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

# On the Effects of Data Heterogeneity on the Convergence Rates of Distributed Linear System Solvers

Boris Velasevic* MIT
Rohit Parasnis* Purdue
Christopher G. Brinton Purdue
Navid Azizan MIT

2023 62nd IEEE Conference on Decision and Control (CDC) | 979-8-3503-0124-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CDC49753.2023.10383603

*Abstract*— **We consider the fundamental problem of solving a large-scale system of linear equations. In particular, we consider the setting where a taskmaster intends to solve the system in a distributed/federated fashion with the help of a set of machines, who each have a subset of the equations. Although there exist several approaches for solving this problem, missing is a rigorous comparison between the convergence rates of the projection-based methods and those of the optimization-based ones. In this paper, we analyze and compare these two classes of algorithms with a particular focus on the most efficient method from each class, namely, the recently proposed Accelerated Projection-Based Consensus (APC) [1] and the Distributed Heavy-Ball Method (D-HBM). To this end, we first propose a geometric notion of data heterogeneity called *angular heterogeneity* and discuss its generality. Using this notion, we bound and compare the convergence rates of the studied algorithms and capture the effects of both cross-machine and local data heterogeneity on these quantities. Our analysis results in a number of novel insights besides showing that APC is the most efficient method in realistic scenarios where there is a large data heterogeneity. Our numerical analyses validate our theoretical results.**

## I. INTRODUCTION

The emergence of big data and the spate of technological advancements over the last few decades have resulted in numerous computational tasks and algorithms being distributed over networks of processing units that may or may not be centrally coordinated by a server or a taskmaster [2]–[5]. Compared to fully centralized architectures, such distributed implementations often enable more efficient solutions to complex problems while facing fewer memory issues.

Of significant interest among these are distributed approaches to the problem of solving a large-scale system of linear equations. This is among the most fundamental problems in distributed computation because systems of linear equations form the backbone of innumerable algorithms in engineering and the sciences. Unsurprisingly, then, there exist multiple approaches for solving linear equations distributively. These can be broadly categorized as (a) approaches based on distributed optimization and (b) those specifically aimed at solving systems of linear equations.

Algorithms belonging to the former category rely on the observation that solving a linear system can be expressed as an optimization problem (a linear regression) in which the loss function is separable in the data (i.e., the coefficients) but not in the variables [1]. Therefore, this category includes popular gradient-based methods such as Distributed Gradient Descent (DGD) and its variants [6]–[8], Distributed Nesterov's Accelerated Gradient Descent (D-NAG) [9], Distributed Heavy Ball-Method (D-HBM) [10], and some recently proposed algorithms such as Iteratively Pre-conditioned Gradient-Descent (IPG) [11]. Besides, the Alternating Direction Method of Multipliers [12], a well-known algorithm that is significantly slower than the others for this problem, also falls into this category.

As for the second category, i.e., approaches that are specific to solving linear systems, the most popular is the Block-Cimmino Method [13]–[15], which is essentially a distributed version of the Karczmarz method [16]. In addition, there exist some recent approaches such as those proposed in [17]–[19], and Accelerated Projection-based Consensus (APC) [1].

Among all these methods from either category, of interest to us are algorithms whose convergence rates are linear (i.e., the error decays exponentially in time) and whose computation and communication complexities are linear in the number of variables. These include DGD, D-NAG, D-HBM, Block-Cimmino Method, APC, and the projection-based distributed solver proposed in [19]. It has been shown analytically in [1] that D-HBM has a faster rate of convergence to the true solution than the other two gradient-based methods, i.e., DGD and D-NAG, and that APC converges faster than the other two projection-based methods, namely Block-Cimmino Method and the algorithm of [19].

However, which among the aforementioned methods is the fastest remains hitherto unknown, because it has so far proven challenging to characterize the relationships between the optimal convergence rates of the gradient-based approaches with those of the projection-based approaches. This is because the optimal convergence rates of the latter class of methods depend on how the global system of linear equations is partitioned for the distribution of these equations among the machines in the network, and any precise characterization of this dependence is bound to be an inherently complex combinatorial problem.

To circumvent this complexity, we propose a novel approach to capture the effect of the partitioning of the equations among the machines on the convergence rates of the aforementioned gradient-based and projection-based methods. Our analysis is based on a new notion of data heterogeneity (a concept used in federated learning to quantify

*Both authors contributed equally.

This work was supported in part by MathWorks, by the MIT-IBM Watson AI Lab, and by NSF grant CNS-2146171.

the diversity of local data across machines) called *angular heterogeneity*. This concept enables us to compare algorithms from both the classes of interest and to show that APC converges faster than all other methods when the degree of cross-machine angular heterogeneity is significant, as is often the case in real-world scenarios.

