Decentralized Computation of Homology Groups in Networks by Gossip

Abubakr Muhammad and Ali Jadbabaie
Department of Electrical and Systems Engineering
University of Pennsylvania, Philadelphia, PA
{abubakr,jadbabai}@seas.upenn.edu

***Abstract*— In this paper, we present an approach towards the computation of certain topological invariants in real sensor networks. As shown by many researchers, these invariants are relevant for modeling certain properties of the network such as coverage and routing. What has been lacking so far is a concrete decentralized method to compute these invariants for proper implementation. In this paper, we give an approach towards such an implementation. The main tools being used here are the the so-called higher order Laplacian operators and distributed methods for their spectral analysis that resemble gossip algorithms.**

I. Introduction and Motivation

Recently, several properties in networked sensing and distributed systems have been modeled by various researchers [3], [4], [7], [9], [13], [11] using topological spaces and their topological invariants. The unifying theme in these approaches has been that the local properties of a network, as dictated by local interactions among agents, can be captured by certain topological spaces. These spaces are mostly combinatorial in nature and are a generalization of the more familiar graphical models. Moreover, the global properties of the network characteristics correspond to certain topological invariants of these spaces. Examples of such modeling attempts include coverage problems for sensor networks [3], [4], [9]; concurrency modeling in asynchronous distributed systems [11]; and routing in networks without geographical information [7].

For a practical utilization of these studies on distributed systems, it is vitally important that the global topological invariants can be computed in a decentralized manner with manageable complexity and scalability. In particular, the distributed computation of homology groups is important for many of these problems [4]. In [13], the first steps towards this goal has been taken. In this paper, we will utilize many basic ideas generated in that work. We focus on methods of computing homology groups that are not only decentralized but implementable on real networks in the form of tokenized protocols that take into account network characteristics such as medium access and bandwidth limitations.

One should point out from the start, that the idea of a decentralized algorithm for computing a global invariant is seemingly contradictory. If the size of the hole is very large, one might need to accumulate all local information at one point before computing the global knowledge of the topology. This apparently makes the whole idea of decentralization irrelevant, as the node that carries all the local information can execute a standard centralized homology algorithm and compute the global topology. However, this goes against the spirit of networked sensing, in which nodes are assumed to have little computational power, and the accumulation of network data at one or few nodes is disliked because of the constraints on network resources.

To resolve this dilemma, we study certain probabilistic methods. The main idea is to create a random process at the node level which generates enough mixing between local estimates so that the local estimates converge to the global knowledge with a high probability [1]. The nodes observe the random process only locally and from their local estimates distill the global information with a certain amount of confidence.

It has been shown in [13] that the spectral decomposition of the so called higher order Laplacian operators on simplicial complexes is one way to compute homology groups. These Laplacians are essentially local averaging or mixing operations, and therefore work in the spirit of gossip algorithms [2]. The flow of these Laplacians has been used in [13] to detect absence of holes or a single hole. But it is not clear how to distinguish between multiple holes using the flow alone unless there is direct method of doing a decentralized spectral decomposition.

In this paper, we build on this theme by first giving explicit protocols at the node level for building the simplicial complexes and then to implement the Laplacian at the simplex level (Section III). We then utilize the results from the work of Kempe at al. [12], where a method for doing a decentralized spectral decomposition of a matrix has been studied (Section III-I). A mixing or gossip-like algorithm has been used to recover a global estimate of certain matrices which cannot be computed locally. These estimates are then used to compute the spectral decomposition locally. We use this idea to compute a decentralized spectral decomposition of the Laplacians. We start with a brief introduction to simplicial complexes, homology groups and their relation to higher order Laplacians.

II. Higher Order Laplacians and their Spectra

A. Simplicial Complexes and Homology

Simplicial complexes are a class of topological spaces that are made of simplices of various dimensions. Given a set of points V, an ordered k-simplex is a subset $[v_0, v_1, \ldots, v_k]$ where $v_i \in V$ and $v_i \neq v_j$ for all $i \neq j$. The *faces* of this k-simplex consist of all $(k-1)$-simplices in it. A *simplicial complex* is a collection of simplices which is closed with respect to the inclusion of faces. The orderings of the vertices correspond to an *orientation*.

