Distributed-Infrastructure Multi-Robot Routing using a Helmholtz-Hodge Decomposition

Peter Kingston and Magnus Egerstedt

***Abstract*— Using graphs and simplicial complexes as models for an environment containing a large number of agents, we provide distributed algorithms based on the Helmholtz-Hodge decomposition that, given desired flow rates on edges or across faces, produce incompressible approximations to the specified flows. These flows are then "lifted" to produce hybrid controllers for the agents, and a related algorithm is described that computes continuous streamfunctions over the environment, also in a distributed way.**

I. Introduction

It is commonly appreciated that many operators on graphs have strong physical and mathematical analogues on differentiable manifolds. Foremost among these is the *graph Laplacian*, whose study is particularly popular in the area of multiagent control. Yet despite this understanding, a number of related physical analogues appear to have been left unexplored in the multiagent systems literature. In this paper, we investigate one of these, a fluid-mechanical–inspired method by which vehicles – e.g., airplanes – can be routed within and between regions of an environment, in a manner that mimics incompressible flow.

The two main contributions of this paper are (1) a distributed, continuous-time algorithm for producing incompressible flows on graphs, and a connection to the well-known consensus algorithm, and (2) a simple method for "lifting" these flows to higher-dimensional models of the environment, to produce either (a) hybrid control laws, or (b) global streamfunctions (via another distributed algorithm), that are closely related.

A number of ideas inform and motivate this work.

The first of these is the recognition that real implementations of multiagent algorithms will often require infrastructure, in the form of wireless communications hubs, air traffic control towers, or other base stations. In these situations, it is natural to think of the static infrastructure as having some control authority over mobile agents – e.g., aircraft – that operate with its assistance. One then obtains *Eulerian* models for the multiagent system, a concept explored in [1].

A second set of ideas comes from the simulation of fluids (see e.g. [2], [3]), where the pressure in a fluid arises as the Lagrange multipliers corresponding to an incompressibility constraint. Fluid flow has been the inspiration for other work in multi-robot navigation, including [4] which models robots as an adiabatic gas (thus relaxing the incompressibility constraint) using smoothed particle hydrodynamics (a Lagrangian simulation technique), and [5], which computes continuous streamfunctions for the avoidance of individual static and moving obstacles.

The third concept is the Helmholtz-Hodge decomposition, both of a vector field on a smooth manifold (see e.g. [6]), and of a chain on a simplicial complex (as in [7] or [8]). The latter is the subject of *discrete exterior calculus* (discussed in [9]), which has found application in a number of areas including computer graphics (e.g. [10]), image processing and clustering (e.g. [11]), computational physics [12], statistical ranking [13], and multiagent control, including [14] where a connection to continuous PDEs is made, [15] which explores a related Laplacian-like operator, [16], which uses higher-order Laplacian dynamics to probe the homology of the complex, and [17], which additionally gives subgradient algorithms to find sparse representatives of the homology groups.

The formalism used in this paper closely parallels that of [16] and [17]. Philosophically, however, the goals are very different – in [16] and [17], one seeks to locate holes in a network; here, we look to direct agents throughout an environment. Technically, there are also important differences: We are not projecting 1-chains onto the harmonic subspace, and indeed we have no interest in separating the harmonic component from the rotational component at all, so we are able to work with lower-dimensional Laplacians. More importantly, streamfunc-

This work was supported in part by the Office of Naval Research through MURI-HUNT.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology. Emails: kingston@gatech.edu, magnus@ece.gatech.edu

tions and Hamiltonian vector fields appear nowhere in that work.

We would also like to mention [18], to which this work is indirectly related. There, an extremely interesting Dual Lyapunov approach is explored in which the divergence of state-space mass flows is used to analyze the stability of systems; those ideas are orthogonal (literally, in some ways!) to those of this paper.

In the remainder of this paper, we review a number of definitions that will be useful to us, emphasize a set of analogies that motivate this work, and describe the Helmholtz-Hodge decomposition, before giving distributed algorithms for computing incompressible flows and lifting them to higher-dimensional models of the environment. We conclude by discussing an application to air traffic management, and showing an example from simulation demonstrating the proposed methods.

II. Definitions

A. Abstract Simplicial Complex

The basic object with which we will model the environment is the abstract simplicial complex. In this section, we provide a brief review of this subject, mainly to introduce the notation and terminology that will be used in the rest of the paper. The interested reader may wish to refer to [7] or [8] for more intuition (although the formal definitions used in each are slightly different), as well as the introductions to [13] (which uses a dual formulation) and [17]. The definitions that follow in this section are more-or-less standard.

