SURVEY

A New Perspective on Cooperative Control of Multi-Agent Systems through Different Types of Graph Laplacians

Kai Cai[a] and Masaaki Nagahara[b]

[a]Department of Core Informatics, Osaka Metropolitan University, Osaka, Japan;
[b]Institute of Environmental Science and Technology, The University of Kitakyushu, Kitakyushu, Japan

ARTICLE HISTORY
Compiled June 17, 2022

ABSTRACT
In the field of systems and control, many cooperative control problems of multi-agent systems have been actively studied in the past two decades. This article aims to organize extensive existing work on different cooperative control problems into three categories, based on three different types of graph Laplacian matrices involved. A Laplaican matrix is an important representation of graph topology, and depending on the field of its entries, there are three types: ordinary Laplacian (nonnegative diagonal entries and nonpositive off-diagonal entries), signed Laplacian (arbitrary real entries), and complex Laplacian (arbitrary complex entries). Each type of graph Laplacian is useful in modeling and solving a different set of cooperative control problems. In particular, their algebraic properties are fundamental in characterizing stability and performance of the respective solution algorithms. To our best knowledge, organizing the literature on multi-agent cooperative control through the lens of different graph Laplacians is new.

KEYWORDS
Graphs, Laplacians, Multi-Agent Systems, Cooperative Control

1. Introduction

Cooperative control of multi-agent systems has been actively studied in the field of systems and control in the past two decades. Such systems typically consist of a large number of distributed agents, which locally interact with one another such that they jointly pursue a global goal. Research results on cooperative control of multi-agent systems have found wide applications in robotics (swarms of vehicles/drones) [10,34,44], engineering (sensor/power networks) [6,13,38], physics (systems of oscillators) [15,40,45], epidemics (spreading processes) [25,36,49], and social/political science (opinion dynamics) [1,17,50]. The literature has grown in near-intractable volumes, but excellent textbooks (e.g. [3,5,16,33,42]) and surveys (e.g. [9,14,18,35,39]) have kept the content in organized manners.

This article aims to add to the existing surveys a new perspective of organizing an important subset of work on multi-agent systems. This perspective is based on different types of *graph Laplacian matrices*. The conventional Laplacian matrix is defined based

CONTACT Kai Cai. Email: cai@omu.ac.jp

on a nonnegative *adjacency matrix* [4,19], which describes the interaction (graph) topology of the multi-agent system. This type of Laplacian matrix is fundamental in describing the dynamics of a number of multi-agent cooperative control problems including consensus, averaging, synchronization, regulation, flocking, and optimization [7,8,22–24,32,37,41,46,48,51]. The algebraic properties of this type of Laplacian matrix has been found to characterize the stability and performance of the corresponding cooperative control algorithms. These algebraic properties are also closely related to the connectivity properties of the interaction graph.

More recently, two other types of Laplacian matrices have been proposed in designing cooperative control algorithms. One type is defined from a complex-valued (entry-wise) adjacency matrix, and is called *complex Laplacian.* A complex Laplacian matrix has been found useful in solving a class of formation control and localization in the 2D plane [26–28,30,31]. The other type of Laplacian matrix is defined from a general real adjacency matrix which need not be nonnegative. This type of Laplacian matrix is called *signed Lapalcian*, and has been found effective in designing cooperative control algorithms to solve formation control and localization in 3D and higher-dimensional space [11,12,21,29,52]. For both types – complex and signed Laplacian matrices – their algebraic properties are again essential in characterizing the stability and performance of the corresponding cooperative control algorithms. In addition, these algebraic properties are also related to certain connectivity properties of the interaction graph.

The above works based on different types of Laplacian matrices thus provide us with a new angle to overview the relevant literature on multi-agent cooperative control. Although there are many different cooperative control problems in their appearances, they have a few basic points in common. The interaction topology of the agents can be described by graphs, the dynamics of multi-agent systems is hence underlied by Laplacian matrices, and the algebraic properties of these Laplacian matrices dictate stability/performance of the corresponding cooperative control algorithms. These common points therefore allow us to interlink and organize different cooperative control problems and their solutions by different types of Laplacian matrices and the corresponding algebraic properties. To the best of our knowledge, this way of organizing the literature on multi-agent cooperative control is new.

We note that although this new perspective based on different types of Laplacian matrices can effectively organize much existing work on multi-agent cooperative control, it also has a number of limitations. First, dynamics of multi-agent systems that can be described using Laplacian matrices is linear; thus works on systems with nonlinear, discrete, or hybrid dynamics do not fall under this umbrella. Second, since our focus is on the algebraic properties of different Laplacian matrices, only time-invariant cases are considered. This means that the interaction topology of the agents does not vary over time. Third, works on the effect of various communication/sensing issues such as time delay, packet loss, link failure, and quantization are also not considered. Besides the above, this article makes a further simplification that the dynamic models of agents are single integrators. This choice is deliberately made inasmuch as it conveniently allows us to emphasize the role of Laplacian matrices without being distracting by other technical details (related but insignificant for the interest of this article).

Notation

We denote by I_d the identity matrix of size $d \times d$, and $\mathbf{1}$ the vector of all ones, that is, $\mathbf{1} = [1 \ \cdots \ 1]^\top$. A matrix A with (i,j)-th entry a_{ij} is simply denoted by $A = [a_{ij}]$. For two matrices $A = [a_{ij}]$ of size $m \times n$ and B of any size, $A \otimes B$ denotes the Kronecker product defined by

$$A \otimes B := \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{1}$$

For a vector $v = [v_1 \ \cdots \ v_n]^\top$, we denote by $\mathrm{diag}(v)$ the $n \times n$ diagonal matrix with diagonal entries $v_1, \ldots, v_n$. Finally we write $[1, n]$ for the set $\{1, \ldots, n\}$.

