO+21] Yongsun Lee, Jongshin Lee, Soo Min Oh, Deokjae Lee, and B Kahng. Homological percolation transitions in growing simplicial complexes. Chaos: An Interdisciplinary Journal of Nonlinear Science, 31(4), 2021.

[LN21a] Davide Lofano and Andrew Newman. The worst way to collapse a simplex. Israel Journal of Mathematics, 244(2):625–647, 2021.

[LN21b] Davide Lofano and Andrew Newman. The worst way to collapse a simplex. Israel Journal of Mathematics, 244(2):625–647, 2021.

[Man80] Thomas A Manteuffel. An incomplete factorization technique for positive definite linear systems. Mathematics of computation, 34(150):473–497, 1980.

[MDP21] Matthew J McDermott, Shyam S Dwaraknath, and Kristin A Persson. A graph-based network for predicting chemical reaction pathways in solid-state materials synthesis. Nature communications, 12(1):3097, 2021.

[MSOI+02] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. Science, 298(5594):824–827, 2002.

[MV07] Oliver Mason and Mark Verwoerd. Graph theory and networks in biology. IET systems biology, 1(2):89–119, 2007.

[New06] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. Physical review E, 74(3):036104, 2006.

[NKL19] Buddhika Nettasinghe, Vikram Krishnamurthy, and Kristina Lerman. Diffusion in social networks: Effects of monophilic contagion, friendship paradox, and reactive networks. IEEE Transactions on Network Science and Engineering, 7(3):1121–1132, 2019.

[NN16] Artem Napov and Yvan Notay. An efficient multigrid method for graph laplacian systems. Electron. Trans. Numer. Anal, 45:201, 2016.

[OPT+17] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. EPJ Data Science, 6:1–38, 2017.

[OPW22] Braxton Osting, Sourabh Palande, and Bei Wang. Spectral sparsification of simplicial complexes for clustering and label propagation. Journal of computational geometry, 11(1), 2022.

[RGWC+23] Emily Ribando-Gros, Rui Wang, Jiahui Chen, Yiying Tong, and Guo-Wei Wei. Combinatorial and Hodge Laplacians: Similarity and difference, 2023.

[Saa85] Youcef Saad. Practical use of polynomial preconditionings for the conjugate gradient method. SIAM Journal on Scientific and Statistical Computing, 6(4):865–881, 1985.

[Saa03] Yousef Saad. Iterative methods for sparse linear systems. SIAM, 2003.

[SB07] Thomas Schlitt and Alvis Brazma. Current approaches to gene regulatory network modelling. BMC bioinformatics, 8:1–22, 2007.

[SBH+20] Michael T. Schaub, Austin R. Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random Walks on Simplicial Complexes and the Normalized Hodge 1-Laplacian. SIAM Review, 62(2):353–391, January 2020.

[SMP+22] Michael Szell, Sayat Mimar, Tyler Perlman, Gourab Ghoshal, and Roberta Sinatra. Growing urban bicycle networks. Scientific reports, 12(1):6765, 2022.

[SOMMA02] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of escherichia coli. Nature Genetics, 31(1):64–68, 2002.

[SS08] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In Proceedings of the fortieth annual ACM symposium on Theory of computing, pages 563–568, 2008.

[SSF+22] Michael T Schaub, Jean-Baptiste Seby, Florian Frantzen, T Mitchell Roddenberry, Yu Zhu, and Santiago Segarra. Signal processing on simplicial complexes. In Higher-Order Systems, pages 301–328. Springer, 2022.

[ST14] Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. SIAM Journal on Matrix Analysis and Applications, 35(3):835–885, 2014.

[Stü01] Klaus Stüben. A review of algebraic multigrid. Numerical Analysis: Historical Developments in the 20th Century, pages 331–359, 2001.

[Tan10] Martin Tancer. d-collapsibility is np-complete for d greater or equal to 4. Chicago Journal OF Theoretical Computer Science, 3:1–28, 2010.

[Tan16a] Martin Tancer. Recognition of collapsible complexes is np-complete. Discrete & Computational Geometry, 55:21–38, 2016.

[Tan16b] Martin Tancer. Recognition of collapsible complexes is NP-complete. Discrete & Computational Geometry, 55:21–38, 2016.

[TB20] Joaquín J Torres and Ginestra Bianconi. Simplicial complexes: higher-order spectral dimension and dynamics. Journal of Physics: Complexity, 1(1):015002, 2020.

[TBP21] Francesco Tudisco, Austin R Benson, and Konstantin Prokopchik. Nonlinear higher-order label spreading. In Proceedings of the Web Conference 2021, pages 2402–2413, 2021.

[TH18] Francesco Tudisco and Matthias Hein. A nodal domain theorem and a higher-order Cheeger inequality for the graph p-Laplacian. Journal of Spectral Theory, 8(3):883–908, 2018.

[TH21] Francesco Tudisco and Desmond J Higham. Node and edge nonlinear eigenvector centrality for hypergraphs. Communications Physics, 4(1):201, 2021.

[Tro19] Joel A Tropp. Matrix concentration & computational linear algebra. 2019.

[TZB96] Oleg N Temkin, Andrew V Zeigarnik, and DG Bonchev. Chemical reaction networks: a graph-theoretical approach. CRC Press, 1996.

[Vig16] Sebastiano Vigna. Spectral ranking. Network Science, 4(4):433–445, 2016.

[W+01] Douglas Brent West et al. Introduction to graph theory, volume 2. Prentice hall Upper Saddle River, 2001.

[Whi39a] John Henry Constantine Whitehead. Simplicial spaces, nuclei and m-groups. Proceedings of the London mathematical society, 2(1):243–327, 1939.

[Whi39b] John Henry Constantine Whitehead. Simplicial spaces, nuclei and m-groups. Proceedings of the London mathematical society, 2(1):243–327, 1939.

[WWLX22] Ronald Wei, Junjie Wee, Valerie Laurent, and Kelin Xia. Hodge theory-based biomolecular data analysis. Scientific Reports, 12, 06 2022.

[Y+02] Ulrike Meier Yang et al. Boomeramg: A parallel algebraic multigrid solver and preconditioner. Applied Numerical Mathematics, 41(1):155–177, 2002.

[YL14] Jaewon Yang and Jure Leskovec. Overlapping communities explain core–periphery organization of networks. Proceedings of the IEEE, 102(12):1892–1902, 2014.

[ZCH+20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. AI open, 1:57–81, 2020.

[ZJL+22] Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günnemann, Neil Shah, and Meng Jiang. Graph data augmentation for graph machine learning: A survey. arXiv preprint arXiv:2202.08871, 2022.