Given a finite set $V(K)$ of *vertices*, a *simplex* $\Delta \subset V(K)$ is a subset of $V(K)$. If the cardinality of that subset is $k+1$, then the *order* of Δ is said to be k, and it is called a k-simplex. Any $(k-1)$-simplex $\sigma \subset \Delta$ is a *face* of Δ. A *simplicial complex* K is a finite set of simplices that is closed with respect to taking faces; i.e., if $\Delta \in K$ and σ is a face of Δ, then $\sigma \in K$. A simplicial k-complex K is said to be *pure* if all simplices whose order is less than k are faces of higher-order simplices. We denote the k-simplices of K by $\Sigma_k(K)$. A simplex $\Delta \in K$ is a *coface* of $\sigma \in K$ if σ is a face of Δ. Two simplices σ_1, σ_2 are *lower-adjacent* (denoted $\sigma_1 \smile \sigma_2$) if they share a face, and *upper-adjacent* (denoted $\sigma_1 \frown \sigma_2$) if they share a coface.

An *orientation* of a simplex is a total order over its vertices, modulo even permutations, with a formal sign.[1] If the set $\Delta = \{v_0, \cdots, v_k\}$ is a simplex, we denote an orientation of Δ by an ordered list, e.g. $[v_0, \cdots, v_k]$. Two orientations related by an odd permutation are said to be *opposite*, and this is written with a minus sign; for instance $[v_0, v_1, v_2] = -[v_1, v_0, v_2]$. Finally, an orientation of a simplex induces an orientation on its faces; the i-th oriented face of an oriented simplex $\Delta = [v_0, \cdots, v_k]$ is,

$$\begin{aligned} F_i(\Delta) &= (-1)^i [v_0, \cdots, v_{i-1}, v_{i+1}, \cdots, v_k] \\ &= (-1)^i \Delta / v_i \ . \end{aligned} \tag{1}$$

Likewise, an orientation of a simplicial k-complex is an assignment of an orientation to each of its k-simplices. A simplicial k-complex is *consistently oriented* if, for every pair of lower-adjacent k-simplices Δ_1, Δ_2 sharing a face σ, Δ_1 and Δ_2 induce opposite orientations on σ.

A *k-chain* $c \in C_k(K)$ over an oriented simplicial complex K is a formal sum of elements from $\Sigma_k(K)$ taking coefficients from some commutative ring; we use the real numbers, $\mathbb{R}$. For instance, the formal sum $1.2v_0 + 2.6v_1 - 0.5v_4$ is a 0-chain over an appropriate simplicial complex. Formal sums can be added and multiplied by scalars in the natural way, so $C_k(K)$ forms a finite-dimensional real vector space. Additionally, we equip $C_k(K)$ with an inner product, $\langle \cdot, \cdot \rangle$, defined by

$$\left\langle \sum_{i=0}^{N} a_i \sigma_i, \sum_{i=0}^{N} b_i \sigma_i \right\rangle = \sum_{i=0}^{N} a_i b_i \tag{2}$$

where $\Sigma_k(K) = \{\sigma_0, \cdots, \sigma_N\}$, and $a_i, b_i \in \mathbb{R}\ \forall i$ are the chain coefficients.

Boundary operators will be central to this work. The k-th boundary operator $\delta_k(K) : C_k(K) \to C_{k-1}(K)$ on the oriented simplicial complex K is defined,

$$\delta_k(K) \left(\sum_{i=0}^{N} a_i \sigma_i \right) = \sum_{i=0}^{N} a_i \sum_{j=0}^{k} F_j(\sigma_i) \ ; \tag{3}$$

by convention, $\delta_0(K) = 0$. The null space of $\delta_k(K)$ is called the *k-cycles* of K and denoted $Z_k(K)$; the image of $\delta_{k+1}(K)$ is called the *k-boundaries* and denoted $B_k(K)$. The k-th homology group is the quotient space $H_k(K) = Z_k(K)/B_k(K)$; its dimension is the *k-th Betti Number* of K. Finally, the *k-th combinatorial Laplacian* is defined, $\mathcal{L}_k(K) = \delta_k^*(K)\delta_k(K) + \delta_{k+1}(K)\delta_{k+1}^*(K)$, where $\delta_k^*(K)$ denotes the adjoint operator to $\delta_k(K)$, called the *k-th coboundary operator*. The matrix representations of $\delta_k(K)$ and $\delta_k^*(K)$ are transposes of one another.

For the special case when K is 1-dimensional and so isomorphic to a graph, we may also use terminology

[1] The formal sign is necessary only to allow 0-simplices to have two orientations.

from graph theory [19].[2] There, $C_0(K)$ is called the *vertex space*, $C_1(K)$ is the *edge space*, $\delta_1(K)$ is the *cycle space* (and its dimension is the *cyclomatic number*), and $\operatorname{image} \delta_1^*(K)$ is the *cut space*.

A *realization* of a simplicial complex K is an isomorphic complex K' whose vertex set $V(K')$ is a finite subset of $\mathbb{R}^n$ for some $n \in \mathbb{N}$, and its *Rips Shadow* $\mathcal{R}(K') \subset \mathbb{R}^n$ is the union of the convex hulls of its simplices' vertex sets.

B. Analogies

The remainder of this paper is strongly motivated by close analogies between k-chains of different orders, and objects defined on differentiable manifolds.