2. Definition and Categorization of Laplacian Matrices

Consider a system of interconnected agents modeled by the following differential equation (continuous time t) or difference equation (discrete time k):

$$\dot{x}_i(t) = u_i(t), \quad i \in [1, n], \quad t \geq 0 \tag{2}$$

$$x_i(k+1) = x_i(k) + u_i(k), \quad i \in [1, n], \quad k = 0, 1, \ldots \tag{3}$$

Here x_i is the *state* of agent i, and u_i is its *control input*. In this article, state x_i and control u_i are either real vectors (in $\mathbb{R}^d$, $d \geq 1$) or complex scalars (in $\mathbb{C}$).[1] In (2) the agents' states change continuously, like the change of positions/velocities of robots moving in the 2D/3D space. On the other hand, in (3) the agents' states are updated in discrete steps, like the iterative updates of estimated parameters in sensor networks.

The interconnection of the agents is modeled by a graph. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a node set $\mathcal{V} = \{v_1, \ldots, v_n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ (e.g. [4,19]). Each node represents an agent, and an edge between two nodes denotes the interaction between the two corresponding agents. More precisely, an edge $(v_j, v_i) \in \mathcal{E}$ means that agent i can obtain information from agent j (through sensing or communication). It is assumed (as a convention) that $(v_i, v_i) \notin \mathcal{E}$, namely there is no selfloop edges. In general a graph $\mathcal{G}$ is directed (called *digraph*). If $(v_j, v_i) \in \mathcal{E}$ implies $(v_i, v_j) \in \mathcal{E}$ for all i, j, then the graph is undirected (or bidirectional).

The local pattern of a graph can be described by neighbor sets. For a node v_i, the set of its in-neighbors is $\mathcal{N}_i = \{v_j \mid (v_j, v_i) \in \mathcal{E}\}$, while the set of out-neighbors is $\mathcal{N}_i^o = \{v_j \mid (v_i, v_j) \in \mathcal{E}\}$. If the graph is undirected, then $\mathcal{N}_i = \mathcal{N}_i^o$ for all i.

If for every agent i its control u_i uses information only from $\mathcal{N}_i$, we say that the control is *distributed*. In this article, we consider the distributed control

$$u_i = c_i \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i). \tag{4}$$

Here c_i is the control gain and a_{ij} the weight of interaction that agent i places on the information received from j. Both c_i and a_{ij} are either real or complex non-zero

[1] We consider $x_i, u_i \in \mathbb{C}$ only for the cases of planar formation and localization where these scalar complex variable can provide more compact modeling than treating them as two-dimensional real vectors, i.e. $x_i, u_i \in \mathbb{R}^2$.

scalars.

With weight a_{ij} associated to edge $(v_j, v_i) \in \mathcal{E}$, graph $\mathcal{G}$ is called a *weighted graph* [5]. Note that $a_{ij} \neq 0$ if and only if $(v_j, v_i) \in \mathcal{E}$. The *adjacency matrix* [4,19] of a weighted graph $\mathcal{G}$ is an $n \times n$ matrix $A = [a_{ij}]$. Since $(v_i, v_i) \notin \mathcal{E}$ for all i, the diagonal entries of A are 0. If A is symmetric (i.e. $A = A^\top$), then $\mathcal{G}$ is said to be *weight balanced* [5]. In this article, we consider three types of adjacency matrices depending on the field of their entries.

- If $a_{ij} \geq 0$, A is a nonnegative matrix.
- If $a_{ij} \in \mathbb{R}$, A is an arbitrary real matrix.
- If $a_{ij} \in \mathbb{C}$, A is an arbitrary complex matrix.

Let A be the adjacent matrix of $\mathcal{G}$. Then $D := \mathrm{diag}(A\mathbf{1})$ is the degree matrix, where $\mathbf{1}$ is the vector of all ones. The *Laplacian matrix* [4,5,19] of a weighted digraph $\mathcal{G}$ is $L := D - A$. In general L is not symmetric; it is symmetric if and only if $\mathcal{G}$ is weight balanced. The rationale of naming this matrix L "Laplacian" is due to its relation to the *Laplace differential operator* [2,43], explained in the Appendix.

By definition $L\mathbf{1} = 0$; namely each row of L sums to zero. Thus 0 is an eigenvalue of L, with a corresponding eigenvector $\mathbf{1}$. This is a basic algebraic property shared by all types of graph Laplacians introduced below.

According to the three types of adjacency matrices, we distinguish three types of Laplacian matrices. Each type is useful for a set of cooperative control problems.

- If A is nonnegative, then L has nonnegative diagonal entries and nonpositive off-diagonal entries. This L is the conventional Laplacian matrix, and in this article we refer to it as *ordinary Laplacian* (OL).[2]
- If A is (arbitrary) real, then L is called *signed Laplacian* (SL).
- If A is (arbitrary) complex, then L is called *complex Laplacian* (CL).

With a Laplacian matrix L and letting $x := [x_1^\top \ \cdots \ x_n^\top]$, $C := \mathrm{diag}(c_1, \ldots, c_n)$, equations (2)–(4) can be written in a vector-matrix form in continuous time

$$\dot{x}(t) = (-(CL) \otimes I_d)x(t) \tag{5}$$

where $\otimes$ is the Kronecker product. In particular, for a scalar case (i.e., $x_1, \ldots, x_n$ are real or complex scalars), it is simply written as

$$\dot{x}(t) = -CLx(t). \tag{6}$$

Similarly, in discrete time, we have

$$x(k+1) = ((I_n - CL) \otimes I_d)x(k) \tag{7}$$

for the d-dimensional case, and

$$x(k+1) = (I_n - CL)x(k) \tag{8}$$

for a scalar case.

[2]The term "ordinary Laplacian" is not standard in the literature, but convenient for us to distinguish with the other two types of Laplacian matrices in this article.

