# Effects of Data Heterogeneity on Distributed Linear System Solvers

by

Boris Velasevic

S.B. in Computer Science and Engineering and in Mathematics
Massachusetts Institute of Technology (2023)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024



Authored by: Boris Velasevic
Department of Electrical Engineering and Computer Science
May 10, 2024

Certified by: Navid Azizan
Edgerton Assistant Professor of Mechanical Engineering, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair
Master of Engineering Thesis Committee

## Effects of Data Heterogeneity on Distributed Linear System Solvers

by

Boris Velasevic

Submitted to the Department of Electrical Engineering and Computer Science
on May 10, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

### ABSTRACT

We focus on the fundamental problem of solving a system of linear equations. In particular, we are interested in distributed linear system solvers, where one taskmaster coordinates any number of workers to attain a solution. There are two predominant and fundamentally different ways of doing this: optimization-based and projection-based solvers. Although there is extensive literature on both classes of algorithms, a rigorous analytical comparison of their performance is lacking. Consequently, there is no concrete understanding of why numerical experiments show that projection-based solvers tend to perform better in many real and synthetic scenarios. In this work, we develop a framework for such analysis, and we use that framework to investigate the comparison of optimization-based and projection-based solvers.

Thesis supervisor: Navid Azizan
Title: Edgerton Assistant Professor of Mechanical Engineering

# Acknowledgments

I would love to express my deep appreciation to the following people, without whom the completion of this work would not have been possible:

First and foremost, I would love to thank my thesis supervisor and principal investigator, Professor Navid Azizan, for his outstanding guidance since I joined his lab in October 2022. His kindness and trust towards all lab members, from undergraduate to post-doctoral researchers, is truly impressive. I was extremely fortunate to have found a mentor who recognized my technical strengths and gave me stimulating problems to attempt to solve. I am endlessly indebted to him for everything he has done for me, from teaching me how to do research and write a research paper to providing me with valuable and unparalleled support in my PhD applications.

Then, I would love to thank my closest collaborator over the past year and a half, Rohit Parasnis. I appreciate the countless linear algebra tips and tricks Rohit taught me, which made me a better mathematician. He also helped me formalize and develop many ideas presented in this work and the paper we published on this topic.

I would also love to thank all the members of Azizan Lab for their insightful comments on my work. They had many excellent ideas, although some of them remain untested due to time constraints and the technically challenging scope of the project. I do not doubt that we will continue working together on this and other projects even after my graduation.

Finally, I would love to thank my friends at MIT, who have been there with me through all the ups and downs during the last five years. Last but not least, I am forever grateful to my mom for her unconditional love and infinite support during my studies at MIT and throughout my life.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Solving a system of linear equations is one of the fundamental problems in sciences and engineering. As the systems we study become increasingly complicated over time, especially with recent advancements in big data, so do the systems of linear equations become high-dimensional, large-scale, and memory-intensive. Consequently, the theory of algorithms for solving such systems has evolved significantly over the past centuries.

A natural way of attempting to perform any computational task faster and alleviate some of the memory requirements is to distribute it over several machines. There are numerous models of such computation, but in this work, we assume that there exists a central taskmaster that coordinates the machines.

There are many approaches to constructing distributed algorithms for solving linear systems. Despite the existence of analysis and convergence guarantees for individual approaches, the distinct nature of algorithms that arise when solving systems of linear equations makes it difficult to compare rigorously the efficiency of different algorithms. We aim to present a unifying framework that enables comparing and contrasting different distributed algorithms for solving linear systems.

## 1.1 Problem Formulation

A general system of linear equations can be represented compactly as

$$Ax = b$$

where $A \in \mathbb{R}^{n\times m}$ is a coefficient matrix, and $b \in \mathbb{R}^n$ is a target vector. In this work, we assume that $A$ is a square matrix (that is, $n = m$) and that a system has a unique solution.

When we work in distributed settings, we refer to a model where a taskmaster coordinates $m$ machines where $i^{\text{th}}$ machines have $p_i$ equations stored. In practical settings, full data might be stored in individual machines, as in the case of servers responsible for collecting that subset of the data. Hence, this setting can be represented as:

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} x = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

and each $A_i \in \mathbb{R}^{p_i\times n}, b_i \in \mathbb{R}^{p_i}$. For future brevity, we denote by $\mathcal{R}(A_i)$ the $p_i$-dimensional subspace of $\mathbb{R}^n$ spanned by the row vectors in $A_i$. A schematic representation of this model is shown in Figure 1.1.

When evaluating the performance of an algorithm that solves a system of linear equations, we consider two things. The first is the computational complexity or the number of operations the algorithm needs to perform to terminate. The second one is emblematic of approaches that compute an arbitrarily good approximation to the solution - the convergence rate of the algorithm.

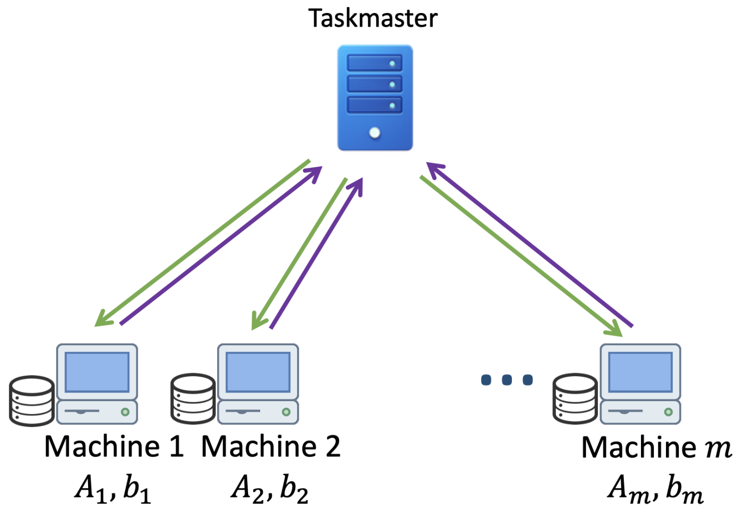


Figure 1.1: Schematic representation of the taskmaster and the $m$ machines/cores. Each machine $i$ has only a subset of the equations, i.e., $[A_i, b_i]$.