A first set of analogies relates to the use of graphs as models for environments. Here, a vertex is the analogue of a point on a smooth manifold, and an edge is the analogue of an arclength-parametrized curve or unit tangent vector; here, *upper*-adjacency represents topology. A 0-chain is the analogue of a scalar field; its coefficients are values assigned to the corresponding vertices. A 1-chain is the analogue of a vector field; it can be thought of as assigning a directed flow to each edge. The coboundary operator $\delta_1^* : C_0(K) \to C_1(K)$ is the analogue of the gradient operator, and the boundary operator $\delta_1 : C_1(K) \to C_0(K)$ is the analogue of the divergence operator. Just as the Laplacian on $\mathbb{R}^n$ factors as $\nabla = \operatorname{div} \operatorname{grad}$, so too does the zeroeth combinatorial Laplacian factor into the analogous combinatorial operators, as $\mathcal{L}_0 = \delta_1^* \circ \delta_1$.

We will make a dual analogy for simplicial 2-complexes. Here, a triangle is the analogue of a point on a smooth manifold, and an edge or face is the analogue of a unit tangent vector; here, *lower*-adjacency represents topology. A 2-chain is the analogue of a scalar field. A 1-chain is the analogue of a vector field; it represents a directed flux across each face. The boundary operator $\delta_2^* : C_2(K) \to C_1(K)$ is the analogue of the gradient operator, and the coboundary operator $\delta_1^* : C_1(K) \to C_2(K)$ is the analogue of the divergence operator.

III. Helmholtz-Hodge Decomposition

The Helmholtz-Hodge decomposition of a vector field $v : \mathbb{R}^3 \to \mathbb{R}^3$ its unique representation as the sum

$$v = v_c + v_r + v_h \tag{4}$$

with $\operatorname{div} v_c \neq 0$, $\operatorname{curl} v_c = 0$; $\operatorname{div} v_r = 0$, $\operatorname{curl} v_r \neq 0$; and $\operatorname{div} v_h = 0$, $\operatorname{curl} v_h = 0$. From a functional analysis perspective, the three terms are projections of v onto three orthogonal linear subspaces of the space of vector fields on $\mathbb{R}^3$. The three terms are the the *curl-free*, *divergence-free*, and *harmonic* components, respectively. The first represents sources and sinks, the second vortices, and the third global flows representing the topology of the space, and illustrated in Figure 1.

On a simplicial 1-complex (i.e., a graph) G, we can compute an analogous decomposition of a 1-chain $v \in C_1(G)$ as

$$v = v_\mathrm{c} + v_\mathrm{r} \tag{5}$$

with $v_\mathrm{c} \perp v_\mathrm{r}$ under the inner product (2); this is the subject of section III-A. Note that by working with the 3-clique complex of a graph – a simplicial 2-complex – it is possible to further decompose v into a total of three components, including an analogue to the harmonic component of (4); this is the path taken in e.g. [13], but it comes at the cost of treating edges rather than nodes as the agents that perform computation, and, since we are not interested in distinguishing the harmonic component, it is not necessary for our purposes.

A. Hodge Decomposition on Graphs

From Hilbert's Projection Lemma, we know that orthogonal projections are least-squares solutions to linear equations. In particular, the orthogonal projection of a 1-chain $v \in C_1(G)$ onto its curl-free component can be found from the least-squares solution to the equation,

$$\delta_1^*(G) p = v \ . \tag{6}$$

We use $p \in C_0(G)$ for the unknown variable because it corresponds to pressure in fluid dynamics. The solution is readily found to be,

$$\begin{aligned} p &= (\delta_1(G)\delta_1^*(G))^\dagger \delta_1(G) v && (7) \\ &= \mathcal{L}_0^\dagger(G) \delta_1 v && (8) \end{aligned}$$

where $(\cdot)^\dagger$ denotes the pseudoinverse operation. [3] Once p is known, the curl-free component is reconstructed easily as

$$v_c = \delta_1^*(G) p \ . \tag{9}$$

What is interesting is that consensus dynamics solve the equation (6), as described in the following theorem:

[2] In graph theory, it is more common to use the two-element field $F_2 = \{0, 1\}$ (i.e., XOR serves as the addition operation) instead of $\mathbb{R}$.

[3] For the (matrix representation of the) graph Laplacian of a connected graph, this is the inverse restricted to $\operatorname{span}\{\mathbf{1}\}^\perp$. I.e., $L^\dagger = (L + \frac{1}{n}\mathbf{1}\mathbf{1}^T)^{-1} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$.