### A. Summary of Contributions

Our contributions are summarized below.

1) ***A Geometric Notion of Data Heterogeneity:*** We propose the concept of angular heterogeneity, which extends the notion of cosine similarity to certain vector spaces associated with the distribution of global data among the machines. The generality of this concept and its scale-invariant nature make it a potentially useful measure of data heterogeneity in distributed learning.

2) ***Convergence Rate Analysis:*** We derive bounds on the optimal convergence rates of (a) three gradient descent-based methods, namely D-NAG, D-HBM, and DGD, and (b) three projection-based methods, namely the Block-Cimmino Method, APC, and the algorithm proposed in [19]. Moreover, we show that the greater the level of cross-machine angular heterogeneity, the greater are the optimal convergence rates of APC and the Block-Cimmino Method.

3) ***Experimental Validation:*** We validate our theoretical results numerically and demonstrate how the optimal convergence rate of the most efficient method (APC) is bounded with respect to the dimension of the data. We also show how this convergence rate depends on the number of machines.

*Notation:* We let $\mathbb{R}$ denote the set of real numbers. Given two natural numbers $n$ and $m$, we let $[n] := \{1, 2, \ldots, n\}$ denote the set of the first $n$ natural numbers, $\mathbb{R}^n$ denotes the space of all $n$-dimensional column vectors with real entries, and $\mathbb{R}^{m\times n}$ to denotes the space of real-valued matrices with $m$ rows and $n$ columns. Besides, $I_{n\times n} \in \mathbb{R}^{n\times n}$ denotes the identity matrix and $O_{n\times n} \in \mathbb{R}^{n\times n}$ denotes the matrix with every entry equal to 0, where the subscripts are dropped if they are clear from the context. for each $k \in [n]$, we let $e_k$ denote the $k$-th canonical basis vector of $\mathbb{R}^n$.

For a vector $v \in \mathbb{R}^n$, we let $v_k$ denote the $k$-th entry of $v$ for each $k \in [n]$, and $\|v\| := \sqrt{\sum_{k=1}^n v_k^2}$ denotes the Euclidean norm of $v$. Given a matrix $M \in \mathbb{R}^{m\times n}$, we let $M^\top$ denote the transpose of $M$, we let $\|M\| := \sup_{\|z\|=1} \|Mz\|$ denote the spectral matrix norm of $M$, and $\rho(M)$ denotes the spectral radius (the absolute value of the eigenvalue with the greatest absolute value) of $M$. In addition, for a square matrix $M \in \mathbb{R}^{n\times n}$, we let $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$ denote, respectively, the maximum and the minimum eigenvalues of $M$. Furthermore, if $M$ is invertible, then $\kappa(M)$ denotes the condition number of $M$ as defined with respect to the spectral norm, i.e., $\kappa(M) := \|M\| \cdot \left\|M^{-1}\right\|$. It is well-known [20] that $\kappa(M)$ equals the ratio of the greatest and the smallest singular values of $M$. All matrix inequalities hold entry-wise.

Two linear subspaces $\mathcal{U}$ and $\mathcal{V}$ are said to be orthogonal if $u^\top v = 0$ for all $u \in \mathcal{U}$ and all $v \in \mathcal{V}$, to express which we write $\mathcal{U} \perp \mathcal{V}$. Finally, we define the inner product of two vectors $u, v \in \mathbb{R}^n$ as $\langle u, v\rangle := u^\top v$.

## II. PROBLEM FORMULATION

We aim to compare the efficiencies of two classes of distributed linear system solvers, namely, gradient-based algorithms and projection-based algorithms. We first introduce the problem setup, describe the most efficient algorithms from each class, and reproduce some known results on their optimal convergence rates. We then introduce a few geometric notions of data heterogeneity to formulate our problem precisely.

### A. The Setup

Consider a large-scale system of linear equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{N\times n}$, $x \in \mathbb{R}^n$, and $b \in \mathbb{R}^N$. Throughout this paper, we assume $N = n$ for the sake of simplicity. In other words, the coefficient matrix $A$ is assumed to be a square matrix, as we believe that the results can be extended to more general cases using very similar arguments and proof techniques. In addition, we assume $A$ to be invertible, which implies that (1) has a unique solution $x^*$ (so that $Ax^* = b$).

To solve the $n$ equations specified by (1) distributively over a network of $m \leq n$ edge machines, the central server partitions the global system (1) into $m$ linear subsystems as

$$\begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix} x = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix},$$

where for each $i \in [m]$, the $i$-th subsystem $A_i x = b_i$ (equivalently, the *local data* pair $[A_i, b_i]$ where $A_i \in \mathbb{R}^{p_i\times n}$ and $b_i \in \mathbb{R}^{p_i}$) consists of $p_i$ equations and is accessible only to machine $i \in [m]$. Unlike the analysis in [1], we do not impose any restrictions on the number of equations in any of these local subsystems. Note, however, that if $m \gg 1$, then each of these local systems is likely to be highly undetermined with infinitely many solutions because the number of local equations is likely much smaller than $n$.

