IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Distributed Algorithm for Solving Linear Equations in Clustered Multi-Agent Systems

Ayush Rai, Shaoshuai Mou, and Brian D. O. Anderson

***Abstract*— A new approach to solving the linear algebraic equation $Ax = b$ is presented by introducing a leaderless clustered multi-agent system. Each agent is associated with a certain submatrix of A and a vector obtained from b by a certain decomposition process. A distributed algorithm has each agent processing information solely with its own information about A and b, but sharing a time-varying estimate of part of the solution of $Ax = b$ with its neighbors, as defined by a graphical structure overlaying the agents. This graphical structure divides the agents into clusters, with agents in one cluster being associated with one block column of A, and different rows of that block column; with each intra-cluster graph being connected. The update law uses consensus within individual clusters, utilizing their estimated states together with a process of inter-cluster conservation to ensure that the concatenated sub-solutions reached by the clusters solve the overall system of linear equations. Unlike previous literature that uses clustered multi-agent systems or double-layered networks, this framework does not require the presence of an aggregator or central node in the network's clusters. The algorithm demonstrates exponential convergence, evidenced by both theoretical proof and numerical simulations.**

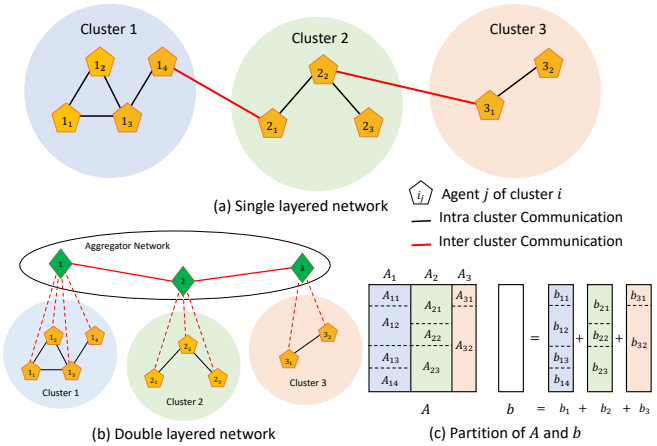


Fig. 1: Communication network and information distribution: (a) Single layer clustered multi-agent system used in our work (b) Double layered clustered multi-agent system used in [13] (c) Partition of A and b among agents

I. Introduction

Distributed algorithms for solving linear equations in multi-agent systems [1]–[10] have recently attracted a significant amount of attention due to their extensive applications in many engineering areas. By decomposing a large system of linear equations into smaller ones that can be cooperatively solved by multiple agents, these distributed algorithms are able to achieve solutions or specific solutions (such as solution with minimum L1 norm [11] or L2 norm [12]) to the overall linear equations in a multi-agent setting with communication constraints [9]. The key idea of these distributed algorithms is consensus, i.e. each agent solves its own linear equation while reaching a consensus with all other agents' solutions. This naturally requires each agent to know at least a complete row of the overall linear equation.

To reduce the amount of information that each agent needs to know and with the aim of developing a more scalable algorithm, the authors of [13], [14] have further partitioned the linear equation $Ax = b$ into smaller blocks represented by A_{ij}, b_{ij} as in Fig. 1c and introduced a double-layered clustered multi-agent systems composed of clusters where each cluster consists of one aggregator and a network of agents, as in Fig. 1b. Each agent i_j (i.e. the jth agent in cluster i) only needs to know A_{ij}, b_{ij} and communicates with its nearby neighbors within the same cluster. Note that the clustered multi-agent system introduced in [13], [14] heavily depends on the *aggregator* in each cluster and the communication among aggregators in a double-layered network, i.e. each aggregator serves a central agent in each cluster by collecting and distributing information with all agents within the cluster, and also communicate with aggregators of its nearby clusters. The requirement of an aggregator as a central node in each cluster in [13], [14], however, can be a restrictive condition, especially for clusters with a large number of agents. This motivates us in this paper to develop a distributed algorithm to solve linear equations by single-layered clustered multi-agent systems composed of clusters without aggregators as in Fig. 1a. Such single-layered clustered multi-agent systems have been deployed in [15]–[17] for unconstrained consensus, but not for solving linear equations.

In this paper, we aim to employ the single-layered clustered multi-agent systems to develop a distributed algorithm for solving linear equations, in which each agent only knows smaller blocks A_{ij}, b_{ij}, which can be two scalars when the number of partitions is sufficient. Similar to [13], the key idea of the proposed algorithm comes from the integration of the classical concept of consensus with conservation [18],

The research is supported in part by a grant from the NASA University Leadership Initiative (80NSSC20M0161) and a gift funding from Northrop Grumman Corporation.

Rai and Mou are with School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906 USA; (email: rai29,mous@purdue.edu); Anderson is with the School of Engineering, The Australian National University, Acton, ACT 2601, Australia; (email: brian.anderson@anu.edu.au). Corresponding Author: Shaoshuai Mou.

but leads to a distributed algorithm with more advantages, as summarized in the following:

1) The proposed algorithm does not require each agent to know complete rows of the overall equation or agents' states to be in the same dimension while consensus-based distributed algorithms for solving linear equations [1]–[8], [11], [12] do.
2) The proposed algorithm does not require an aggregator to serve as the central node in each cluster while the algorithm in [13], [14] does.