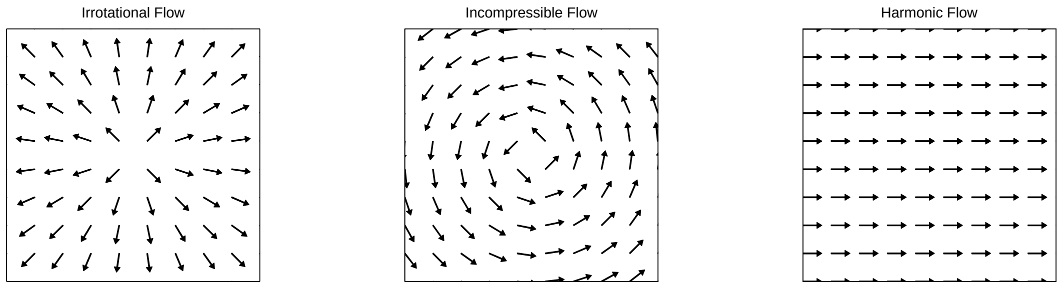

Fig. 1. Prototypical irrotational (left), incompressible (center), and harmonic (right) vector fields on $\mathbb{R}^2$.

Theorem 1: The forced Laplacian dynamics

$$\dot{p} = -\mathcal{L}_0(G)p + \delta_1(G)v \tag{10}$$

converge asymptotically to the solution (8) of (6) if $p(0) = 0$.

Proof : The ODE (10) can be written as

$$\dot{p} = -\text{grad}_p \, \frac{1}{2}||\delta_1^*(G)p - v||^2 \tag{11}$$

which are precisely the gradient descent dynamics needed to solve (6) (Here, the norm is that induced by the inner product (2)). Since the quadratic form is convex on $C_0(G)/\,\text{null}(\mathcal{L}_0(G))$, gradient descent converges in that quotient space regardless of initial condition, and since $p(0) = 0$, the component of p in $\text{null}(\mathcal{L}_0(G))$ remains zero for all time. ■

The important message is that the familiar Laplacian dynamics, when forced, solve the normal equations, and give a spatially-distributed way to asymptotically compute p.

The divergence-free component of the 1-chain v, likewise, is the projection of v onto $\text{image}\{\delta_1^*(G)\}^\perp$. Hence it can be found as,

$$v_r = v - v_c = v - \delta_1^* p \tag{12}$$

from the same p.

IV. Two-Dimensional Models

We now shift our attention from one- to two- dimensional models of the environment; these described by simplicial 2-complexes. We will describe a method for generating incompressible vector fields in their Rips Shadows as Hamiltonian vector fields, and for computing a single global streamfunction that generates these.

In this line of thought, *agents are 2-simplexes*. For the case of air traffic control, this represents the idea that each simplex is a region of airspace under the authority of a particular controller on the ground, and that it is the job of these automated ground controllers to agree in a distributed way how airplanes should be routed among themselves.

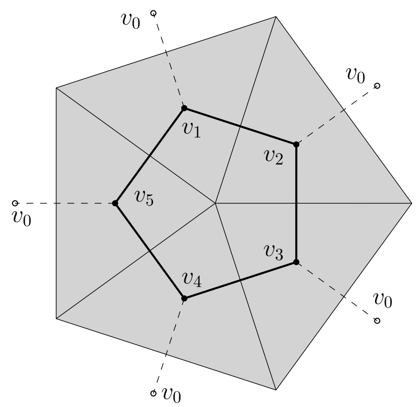

Fig. 2. Given a planar simplicial 2-complex K (gray), G is the lower-adjacency graph (bold lines) of the triangles. It is a subgraph of the dual graph $\mathcal{G}$ (bold and dashed lines) to the 1-skeleton of K (thin solid lines), denoted $\mathcal{G}^*$. (Note that the five copies of v_0 (circles) are identified.)

We will assume that the graph G of the previous sections is the lower-adjacency graph of the triangles of a pure simplicial 2-complex – i.e., that, given a 2-complex K, $V(G) = \Sigma_2(K)$, and (Δ_1, Δ_2) is an edge of G if and only if $\Delta_1 \smile \Delta_2$ in K. Equivalently, G is the subgraph of the dual graph to the 1-skeleton of K obtained by deleting the "outside vertex" (v_0 in Figure 2).

In what follows, we will produce an incompressible flow over $\mathcal{R}(K)$ by computing a particular 0-chain over K. To do this, we first introduce a family of local flows defined on the individual k-simplices (this is the subject of Section IV-A), and then compute a global 0-chain over K (Section IV-C) representing a streamfunction.

A. Local vector fields

In this section we will describe the individual building blocks for our global vector field. In particular, given a 0-chain over the vertices of a simplex, we will produce an incompressible flow within the simplex. This is done by using barycentric interpolation to create a streamfunction over the simplex, and defining a Hamiltonian vector field along this streamfunction.