We now describe the algorithms of interest. For each of these algorithms, there exists an optimal *convergence rate* $\rho > 0$ such that the convergence error vanishes at least as fast as $\rho^t$ vanishes in the limit as $t$ goes to $\infty$. We focus particularly on APC, the projection-based method with the fastest convergence behavior, and D-HBM, the gradient-based method with the fastest convergence behavior.

### B. Accelerated Projection-Based Consensus

Originally proposed in [1], accelerated projection-based consensus (APC) is essentially a distributed linear system solver in which every iteration consists of a local projection-based consensus step followed by a global averaging step, both of which incorporate momentum terms that accelerate the convergence of the algorithm to the global solution of (1).

We now describe the APC algorithm in detail. At all times $t$, the server as well as all of the $m$ machines store their estimates $\{\bar{x}(t)\} \cup \{x_i(t)\}_{i=1}^m$ of the global solution $x^*$, and these estimates are initialized and updated as follows. Each machine $i$ sets its initial estimate $x_i(0)$ of $x^*$ to one of the infinitely many solutions of $A_i x = b_i$, which can be easily computed in $O(p_i^3)$ steps. The machine then transmits $x_i(0)$ to the server, which then computes its own initial estimate of $x^*$ as $\bar{x}(0) := \frac{1}{m}\sum_{i=1}^m x_i(0)$. In subsequent iterations, $x_i(t)$ and $\bar{x}(t)$ are updated as follows.

*1) Projection-Based Consensus Step:* In iteration $t+1$, every machine $i$ receives $\bar{x}(t)$, the server's most recent estimate of $x^*$. The machine then updates its own estimate $x_i(t)$ by performing the following projection-based consensus step:

$$x_i(t+1) = x_i(t) + \gamma P_i(\bar{x}(t) - x_i(t)), \tag{2}$$

where $\gamma \geq 1$ is a fixed momentum and the matrix $P_i := I - A_i^\top (A_i A_i^\top)^{-1} A_i$ is the orthogonal projector onto the nullspace of $A_i$. In other words, the machine takes an accelerated step in a direction that is orthogonal to the coefficient vectors of its local system of equations (i.e., the rows of $A_i$). This ensures that $A_i x_i(t+1) = A_i x_i(t) = A_i x_i(0) = b_i$, i.e., $x_i(t)$ never leaves the local solution space of machine $i$.

*2) Global Averaging Step:* The next step in iteration $t+1$ is the global averaging step performed by the server as

$$\bar{x}(t+1) = \frac{\eta}{m}\sum_{i=1}^m x_i(t+1) + (1-\eta)\bar{x}(t), \tag{3}$$

where $\eta > 1$ is a fixed momentum and $(1-\eta)\bar{x}(t)$ is the memory term.

In sum, (2) and (3) are the update steps that define APC.

*Convergence Rate:* It was shown in [1] that the convergence rate of APC depends on the arithmetic mean of the projectors $\{P_i\}_{i=1}^m$. More precisely, let $S := \sum_{i=1}^m (I - P_i) = \sum_{i=1}^m (A_i^\top (A_i A_i^\top)^{-1} A_i)$. We then know from [1, Theorem 1][1] that there exist values of $\gamma$ and $\eta$ that result in the optimal convergence rate of APC being given by

$$\rho_{\text{APC}} := 1 - 2\left(\sqrt{\kappa(S)} + 1\right)^{-1}. \tag{4}$$

*Other Projection-Based Methods:* It was shown in [1] that the optimal convergence rate of the Block-Cimmino Method (BCM) is given by the convergence rate $\rho_{\text{BCM}} := 1 - \frac{2}{\kappa(S)+1}$. Besides, the algorithm proposed in [19] has an optimal convergence rate given by $\rho_{\text{MLM}} := 1 - \frac{1}{m}\lambda_{\min}(S)$, where *MLM* stands for Mou, Liu, and Morse. As shown in [1], we have $\rho_{\text{APC}} \leq \rho_{\text{BCM}} \leq \rho_{\text{MLM}}$.