Notations: The transpose for vectors and matrices is denoted by $(\cdot)^\top$, while $\ker(\cdot)$ and $\mathrm{Im}\,(\cdot)$ denote the kernel and image of a matrix, respectively. A column stack of vectors $x_i, i = 1, 2, \ldots, r$ is written as $\mathrm{col}\,\{x_1, \ldots, x_r\}$ and $\mathrm{diag}\{A_1, \ldots, A_r\}$ denotes a block diagonal matrix with $A_i, i = 1, 2, \ldots, r$ as the ith block diagonal entry. The notation $\mathbf{1}_r$ represents a vector in $\mathbb{R}^r$ where all its entries are equal to 1. The symbol $\otimes$ denotes the Kronecker product. A square matrix eigenvalue is *non-defective* when its algebraic and geometric multiplicities are the same.

II. Problem Formulation

We first describe the overall system in graph theoretic terms. Consider a clustered multi-agent system of p clusters. Each cluster $i = 1, \ldots, p$, has p_i agents, and communication within this cluster is characterized by a connected undirected local graph $\mathbb{G}_i = \{\mathcal{V}_i, \mathcal{E}_i\}$, where the vertex set, $\mathcal{V}_i = \{i_1, ..., i_{p_i}\}$, denotes the agents and the edge set, $\mathcal{E}_i \subset \mathcal{V}_i \times \mathcal{V}_i$, denotes the edges between them. For each agent j in cluster i, we use $\mathcal{N}_{ij}^l$ to denote its neighbors in the local graph $\mathbb{G}_i$. Each cluster includes at least one agent capable of exchanging information with agents outside their cluster, referred to as *communicators*.

The total number of agents in the clustered network is $\sum_{i=1}^p p_i$, denoted by $\bar{p}$. The communication in the network as a whole is characterized by an undirected $\bar{p}$-node global graph $\mathbb{G} = \{\mathcal{V}, \mathcal{E}\}$, where the vertex set, $\mathcal{V} = \bigcup_{i=1}^p \mathcal{V}_i$ is the set of all the agents in the network and the edge set, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, denotes the edges between them. Note that the set $\mathcal{E}$ encompasses both communications within the clusters and communication between the clusters. We define an *inter-cluster graph* $\mathbb{G}_c$ to be the spanning subgraph of $\mathbb{G}$ obtained by removing all the local intra-cluster edges, thus $\mathbb{G}_c = \mathbb{G} \setminus \bigcup_{i=1}^p \mathcal{E}_i$. The vertex set of $\mathbb{G}_c$ remains unchanged from that of $\mathbb{G}$. The sets of neighboring agents of agent j in cluster i in the global graph $\mathbb{G}$ and the inter-cluster graph $\mathbb{G}_c$ are represented by $\mathcal{N}_{ij}$ and $\mathcal{N}_{ij}^c$ respectively.

The linear equation we seek to solve is given by $Ax = b$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. In the given architecture, no single agent has complete knowledge of either A or b. The matrix A is decomposed column-wise into matrices $A_i \in \mathbb{R}^{m \times n_i}$, where $\sum_{i=1}^p n_i = n$, such that

$$A = \begin{bmatrix} A_1 & A_2 & \ldots & A_p \end{bmatrix}. \tag{1a}$$

There is no requirement for the A_i to have equal numbers of columns). The vector b is additively decomposed[1] into p-vectors $b_i \in \mathbb{R}^m$, chosen arbitrarily with the constraint that

$$b = \sum_{i=1}^p b_i. \tag{1b}$$

In our setup, agent j of cluster i is given knowledge of certain rows of A_i, in particular a matrix $A_{ij} \in \mathbb{R}^{m_{ij} \times n_i}$, and it is also given knowledge of a vector $b_{ij} \in \mathbb{R}^{m_{ij}}$, a subvector of b_i (refer to Fig. 1c), so that

$$\begin{bmatrix} A_{i1} \\ A_{i2} \\ \vdots \\ A_{ip_i} \end{bmatrix} = A_i \qquad \begin{bmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ b_{ip_i} \end{bmatrix} = b_i,$$

where $\sum_{j=1}^{p_i} m_{ij} = m$. Each such agent j of cluster i is also regarded as controlling a time-varying solution state vector, denoted by $x_{ij}(t) \in \mathbb{R}^{n_i}$. Overall, we aim to attain convergence of the states of all the agents to a constant vector $x_{ij}^* \in \mathbb{R}^{n_i}$ under two conditions-

- Local consensus: Since agents in a given cluster hold state vectors corresponding to the same subvector of x (the entries of x multiplied by A_i in the equation $Ax = b$), they need to come to a consensus to provide a common solution x_i^*. This gives the local consensus condition

$$x_{i1}^* = x_{i2}^* = \cdots = x_{ip_i}^* = x_i^*. \tag{2}$$

- Global conservation: We define $x^* = \mathrm{col}\,\{x_1^*, x_2^*, \ldots, x_p^*\}$. For x^* to be the solution of the linear equation $Ax = b$, it has to satisfy the equation as $Ax^* = b$. Using the definitions (1a) and (1b), the equation $Ax^* = b$, can be re-written as $\sum_{i=1}^p A_i x_i^* = \sum_{i=1}^p b_i$ giving us

$$\sum_{i=1}^p (A_i x_i^* - b_i) = 0. \tag{3}$$

This is the global conservation condition.