Two k-simplices σ_i and σ_j of a simplicial complex X are *upper adjacent* if both are faces of some $k+1$-simplex in X. We denote this adjacency by $\sigma_i \frown \sigma_j$. The *upper degree* of a k-simplex, denoted $\deg_u(\sigma)$, is the number of $k+1$-simplices in X of which σ is a face. Now, give X an orientation and suppose $\sigma_i \frown \sigma_j$ with a common $k+1$-simplex ξ. If the orientations of σ_i and σ_j agree with the ones induced by ξ then σ_i, σ_j are said to be *similarly oriented* with respect to ξ. If not, we say that the simplices are *dissimilarly oriented*. In a similar fashion, we also define *lower adjacency* and *lower degree* of simplices. Two k-simplices σ_i and σ_j of a simplicial complex X are *lower adjacent* if both have a common face, denoted by $\sigma_i \smile \sigma_j$. The *lower degree* of a k-simplex, denoted $\deg_l(\sigma)$, is equal to the number of faces in σ.

For each $k \geq 0$, denote by $C_k(X)$ the vector space, whose basis is the set of oriented k-simplices of X. The elements of these vector spaces are called as *chains*, which are linear combinations of the basis elements. The *boundary maps* are defined to be the linear transformations $\partial_k : C_k(X) \rightarrow C_{k-1}(X)$ which acts on basis elements $[v_0, \ldots, v_k]$ via

$$\partial_k[v_0, \ldots, v_k] := \sum_{i=0}^{k} (-1)^i [v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k].$$

When dealing with a finite simplicial complex X, the vector spaces $C_i(X)$ are also of finite dimension. Therefore we can represent the boundary maps $\partial_i : C_i(X) \rightarrow C_{i-1}(X)$ in matrix form. By doing a simple calculation of $\partial_k \partial_{k-1}$ using Equation 1, it is easy to see that the composition $C_k(X) \xrightarrow{\partial_k} C_{k-1}(X) \xrightarrow{\partial_{k-1}} C_{k-2}(X)$ is zero. From this, it follows that im $\partial_k \subset \ker \partial_{k-1}$. The *k-th homology group* of the space X is defined as $H_k(X) = \ker \partial_k / \text{im } \partial_{k+1}$. Homology groups are used to distinguish topological spaces from one another by identifying the number of 'holes' of various dimension, contained in these spaces.

B. Combinatorial k-Laplacians

Since the boundary operators ∂_k have finite matrix representations, one can define the dual operators $\partial_k^* : C_{k+1}(X) \rightarrow C_k(X)$. This allows us to define the operators $\mathcal{L}_k : C_k(X) \rightarrow C_k(X)$, from the k-chains to themselves by $\mathcal{L}_k = \partial_{k+1}\partial_{k+1}^* + \partial_k^*\partial_k$. These operators are called as the discrete k-Laplacians of the simplicial complex. They were introduced by Eckmann in 1945 [6], and have been studied since then under the name of *combinatorial Laplacians* [5], [8], [14]. Eckmann noted that each $C_k(X)$ decomposes into orthogonal subspaces

$$C_k(X) = \mathcal{H}_k(X) \oplus \text{im}(\partial_{k+1}) \oplus \text{im}(\partial_k^*),$$

where $\mathcal{H}_k(X) = \{c \in C_k(X) : \mathcal{L}_k c = 0\} = \ker \mathcal{L}_k$. From this follows that for each k, there is an isomorphism $H_k(X) \cong \mathcal{H}_k(X)$.

This means that in order to compute the homology groups of a simplicial complex, it is enough to study the null space of the matrix $\mathcal{L}_i$. One can recognize that the more familiar *graph Laplacian* is synonymous with $\mathcal{L}_0 : C_0(X) \rightarrow C_0(X)$. Since there are no simplices of negative dimension, $C_{-1}(X)$ is assumed to be 0. Also, the maps ∂_0 and ∂_0^* are assumed to be zero maps so that $\mathcal{L}_0 = \partial_1\partial_1^*$, where ∂_1 is the incidence matrix of a graph.

III. Homology Groups and k-Laplacians for Realistic Sensor Networks

In this section, we discuss the practical issues in interpreting and realizing homology groups and k-Laplacians in real networks.

A. Connectivity Graphs and Rips Complexes

We first explain the relevance of simplicial complexes in the context of networked sensing and control. As discussed in [3], [4], [9], the sensing and communication properties of sensor networks exhibit various aspects, that can be captured by simplicial complexes and their homology groups. Consider, for example, the connectivity graph of a sensor network of identical agents, each having an isotropic radio of range, say ϵ. In this graph, the vertices correspond to the sensing nodes, and an edge between two vertices signifies that the two nodes are within a distance ϵ of each other. We now build what is called the *Rips Complex* induced by this connectivity graph.