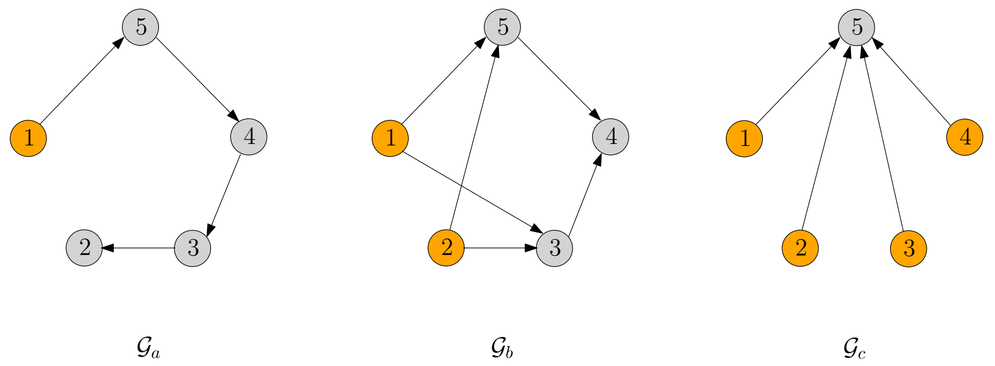
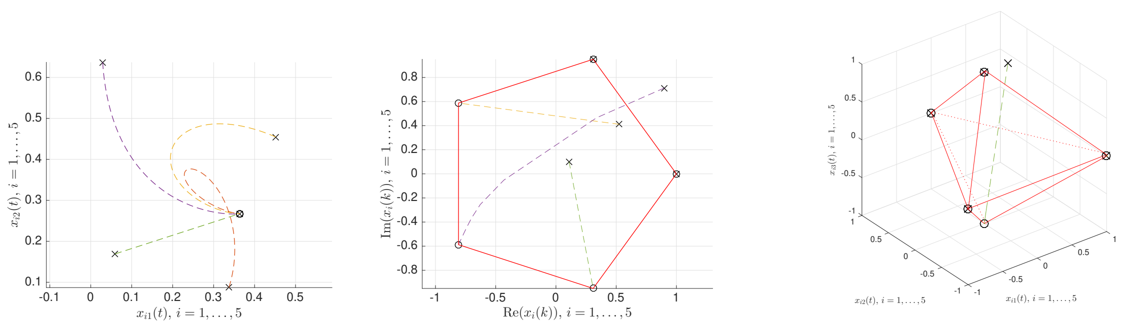


Figure 1. Example 2.1: graphs



(a) Consensus ($\mathcal{G}_a$ and L_a) (b) Localization ($\mathcal{G}_b$ and L_b) (c) Formation ($\mathcal{G}_c$ and L_c)

Figure 2. Example 2.1: simulations (x denotes initial state while $\circ$ final state)

Example 2.1. We provide an example to illustrate that different types of Laplacian matrices can generate different cooperative behaviors. Consider a system of five agents whose interconnection is represented by three different graphs in Figure 1. For $\mathcal{G}_a$ consider the following OL (nonnegative diagonal entries and nonpositive off-diagonal entries):

$$L_a = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

With L_a, $C = I_5$, $d = 2$, a simulation of equation (5)

$$\dot{x} = (-(CL_a) \otimes I_2)x$$

is shown in Figure 2(a). Observe that in the 2D plane, the five agents achieve consensus (i.e. reaching the same point).

Next for $\mathcal{G}_b$ consider the following CL (complex entries):

$$L_b = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 1.309 - 0.9511\mathrm{j} & -0.309 + 0.9511\mathrm{j} & 0 & 0 \\ 0 & 0 & -1 & 1.309 - 0.9511\mathrm{j} & -0.309 + 0.9511\mathrm{j} \\ -1 & 0.5 - 0.3633\mathrm{j} & 0 & 0 & 0.5 + 0.3633\mathrm{j} \end{bmatrix}.$$

With L_b, $C = \mathrm{diag}(0, 0, -0.309 - 0.9511\mathrm{j}, 0.4045 + 0.2939\mathrm{j}, 0.4045 - 0.2939\mathrm{j})$, a simulation of equation (8)

$$x(k+1) = (I_5 - CL_b)x(k)$$

is shown in Figure 2(b). In 2D, agents 3,4,5 achieve localization (i.e. determining their positions in the global reference frame) given the positions of (anchor) agents 1,2.

Finally for $\mathcal{G}_c$ consider the following SL (real entries):

$$L_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -1 & 2 & 2 \end{bmatrix}.$$

With L_c, $C = \mathrm{diag}(0, 0, 0, 0, 0.2)$, $d = 3$, a simulation of equation (5)

$$\dot{x} = (-(CL_c) \otimes I_3)x$$

is shown in Figure 2(c). In 3D, the five agents achieve a hexahedron formation.

The above examples showcase the possibilities of using different types of Laplacian matrices to generate different cooperative behaviors. In the subsequent sections, we examine each type of Laplacian matrices in order. For each type, among multiple cooperative control problems that may be solved using that type of Laplacian, we present $1 \sim 2$ representative problems. Then we introduce key algebraic properties of the respective Laplacian matrices that guarantee achieving the desired cooperative behaviors. Note that we choose the representative problems with the purpose of emphasizing the differences between different types of Laplacians; we do not, however, intend to exhaust all problems that can be resolved by each type of Laplacian.

3. Ordinary Laplacian

OL is first and most widely used in cooperative control of multi-agent systems (consensus, averaging, synchronization, regulation, flocking, and optimization [7,8,22–24,32,37,41,46,48,51]). The basic problems are consensus (or agreement, rendezvous) and averaging, which are later extended to consensus-based estimation, synchronization, regulation, and averaging-based optimization. In this section, we choose to present these two basic problems – consensus and averaging – and both the algebraic and the graphical conditions that characterize their solutions.

Problem 3.1 (**Continuous-time consensus**). *Consider a system of n agents as in (2) with $x_i, u_i \in \mathbb{R}^d$, which are interconnected through a graph $\mathcal{G}$. Design a distributed control (4) such that for every $i \in [1, n]$ and every $x_i(0) \in \mathbb{R}^d$ there exists $x^* \in \mathbb{R}^d$ such that*

$$\lim_{t\to\infty} x_i(t) = x^*.$$

In terms of vectors/matrices and the differential equation $\dot{x}(t) = (-(CL) \otimes I_d)x(t)$ in (5), this means that for any initial state $x(0) \in \mathbb{R}^{nd}$ there exists $x^* \in \mathbb{R}^d$ such that

$$\lim_{t\to\infty} x(t) = \mathbf{1} \otimes x^*.$$

Problem 3.2 (**Discrete-time averaging**). *Consider a system of n agents as in (3) with $x_i, u_i \in \mathbb{R}^d$, which are interconnected through a graph $\mathcal{G}$. Design a distributed control (4) such that for every $i \in [1, n]$ and every $x_i(0) \in \mathbb{R}^d$*

$$\lim_{k\to\infty} x_i(k) = \frac{1}{n}\sum_{i=1}^{n} x_i(0).$$

In terms of vectors/matrices and the difference equation $x(k+1) = ((I_n - CL) \otimes I_d)x(k)$ in (7), this means that for any initial state $x(0) \in \mathbb{R}^{nd}$

$$\lim_{k\to\infty} x(k) = \mathbf{1} \otimes \frac{1}{n}\sum_{i=1}^{n} x_i(0).$$

Both consensus and averaging problems can be formulated in continuous-time or discrete-time (e.g. [5,42]). The above choices are for convenience and the other versions are similar.