Let $x_1, x_2, x_3 \in \mathbb{R}^2$ be the vertices of a realization of an oriented 2-simplex $\Delta = [v_0, v_1, v_2]$, Defining $X = [x_1, x_2, x_3] \in \mathbb{R}^{2\times 3}$, the barycentric coordinates $b \in \mathbb{R}^3$ of a point $x \in \mathbb{R}^2$ are the unique solution to the equations,

$$Xb = x \tag{13}$$

$$\mathbf{1}^T b = 1 \ . \tag{14}$$

It is also convenient to define the inverse matrices $B_1 \in \mathbb{R}^{3\times 2}$ and $B_2 \in \mathbb{R}^{3\times 1}$ by[4]

$$\begin{bmatrix} X \\ \mathbf{1}^T \end{bmatrix}^{-1} = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \ . \tag{15}$$

Then, letting $c_0 v_0 + c_1 v_1 + c_2 v_2$ be a 0-chain on Δ and $c = (c_0, c_1, c_2) \in \mathbb{R}^3$, we define a scalar field $\phi(\Delta) : \mathbb{R}^2 \to \mathbb{R}$ over the Rips Shadow of Δ by

$$\phi(\Delta)(x) = c^T \left(B_1 x + B_2\right) \ . \tag{16}$$

We will call $\phi(\Delta)$ the *local streamfunction* corresponding to the simplex Δ.

Finally, the Hamiltonian dynamics corresponding to $\phi(\Delta)$ are defined, in Cartesian coordinates, as

$$\begin{aligned} \dot{x} &= J \operatorname{grad} \phi(\Delta) \\ &= J B_1^T c \end{aligned} \tag{17}$$

or in barycentric coordinates as,

$$\begin{aligned} \dot{b} &= B_1 J B_1^T c \\ &\triangleq A(\Delta) \end{aligned} \tag{18}$$

where $J \in \mathbb{R}^{2\times 2}$ is the matrix representation of the symplectic form $(a, b) \mapsto \det([a, b])$. [5]

Lemma 1: The vector field (17) is divergence-free within each triangle.

Proof : The vector field $x \mapsto J B_1^T c$ is constant in x, so its divergence is zero. ■

[4]The inverse has a nice interpretation: b_i is the ratio of the volume of the simplex with x substituted for x_i, to that of the original simplex.

[5]I.e., $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

We will now use these per-simplex building blocks to assemble a single global vector field on K.

B. A global vector field

Under the assumption that the interiors of the Rips Shadows of all the simplices are disjoint, we define the piecewise vector field $\nu : \mathcal{R}(K) \to \mathbb{R}^3$ in barycentric coordinates by,

$$\nu(x) = \left\{ \begin{array}{lll} A(\Delta) & \text{if} & x \in \mathcal{R}(\Delta) \ \forall \Delta \in K \end{array} \right. \ . \tag{19}$$

In the section that follows, we will show that this vector field is *globally* divergence-free by demonstrating the existence of a single global streamfunction. Moreover, we will give a distributed algorithm to compute this streamfunction.

Before proceeding, however, we would like to point out that, already, (19) by itself constitutes a single hybrid controller for the vehicles: Each vehicle looks up which 2-simplex Δ it is in, requests the vector $A(\Delta)$ from Δ, and then follows that vector field.

C. The global stream function

We would like to construct a global streamfunction $\phi : \mathcal{R}(K) \to \mathbb{R}$ of the form,

$$\phi(x) = \left\{ \begin{array}{lll} \phi(\Delta)(x) & \text{if} & x \in \mathcal{R}(\Delta) \ \forall \Delta \in K \end{array} \right. \tag{20}$$

that produces the vector field 19 – for some *global* 0-chain over K. In the following sections, we prove that such a 0-chain exists, and give algorithms for computing it.

1) Existence and Properties:

Definition 4.1: Given an oriented simplicial k-complex K, a vector field (in barycentric coordinates) $v : \mathcal{R}(K) \to \mathbb{R}^3$ *agrees with* a $(k-1)$-chain v if, for each simplex $\Delta \in \Sigma_{k-1}(K)$, the flux of v across $\mathcal{R}(\Delta)$ equals $\langle v, \Delta \rangle$.

Theorem 2: If v is a divergence-free 1-chain over G, then there exists a 0-chain over K that induces a Hamiltonian vector field agreeing with v on the Rips Shadow of K.

Proof : Since the edge flow v is in the cycle space of G and $G \subset \mathcal{G}$, it is in the cycle space of $\mathcal{G}$. Then, by cycle-cut duality, it is in the cut space of $\mathcal{G}^*$, the 1-skeleton of K. Consequently there exists a vector c' in the vertex space of G^*, or equivalently a 0-chain c over K, whose coboundary is v. ■

2) Distributed computation of a global stream function:

a) Method 1: This first method serves to motivate the second. As in section III-A, we are faced with the problem of computing a 0-chain whose boundary best approximates a given 1-chain; hence the global 0-chain $c \in C_0(K)$ can be computed using the gradient descent dynamics,

$$\dot{c} = -\mathcal{L}_0(K)c + \delta_1(K)v \tag{21}$$

where now c is a 0-chain over the vertices of K rather than of G, and the operators $\mathcal{L}_0, \delta_1$ likewise correspond to K. An issue with this approach is that vertices of K are shared by multiple agents – triangles – so an additional synchronization protocol is required for an actual implementation. The next method avoids this messiness, and is much more compatible with the reality that it is triangles, not vertices, that represent agents.