Definition III-B: Given a set of points $x_1, \ldots, x_N$ in $\mathbb{R}^d$ and $\epsilon > 0$, the *Veitoris-Rips complex* $\mathcal{R}$, is the simplicial complex whose k-simplices correspond to the unordered $(k+1)$-tuples of the points which are pairwise within a distance ϵ of each other.

Thus, the $0-$ and $1-$ simplices of the Rips complex are exactly the nodes and vertices of the connectivity graph. The dimension of the zeroth homology group $H_0(\mathcal{R})$ counts the number of connected components of the network. Clearly, $H_0(\mathcal{R})$ has dimension 1 if and only if the corresponding connectivity graph is connected. The homology group $H_1(\mathcal{R})$ counts the number of network holes in the connectivity graph. As shown in [3], [4], [9], these homology groups are related to certain coverage properties of a sensor network.

Since these coverage results require explicit computation of the homology groups, it is important to know whether these computations can be performed in a distributed manner over a real sensor network. Thus one would desire an algorithm that operates at the node level and interacts only locally with its neighbors to manage complexity. In the approach taken by us in this paper, we describe the algorithms at the simplex level and also describe a procedure to project the simplices further down to the node level.

The first step in constructing a distributed algorithm is to obtain a local representation of the Rips complex. Since a distributed algorithm essentially runs at the node level, there are two steps in obtaining such a representation.

1) *Simplex Membership:* Each node should discover what higher simplices it is a part of. Thus a node can be a member of many simplices in the complex. A

procedure, in the form of a network protocol is needed to discover this membership.

2) *Simplex Ownership:* By definition, a k-simplex has $k+1$ nodes as its members. Thus all these nodes can claim ownership to a particular simplex. If the algorithm is described at the simplex level, either one requires a procedure for synchronization between different copies of the same simplex at various nodes, or a network protocol is needed to resolve ownership. Since the former approach potentially requires greater communication overhead, we adopt the later approach of specifying simplex ownership by a network protocol.

C. Protocols for Simplex Membership

Suppose that each agent carries a unique identification tag in the form of a natural number. Also, it is capable of communicating its identification tag to its neighboring agents along with some other information of interest. Each agent also maintains an array of lists of identification tags, each list corresponding to a simplex that the agent is a part of.

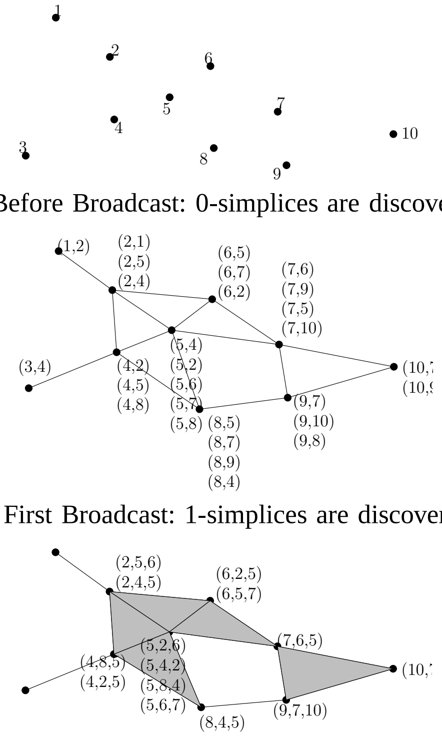
In the start, each agent is aware of its own identification alone. Therefore the first entry is the identification tag of the agent alone. This generates the 0-simplices. Now the agents simultaneously broadcast their identifications. The agents within the communication radius receive this information and pair the received tag with their own tags, and store this information in their respective arrays. This generates the 1-simplices or the edges in the simplicial complex. Once again, the agents broadcast, but this time the list of edges that possess. After reception, each agent compares the received list of edges to its own array of edges and searches for a cycle. If it finds one, it appends each three cycle to its array, thus generating the 2-simplices. In the next broadcast, the lists of 2-simplices is shared between neighboring agent, thereby discovering 3-simplices. In this way, the all simplices of dimension k or lower are discovered in k broadcasts. This simplex discovery mechanism has been illustrated by a simple example in Figure 1. The protocol can be formally specified as follows.