### C. Distributed Heavy-Ball Method

Introduced in [10], the Distributed Heavy-Ball Method (D-HBM) is a distributed linear system solver that performs the following momentum-enhanced updates in each iteration $t$:

$$z(t+1) = \beta z(t) + \sum_{i=1}^m A_i^\top (A_i x(t) - b_i) \tag{5}$$

$$x(t+1) = x(t) - \alpha z(t+1), \tag{6}$$

Here, $\beta > 0$ and $\alpha > 1$ are the momentum and step size parameters, respectively, and $A_i^\top (A_i x(t) - b_i)$ is the gradient of the function $f_i : \mathbb{R}^n \to [0, \infty)$ defined by $f_i(y) = \|A_i y - b_i\|^2$. This gradient is evaluated at the global estimate $x(t)$ by machine $i$. Thus, each iteration of D-HBM consists of a memory-augmented gradient update followed by an accelerated gradient descent step.

*Convergence Rate:* We know from [21] that the global estimate $x(t)$ in D-HBM converges to $x^*$ as fast as $\rho_{\text{HBM}}^\top$ vanishes, where

$$\rho_{\text{HBM}} := 1 - 2\left(\sqrt{\kappa(A^\top A)} + 1\right)^{-1}. \tag{7}$$

*Other Gradient-Based Methods:* It was shown in [1] that DGD has an optimal convergence rate given by the convergence rate $\rho_{\text{DGD}} := 1 - \frac{2}{\kappa(A^\top A)}$, and it was shown in [21] that the optimal convergence rate of D-NAG is $\rho_{\text{NAG}} := 1 - \frac{2}{\sqrt{3\kappa(A^\top A)+1}}$. Thus, $\rho_{\text{HBM}} \leq \rho_{\text{NAG}} \leq \rho_{\text{DGD}}$.

## III. Geometric Notions of Data Heterogeneity

In this section, we develop two geometric notions of data heterogeneity. The first notion is based on the following concepts of local data spaces and cosine similarities between local data.

**Definition 1** (**Local Data Spaces**). *Given a machine $i \in [m]$, the row space of $A_i$, denoted by $\mathcal{R}(A_i)$, is called the local data space of machine $i$.*

Note that the linear span of the coefficient vectors (the rows of $A$) stored at machine $i$ equals the span of the rows of $A_i$, which is precisely the local data space of the machine.

**Definition 2** (**Cosine Similarity**). *For any two machines $i, j \in [m]$, let $\theta_{ij}$ denote the minimum angle between their local data spaces, i.e.,*

$$\theta_{ij} := \cos^{-1} \max_{u \in \mathcal{R}(A_i), v \in \mathcal{R}(A_j)} \left(|u^\top v| \, \|u\|^{-1} \, \|v\|^{-1}\right). \tag{8}$$

*Then $\cos\theta_{ij}$ is called the cosine similarity between the local data of machines $i$ and $j$.*

**Remark 1.** *Making the local data spaces $\mathcal{R}(A_i)$ and $\mathcal{R}(A_j)$ uni-dimensional in* (8) *and setting $\|u\| = \|v\| = 1$ would result in an expression for the cosine similarity between two unit-norm vectors. Definition 2, therefore, generalizes the standard definition of cosine similarity.*

On the basis of Definition 2, we now define cross-machine angular heterogeneity.

**Definition 3** (**Cross-machine Angular Heterogeneity**). *The angle defined as the inverse cosine of the maximum of all pairwise cosine similarities, i.e., $\theta_H := \cos^{-1}(\max_{1 \leq i < j \leq m} \cos\theta_{ij})$, is called the cross-machine angular heterogeneity of the network.*

Note that we always have $0 \leq \theta_{ij}, \theta_{\text{H}} \leq \frac{\pi}{2}$. Also, note that the more the local data spaces of the machines diverge from each other in the angular sense, the greater is the cross-machine angular heterogeneity of the network. At the

[1] Note that $S = mX$ for the matrix $X$ defined in [1, Eq. (4)].

same time, however, a salient feature of this notion of data heterogeneity is that its value is invariant with respect to any scaling applied to the rows of $A$. This is especially useful in the context of solving a linear system, because the true solution of such a system is unaffected by scaling any subset of the equations.

Besides cross-machine heterogeneity, we define another geometric notion of data heterogeneity to quantify the total angular spread of the local data at each machine.

**Definition 4** (**Local Angular Heterogeneity**). *The local angular heterogeneity $\phi_i$ of machine $i$ is the minimum angle between any two of its feature vectors (i.e., the rows of $A_i$). Hence,* $\phi_i := \cos^{-1} \max_{1\leq k<\ell\leq p_i} \left( \frac{|e_k^\top A_i A_i^\top e_\ell|}{\|A_i^\top e_k\| \|A_i^\top e_\ell\|} \right)$.

Our goal now is (a) to analyze the effects of both local and non-local angular heterogeneity on the convergence rates $\rho_{\text{APC}}$ and $\rho_{\text{HBM}}$, and (b) to use the results of our analysis to compare the efficiencies of APC and D-HBM.