Hence our objective is to design a distributed update strategy that converges to the solution of the given linear system, satisfying the local consensus (2) and global conservation condition (3).

III. Main results

In this section, we present the result for solving linear equations in a decentralized clustered manner without relying on aggregators. A significant obstacle in developing such a strategy is ensuring that information can be transmitted efficiently throughout the network. Additionally, to achieve global conservation across different clusters, each agent must estimate the residual error (a concept explained later) from other clusters that it may not have direct communication with. To address these issues, our approach utilizes an

[1]The additive decomposition of b into b_is is flexible since any values of b_i can be chosen as long as their sum equals b. The numerical study suggests that the convergence rate varies depending on the specific choices of b_i values.

auxiliary state variable that serves two functions: enabling the flow of information across the entire network and providing an estimate of the residual error. It is worth noting that only a portion of the auxiliary state information is employed by each agent to update its solution state, while the rest is used for information flow.

We introduce an *auxiliary coordination state* $z_{ij} \in \mathbb{R}^m$ associated with each agent. To this end, we define a *selector matrix* $E_{ij} \in \mathbb{R}^{m_{ij} \times m}$ which consists of rows of the identity matrix, such that

$$A_{ij} = E_{ij} A_i. \tag{4}$$

This matrix serves the purpose of determining the location of the sub-matrix that each individual agent directly uses, relative to the matrix that the entire cluster is aware of. It is assumed that every agent is aware of their own selector matrix. As a result, agent j of cluster i has two states associated with it: The solution state x_{ij} and the auxiliary coordination state z_{ij}.

For agent j of cluster i, we propose the following distributed update rule[2]:

$$\dot{x}_{ij} = - A_{ij}^\top \left(A_{ij} x_{ij} - b_{ij} - E_{ij} \sum_{\bar{i}\bar{k} \in \mathcal{N}_{ij}} (z_{ij} - z_{\bar{i}\bar{k}}) \right) - \sum_{ik \in \mathcal{N}_{ij}^l} (x_{ij} - x_{ik}), \tag{5a}$$

$$\dot{z}_{ij} = E_{ij}^\top (A_{ij} x_{ij} - b_{ij}) - \sum_{\bar{i}\bar{k} \in \mathcal{N}_{ij}} (z_{ij} - z_{\bar{i}\bar{k}}). \tag{5b}$$

One can observe that the summation term of the auxiliary coordination state can be separated into two components - the sum over the local graph and the sum over the inter-cluster graph, as follows-

$$\sum_{\bar{i}\bar{k} \in \mathcal{N}_{ij}} (z_{ij} - z_{\bar{i}\bar{k}}) = \sum_{ik \in \mathcal{N}_{ij}^l} (z_{ij} - z_{ik}) + \sum_{\bar{i}\bar{k} \in \mathcal{N}_{ij}^c} (z_{ij} - z_{\bar{i}\bar{k}}).$$

The second term on the right-hand side of this equation will be zero for all the non-communicators.

Remark 3.1: In equation (5b), the term $\sum_{\bar{i}\bar{k} \in \mathcal{N}_{ij}} (z_{ij} - z_{\bar{i}\bar{k}})$ estimates the residual error of the conservation condition accessible to agent j in cluster i. Note that the absence of an aggregator, as in [13], necessitates the dimension of the auxiliary coordination state z_{ij} to be $\mathbb{R}^m$, rather than $\mathbb{R}^{n_i}$.

Remark 3.2: In the update law, we observe that the solution state x_{ij} of any one agent is only shared within its own cluster, thus offering a degree of security and privacy. Only the auxiliary coordination state z_{ij} is shared across the entire network.

Before presenting the main theorem that guarantees the exponential convergence of the proposed algorithm to the exact solution, we first state a lemma that will be utilized in the proof of the theorem.

[2]We later introduce certain scalar positive gains in the update law for inter and intra-cluster communications. However, we omit these gains from the analysis for the sake of simplicity.

Lemma 3.3: Consider a matrix with the structure

$$S = \begin{bmatrix} -S_1^\top S_1 - S_2 & S_1^\top S_3 S_4 \\ S_3^\top S_1 & -S_4 \end{bmatrix}, \tag{6}$$

where S_i are real $\forall\, i = 1, \dots 4$, the matrices S_2 and S_4 are positive semidefinite, and $S_3 S_3^\top = I$. Then all eigenvalues of S are negative real or 0. Also, if S has 0 as an eigenvalue, then it must be non-defective.

The lemma's proof is available in the journal version of this paper [19]. Next, we state the main theorem along with its proof.