First we give a procedure *FindSimplices()* that generates lists of $k+1$-simplices, from a list of k-simplices.

Algorithm *FindSimplices*
Input: k; List of k-simplices L_k.
Output: List of $k+1$ simplices L_{k+1}

$K \leftarrow$length of L_k;
$L_{k+1} = \emptyset$;
$\mathcal{C} \leftarrow$all K choose k combinations from $\{1,\ldots,K\}$;
for $i \leftarrow 1$ **to** $|\mathcal{C}|$
  $\{\sigma_1,\ldots,\sigma_k\}$ $\leftarrow$simplices corresponding to combination $\mathcal{C}(i)$;
  $\omega \leftarrow$the chain $\sigma_1+\ldots+\sigma_k$;
  if $\partial_k\omega = 0$, Append $\{\sigma_1,\ldots,\sigma_k\}$ to L_{k+1}
end

For properly implementing this algorithm on the sensing nodes, we specify a protocol *SimplexMember*. For computing a homology group of dimension n, we only need simplices of dimensions $n-1$,n and $n+1$. The protocol described below, applies the procedure *FindSimplices()* iteratively at each node, until all simplices of a desired dimension are discovered. Since, this iterative procedure is implemented at the node level as a broadcast protocol, there is a potential deadlock when many nodes try to broadcast their lists at the same time. We therefore assume the presence of an underlying medium access control (MAC) protocol that tells the node when to transmit and when all broadcasts have been received from its neighbors, so that the next level of simplices can be discovered.



(a) Before Broadcast: 0-simplices are discovered.

(b) First Broadcast: 1-simplices are discovered.

(c) Second Broadcast: 2-simplices are discovered.

Fig. 1. Discovery of the Rips complex via a broadcast protocol.

Algorithm *SimplexMember*
Input: Identification tag of node j; largest dimension of simplices to discover k.
Output: Lists of simplices $\{L_1,\ldots,L_k\}$

$L_0 \leftarrow \{j\}$;
for $i \leftarrow 1$ **to** k
  $L_i \leftarrow \emptyset$;
  repeat
    WAIT;
  until MAC allows TRANSMIT;
  TRANSMIT(L_{i-1})
  repeat
    WAIT;
  until MAC indicates all messages received;
  Get messages $\mathcal{M}_{i-1} \leftarrow \{M_1,\ldots,M_{\deg n_j}\}$;
  Append L_{i-1} to $\mathcal{M}_{i-1}$;
  L_i = FindSimplices($\mathcal{M}_{i-1}$, i);
end

D. Protocols for Simplex Ownership

We now move on to our next problem of resolving ownership for simplices by nodes. As mentioned above, $n+1$ nodes can claim ownership to an n-simplex. To resolve this we propose a simple protocol *SimplexOwn*. The result of this protocol is a sub-complex $\mathcal{R}_j \subset \mathcal{R}$, owned by a node with identification tag j. The full Rips complex is a union of these sub-complexes, given by $\mathcal{R} = \bigcup_{j=1}^{N} \mathcal{R}_j$.

Algorithm *SimplexOwn*
Input: Identification tag of node j; largest dimension of simplices to discover k.
Output: Sub-complex $\mathcal{R}_j \subset \mathcal{R}$ owned by n_j

$\{L_1, \ldots, L_k\} \leftarrow$SimplexMember($j$, k) ;
$\mathcal{R}_j \leftarrow \emptyset$;
for $p \leftarrow 1$ **to** k
  for $q \leftarrow 1$ **to** $|L_p|$
    $\sigma \leftarrow L_p(q)$;
    $\{v_0, \ldots v_k\} \leftarrow$nodes of σ;
    $v_{min} = \min(v_0, \ldots, v_k)$;
    if $v_{min} = j$, append σ to $\mathcal{R}_j$;
  end
end

Essentially, this protocol grants ownership of a simplex to a node if that node has the smallest identification tag in the simplex. It is easy to see that this simple protocol has no deadlocks in resolving ownership among nodes. This protocol however is not unique and there can be many other protocols for resolving ownership. Of course, proving the absence of deadlocks becomes non-trivial for more sophisticated protocols.

The special properties of the Rips complex lead to the following observations.