## IV. MAIN RESULTS

To see how data heterogeneity affects the convergence rates $\rho_{\text{APC}}$ and $\rho_{\text{HBM}}$, we first bound the condition numbers $\kappa(S)$ and $\kappa(A^\top A)$ (with $S$ as defined in Subsection II-B) in terms of the angular heterogeneity measures $\theta_{\text{H}}$ and $\{\phi_i\}_{i=1}^m$. We then use (4) and (7) to compare $\rho_{\text{APC}}$ with $\rho_{\text{HBM}}$. We relegate all the proofs to the extended version [22].

**Theorem 1.** *For any $n \times n$ system of equations and $m$ machines with a given cross-machine angular heterogeneity $\theta_H > \cos^{-1}\left(\frac{1}{m-1}\right)$, the following bound holds independent of the number of equations $n$:*

$$\kappa(S) \leq (1 + (m-1)\cos\theta_H)(1 - (m-1)\cos\theta_H)^{-1}.$$

Theorem 1 shows that the condition number of $S$ (and hence also the convergence rate $\rho_{\text{APC}}$) is upper-bounded by an expression that is independent of the data dimension $n$. As expected, the higher the cross-machine angular heterogeneity, the tighter is the bound and the greater is the likelihood of APC converging faster to the true solution. Moreover, the result suggests that increasing the number of machines may slow down the convergence rate, which is in agreement with our intuition that packing more local data spaces into the same global space $\mathbb{R}^n$ may result in reducing the angular divergence between the data spaces.

Having examined $\kappa(S)$, which determines $\rho_{\text{APC}}$, we now examine $\kappa(A^\top A)$, which determines $\rho_{\text{HBM}}$.

**Theorem 2.** *Let $a_k^\top \in \mathbb{R}^{1\times n}$ denote the k-th row of $A$ for each $k \in [n]$. We have*

$$\kappa(A) \geq \left(\max_{k\in[n]} \|a_k\|\right) \left(\min_{\ell\in[n]} \left\{ \|a_\ell\| \sin\theta_{\min}^{(\ell)} \right\}\right)^{-1}, \quad (9)$$

*where* $\theta_{\min}^{(\ell)} := \cos^{-1} \max_{k\in[n]\setminus\{\ell\}} \left( |a_k^\top a_\ell| \, \|a_k\|^{-1} \, \|a_\ell\|^{-1} \right)$ *is the minimum angle between $a_\ell^\top$ and any other row of $A$.*

Theorem 2 provides a bound on $\kappa(A^\top A) = (\kappa(A))^2$ not only in terms of the minimum local angular heterogeneity $\phi_{\min} := \min_{i\in[m]} \phi_i$, but also in terms of the variation in the norms of the rows of $A$.

To see the dependence on $\phi_{\min}$, we first observe that for every $\ell \in [n]$, there exists a machine $i \in [m]$ that stores $a_\ell$, which, by the definitions of $\theta_{\min}^{(\ell)}$ and $\phi_i$, implies that $\theta_{\min}^{(\ell)} \leq \phi_i$. Using this, we deduce from (9) that

$$\kappa(A^\top A) \geq \left(\sin^2\phi_{\min}\right)^{-1} \quad (10)$$

Thus, it suffices to have just one machine with low local data heterogeneity for the condition number of $A^\top A$ to be large.

To see the dependence on the variation in the norms of $\{a_k : k \in [n]\}$, one can easily verify that (9) implies that

$$\kappa(A^\top A) \geq \left(\max_{k\in[n]} \|a_k\|^2\right) \left(\min_{\ell\in[n]} \|a_\ell\|^2\right)^{-1}. \quad (11)$$

Therefore, a single row of $A$ with an atypically large (or small) norm suffices to make $\kappa(A^\top A)$ large.

Besides, it is worth noting that Theorem 2 is a general result on condition numbers as it does not make any assumptions on $A$ other than that it is a square matrix. Hence, this result may be of independent interest to the reader. Finally, the theorem leads to a lower bound on $\kappa(S)$, as shown below.

**Corollary 1.** *We always have $\kappa(S) \geq \frac{1}{\sin^2\theta_H}$ independently of the number of equations $n$ and the number of machines $m$.*

We now combine the bound derived in Theorem 1 with the expression for $\rho_{\text{APC}}$ provided in (4) and simplify the result in order to obtain an upper bound on $\rho_{\text{APC}}$. Similarly, combining Corollary 2 with (4) results in a lower bound on $\rho_{\text{APC}}$. We repeat these steps with the closed-form expressions we provided in Section II-B for the optimal convergence rates $\rho_{\text{MLM}}$ and $\rho_{\text{BCM}}$ to obtain similar bounds, which we summarize in Table I.