Theorem 3.4: Suppose the linear equation $Ax = b$ has at least one solution and the local and global graphs, $\mathbb{G}_i \;\forall\, i = 1, \dots, p$, and $\mathbb{G}$, of the clustered network are connected. Then the update law given by (5a) and (5b) yields solutions asymptotically achieving local consensus and global conservation. In particular, $\lim_{t\to\infty} x_{ij} = x_i^* \;\forall\; j$ (convergence being exponentially fast) with $x^* = \text{col}\,\{x_1^*, x_2^*, \dots, x_p^*\}$ satisfying $Ax^* = b$.[3][4]

Proof: The proof follows a similar structure to the one outlined in [13]. It consists of three main steps:

1) The derivation of a simplified and concise form of the update rules (5a) and (5b), resulting in a large linear system.
2) The demonstration of the existence of an equilibrium point for the new linear system and the exponential convergence of the system towards an equilibrium set.
3) The final step is to show every point in the equilibrium set satisfies both the local consensus and the global conservation conditions.

Step 1. Equation (5) specifies the update strategy for individual agents. By incorporating each matrix in a block diagonal form and stacking the vectors column-wise, we can combine these update laws for the entire cluster. We define $x_i \in \mathbb{R}^{p_i n_i}$ and $z_i \in \mathbb{R}^{m p_i}$ as the combined states of all agents in cluster i, i.e. the collection of states x_{ij} and z_{ij} of all the agents in the cluster as

$$x_i = \text{col}\,\{x_{i1}, x_{i2}, \dots, x_{ip_i}\};\; z_i = \text{col}\,\{z_{i1}, z_{i2}, \dots, z_{ip_i}\}.$$

Additionally, let $x \in \mathbb{R}^{\bar{n}}$ where $\bar{n} = \sum_{i=1}^p p_i n_i$, and $z \in \mathbb{R}^{m\bar{p}}$ represent the collection of states x_i and z_i of all clusters in the clustered multi-agent system, respectively, defined as

$$x = \text{col}\,\{x_1, x_2, \dots, x_p\};\; z = \text{col}\,\{z_1, z_2, \dots, z_p\}.$$

Also, let

$$\begin{aligned} \overline{A}_i &= \text{diag}\,\{A_{i1}, A_{i2}, \dots, A_{ip_i}\} \in \mathbb{R}^{m \times p_i n_i}, \\ E_i &= \text{diag}\,\{E_{i1}, E_{i2}, \dots, E_{ip_i}\} \in \mathbb{R}^{m \times m p_i}, \\ \overline{L}_{\mathbb{G}_i} &= L_{\mathbb{G}_i} \otimes I_{n_i} \in \mathbb{R}^{p_i n_i \times p_i n_i}, \\ \overline{L}_{\mathbb{G}} &= L_{\mathbb{G}} \otimes I_m \in \mathbb{R}^{m\bar{p} \times m\bar{p}}, \end{aligned}$$

[3]In cases where there are multiple solutions to the equation $Ax = b$, the algorithm will converge to one of those solutions (rather than move within the set of solutions).

[4]The auxiliary coordination state z_{ij} also converges exponentially to a fixed value, however, this value is not of much relevance to the problem being solved.

where $L_{\mathbb{G}_i} \in \mathbb{R}^{p_i \times p_i}$ and $L_{\mathbb{G}} \in \mathbb{R}^{\bar{p} \times \bar{p}}$ are the Laplacian matrix of the connected graphs, $\mathbb{G}_i$ and $\mathbb{G}$, respectively. We use the notation $[.]_{mp_i}$ to represent the mp_i rows of the matrix $[.]$ following after the $\sum_{j=1}^{j=i-1} mp_j$-th row. Using this notation the update law (5) can be written as

$$\dot{x}_i = -\overline{A}_i^\top \left\{ \overline{A}_i x_i - b_i - E_i \left[\overline{L}_{\mathbb{G}} z \right]_{mp_i} \right\} - \overline{L}_{\mathbb{G}_i} x_i, \quad (7a)$$

$$\dot{z}_i = E_i^\top (\overline{A}_i x_i - b_i) - \left[\overline{L}_{\mathbb{G}} z \right]_{mp_i}. \quad (7b)$$

Next, we define

$$\widehat{A} = \text{diag}\,\{\overline{A}_1, \overline{A}_2, \ldots, \overline{A}_p\} \in \mathbb{R}^{mp \times \bar{n}},$$
$$\widehat{b} = \text{col}\,\{b_1, b_2, \ldots, b_p\} \in \mathbb{R}^{mp},$$
$$E = \text{diag}\,\{E_1, E_2, \ldots, E_p\} \in \mathbb{R}^{mp \times m\bar{p}},$$
$$\widehat{L} = \text{diag}\,\{\overline{L}_{\mathbb{G}_1}, \overline{L}_{\mathbb{G}_2}, \ldots, \overline{L}_{\mathbb{G}_p}\} \in \mathbb{R}^{\bar{n} \times \bar{n}}.$$

With these matrices and rows defined, the update equations (7a) and (7b) for the individual clusters can be combined to form an update strategy for the entire network, as follows:

$$\dot{x} = -\widehat{A}^\top \left\{ \widehat{A}x - \widehat{b} - E\overline{L}_{\mathbb{G}} z \right\} - \widehat{L}x, \quad (8a)$$

$$\dot{z} = E^\top (\widehat{A}x - \widehat{b}) - \overline{L}_{\mathbb{G}} z. \quad (8b)$$

This gives us a simpler and more concise expression of the update rules (5a) and (5b) for the entire network

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = Q \begin{bmatrix} x \\ z \end{bmatrix} + R, \quad (9)$$

where $Q = \begin{bmatrix} -\widehat{A}^\top \widehat{A} - \widehat{L} & \widehat{A}^\top E \overline{L}_{\mathbb{G}} \\ E^\top \widehat{A} & -\overline{L}_{\mathbb{G}} \end{bmatrix}$ and $R = \begin{bmatrix} \widehat{A}^\top \widehat{b} \\ -E^\top \widehat{b} \end{bmatrix}$.