Comparing consensus and averaging, both problems require all the agents' states to converge to a common vector (aka. *consensus vector*). While in consensus this vector is not specified, in averaging it is the (element-wise) average of the initial states. Consensus has found applications in multi-robot rendezvous and flocking [23,37,41], while averaging is key to solution of distributed optimization problems [47,51] which apply to multi-robot motion coordination and planning [20]. Averaging has also been applied to distributed estimation, load balancing, and opinion dynamics [5].

The key to the solvability of these two problems is certain algebraic properties of the OL L, in particular its eigenstructure. The latter is in turn determined by the *connectivity* of the graph $\mathcal{G}$.

We introduce several graph notions related to connectivity, followed by a central result linking graph connectivity with an algebraic property of OL. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a path is a sequence of nodes

$$v_1 v_2 \cdots v_l \quad (l \geq 1)$$

such that $(v_i, v_{i+1}) \in \mathcal{E}$ for every $i = 1, 2, \ldots, l-1$. The path is said to be from v_1 to v_l. Let $u, v \in \mathcal{V}$ be two nodes of $\mathcal{G}$. We say that v is *reachable* from u if there is a path from u to v. Every node v is assumed trivially reachable from itself. A node $r \in \mathcal{V}$ is called a *root* if every node in $\mathcal{V} \setminus \{r\}$ is reachable from r. For example, consider $\mathcal{G}_a$ in Figure 1; node 1 is a root and the other nodes are not.

The following result is fundamental (e.g. [5,16,42]).

Proposition 3.1. *Let $\mathcal{G}$ be a weighted graph with n nodes and L the corresponding OL matrix. Then* $\mathrm{rank}(L) = n - 1$ *if and only if $\mathcal{G}$ contains a root.*

Proposition 3.1 asserts that existence of a root is necessary and sufficient for that 0 is a simple eigenvalue of L; that is, the null space of L is exactly one-dimensional spanned by the eigenvector $\mathbf{1}$ of the eigenvalue 0. For example, the OL L_a in Example 2.1 satisfies $\mathrm{rank}(L_a) = 4$ and the eigenvalue 0 is simple. Moreover, by Gershgorin disc theorem, all the other (nonzero) eigenvalues of L have positive real parts. This result leads to the solutions to the consensus and averaging problems.

Theorem 3.1 (Solution to consensus). *There exists a control gain matrix C such that Problem 3.1 is solvable if and only if $\mathcal{G}$ contains a root. Moreover, the consensus vector $x^* = (\mathbf{1} \otimes w)^\top x(0)$, where w is a left eigenvector of L with respect to the (simple) eigenvalue* 0.

Theorem 3.2 (Solution to averaging). *Suppose that $\sum_{j \in \mathcal{N}_i} a_{ij} < 1$ for all $i \in [1, n]$. There exists a control gain matrix C such that Problem 3.2 is solvable if and only if $\mathcal{G}$ is weight-balanced (i.e. $L = L^\top$) and every node $v_i \in \mathcal{V}$ is a root.*

Several remarks on the above results are in order.

- Theorem 3.1 is adapted from e.g. [33,42], and Theorem 3.2 from e.g. [5].
- In both Theorems 3.1 and 3.2, a solution control gain matrix is $C = I_n$. The second statement in Theorem 3.1 shows that the consensus vector is a *weighted average* of the initial state.
- The additional assumption in Theorem 3.2 ($\sum_{j \in \mathcal{N}_i} a_{ij} < 1$ for all $i \in [1, n]$) ensures that the matrix $I_n - L$ is nonnegative (and thus row stochastic). Moreover, for weight-balanced graphs, the matrix $I_n - L$ is also column stochastic (hence doubly stochastic).
- For undirected graphs, existence of a root is in fact equivalent to that all nodes are roots (indeed the graph is called *connected*).
- For directed graphs, when all nodes are roots, the graph is said to be *strongly connected*. Indeed, strongly connected and weight-balanced are necessary and sufficient for the weight vector $w = \frac{1}{n}\mathbf{1}$ in Theorem 3.1.

4. Complex Laplacian

CL is revealed to be useful in formulating and solving a class of localization and formation control problems in 2D [26–28,30,31]. Both the localization and the formation control problems bear certain similarities, and introducing both would be repetitive. In this section we choose to introduce the 2D localization problem (and the formation control problem in higher dimensions will be introduced in the subsequent section). Then we present both the algebraic and the graphical conditions that characterize its solution.

In 2D localization, a system of n agents are stationary in the plane with a global reference frame Σ. Among the n agents, two agents are called *anchors* whose positions in Σ are known, while the rest called *free agents* need to determine their positions based only on their local reference frames that are generally not aligned with Σ. In describing this 2D localization problem below, we denote the position of agent i at time k by a complex number $x_i(k)$, so that the real part $\mathrm{Re}(x_i(k))$ and the imaginary part $\mathrm{Im}(x_i(k))$ are the two perpendicular components of agent i's position in the global frame Σ. Using this complex representation of agents' positions in 2D is not only more compact (than the real-vector counterpart), but gives rise to the use of CL (the focus of this section) in describing the multi-agent system's dynamics.