Likewise, we combine the bounds established in (10) and (11) with the expression for $\rho_{\text{HBM}}$ provided in (7) in order to obtain the following lower bounds on $\rho_{\text{HBM}}$: $\rho_{\text{HBM}} = 1 - \frac{2}{\sqrt{\kappa(A^\top A)}+1} \overset{(a)}{\geq}= \frac{2}{1+\sin\phi_{\min}} - 1$, and $\rho_{\text{HBM}} = 1 - \frac{2}{\sqrt{\kappa(A^\top A)}+1} \overset{(b)}{\geq} \frac{\max_k\|a_k\| - \min_\ell\|a_\ell\|}{\max_k\|a_k\| + \min_\ell\|a_\ell\|}$, where $(a)$ follows from (10) and the fact that $\phi_i \in [0, \frac{\pi}{2}]$ for all $i \in [m]$, and $(b)$ follows from (11). We repeat these steps with the closed-form expressions we provided in Section II-C for $\rho_{\text{DGD}}$ and $\rho_{\text{NAG}}$ to obtain similar convergence rate bounds, which we summarize in Table I.

### A. Comparison of $\rho_{APC}$ and $\rho_{HBM}$

To compare $\rho_{\text{APC}}$ with $\rho_{\text{HBM}}$ in settings with different levels of local and cross-machine angular data heterogeneity, we first deduce from (4) and (7) that $\rho_{\text{APC}} \leq \rho_{\text{HBM}}$ if and only if $\kappa(S) \leq \kappa(A^\top A)$. Hence, we obtain the following result as an immediate consequence of Theorem 1 and (10).

**Corollary 2.** *A sufficient condition for $\rho_{APC} \leq \rho_{HBM}$ is the following: $(m-1)\cos\theta_H \leq \cos^2\phi_{min}$.*

We now consider two realistic cases for $\phi_{\min}$ and $\theta_{\text{H}}$.

TABLE I: Summary of our bounds on the optimal convergence rates of projection-based and gradient-based methods. MLM: Mou, Liu, and Morse [19]; BCM: Block-Cimmino Method [13]–[15]; APC: Accelerated Projection-Based Consensus [1]; DGD: Distributed Gradient Descent [8]; D-NAG: Distributed Nesterov's Accelerated Gradient Descent [9]; D-HBM: Distributed Heavy-Ball Method [10]

| |
|---|
| $1 - \sin^2 \theta_{\mathrm{H}} \leq \rho_{\mathrm{MLM}} \leq \left(1 - \frac{1}{m}\right)\left(1 + \cos\theta_{\mathrm{H}}\right)$ |
| $\frac{2}{1+\sin^2 \theta_{\mathrm{H}}} - 1 \leq \rho_{\mathrm{BCM}} \leq (m-1)\cos\theta_{\mathrm{H}}$ |
| $\frac{2}{1+\sin \theta_{\mathrm{H}}} - 1 \leq \rho_{\mathrm{APC}} \leq \frac{(m-1)\cos\theta_{\mathrm{H}}}{1+\sqrt{1-(m-1)\cos^2\theta_{\mathrm{H}}}}$ |
| $\rho_{\mathrm{DGD}} \geq \frac{\max_k \lVert a_k \rVert^2 - \min_\ell \lVert a_\ell \rVert^2}{\max_k \lVert a_k \rVert^2 + \min_\ell \lVert a_\ell \rVert^2}$, $\rho_{\mathrm{DGD}} \geq \frac{2}{1+\sin^2 \phi_{\mathrm{min}}} - 1$ |
| $\rho_{\mathrm{NAG}} \geq 1 - \frac{2}{\sqrt{3\frac{\max_k \lVert a_k \rVert}{\min_\ell \lVert a_\ell \rVert}+1}}$, $\rho_{\mathrm{NAG}} \geq 1 - \frac{2\sin\phi_{\mathrm{min}}}{\sqrt{3+\sin^2\phi_{\mathrm{min}}}}$ |
| $\rho_{\mathrm{HBM}} \geq \frac{\max_k \lVert a_k \rVert - \min_\ell \lVert a_\ell \rVert}{\max_k \lVert a_k \rVert + \min_\ell \lVert a_\ell \rVert}$, $\rho_{\mathrm{HBM}} \geq \frac{2}{1+\sin\phi_{\mathrm{min}}} - 1$ |

*1) Small $\phi_{min}$ and Large $\theta_H$:* This is the case of high cross-machine heterogeneity accompanying low local heterogeneity. Hence, this corresponds to most real-world scenarios in which different machines being exposed to different environments results in significant data variation across machines rather than within any local dataset. From Corollary 2 and the preceding discussion, it is clear that APC is likely to outperform D-HBM.