Step 2. Next, we prove that (9) has at least one equilibrium point and the system converges exponentially to an equilibrium set. We assumed that $Ax = b$ has at least one solution and let one such solution be $y \in \mathbb{R}^n$. The vector y can be partitioned to yield $y_i \in \mathbb{R}_i^n$ such that $y = \text{col}\,\{y_1, y_2, \ldots, y_p\}$. Hence, using the matrix properties and definition of A_i and b_i, the equation $Ay = b$ can be re-written as $\sum_{i=1}^{p} (A_i y_i - b_i) = 0$, which can be further expanded using the definition of A_{ij} and b_{ij} a

$$\sum_{i=1}^{p} \left(\begin{bmatrix} A_{i1} \\ A_{i2} \\ \vdots \\ A_{ip_i} \end{bmatrix} y_i - \begin{bmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ b_{ip_i} \end{bmatrix} \right) = 0.$$

Using the selector matrix, this can be written in a compact form as

$$\sum_{i=1}^{p} \sum_{j=1}^{p_i} E_{ij}^\top (A_{ij} y_i - b_{ij}) = 0.$$

We define $\overline{y}_i = \mathbf{1}_{p_i} \otimes y_i \in \mathbb{R}^{p_i n_i}$ and $\overline{x} = \text{col}\,\{\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_p\} \in \mathbb{R}^{\bar{n}}$, which helps in simplifying the equation as $\sum_{i=1}^{p} (\mathbf{1}_{p_i}^\top \otimes I_m) E_i^\top (\overline{A}_i \overline{y}_i - b_i) = 0$. Using the definition of E, this can be written as a standard linear matrix equation given by

$$(\mathbf{1}_{\bar{p}}^\top \otimes I_m) E^\top (\widehat{A}\overline{x} - b) = 0. \quad (10)$$

Recall that we defined $\overline{L}_{\mathbb{G}} = L_{\mathbb{G}} \otimes I_m$ where $L_{\mathbb{G}}$ was the Laplacian of the connected global graph $\mathbb{G}$. The Laplacian property implies that $\mathbf{1}_{\bar{p}}$ is an eigenvector of $L_{\mathbb{G}}$ with zero eigenvalue, giving us $L_{\mathbb{G}} \mathbf{1}_{\bar{p}} = 0$, which can be re-written as $(\mathbf{1}_{\bar{p}}^\top \otimes I_m) \overline{L}_{\mathbb{G}} = 0$. This can also be stated as $\text{Im}\,(\overline{L}_{\mathbb{G}}) = \ker(\mathbf{1}_{\bar{p}}^\top \otimes I_m)$. With the use of equation (10), it can be deduced that $E^\top (\widehat{A}\overline{x} - b) \in \text{Im}\ \overline{L}_{\mathbb{G}}$. Hence, there exist some vector, say $\overline{z}$, such that

$$E^\top (\widehat{A}\overline{x} - b) - \overline{L}_{\mathbb{G}} \overline{z} = 0. \quad (11)$$

If we pre-multiply by E on both the sides and noting the fact that $EE^\top = I_{pm}$, we get

$$\widehat{A}\overline{x} - b - E\overline{L}_{\mathbb{G}} \overline{z} = 0. \quad (12)$$

Using the definitions of $\widehat{L}$ as the matrix composed of the diagonal blocks $\overline{L}_{\mathbb{G}_1}, \overline{L}_{\mathbb{G}_2}, \ldots, \overline{L}_{\mathbb{G}_p}$ and $\overline{y}_i$ as the vector obtained by stacking the elements of $\mathbf{1}_{p_i} \otimes y_i$, we find that $\overline{L}_{\mathbb{G}_i} \overline{y}_i = 0$. This, combined with the definition of $\overline{x}$ as the vector obtained by stacking the $\overline{y}_i$ column-wise, leads us to

$$\widehat{L}\overline{x} = 0. \quad (13)$$

Using (11), (12), and (13) we see that $\begin{bmatrix} \overline{x}^\top & \overline{z}^\top \end{bmatrix}^\top$ is an equilibrium point of linear system (9). We can define the convergence error as

$$e(t) = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} - \begin{bmatrix} \overline{x} \\ \overline{z} \end{bmatrix}.$$