Proposition III-E: If $\mathcal{R}_j \subset \mathcal{R}$ is a sub-complex generated by running the protocol *SimplexOwn* on node n_j, then $\mathcal{R}_j$ has trivial homology in all dimensions.

Proof: Suppose that, the simplices $\sigma_1, \ldots, \sigma_n$ in $\mathcal{R}_j$ span $C_k(\mathcal{R}_j)$, where $k \geq 0$. Since all simplices have j as the common node, it is easy to see that any chain in $C_k(\mathcal{R}_j)$ is a star-neighborhood of j. Moreover, if there is a cycle induced by a simplex in $k+1$ dimension, then by the properties of the Rips-complex, it is always filled in due to the simplex itself. Thus, $\mathcal{R}_j$ is always connected and $H_k(\mathcal{R}) \cong 0$, for $k > 0$. ∎

Note, that this result holds for *any* ownership protocol that projects simplices down to the node level. Thus it is due to the special properties of the Rips complex and not the protocol, that the above result holds. This result also hints towards methods of combining the sub-complexes at various nodes to compute the global homology classes.

Proposition III-F: If the $k+1, k, k-1$ dimensional simplices of $\mathcal{R}$ are completely contained in two sub-complexes $\mathcal{R}_i$ and $\mathcal{R}_j$, generated by running a simplex ownership protocol on nodes i, j, then the k-th homology of $\mathcal{R} = \mathcal{R}_i \cup \mathcal{R}_j$ is isomorphic to the homology of $\mathcal{R}_i \cap \mathcal{R}_j$ at one dimension less.

Proof: We can prove this by a direct application of the so-called Mayer-Veitoris exact sequence [10]. The Mayer-Veitoris exact sequence states that in the following series of maps,

$$\begin{aligned} \cdots \to H_k(\mathcal{R}_i) \oplus H_k(\mathcal{R}_j) \quad &\xrightarrow{\Psi} \quad H_k(\mathcal{R}) \xrightarrow{\partial} \ldots \\ \ldots H_{k-1}(\mathcal{R}_i \cap \mathcal{R}_j) \quad &\xrightarrow{\Phi} \quad H_{k-1}(\mathcal{R}_i) \oplus H_{k-1}(\mathcal{R}_j) \to \end{aligned}$$

the image of each map is isomorphic to the kernel of its proceeding map. From Proposition III-E, we obtain $H_k(\mathcal{R}_i) \oplus H_k(\mathcal{R}_j) \cong 0$ and $H_{k-1}(\mathcal{R}_i) \oplus H_{k-1}(\mathcal{R}_j) \cong 0$. By exactness, ∂ is an isomorphism, and thus $H_k(\mathcal{R}_i \cup \mathcal{R}_j) \cong H_{k-1}(\mathcal{R}_i \cap \mathcal{R}_j)$. ∎

As a consequence of this result, the homology cycles in dimension 1 can be computed by counting the number of connected components (i.e. the 0-th homology) of $\mathcal{R}_i \cap \mathcal{R}_j$. Similar results hold for higher dimensions. A generalization of this result to an arbitrary number of sub-complexes and a study of network protocols for implementing these ideas are subjects of future research.

G. k-Laplacians as Local Averaging

Let us compute a formula for the k-Laplacian at the simplex level. Let $\sigma_1, \cdots, \sigma_n$ be the oriented k-simplices of a simplicial complex X. By computing the terms $(\partial_{k+1}\partial_{k+1}^*)_{ij}$ and $\partial_k^*\partial_k)_{ij}$ for a fixed pair i, j, we obtain

$$\begin{aligned} (\mathcal{L}_k)_{ij} &= (\partial_k^*\partial_k)_{ij} + (\partial_{k+1}\partial_{k+1}^*)_{ij} \\ &= \begin{cases} \deg_u(\sigma_i) + k + 1 & \text{if } i = j, \\ 1 & \text{if } i \neq j,\ \sigma_i \not\frown \sigma_j, \text{ and} \\ & \sigma_i \smile \sigma_j \text{ with similar} \\ & \text{orientation,} \\ -1 & \text{if } i \neq j,\ \sigma_i \not\frown \sigma_j, \text{ and} \\ & \sigma_i \smile \sigma_j \text{ with dissimilar} \\ & \text{orientation,} \\ 0 & \text{if } i \neq j \text{ and either} \\ & \sigma_i \frown \sigma_j \text{ or } \sigma_i \not\smile \sigma_j. \end{cases} \end{aligned}$$