*2) Large $\phi_{min}$ and Small $\theta_H$:* This may happen in federated learning scenarios in which the distribution of the global data across the machines is implemented in an i.i.d. manner, which results in the local data being highly representative of the global data. Consequently, if the global data are diverse, then so is every local dataset. This may lead to the value of $\phi_{\mathrm{min}}$ being large. On the other hand, since the local datasets are similar to the global dataset, they are also similar to each other. This may result in a small $\theta_{\mathrm{H}}$. In light of Corollary 2, this means that APC converges slowly in this case. At the same time, however, highly diverse global data are likely to result in a large variation in the norms of $\{a_k : k \in [n]\}$ (the rows of $A$). This leads to a large $\kappa(A^\top A)$, and consequently, a poor convergence rate for D-HBM too.

Nonetheless, APC is likely to converge faster than D-HBM in most real-world scenarios, which are subsumed by Case 1. Moreover, as Theorems 1 and  2 suggest, APC has the added advantage of its convergence rate being insensitive to any diversity in the Euclidean lengths of the coefficient vectors (the rows of $A$).

### *B. Comparison of Other Optimal Convergence Rates*

From the definitions of $\rho_{\mathrm{BCM}}$ and $\rho_{\mathrm{DGD}}$, it is clear that the condition stated in Corollary 2 is also a sufficient condition for the Block-Cimmino Method to converge faster than DGD. Therefore, the Block-Cimmino Method can be compared with DGD in the exact same manner in which we compared APC with D-HBM in Section IV-A. Moreover, Table I shows that any comparison between a projection-based method and a gradient-based method will be qualitatively similar to the preceding comparisons.

## V. Experiments

In this section, we validate our theoretical results with the help of two sets of Monte-Carlo experiments:

1) In the first experiment, we keep $n$, the number of equations in the global system (1), fixed, and we investigate how the convergence rate of APC compares with that of D-HBM for a given number of machines $m$.

2) We keep $m$ fixed and investigate how the convergence rate of APC compares with that of D-HBM.

We now describe the experiments in detail. In the following, $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. Furthermore, in every experiment, we set $p_i = \frac{n}{m}$, where $n$ is the number of equations and $m$ is the number of machines.

### *Experiment 1: Dependence of $\rho_{APC}$ and $\rho_{HBM}$ on $m$*

We set $n = 120$ and generate multiple independent realizations of $A \in \mathbb{R}^{n \times n}$, whose entries are i.i.d. random variables generated according to $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma = 1$. We then compute the condition numbers of $S$ and $A^\top A$. To make our simulations stable, we drop the samples where $\kappa(A^\top A) > 10^7$. Nevertheless, we make sure to obtain $T = 300$ samples, and we compute the empirical expectations of $\rho_{\mathrm{HBM}}$ (which is independent of the number of machines) and $\rho_{\mathrm{APC}} = \rho_{\mathrm{APC}}(m)$ for $m \in [n]$.

Next, we repeat all of the above steps with $\mu = 1$ in order to examine the phenomenon of large-mean distributions leading to more pronounced differences between the convergence rates of APC and D-HBM, as described in [1]. Figure 1 plots the results of Experiment 1 for $\mu \in \{0, 1\}$.

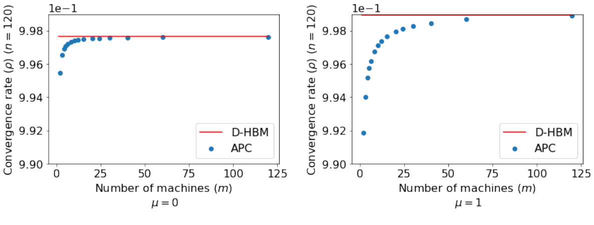

Fig. 1: Variation of $\rho_{\mathrm{APC}}$ and $\rho_{\mathrm{HBM}}$ with $m$ for $n = 120$ and $\mu \in \{0, 1\}$.

*Key Inferences:*

1) APC clearly outperforms D-HBM in both cases. This validates the conclusions drawn in Section IV.

2) As the number of machines increases, the optimal convergence rate of APC deteriorates and approaches that of D-HBM. This is as expected: as we increase $m$, we increase the number of local data spaces being packed into $\mathbb{R}^n$ (the

universal data space), possibly reducing the angles between some of the local spaces. This reduces the cross-machine angular heterogeneity with some positive probability and leads to an increase in the expected value of the upper bound on $\kappa(S)$ established in Theorem 1.
3) APC converges faster when the coefficient mean $\mu$ is increased. This is consistent with the findings of [1] and requires further investigation.