Problem 4.1 (Discrete-time 2D localization). *Consider a system of n agents interconnected through a graph $\mathcal{G}$. The first 2 agents are anchors whose positions $x_a^* = [x_1^* \ x_2^*]^\top \in \mathbb{C}^2$ in the global reference frame Σ are known, while the rest $n-2$ agents are free whose positions $x_f^* = [x_3^* \ \cdots \ x_n^*]^\top \in \mathbb{C}^{n-2}$ in the global reference frame Σ are unknown. Consider that each agent i uses a scheme for its position estimation as in (3) where $x_i \in \mathbb{C}$ is the estimated position and $u_i \in \mathbb{C}$ is based on its local reference frame Σ_i (which are not aligned with Σ in general). Design a distributed control (4) such that $x(k+1) = (I_n - CL)x(k)$ in (8) satisfies: for any initial state $x(0) \in \mathbb{C}^n$*

$$\lim_{k\to\infty} x(k) = x^* := \begin{bmatrix} x_a^* \\ x_f^* \end{bmatrix}.$$

To solve the above 2D localization problem, a certain algebraic property of the CL L is key. The latter is in turn determined by a graphical notion that generalizes that of root. Let $v_1, v_2 \in \mathcal{V}$ and write $\mathcal{R} = \{v_1, v_2\}$. For another node $v \in \mathcal{V} \setminus \mathcal{R}$, we say that v is 2-*reachable* from $\mathcal{R}$ if v is reachable from v_1 or v_2 after removing an arbitrary node except for v itself. Intuitively 2-reachability of v requires that there be two independent paths (sharing no nodes) from $\mathcal{R}$ to v. Call $\mathcal{R} = \{v_1, v_2\}$ a 2-root set if every node in $\mathcal{V} \setminus \mathcal{R}$ is 2-reachable from $\mathcal{R}$. For example, consider $\mathcal{G}_b$ in Figure 1; the set $\{1, 2\}$ is the only 2-root set.

Existence of a 2-root set turns out to be sufficient to ensure that the rank of CL L is at least $n-2$ (in a generic sense), as asserted below (adapted from [31, Lemma 3.1].

Proposition 4.1. *Let $\mathcal{G}$ be a weighted graph with n nodes and L the corresponding CL matrix. If $\mathcal{G}$ contains a 2-root set, then $\mathrm{rank}(L) \geq n-2$ for L with almost all complex entries.*

Return to the 2D localization problem. Since the two anchor agents do not need to update their states, we set $x_1(k) = x_1(0)$ and $x_2(k) = x_2(0)$ for all $k \geq 0$. Hence the first two rows of the CL L are zeros. This means that $\mathrm{rank}(L) \leq n-2$. Together with Proposition 4.1, we have $\mathrm{rank}(L) = n-2$, namely the null space of L are exactly two-dimensional. One basis of this null space is $\mathbf{1}$ by definition. The other basis is in fact $x^* = [(x_a^*)^\top \ (x_f^*)^\top]^\top$, the position vector in the global reference frame Σ, provided that x^* is generic (i.e. not satisfying any linear algebraic equation with rational coefficients). For example, the CL L_b in Example 2.1 satisfies $\mathrm{rank}(L_a) = 3$ and the two bases for the null space are $\mathbf{1}$ and

$$\xi = [1 \ \ \mathrm{e}^{\frac{\pi}{5}\mathrm{j}} \ \ \mathrm{e}^{\frac{2\pi}{5}\mathrm{j}} \ \ \mathrm{e}^{\frac{3\pi}{5}\mathrm{j}} \ \ \mathrm{e}^{\frac{4\pi}{5}\mathrm{j}}]^\top.$$

The vector ξ is a shape of a regular pentagon, which is generic in $\mathbb{C}$.

Below we present the result of solving the 2D localization problem (adapted from [27, Theorem 1]).

Theorem 4.1. *Suppose that the global position vector* $x^* = [x_1^* \; x_2^* \; \cdots \; x_n^*]^\top$ *is generic in* $\mathbb{C}$. *There exists a control gain matrix* C *such that Problem 4.1 is solvable if* $\mathcal{G}$ *contains a 2-root set* $\{v_1, v_2\}$ *and* $\mathcal{N}_1 = \mathcal{N}_2 = \emptyset$.

We remark that the condition $\mathcal{N}_1 = \mathcal{N}_2 = \emptyset$ implies that the two anchor agents do not receive information and thereby do not update their states, i.e. $x_1(k) = x_1(0)$ and $x_2(k) = x_2(0)$ for all $k \geq 0$. This condition is, by contrast, not required in the similar formation control problem [31], because there the two root agents, playing the role of leaders, do not know their positions in the global frame Σ and therefore generally can and need to update their states.

Unlike the case of OL where the stability of the nonzero eigenvlaues is determined by the Gershgorin disc theorem, the nonzero eigenvalues of CL are generally unstable. Thus one needs to design a suitable control gain matrix C to stabilize CL. Fortunately, such a stabilizing matrix C always exist for CL of graphs containing a 2-root set.

5. Signed Laplacian

While CL in the preceding section is instrumental for localization and formation control in 2D, it is not applicable to similar problems in higher dimensions. For a class of localization, formation control, and formation maneuvering problems in higher dimensions, SL instead provides a useful approach for their formulations and solutions [11,12,21,29,52].

Like in Section 4, these localization, formation control, and formation maneuvering problems bear certain similarities, and thus introducing one would be representative. In this section we choose to introduce a d-dimensional ($d \geq 3$) affine formation control problem, and present both the algebraic and the graphical conditions that characterize its solution.

In this formation control problem, a system of n agents move in a d-dimensional space and is tasked to form a formation shape that is 'affine' to a given desired configuration $\xi \in \mathbb{R}^{nd}$: namely the formed formation can be obtained from ξ by translation, rotation, and scalings in each of the d dimensions.