Let $\epsilon_{ij} \in \{-1, 1\}$ capture the similarity or dissimilarity of orientation between simplices σ_i, σ_j, then the formula for the k-Laplacian can be explicitly written at the simplex level as,

$$\mathcal{L}_k(\sigma_i) = (\deg_u(\sigma_i)+k+1)\sigma_i + \sum_{\sigma_i \smile \sigma_j} \epsilon_{ij}\sigma_j - \sum_{\sigma_i \frown \sigma_m} \epsilon_{im}\sigma_m.$$

This formula suggests that the k-Laplacians work in the spirit of a certain class of distributed algorithms known as gossip algorithms [2], whereby the local estimates are updated using estimates from simplices that are upper or lower adjacent only. From the point of view of an actual representation inside a network, let us observe the following connectivity condition between two simplices that are adjacent. We give the result in terms of communication *hop-count*, which is defined as the smallest number of communication links between two nodes in a network.

Proposition III-H: Let σ_1, σ_2 be two k-simplices in $\mathcal{R}$ of a network. Suppose that $\sigma_1 \in \mathcal{R}_{j_1}$ and $\sigma_2 \in \mathcal{R}_{j_2}$, where the sub-complexes $\mathcal{R}_{j_1}, \mathcal{R}_{j_2} \subset \mathcal{R}$ are owned by the nodes j_1, j_2 by executing the protocol *SimplexOwn*. Also, let $d(i,j)$ be the hop-count between two nodes i, j in the network.

1) If $\sigma_1 \frown \sigma_2$, then $d(j_1, j_2) \leq 1$.
2) If $\sigma_1 \smile \sigma_2$, then $d(j_1, j_2) \leq 2$.

Proof: First note that if the two simplices are owned by the same node, then $d(j_1, j_2) = 0$. If $\sigma_1 \frown \sigma_2$, then by definition both k-simplices belong to a $k+1$-simplex. Since in a simplex any two vertices are adjacent to each other, $d(j_1, j_2) \leq 1$.

For the case when $\sigma_1 \smile \sigma_2$, take any pair of vertices, one from each simplex. The two vertices can be at most one edge distant from the common $k-1$-simplex. Thus the two nodes are at most 2 edges distant from each other, which proves that $d(j_1, j_2) \leq 2$. ∎

Thus the k-Laplacian can be implemented at the node level using at the most 2-hops of communications between neighboring nodes. We only need the existence of a routing protocol that delivers packets between two nodes that are at most 2 hops apart. Thus the higher order Laplacian is essentially a local operation and needs a small communication overhead for implementation on a real network.

Due to the local nature of the Laplacians, it has been suggested in [13] to use these higher-order Laplacians for computing homology groups in a decentralized manner. A class of dynamical systems, based on the flow of the these Laplacians, has been studied in [13]. Crudely speaking, the convergence properties of the dynamical system

$$\frac{\partial \omega(t)}{\partial t} = -\mathcal{L}_k \omega(t), \quad (1)$$

depend on the k-th homology group of the underlying simplicial complex X. One can think of it as a higher dimensional analog of a discretized heat equation.

This system has been shown to be generally semi-stable, and has asymptotic stability for any choice of initial conditions if and only if $H_k(X) \cong 0$. Thus the Laplacian flow is a decentralized way of detecting whether the k-homology is trivial (i.e. no holes). If X has holes, then depending on the initial condition, the dynamical system converges to the subspace spanned by the homology classes of $\mathcal{L}_k$. The convergence rate of this system is controlled by the smallest non-zero eigenvalue of $\mathcal{L}_k$. If there is exactly one hole in dimension k, we have a decentralized method of computing that homology class by letting the dynamical system converge to its harmonic representation. It should be noted that the system converges to that homology class for any choice of initial conditions.

For two holes and more, this method fails to compute useful representations of the multiple homology classes explicitly. For an example, examine the Rips complex drawn in Figure 2 and the eigenvectors of $\mathcal{L}_1$. The thickness of the edges represent the magnitude of the corresponding coefficients of the eigenvectors. These magnitudes are clearly indicative of the proximity of a hole. In this case, the two eigenvectors can help distinguish the two holes quite clearly.