b) Method 2: Within a *single* oriented 2-simplex Δ, the problem of computing 0-chains with given boundaries is straightforward. Let $c \in C_0(\Delta)$ and $v \in C_1(\Delta)$ be 0- and 1-chains over Δ representing streamfunction values and face fluxes, respectively. The problem is that of solving the equation

$$\delta_1^*(\Delta)c = v, \tag{22}$$

where $\delta_1(\Delta)$ has the matrix representation

$$E_3 \triangleq \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} . \tag{23}$$

Since the matrix E_3^T has a 1-dimensional null space spanned by $\mathbf{1}$, there is a family of solutions,

$$c = [\delta_1^*(\Delta)]^\dagger v + \mathbf{1}s \tag{24}$$

where $\mathbf{1} \in C_0(\Delta)$ is the 0-chain that assigns a 1 to each vertex.[6] What this means is that, if a single agent – a triangle – knows its face fluxes, then it can independently determine what the 0-chain over its vertices should be, up to a constant. The coordination problem then is only to determine that scalar s for each triangle – i.e., a 2-chain over K, or, equivalently, a 0-chain over G.

What need the values $s_1, \cdots, s_N$ of the different triangles satisfy? Namely, for two consistently-oriented simplices indexed i and j, sharing a face that is the k^{th} face of simplex i and the l^{th} face of simplex j,

$$s_i - s_j = -\frac{1}{6}\left[D_k(\bar{v}_i) - D_l(\bar{v}_j)\right] \triangleq w_{ij} \tag{25}$$

[6] Note that the matrix representation of the pseudoinverse in (24) is particularly simple: $(E_3^T)^\dagger = \frac{1}{3}E_3$.

where $\bar{v}_j \in \mathbb{R}^3$ is the vector representation of the restriction of the 1-chain v to the simplex j, and $D_k(\bar{v})$ is defined by,

$$[D_0(\bar{v}), D_1(\bar{v}), D_2(\bar{v})]^T = E_3[\bar{v}_0, \bar{v}_1, \bar{v}_2]^T . \tag{26}$$

The skew-symmetric matrix $W = [w_{ij}]_{ij}$ itself encodes a 1-chain over $\mathcal{G}$. The problem has thus been reduced to computing a 0-chain $s \in C_0(G)$ – that with coefficients $s_1, \cdots, s_N$ – given a 1-chain, $w \in C_1(G)$ – whose coefficients come from W – that is to be its boundary. Hence, s can be computed asymptotically by the system,

$$\dot{s} = -\mathcal{L}_0(G)s + \delta_1(G)w \tag{27}$$

much as before.

V. A Combined Algorithm

The two distributed computations described in the previous sections can be performed simultaneously within the network, and stability properties are maintained. This is the subject of the following theorem.

Theorem 3: The ODE

$$\begin{bmatrix} \dot{s} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} -\mathcal{L}_0 & -\delta_1\mathcal{D}\delta_1^* \\ 0 & -\mathcal{L}_0 \end{bmatrix}\begin{bmatrix} s \\ p \end{bmatrix} + \begin{bmatrix} \delta_1\mathcal{D} \\ \delta_1 \end{bmatrix} v \tag{28}$$

(where $\mathcal{D}$ is the linear operator that produces the 1-chain w following (25)), converges asymptotically to a vector in $C_0(G) \times C_0(G)$ that solves the equations (8) and (25).

Proof : The system matrix in (28), which we will refer to as A, is block-upper-triangular, so its eigenvalues are those of its diagonal blocks. Those in turn are graph Laplacians, which are known to be positive semi-definite (see e.g. [20]). Consequently, (28) converges asymptotically to a solution (s, p) provided it has no Jordan blocks larger than 1×1 – a possibility that is ruled out since $\mathrm{image}(\delta_1\mathcal{D}\delta_1^*) \perp \mathrm{null}(\mathcal{L}_0)$. ∎

VI. Example Application: Air Traffic Control

In this section, we explore the use of the the vector fields obtained in the previous sections to direct air traffic throughout an environment, in order to give a flavor for how the preceding theory can be applied. The idea will be to project control inputs onto the divergence-free subspace, using consensus dynamics as in Section III-A; this ensures that aircraft don't "pile up" anywhere.

The aircraft are assumed to inhabit the nodes of the graph G (which correspond to different regions of airspace), and the edges encode which regions are adjacent. Let $M \in C_0(G)$ be a constant scalar field on G representing the capacity of each vertex – i.e., the number of aircraft that can safely share that airspace – and let $m : \mathbb{R}_+ \to C_0(G)$ be a time-varying 0-chain on G representing the number of aircraft at each vertex. Assuming $m(0) = M$ – i.e., that the airspace is initially filled to capacity – we investigate the general problem of directing the aircraft between them while maintaining the safety constraint $m(t) \leq M \ \forall t$.

For this example, we restrict our attention to the case when safety is ensured by maintaining $m(t) = M \ \forall t$ with equality. This is guaranteed by maintaining $\dot{m}(t) = 0 \ \forall t$, which in turn is satisfied by ensuring that the 1-chain describing the air traffic is divergence free. Two such problems naturally arise, from different projection operations, described in the following sections.