This, along with equation (9) and the fact that $[\bar{x}^\top\ \bar{z}^\top]^\top$ is an equilibrium point of the same equation, results in the error dynamics being described as $\dot{e} = Qe$. As both $\widehat{L}$ and $\overline{L}_{\mathbb{G}}$ are obtained using Laplacian of connected graphs, they are positive definite. Using the fact that $EE^\top = I$, we observe that the structure of the matrix Q matches the structure in equation (6). By invoking Lemma 3.3, we can conclude that all the eigenvalues of Q are either negative real numbers or 0. Also, if there is a 0 eigenvalue, it must be non-defective. This implies that the error $e(t)$ converges *exponentially* to either a zero vector or a constant vector $q \in \ker\ Q$. We define a set

$$\overline{S} = \left\{ \begin{bmatrix} \overline{x}^* \\ \overline{z}^* \end{bmatrix} \middle| \begin{bmatrix} \overline{x}^* \\ \overline{z}^* \end{bmatrix} = \begin{bmatrix} \overline{x} \\ \overline{z} \end{bmatrix} + q\ \forall\ q \in \ker\ Q \right\}.$$

As q lies in the kernel of Q, every point in the set $\overline{S}$, namely $\begin{bmatrix} \overline{x}^{*\top} & \overline{z}^{*\top} \end{bmatrix}^\top$, is also an equilibrium point of the dynamics described by equation (9). Hence, the set $\overline{S}$ serves as the equilibrium set.

Step 3 To conclude the proof, we demonstrate that any constant vector $\begin{bmatrix} \overline{x}^{*\top} & \overline{z}^{*\top} \end{bmatrix}^\top \in \overline{S}$ adheres to the requirements of both local consensus and global conservation. To do this, we first break down $\overline{x}^*$ into components, partitioning it as $\overline{x}^* = \text{col}\,\{\overline{x}_1^*, \overline{x}_2^*, \ldots, \overline{x}_p^*\}$, and then further partitioning each component as $\overline{x}_i^* = \text{col}\,\{\overline{x}_{i1}^*, \overline{x}_{i2}^*, \ldots, \overline{x}_{ip_i}^*\}$, where $\overline{x}_i^* \in \mathbb{R}^{p_i n_i}$ and $\overline{x}_{ij}^* \in \mathbb{R}^{n_i}$. Since $\begin{bmatrix} \overline{x}^{*\top} & \overline{z}^{*\top} \end{bmatrix}^\top$ is an

equilibrium point, we have-

$$-\widehat{A}^\top \left\{ \widehat{A}\overline{x}^* - \widehat{b} - E\overline{L}_{\mathbb{G}}\overline{z}^* \right\} - \widehat{L}\overline{x}^* = 0, \tag{14a}$$

$$E^\top(\widehat{A}\overline{x}^* - b) - \overline{L}_{\mathbb{G}}\overline{z}^* = 0 \tag{14b}$$

Equation (14a) and (14b) together gives us $\widehat{L}\overline{x}^* = 0$, thus $\overline{L}_{\mathbb{G}_i}\overline{x}_i^* = 0$. Using the properties of Laplacian and the fact that $\overline{L}_{\mathbb{G}_i} = L_{\mathbb{G}_i} \otimes I_{n_i}$, there exist a constant $x_i^* \in \mathbb{R}^{n_i}$ such that $(L_{\mathbb{G}_i} \otimes I_{n_i})(\mathbf{1}_{p_i} \otimes x_i^*) = 0$, which gives us $\overline{x}_i^* = \mathbf{1}_{p_i} \otimes x_i^*$, resulting in-

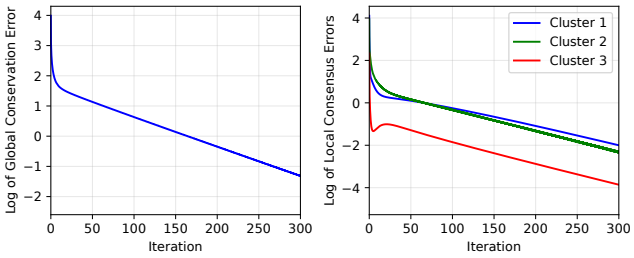
$$x_{i1}^* = x_{i2}^* = \cdots = x_{ip_i}^* = x_i^* \; \forall \; i = 1, 2, \ldots p. \tag{15}$$

This shows that the *local consensus* condition is satisfied by every point in the equilibrium set.

Next, if we pre-multiply equation (14b) on both sides by $(\mathbf{1}_{\bar{p}}^\top \otimes I_m)$ and using the fact that $\mathbb{G}$ is a connected graph, we get $(\mathbf{1}_{\bar{p}}^\top \otimes I_m)E^\top(\widehat{A}\overline{x}^* - b) - (\mathbf{1}_{\bar{p}}^\top \otimes I_m)\overline{L}_{\mathbb{G}}\overline{z}^* = 0$. The second term can be re-written as $(\mathbf{1}_{\bar{p}}^\top L_{\mathbb{G}} \otimes I_m)\overline{z}^*$ which is zero, resulting in $(\mathbf{1}_{\bar{p}}^\top \otimes I_m)E^\top(\widehat{A}\overline{x}^* - b) = 0$. This can be expressed as a summation of p equations giving us $\sum_{i=1}^{p}(\mathbf{1}_{p_i}^\top \otimes I_m)E_i^\top(\overline{A}_i\overline{x}_i^* - b_i) = 0$. Using the fact that $\overline{x}_i^* = \mathbf{1}_{p_i} \otimes x_i^*$, we get

$$\sum_{i=1}^{p}\sum_{j=1}^{p_i} E_{ij}^\top(A_{ij}x_i^* - b_{ij}) = 0, \tag{16}$$

which can be simplified as $\sum_{i=1}^{p}(A_i x_i^* - b_i) = 0$. This proves that every point in the equilibrium set $\overline{S}$ satisfies the global conservation condition. This completes our proof. ■



(a) Global conservation error (b) Local consensus error

Fig. 2: Numerical performance: (a) Evolution of global conservation error in log scale, (b) Evolution of local consensus error in log scale for each cluster of the network.