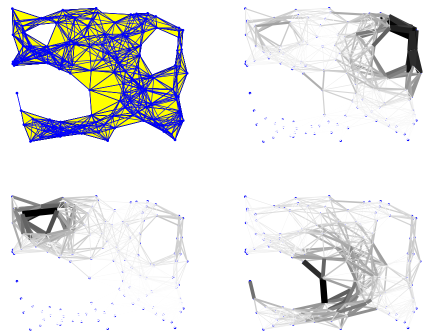


Fig. 2. A simplicial complex [upper left], zero-eigenvectors of the 1-Laplacian [upper right and lower left] and the eigenvector corresponding to the smallest non-zero eigenvalue of $\mathcal{L}_1$ [lower right].

If we employ the method of the Laplacian flow as described in [13], the system converges to a subspace spanned by the two eigenvectors ω_1, ω_2 corresponding to the null-space of $\mathcal{L}_1$. One such simulation has been reproduced in Figure Note that the system converges to a vector $\omega_\infty = \alpha_1\omega_1 + \alpha_2\omega_2$, where $\alpha_1, \alpha_2 \in \mathbb{R}$. By the linearity of the Laplacian, $\mathcal{L}_1\omega_\infty = \alpha_1\mathcal{L}_1\omega_1 + \alpha_2\omega_2 = 0$. So ω_∞ is a member of the first homology class of X. However, in its local representation, ω_∞ is quite useless. At the edge-level, the magnitudes do not relate to the proximity to the holes nor is there a clear relation between this cycle and the holes, unless the initial conditions are chosen close to any one of the eigenvectors ω_1, ω_2. But this choice cannot be made since we are using this method to compute the eigenvectors in the first place. Hence we need an improved method of computing homology groups for an arbitrary number of holes.

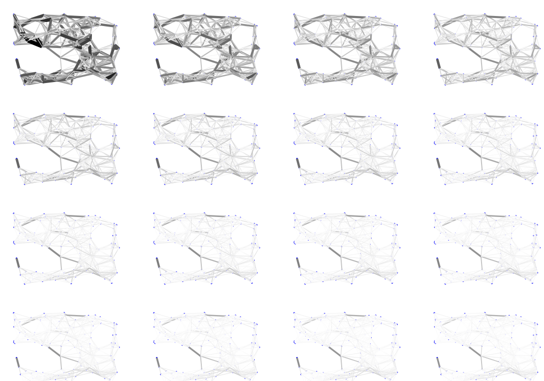


Fig. 3. The 1-Laplacian flow on the simplicial complex of Figure 2 for a randomly chosen initial condition. For this example, the flow cannot distinguish between multiple holes.

I. Decentralized Spectral Decomposition

To address the limitations of the methods explained above, we briefly describe an alternative method that computes a direct spectral decomposition of $\mathcal{L}_k$. In [12], an algorithm to do a decentralized spectral decomposition of the adjacency matrix of a graph has been studied. By the nature of the algorithm, it works for any matrix that can be locally represented in a network. Since we can represent the k-Laplacian at the simplex level by the formulae described in the previous section, we can employ this algorithm for computing the spectral decomposition. We give the details of this algorithm below.

The algorithm is based on the standard orthogonal iteration scheme for computing the k top eigenvectors of a matrix $A \in \mathbb{R}^{n \times n}$. The basic idea of the orthogonal iteration algorithm is to start with an initial estimate for the eigenvectors $e_1, \ldots, e_n$, stacked in $Q_n \in \mathbb{R}^{n \times k}$. Next compute the standard QR-factorization of AQ_n. Let this factorization be given by $Q_{n+1}R$, then we can invert R to get $Q_{n+1} = R^{-1}AQ_n$. For large n, Q_n approximates the actual eigenvectors with a small error.

Suppose now that the matrix A models some property of a network, in which the entries A_{ij} capture some relationship at the local level. The simplest example is the adjacency matrix of a connectivity graph, in which $A_{ij} \in \{0, 1\}$ model the absence or presence of communication links between nodes i and j. For the Rips complex of a network, A could be an upper or lower adjacency matrix between simplices. In all these cases, the spectral decomposition of A gives us useful information about the global properties of the network. Suppose that it is also desired that the network computes this decomposition so that it is self-aware of these global network properties. Let us comment on how the iteration described above can can be implemented in a decentralized manner.