3) Least-squares approximation: Suppose an operator wishes to command the air traffic system with a particular reference vector field. One way in which the system can respond to this command is by providing the vector field that approximates the commanded field optimally in a least-squares sense while satisfying the incompressibility constraint. This is precisely the projection problem of Section III-A, so the problem can be directly solved in a distributed fashion by the algorithm (10). We should note that, in order to do this, the operator need only communicate with two nodes per nonzero commanded edge flow; this is encoded by the product $\delta_1(G)v$.

4) Smallest divergence-free flow containing a particular component: A second way in which an operator's commanded vector fields may be used is by finding the smallest (in an l^2 sense) divergence-free flow containing the commanded flow $\bar{v}$ as a component; this is the lowest-energy safe holding pattern that guarantees a certain amount of traffic on specified edges. In this case, we seek a solution to the constrained optimization problem,

$$\arg\min_{v \in C_1(G)} \frac{1}{2}||v||^2 \tag{29}$$

$$\text{s.t.} \quad \delta_1(G)v = 0 \qquad \text{Divergence-free} \tag{30}$$

$$\langle \bar{v}, v \rangle = ||\bar{v}||^2 \qquad \text{Contains component.} \tag{31}$$

Theorem 4: Let P_r denote the l^2 projection operator for the divergence-free subspace of $C_1(G)$, which is computed by (10). Then for all $\bar{v} \neq 0$, the problem (29-31) either has the solution

$$\frac{||\bar{v}||^2}{\langle P_r \bar{v}, \bar{v} \rangle} P_r \bar{v} \tag{32}$$

or is infeasible.

Proof: If $\bar{v} \perp \ker \delta_1(G)$, then (31) requires $v \notin \ker \delta_1(G)$. This contradicts (30), so in this case the problem is infeasible. Hence, without loss of generality, suppose $\bar{v} \notin (\ker \delta_1(G))^{\perp}$.

Any vector $v \in C_1(G)$ can be decomposed uniquely as $v = v_c + v_r$, with $v_c \in (\ker \delta_1(G))^{\perp}$ and $v_r \in \ker \delta_1(G)$. Furthermore, v_c can be uniquely decomposed as $v_c = v_{c,||} + v_{c,\perp}$, with $v_{c,||} \in \operatorname{span} P_r \bar{v}$ and $v_{c,\perp} \in (\operatorname{span} P_r \bar{v})^{\perp}$; hence a unique decomposition $v_c = v_{c,||} + v_{c,\perp} + v_r$ exists, and by the Pythagorean Theorem, $||v_c||^2 = ||v_{c,||}||^2 + ||v_{c,\perp}||^2 + ||v_r||^2$. By (30), $||v_r||^2 = 0$, and by (31), $||v_{c,||}||^2 = ||\bar{v}||^2$. Only $||v_{c,\perp}||^2$ remains free; the quantity $||v_c||^2$ is minimized when $||v_{c,\perp}||^2 = 0$. To summarize, we know that $v = v_{c,||}$ with $v_{c,||} \in \operatorname{span} P_r \bar{v}$, and that $||v_{c,||}||^2 = ||\bar{v}||^2$. Only one element of the vector space $C_1(G)$ satisfies these properties, and that is (32). ■

VII. Numerical Example

To demonstrate the character of the results obtained with these methods, starting from a simplicial 2-complex K with second lower-adjacency graph G, we computed the divergence-free projection of a commanded 1-chain on G with three nonzero elements, and the corresponding 0-chain on K and streamfunction on the Rips Shadow of K; this is shown in Figure 3. Note that the large commanded flow across a single face at the upper right of the complex is propagated through the "jughandle" at the upper right, and that the commanded flows lower in the complex in less confined areas result in pairs of vortices that have mostly local effects; nevertheless, small flows are produced throughout the complex. These qualitative characteristics are typical of the kinds of flows obtained: Where necessary, flows propagate globally, but otherwise most effects of a command are manifested locally. It is the pressure field that propagates this information; essentially, "shocks" are created across the faces where large flows are commanded, and elsewhere the pressure is smoothed across the complex by diffusion. The nonzero commanded flow at the upper right demonstrates this well; it creates a "shock" in the pressure field (black triangle next to white triangle), which diffusion spreads into linearly-decreasing pressure around the upper right "jughandle." Where vortices are produced, the streamfunction exhibits a pair of local extrema – a maximum for a clockwise vortex and a minimum for a counterclockwise one – as can be observed in the left part of the complex. Vehicles then follow level sets of the streamfunction around the environment.