Problem 5.1 (Continuous-time d-dimensional affine formation). *Consider a system of n agents as in (2) with $x_i, u_i \in \mathbb{R}^d$, which are interconnected through a graph $\mathcal{G}$. Let $\xi \in \mathbb{R}^{nd}$ be the desired formation shape (or configuration), and define the family of all affine formations of ξ:*

$$\mathcal{A}(\xi) := \{\xi' \in \mathbb{R}^{nd} : (\exists M \in \mathbb{R}^{d \times d}, \exists m \in \mathbb{R}^d)\xi' = (I_n \otimes M)\xi + \mathbf{1}_n \otimes m\}.$$

Design a distributed control (4) such that $\dot{x}(t) = (-(CL) \otimes I_d)x(t)$ in (5) satisfies: for any initial state $x(0) \in \mathbb{R}^{nd}$ there exists $\xi' \in \mathcal{A}(\xi)$ such that

$$\lim_{t \to \infty} x(t) = \xi'.$$

To solve the above problem, a certain algebraic property of the SL L is key. The latter is in turn determined by a key graphical notion that generalizes that of 2-root set. Let $\mathcal{R} \subseteq \mathcal{V}$ be a subset of $d+1$ nodes, i.e. $|\mathcal{R}| = d+1$. For an arbitrary node $v \in \mathcal{V} \setminus \mathcal{R}$, we say that v is $(d+1)$-*reachable* from $\mathcal{R}$ if v is reachable from a node after removing arbitrary d nodes except for v itself. Intuitively $(d+1)$-reachability of

v requires that there exist d independent paths (sharing no nodes) from $\mathcal{R}$ to v. Call $\mathcal{R}$ a $(d+1)$-root set if every node in $\mathcal{V} \setminus \mathcal{R}$ is $(d+1)$-reachable from $\mathcal{R}$. For example, consider $\mathcal{G}_c$ in Figure 1; $\{1, 2, 3, 4\}$ is a 4-root set.

Existence of a $(d+1)$-root set turns out to be sufficient to ensure that the rank of SL L is at least $n - d - 1$ (in a generic sense), as stated below (adapted from [29, Lemma 4.1]).

Proposition 5.1. *Let $\mathcal{G}$ be a weighted graph with n nodes and L the corresponding SL matrix. If $\mathcal{G}$ contains a $(d+1)$-root set, then rank$(L) \geq n-d-1$ for L with almost all real entries.*

Proposition 5.1 provides a sufficient condition to ensure $\mathrm{rank}(L) \geq n-d-1$. On the other hand, it can be verified that $\ker(L \otimes I_d) \supseteq \mathcal{A}(\xi)$, which implies $\mathrm{rank}(L) \leq n-d-1$ (since the dimension of $\mathcal{A}(\xi)$ is $d+1$). Hence $\mathrm{rank}(L) = n-d-1$ after all. In fact $\ker(L \otimes I_d) = \mathcal{A}(\xi)$. For example, the SL L_c in Example 2.1 satisfies $\mathrm{rank}(L_c) = 1$ for the affine formation control problem in $\mathbb{R}^3$ ($d = 3$).

We are ready to present the result of solving the d-dimensional affine control problem (adapted from [29, Theorem 4.1]).

Theorem 5.1. *Suppose that the desired formation shape ξ is generic in $\mathbb{R}^d$. There exists a control gain matrix C such that Problem 5.1 is solvable if $\mathcal{G}$ contains a $(d+1)$-root set.*

Note that unlike in Theorem 4.1 it is required that the roots have no neighbors, here the roots are allowed to have neighbors. This means that the root agents in affine formation control can receive information from their neighbors (if there are any) and accordingly update their states (positions in the d-dimensional space).

Regarding the stability of the nonzero eigenvalues of SL, the situation is similar to that of CL. Namely, the nonzero eigenvalues of SL are generally unstable, and one needs to properly design a control gain matrix C to stabilize CL. Fortunately again, such a stabilizing matrix C always exist for SL of graphs containing a $(d+1)$-root set.

6. Summary

In this article we introduced three different types of graph Laplacian matrices based on the field of their entries. For each type of Laplacian matrix, we introduced their use in different multi-agent cooperative control problems, as well as their characteristic algebraic conditions that are key to solve the corresponding problems. Moreover, these algebraic conditions have nice correspondences to conditions on graph connectivities. The content is summarized in the following table.

Table 1. Content summary

	OL	CL	SL
Problems	consensus/averaging	localization in 2D	formation in d-dimension
Algebraic conditions	$\mathrm{rank}(L) = n-1$	$\mathrm{rank}(L) \geq n-2$	$\mathrm{rank}(L) \geq n-d-1$
Graphical conditions	root	2-root set	$(d+1)$-root set

Funding

This work was supported in part by JSPS KAKENHI Grant nos. 21H04875, 20H02172 and 20K21008, and the 2021 Osaka City University Strategic Research Grant to Encourage Applicants for Upper Level KAKENHI.

References

[1] C. Altafini and G. Lini. Predictable dynamics of opinion forming for networks with antagonistic interactions. *IEEE Trans. Autom. Control*, 60(2):342–357, 2015.

[2] L. Alvarez. Images and PDE's. In *Proc. Int. Conf. on Analysis and Optimization of Systems*, pages 3–14, 1996.

[3] H. Bai, M. Arcak, and J. Wen. *Cooperative Control Design.* 2011.

[4] R. B. Bapat. *Graphs and Matrices.* Springer, 2010.

[5] F. Bullo. *Lectures on Network Systems.* 2022.

[6] K. Cai, B. D. O. Anderson, C. Yu, and G. Mao. Local average consensus in distributed measurement of spatial-temporal varying parameters: 1d case. *Automatica*, 52(2):135–145, 2015.

[7] K. Cai and H. Ishii. Quantized consensus and averaging on gossip digraphs. *IEEE Trans. Autom. Control*, 56(9):2087–2100, 2011.

[8] K. Cai and H. Ishii. Average consensus on general strongly connected digraphs. *Automatica*, 48(11):2750–2761, 2012.

[9] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans Industrial informatics*, 9(1):427–438, 2013.

[10] J. Chen, H. Wang, M. Rubenstein, and H. Kress-Gazit. Automatic control synthesis for swarm robots from formation and location-based high-level specifications. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 8027–8034, 2020.

[11] W. Chen, J. Liu, Y. Chen, S. Z. Khong, D. Wang, T. Basar, L. Qiu, and K. H. Johansson. Characterizing the positive semidefiniteness of signed laplacians via effective resistances. In *Proc. IEEE Conf. on Decision and Control*, pages 985–990, 2016.