It is reasonable to assume that each node (or simplex) each simplex only handles a corresponding row in Q_n. Also assume that information from adjacent simplices (upper or lower) can be obtained with a minimal communication overhead (as shown in the previously). Therefore, the computation $V = AQ_n$ is entirely local and directly implementable. However, the matrix R^{-1} does not have a local representation since the inversion requires complete knowledge of the matrix R. To circumvent this problem, first observe that $K := V^T V = RQ^T QR = R^T R$. If each simplex can somehow locally estimate K, then the Cholesky factorization of the estimate $\hat{K}$ will produce an estimate $\hat{R}$ for R, thus enabling the iteration $Q_{n+1} = \hat{R}^{-1}AQ_n$ to be implemented locally.

For computing a local estimate, observe that $K = \sum_\sigma V_\sigma^T V_\sigma$, where V_σ is a row of V corresponding to the simplex σ. However, each simplex can only produce $K_\sigma = V_\sigma^T V_\sigma$. The main idea in [12] is to use a gossip algorithm for local averaging to produce a global estimate for the sum $K = \sum_\sigma K_\sigma$. The gossip algorithm is basically a deterministic simulation of a random walk on the simplices at each step of QR-factorization. We refer the reader to [12] for further details on this. The main result is that by using this gossip algorithm, the estimate converges with probability one, in essentially $O(\tau_{mix} \log^2 n)$ rounds of communication and computation, where τ_{mix} is the mixing time of the random walk on the network. We implemented this algorithm and found it to be consistent with the results from a centralized computation. Convergence is slow but the method gives hope for a more efficient method for the decentralized computation of homology groups. A careful study of the convergence properties of the algorithm for the particular case of the k-Laplacians is currently being investigated by the authors.

IV. CONCLUSIONS

In this paper, we have focussed on some practical issues in computing simplicial complexes and their homology groups for real sensor networks. We have observed that the network routing and medium access properties deserve careful attention when studying these issues. We study the higher order Laplacians as the central idea towards the decentralization of these computations. Moreover, we circumvent the problem of requiring full global knowledge of certain matrices for the spectral decomposition of the Laplacians by using a gossip-like algorithm.

ACKNOWLEDGEMENTS

We gratefully acknowledge many helpful discussions with Vin de Silva, Magnus Egerstedt and Robert Ghrist.

REFERENCES

[1] I. Benjamini and L. Lovasz, "Global Information from Local Observation," in *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[2] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, "Gossip Algorithms: Design, Analysis and Applications," in *Proc. IEEE Infocom 2005*, Vol. 3, pp. 1653–1664, Miami, March 2005.

[3] V. de Silva and R. Ghrist, "Coverage in sensor networks via persistent homology," *Algebraic and Geometric Topology*. (To appear)

[4] V. de Silva, R. Ghrist and A. Muhammad, "Blind Swarms for Coverage in 2-D," *Robotics: Science and Systems*, MIT, Cambridge, MA, 2005.

[5] A. Duval and V. Reiner, "Shifted Simplicial Complexes are Laplacian Integral," *Transactions of the American Mathematical Society*, Vol. 354 Number 11, pp. 4313–4344, 2002.

[6] B. Eckmann, "Harmonische Funktionen und Randwertaufgaben in Einem Komplex," *Commentarii Math. Helvetici*, Vol. 17, 1945.

[7] Q. Fang, J. Gao, L. Guibas, V. de Silva, and L. Zhang, "Glider: gradient landmark-based distributed routing for sensor networks," *Proc. IEEE Infocom*, 2005.

[8] R. Forman, "Combinatorial Differential Topology and Geometry," *New Perspectives in Geometric Combinatorics*, L. Billera et al. (eds.), Cambridge University Press, Math. Sci. Res. Inst. Publ. 38, 1999.

[9] R. Ghrist and A. Muhammad, "Coverage and Hole-Detection in Sensor Networks via Homology," *The Fourth International Conference on Information Processing in Sensor Networks* (IPSN'05), UCLA, Los Angeles, CA, April 25-27, 2005.

[10] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 2002.

[11] M. Herlihy and N. Shavit, "The Topological Structure of Asynchronous Computability," *Journal of the ACM*, vol. 46, no. 6, pp. 858–923, November 1999.

[12] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 561–568, 2004.

[13] A. Muhammad and M. Egerstedt, "Control Using Higher Order Laplacians in Network Topologies," in *Proc. of 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan , 2006.

[14] M.Wachs and X. Dong, "Combinatorial Laplacian of the Matching Complex," *Electronic Journal of Combinatorics*, Vol. 9, 2002.