IV. Numerical Simulations

For the implementation and numerical performance of the algorithm we use the same linear equations as used in [13] given by

$$A = \left[\begin{array}{cc|cc|c} 1 & 2 & 1 & 1 & 2 \\ 2 & -1 & 3 & -3 & 0 \\ -6 & 2 & 6 & 1 & -2 \\ \hline 2 & -6 & 3 & 7 & 0 \\ -3 & -4 & 0 & -2 & 6 \end{array}\right]; b = \begin{bmatrix} 8 \\ 8 \\ 11 \\ -17 \\ -9 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \\ -5 \\ -5 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \\ 5 \\ -9 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 6 \\ -3 \\ -3 \end{bmatrix}$$

which has a unique solution. The color blocks indicate the distribution of matrices and vectors among the clusters and agents. In order to study the numerical performance of the algorithm, we introduce two types of errors: Global conservation and local consensus errors. The global conservation error measures how close each agent's solution estimate is to the unique solution, while the local consensus error reflects the similarity of solution estimates among agents within a cluster. The global conservation error is defined as

$$e^g(t) = \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p_i}\|x_{ij}(t) - x_i^*\|^2,$$

whereas the local consensus error for ith cluster is defined as

$$e_i^l(t) = \frac{1}{2}\sum_{j=1}^{p_i}\sum_{k=1}^{p_i}\|x_{ij}(t) - x_{ik}(t)\|^2.$$

The global conservation error measures the proximity of each agent's solution estimate to the unique solution at any given instant, which reflects the overall accuracy of the estimated solution. On the other hand, the local consensus error provides insight into the similarity of solution estimates among agents within a given cluster, indicating the effectiveness of intra-cluster communication in achieving consensus.

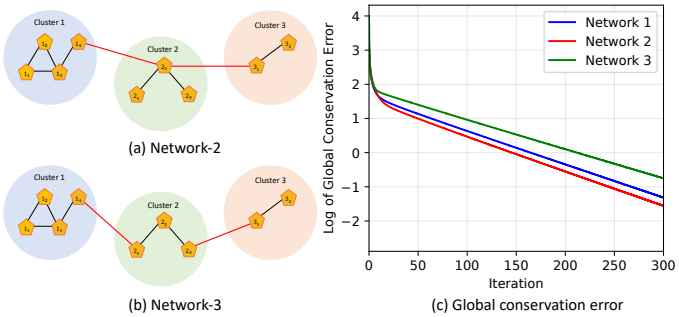


Fig. 3: (a)-(b) Other networks considered; (c) Evolution of global conservation error in log scale for 3 different network setups.

Using $x^* = [0.77; 2.79; 1.98; -1.10; 0, 38]^\top$ as the unique solution to the linear equation, we evaluate the clustered multi-agent system in Fig. 1a. The evolution of the global conservation and local consensus errors in log scale are presented in Fig. 2a and 2b respectively. Both errors converge to zero exponentially, consistent with our theoretical findings. To understand the impact of network structure on the convergence properties, we examined two additional networks in addition to the one shown in Fig. 1, as depicted in Fig. 3(a)-3(b). The global conservation error plots in the log scale are displayed in Fig. 3(c). All three cases exhibit exponential convergence but at different rates. Analysis of the results shows that the distance between communicators within the same cluster plays a crucial role in determining the convergence rate. In network 2, all the clusters have a single communicator, which results in the fastest convergence rate. In contrast, in network 3, the distance between the two communicators in cluster 2 is the largest among the considered scenarios, leading to a slower convergence rate.

We next modify the update law by introducing three positive scalar gains for inter and intra-cluster communications: k_x and k_z^l for local communication of state x and z, respectively, and k_z^g for inter-cluster communication (for communicators) of state z.[5]

TABLE I: Gains for consensus

Case	0	1	2	3	4	5	6	7
k_x	1	10	100	1000	1	1	1	100
k_z^l	1	10	100	1000	1	1	100	1
k_z^g	1	1	1	1	10	100	1	1