[12] W. Chen, D. Wang, J. Liu, T. Basar, and L. Qiu. On spectral properties of signed laplacians for undirected graphs. In *Proc. IEEE Conf. on Decision and Control*, pages 1999–2002, 2017.

[13] F. Dorfler and F. Bullo. Synchronization and transient stability in power networks and non-uniform kuramoto oscillators. *SIAM J. Control and Optimization*, 50(3):1616–1642, 2010.

[14] F. Dorfler and F. Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539–1564, 2014.

[15] F. Dorfler, M. Chertkov, and F. Bullo. Synchronization in complex oscillator networks and smart grids. *Proc. National Academy of Sciences*, 110(6):2005–2010, 2013.

[16] B. A. Francis and M. Maggiore. *Flocking and Rendezvous in Distributed Robotics.* 2016.

[17] N. E. Friedkin, P. Jia, and F. Bullo. A theory of the evolution of social power: Natural trajectories of interpersonal inuence systems along issue sequences. *Sociological Science*, 3:444–472, 2016.

[18] F. Garin and L. Schenato. A survey on distributed estimation and control applications using linear consensus algorithms. In A. Bemporad, M. Heemels, and M. Johansson, editors, *Networked Control Systems, Lecture Notes in Control and Information Sciences, Springer*, pages 75–107, 2010.

[19] C. Godsil and G. Royle. *Algebraic Graph Theory.* Springer, 2000.

[20] T. Halsted, O. Shorinwa, J. Yu, and M. Schwager. A survey of distributed optimization methods for multi-robot systems. *arXiv:2103.12840*, 2021.

[21] T. Han, Z. Lin, R. Zheng, Z. Han, and H. Zhang. A barycentric coordinate based approach to three-dimensional distributed localization for wireless sensor networks. In *Proc. IEEE Int. Conf. on Control & Automation*, pages 600–605, 2017.

[22] T. Ikeda, M. Nagahara, and K. Kashima. Maximum hands-off distributed control for consensus of multiagent systems with sampled-data state observation. *IEEE Trans. Control of Network Systems*, 6(2):852–862, 2019.

[23] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents sing nearest neighbor rules. *IEEE Trans. Autom. Control*, 48(6):988–1001, 2003.

[24] S. Kawamura, K. Cai, and M. Kishida. Distributed output regulation of heterogeneous uncertain linear agents. *Automatica*, 2020.

[25] A. Khanafer, T. Basar, and B. Gharesifard. Stability properties of infection diffusion dynamics over directed networks. In *Proc. IEEE Conf. on Decision and Control*, page 6215–6220, 2014.

[26] Z. Lin, W. Ding, G. Yan, C. Yu, and A. Giua. Leader-follower formation via complex laplacian. *Automatica*, 49:1900–1906, 2013.

[27] Z. Lin, M. Fu, and Y. Diao. Distributed self localization for relative position sensing networks in 2d space. *IEEE Trans. Sig. Proc.*, 63(14):3751–3761, 2015.

[28] Z. Lin, T. Han, R. Zheng, and M. Fu. Distributed localization for 2-d sensor networks with bearing-only measurements under switching topologies. *IEEE Trans. Sig. Proc.*, 64(23):6345–6359, 2016.

[29] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han. Necessary and sufficient graphical conditions for affine formation control. *IEEE Trans. Autom. Control*, 61(10):2877–2891, 2016.

[30] Z. Lin, L. Wang, Z. Han, and M. Fu. Distributed formation control of multi-agent systems using complex laplacian. *IEEE Trans. Autom. Control*, 59(7):1765–1777, 2014.

[31] Z. Lin, L. Wang, Z. Han, and M. Fu. A graph laplacian approach to coordinate-free formation stabilization for directed networks. *IEEE Trans. Autom. Control*, 61(5):1269–1280, 2016.

[32] J. Lunze. Synchronization of heterogeneous agents. *IEEE Trans. Autom. Control*, 57(11):2885–2890, 2012.

[33] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. 2010.

[34] T. Motoyama and K. Cai. Top-down synthesis of multi-agent formation control: an eigenstructure assignment based approach. *IEEE Trans. Control of Network Systems*, 6(4):1404–1414, 2019.

[35] K. K. Oh, M. C. Park, and H. S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[36] K. Oles, E. Gudowska-Nowak, and A. Kleczkowski. Efficient control of epidemics spreading on networks: Balance between treatment and recovery. *PLoS ONE*, 8(6):e63813, 2013.

[37] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control*, 51(3):401–420, 2006.

[38] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Proc. IEEE Conf. on Decision and Control*, pages 5492–5498, 2007.

[39] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE*, 95(1):215–233, 2007.

[40] A. Pikovsky and M. Rosenblum. Dynamics of heterogeneous oscillator ensembles in terms of collective variables. *Physica D: Nonlinear Phenomena*, 240(9):872–881, 2011.

[41] W. Ren. Synchronization of coupled harmonic oscillators with local interaction. *Automatica*, 44(12):3195–3200, 2008.

[42] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. 2008.

[43] J. C. Russ. *The Image Processing Handbook*. Boca Raton, FL: CRC, 1995.

[44] M. Saska, V. Vonasek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *J. Intelligent & Robotic Systems*, 84:469–492, 2016.

[45] L. Scardovi and R. Sepulchre. Synchronization in networks of identical linear systems.

Automatica, 45(11):2557–2562, 2008.
[46] P. Wieland, R. Sepulchre, and F. Allgower. An internal model principle is necessary and sufficient for linear output synchronization. *Automatica*, 47(5):1068–1074, 2011.
[47] R. Xin and U. Khan. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, 2(3):315–320, 2018.
[48] Y. Xu, T. Han, K. Cai, Z. Lin, G. Yan, and M. Fu. A distributed algorithm for resource allocation over dynamic digraphs. *IEEE Trans. Sig. Proc.*, 65(10):2600–2612, 2017.
[49] M. Ye, J. Liu, B. D. O. Anderson, and M. Cao. Applications of the Poincare-Hopf theorem: Epidemic models and Lotka-Volterra systems. *IEEE Trans. Autom. Control*, 2021.
[50] M. Ye, J. Liu, B. D. O. Anderson, C. Yu, and T. Basar. Evolution of social power in social networks with dynamic topology. *IEEE Trans. Autom. Control*, 63(11):3793–3808, 2018.
[51] J. Zhang, K. You, and K. Cai. Distributed conjugate gradient tracking for resource allocation in unbalanced networks. *IEEE Trans. Sig. Proc.*, 68:2186–2198, 2020.
[52] S. Zhao. Affine formation maneuver control of multiagent systems. *IEEE Trans. Autom. Control*, 63(12):4140–4155, 2018.