VIII. Concluding Remarks

Given specified input flows, distributed consensus-like algorithms were described that compute divergence-free

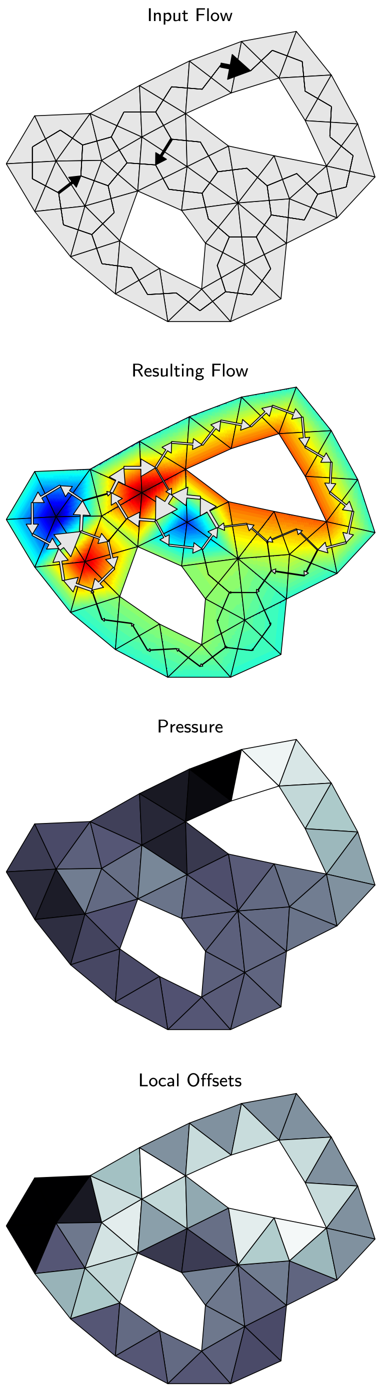

Fig. 3. Computational results are shown. Given a flow as input (first plot; arrow sizes indicate flow magnitudes) on $\mathcal{G}$, a circulant flow on $\mathcal{G}$ and a streamfunction on the Rips Shadow of K are produced (second plot). The Lagrange multipliers for the cycle-space projection (third plot) are a close analogue of pressure in the dynamics of incompressible fluids. The streamfunction is computed locally at each triangle, requiring only the addition of a local offset (fourth plot), which is computed in a distributed fashion.

approximations. Then, these discrete flows were "lifted" to two-dimensional streamfunctions that generate vector fields over the entire Rips Shadows of corresponding simplicial 2-complexes. These flows mimic the behavior of incompressible fluids, and, since vehicles following them will never concentrate in any region, provide a useful method for coordinating collision-free navigation among large numbers of agents.

References

[1] P. Kingston and M. Egerstedt, "Index-free multiagent systems: An eulerian approach," in *Proc. IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, 2010.

[2] Jos Stam, "Stable fluids," in *Proc. SIGGRAPH 99*, August 1999, pp. 121–128.

[3] Antoine McNamara, Adrien Treuille, Zoran Popovic, and Jos Stam, "Fluid control using the adjoint method," in *Proc. SIGGRAPH 2004*, 2004.

[4] L. Pimenta, N. Michael, R. Mesquita, G. Pereira, and V. Kumar, "Control of swarms based on hydrodynamic models.," in *Proc. IEEE International Conference on Robotics and Automation (ICRA).*, May 2008, pp. 1948–1953.

[5] Stephen Waydo and Richard M. Murray, "Vehicle motion planning using stream functions," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2003, pp. 2484–2491.

[6] John M. Lee, *Introduction to Smooth Manifolds*, Springer, 2000.

[7] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 2002, `http://www.math.cornell.edu/~hatcher/AT/ATpage.html`.

[8] James R. Munkres, *Elements of Algebraic Topology*, Addison-Wesley, 1984.

[9] M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden, "Discrete exterior calculus," (preprint, `http://arxiv.org/abs/math/0508341`), 2003.

[10] Yiying Tong, Santiago Lombeyda, Anil N. Hirani, and Mathieu Desbrun, "Discrete multiscale vector field decomposition," 2003.

[11] Leo J. Grady and Jonathan R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*, Springer, 2010.

[12] J. B. Perot and V. Subramanian, "Discrete calculus methods for diffusion," *J. Comput. Phys.*, vol. 224, pp. 59–81, May 2007.

[13] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, "Statistical ranking and combinatorial hodge theory," *Mathematical Programming*, vol. 127, pp. 203–244, 2011.

[14] A. Buffa G. Ferrari-Trecate and M. Gati, "Analysis of coordination in multi-agent systems through partial difference equations. part I: The Laplacian control.," in *16th IFAC World Congress on Automatic Control*, 2005.

[15] D. Zelazo and M. Mesbahi, "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, 2011.

[16] Abubakr Muhammad and Ali Jadbabaie, "Decentralized computation of homology groups in networks by gossip," in *Proc. American Control Conference*, 2007.

[17] Alireza Tahbaz-Salehi and Ali Jadbabaie, "Distributed coverage verification algorithms in sensor networks without location information," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1837–1849, August 2010.

[18] S.G. Loizou and A. Jadbabaie, "Density functions for navigation function based systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 612–617, March 2008.

[19] Richard Diestel, *Graph Theory*, Springer, 2005.

[20] Mehran Mesbahi and Magnus Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.