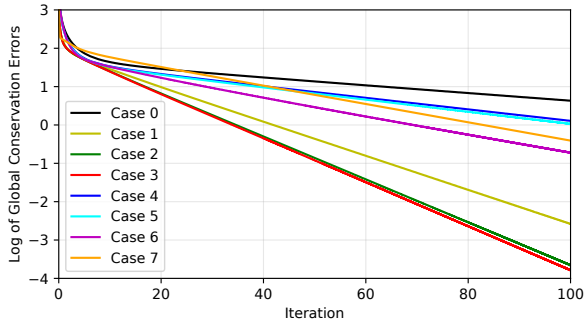


Fig. 4: Effect of different gains on convergence rate

To investigate the impact and *relative* significance of the consensus terms on the algorithm's convergence rate, we varied the positive gains assigned for inter and intra-cluster communication. The numerical study results are presented in Fig. 4, using the parameters listed in Table I. The results reveal several noteworthy observations. Firstly, Case 1, Case 2, and Case 3 show a significantly improved convergence rate compared to other cases. In all three scenarios, only the local communication gains k_x and k_z^l were increased while k_z^g was kept constant. Intuitively, a high local communication gain results in quicker local consensus, but numerically it also leads to quicker conservation. The best convergence rate of the algorithm is obtained when the local gains are 1000 times the inter-cluster gains. This behavior can be attributed to the fact that with high local gain, the cluster behaves as a single entity, resulting in a quicker communication of state z inside the graph that governs conservation. Lastly, by comparing Case 6 and Case 7, we note that a higher local gain for state z plays a more important role in conservation than for state x, as it provides an estimate of the residual error.

V. Conclusions

In this work, we have developed a distributed algorithm to solve algebraic linear equations in clustered multi-agent systems. The proposed algorithm has eliminated the requirement for an aggregator within clusters compared with existing results by introducing an auxiliary coordination state to facilitate information flow and enforce global conservation. Future work includes the generalization of the proposed algorithm for least-squares solutions and for time-varying networks and the investigation of the impact of different communication strategies in the inter-cluster and intra-cluster networks on its convergence.

[5]In the update law, positive gain k_x and k_z^l are added to the term $\sum_{ik\in\mathcal{N}_{ij}^l}(x_{ij}-x_{ik})$ and $\sum_{ik\in\mathcal{N}_{ij}^l}(z_{ij}-z_{ik})$ for local intra-cluster communication of x and z, respectively. A positive gain k_z^g is added to the term $\sum_{\bar{i}k\in\mathcal{N}_{ij}^c}(z_{ij}-z_{\bar{i}k})$ for communication of z among clusters. Note that positive gains do not affect our theoretical convergence results.

References

[1] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2863–2878, 2015.

[2] T. Yang, J. George, J. Qin, X. Yi, and J. Wu, "Distributed least squares solver for network linear equations," *Automatica*, vol. 113, p. 108798, 2020.

[3] D. Meng and Y. Wu, "Control design for iterative methods in solving linear algebraic equations," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5039–5054, 2022.

[4] N. Azizan-Ruhi, F. Lahouti, A. S. Avestimehr, and B. Hassibi, "Distributed solution of large-scale linear systems via accelerated projection-based consensus," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 3806–3817, 2019.

[5] G. Shi, B. D. Anderson, and U. Helmke, "Network flows that solve linear equations," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2659–2674, 2016.

[6] B. D. O. Anderson, S. Mou, U. R. Helmke, and A. S. Morse, "Decentralized gradient algorithm for solution of a linear equation," *Numerical Algebra, Control and Optimization*, vol. 6, no. 3, pp. 319–328, 2016.

[7] X. Wang, J. Zhou, S. Mou, and M. J. Corless, "A distributed algorithm for least square solutions," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4217–4222, 2019.

[8] S. S. Alaviani and N. Elia, "A distributed algorithm for solving linear algebraic equations over random networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2399–2406, 2021.

[9] P. Wang, S. Mou, J. Lian, and W. Ren, "Solving a system of linear equations: From centralized to distributed algorithms," *Annual Reviews in Control*, vol. 47, pp. 306–322, 2019.

[10] S. Kar, J. M. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[11] J. Zhou, X.Wang, S. Mou, and B.D.O.Anderson, "Finite-time distributed linear equation solver for solutions with minimum l1-norms," *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1691–1696, 2019.

[12] X. Wang, S. Mou, and D. Sun, "Improvement of a distributed algorithm for solving linear equations," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3113–3117, 2017.

[13] X. Wang, S. Mou, and B. D. Anderson, "Scalable, distributed algorithms for solving linear equations via double-layered networks," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1132–1143, 2019.

[14] Y. Huang, Z. Meng, and J. Sun, "Scalable distributed least square algorithms for large-scale linear equations via an optimization approach," *Automatica*, vol. 146, p. 110572, 2022.

[15] T. V. Pham, T. T. Doan, and D. H. Nguyen, "Distributed two-time-scale methods over clustered networks," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4625–4630.

[16] E. Bıyık and M. Arcak, "Area aggregation and time-scale modeling for sparse nonlinear networks," *Systems & Control Letters*, vol. 57, no. 2, pp. 142–149, 2008.

[17] M. C. Bragagnolo, I.-C. Morărescu, J. Daafouz, and P. Riedinger, "Reset strategy for consensus in networks of clusters," *Automatica*, vol. 65, pp. 53–63, 2016.

[18] X. Wang and S. Mou, "A distributed algorithm for achieving the conservation principle," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 5863–5867.

[19] A. Rai, S. Mou, and B. D. Anderson, "A distributed algorithm for solving linear equations in clustered multi-agent systems," *IEEE Control Systems Letters*, 2023.