Appendix: Relation to Laplace Differential Operator

In this section, we show the relation between the Laplacian matrix and the Laplace differential operator [2,43]. We will see the graph Laplacian is a result of discretization of Laplace differential operator. For simplicity, we consider the operator in $\mathbb{R}^2$:

$$\Delta := \left(\frac{\partial}{\partial x_1}\right)^2 + \left(\frac{\partial}{\partial x_2}\right)^2, \tag{1}$$

but the relation explained below is still true for any dimensional spaces.

First, we consider an open set $\Omega \subset \mathbb{R}^2$, and a function $\phi(x, y)$ in this domain. Then we discretize Ω by dividing it into sub-domains with a square gird with step size h, as shown in Figure 1.

Then, we approximate the value of $\Delta\phi$ on the grid points by the finite difference method. For this, we name the grid points as $v_1, v_2, \ldots, v_n$, where n is the number of grid points. As shown in Figure 1, a point v_i is connected to four grid points located at (x_{i-1}, y_i), (x_{i+1}, y_i), (x_i, y_{i-1}), and (x_i, y_{i+1}). By the graph notation, these four points form $\mathcal{N}_i$. By using the values of ϕ at these points, we discretize the Laplace differential operator using finite difference method as

$$\begin{aligned}(\Delta\phi)(x_i, y_i) &\approx \frac{\phi(x_{i-1}, y_i) - 2\phi(x_i, y_i) + \phi(x_{i+1}, y_i)}{h^2} + \frac{\phi(x_i, y_{i-1}) - 2\phi(x_i, y_i) + \phi(x_i, y_{i+1})}{h^2} \\ &= -\frac{1}{h^2}\sum_{j\in\mathcal{N}_i}(\phi_i - \phi_j),\end{aligned} \tag{2}$$

where ϕ_j denotes the value of ϕ at grid point v_j. If we define $\psi := [\phi_1 \ \cdots \ \phi_n]^\top$, it is easily shown that

$$(L\psi)_i = \sum_{j\in\mathcal{N}_i}(\phi_i - \phi_j), \tag{3}$$

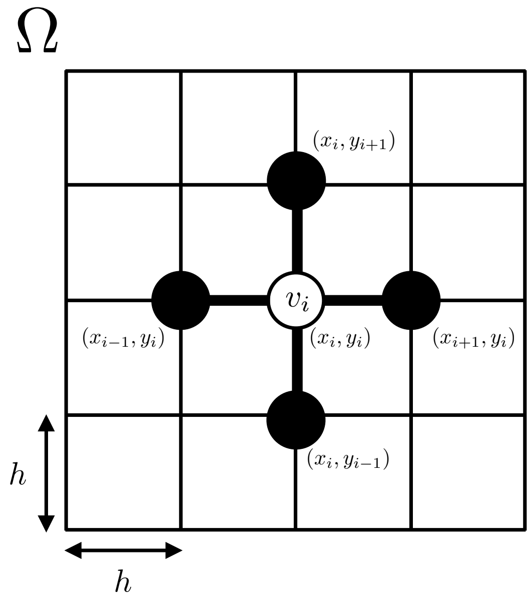


Figure 1. Space discretization with a square grid with step size h. The grid point named v_i located at (x_i, y_i) in Ω is connected to four grid points colored in black.

where $(L\psi)_i$ is the i-th entry of vector $L\psi$. Therefore, vector $-CL\psi$ with $C := \mathrm{diag}(\mathbf{1}/h^2)$ gives approximated values of $\Delta\phi$ at grid points $v_1, \dots, v_n$ as in (2) and (3).

The Laplace differential operator appears in the heat equation

$$\frac{\partial \phi}{\partial t} = \Delta \phi, \tag{4}$$

where $\phi(x, y, t)$ is the temperature at position (x, y) at time t. Define $\Delta_d := -CL$ with $C := \mathrm{diag}(\mathbf{1}/h^2)$. Using this discretized operator, we can take the consensus equation

$$\frac{d\phi}{dt} = \Delta_d \phi = -CL\phi \tag{5}$$

as a *discrete heat equation*. The properties of the consensus discussed in Section 3 can be also understood as the flow of heat over a network.

Define the incidence matrix $B = [b_{ik}] \in \mathbb{R}^{n \times m}$ of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \dots, v_n\}$ and $\mathcal{E} = \{e_1, \dots, e_m\}$ as

$$b_{ik} = \begin{cases} 1, & \text{if } e_k \text{ leaves } v_i, \\ -1, & \text{if } e_k \text{ enters } v_i, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

It can be easily shown that for vector $\psi = [\phi_1 \ \cdots \ \phi_n]^\top \in \mathbb{R}^n$,

$$\begin{aligned} (B\psi)_i &= \sum_{k:\, e_k \text{ leaves } v_i} \phi_k - \sum_{k:\, e_k \text{ enters } v_i} \phi_k, \\ (B^\top \psi)_k &= \sum_{l=1}^{n} b_{lk}\phi_l = \phi_i - \phi_j, \end{aligned} \tag{7}$$

and $L = BB^\top$. At the same time, if we take the finite differential method with the square grid shown in Figure 1, the discretization ∇_d and ∇_d^* respectively of the gradient ∇ and the divergence ∇^* can be obtained as

$$\nabla_d = -\frac{1}{h}B^\top, \quad \nabla_d^* = \frac{1}{h}B. \tag{8}$$

Therefore, the relation $L = BB^\top$ or $\Delta_d = \nabla_d^* \nabla_d$ is a discretized version of the well-known relation $\Delta = \nabla^* \nabla$.