*Experiment 2: Dependence of $\rho_{\text{APC}}$ and $\rho_{\text{HBM}}$ on $n$*

We retain the setup of Experiment 1, except that we now vary the number of equations (i.e., the size of the matrix $A$) and keep the number of machines fixed, and we now draw the entries of $A$ only from $\mathcal{N}(1,1)$. Note that $\rho_{\text{HBM}}$ depends on the matrix size $n \times n$ via $\kappa(A^\top A)$. Therefore, we now expect this convergence rate to vary in our Monte-Carlo simulations. Figure 2 displays the results for $m \in \{10, 20\}$.

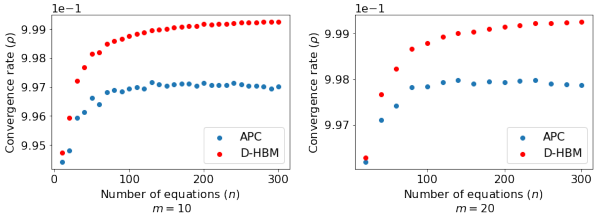

Fig. 2: Variation of $\rho_{\text{APC}}$ and $\rho_{\text{HBM}}$ with $n$ for fixed $m$.

*Key Inferences:*

1) Both $\rho_{\text{APC}}$ and $\rho_{\text{HBM}}$ increase with $n$ because the inherent complexity of (1) increases with the number of equations.
2) The convergence rate of APC is remarkably insensitive to $n$ for large values of $n$. This can be explained with the help of Theorems 1 and 2 as follows: we know from Theorem 1 that $\kappa(S)$ is upper-bounded by a quantity that depends on $n$ only through the cross-machine angular heterogeneity $\theta_{\text{H}}$. Given that the rows of $A$ are i.i.d. Gaussian random vectors, we do not expect $\theta_{\text{H}}$ to decrease with $n$, which implies that $\kappa(S)$ (and hence $\rho_{\text{APC}}$) is bounded with respect to $n$.

## VI. Conclusion and Future Directions

We compared the convergence rates of two classes of distributed linear system solvers that differ greatly in their design, namely, gradient descent-based methods such as D-HBM, and projection-based methods such as APC. To this end, we developed a novel, geometric notion of data heterogeneity called angular heterogeneity and used it to characterize the convergence rates of three distributed linear system solvers belonging to each class. In the process, we established the superiority of APC for typical real-world scenarios theoretically and empirically. We also provided several interesting insights into the effect of angular heterogeneity on the efficiencies of the studied methods. As a by-product of our work, we obtained a tight bound on the condition number of an arbitrary square matrix in terms of the Euclidean norms of its rows and the angles between them.

In the future, we aim to study the effect of the number of machines on the convergence rates of projection-based algorithms such as APC. We also aim to use the condition number bounds derived above to characterize the expected convergence rate of APC in different randomized settings.

## References

[1] N. Azizan, 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. 14, pp. 3806–3817, 2019.

[2] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pp. 63–70, IEEE, 2005.

[3] N. Santoro, *Design and analysis of distributed algorithms*. John Wiley & Sons, 2006.

[4] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–33, 2020.

[5] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[6] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[7] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[8] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[9] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate O(k$^2$)," in *Doklady Akademii Nauk*, vol. 269, pp. 543–547, Russian Academy of Sciences, 1983.

[10] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[11] K. Chakrabarti, N. Gupta, and N. Chopra, "Robustness of Iteratively Pre-conditioned Gradient-Descent Method: The case of distributed linear regression problem," in *2021 American Control Conference (ACC)*, pp. 2248–2253, IEEE, 2021.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[13] I. S. Duff, R. Guivarch, D. Ruiz, and M. Zenadi, "The augmented Block Cimmino distributed method," *SIAM Journal on Scientific Computing*, vol. 37, no. 3, pp. A1248–A1269, 2015.

[14] F. Sloboda, "A projection method of the Cimmino type for linear algebraic systems," *Parallel Computing*, vol. 17, no. 4-5, pp. 435–442, 1991.

[15] M. Arioli, I. Duff, J. Noailles, and D. Ruiz, "A block projection method for sparse matrices," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 1, pp. 47–70, 1992.

[16] S. Karczmarz, "Angenaherte auflosung von systemen linearer gleichungen," *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.*, pp. 355–357, 1937.

[17] 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, 2020.

[18] 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.

[19] 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.

[20] C. D. Meyer, *Matrix analysis and applied linear algebra*, vol. 71. Siam, 2000.

[21] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, 2016.

[22] B. Velasevic, R. Parasnis, C. G. Brinton, and N. Azizan, "On the effects of data heterogeneity on the convergence rates of distributed linear system solvers," *arXiv preprint arXiv:2304.10640*, 2023.