**Definition 1.** *An algorithm is said to have a convergence rate $\rho > 0$ if the convergence error - the difference of the current approximation to the solution - vanishes at least as fast as $\rho^t$ vanishes in the limit as $t \to \infty$.*

We now present some traditional approaches to solving this problem in a non-distributed setting.

## 1.2 Gaussian Elimination

One of the first algorithms studied for solving linear systems of equations is Gaussian elimination. It transforms our system into a system with the same solution, but with a diagonal coefficient matrix, enabling us to solve it row-by-row. We accomplish the transformation by performing a series of operations on the coefficient matrix and the target vector, which does not affect the solution of our system. We now briefly summarize the procedure.

Starting from the first row/column, one iteratively subtracts a multiple of the $i^{\text{th}}$ row/column of the matrix from all the rows/columns with a larger index to make the $i^{\text{th}}$ coefficients of all rows/columns with a larger index equal to zero while making the analogous operations to the $b$ vector.

Following this procedure, starting from the last row/column, one iteratively subtracts multiples of the $i^{\text{th}}$ row/column from all rows/columns with a smaller index to make the $i^{\text{th}}$ coefficient of all rows/columns with a smaller index equal to zero while making the analogous operations to the $b$ vector.

At the end of the process, all of the coefficients in the $i^{\text{th}}$ row are equal to zero except the $i^{\text{th}}$ coefficient, which allows us to obtain immediately a solution to the equation.

An illustrative example of such a process is provided below, for a coefficient matrix $A = \begin{bmatrix} 3 & 1 & 5 & 1 \\ 2 & 8 & 2 & -2 \\ 1 & 2 & 6 & -3 \\ 3 & 4 & -2 & -2 \end{bmatrix}$, and a target vector $b = \begin{bmatrix} 10 \\ 10 \\ 6 \\ 3 \end{bmatrix}$:

$$\left[\begin{array}{cccc|c} 3 & 1 & 5 & 1 & 10 \\ 2 & 8 & 2 & -2 & 10 \\ 1 & 2 & 6 & -3 & 6 \\ 3 & 4 & -2 & -2 & 3 \end{array}\right] \to \left[\begin{array}{cccc|c} 3 & 1 & 5 & 1 & 10 \\ 0 & \frac{22}{3} & \frac{-4}{3} & \frac{-8}{3} & \frac{10}{3} \\ 0 & \frac{5}{3} & \frac{13}{3} & \frac{-10}{3} & \frac{8}{3} \\ 0 & 3 & -7 & -3 & -7 \end{array}\right] \to \left[\begin{array}{cccc|c} 3 & 1 & 5 & 1 & 10 \\ 0 & \frac{22}{3} & \frac{-4}{3} & \frac{-8}{3} & \frac{10}{3} \\ 0 & 0 & \frac{-51}{11} & \frac{30}{11} & \frac{-21}{11} \\ 0 & 0 & \frac{-71}{11} & \frac{-21}{11} & \frac{-92}{11} \end{array}\right] \to$$
$$\left[\begin{array}{cccc|c} 3 & 1 & 5 & 1 & 10 \\ 0 & \frac{22}{3} & \frac{-4}{3} & \frac{-8}{3} & \frac{10}{3} \\ 0 & 0 & \frac{-51}{11} & \frac{30}{11} & \frac{-21}{11} \\ 0 & 0 & 0 & \frac{-97}{17} & \frac{-97}{17} \end{array}\right] \to \left[\begin{array}{cccc|c} 3 & 1 & 5 & 0 & 9 \\ 0 & \frac{22}{3} & \frac{-4}{3} & 0 & 6 \\ 0 & 0 & \frac{-51}{11} & 0 & \frac{-51}{11} \\ 0 & 0 & 0 & \frac{-97}{17} & \frac{-97}{17} \end{array}\right] \to \left[\begin{array}{cccc|c} 3 & 1 & 0 & 0 & 4 \\ 0 & \frac{22}{3} & 0 & 0 & \frac{22}{3} \\ 0 & 0 & \frac{-51}{11} & 0 & \frac{-51}{11} \\ 0 & 0 & 0 & \frac{-97}{17} & \frac{-97}{17} \end{array}\right] \to$$
$$\left[\begin{array}{cccc|c} 3 & 0 & 0 & 0 & 3 \\ 0 & \frac{22}{3} & 0 & 0 & \frac{22}{3} \\ 0 & 0 & \frac{-51}{11} & 0 & \frac{-51}{11} \\ 0 & 0 & 0 & \frac{-97}{17} & \frac{-97}{17} \end{array}\right].$$

Thus, our system $\begin{bmatrix} 3 & 1 & 5 & 1 \\ 2 & 8 & 2 & -2 \\ 1 & 2 & 6 & -3 \\ 3 & 4 & -2 & -2 \end{bmatrix} x = \begin{bmatrix} 10 \\ 10 \\ 6 \\ 3 \end{bmatrix}$ has the same solution as the system $\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & \frac{22}{3} & 0 & 0 \\ 0 & 0 & \frac{-51}{11} & 0 \\ 0 & 0 & 0 & \frac{-97}{17} \end{bmatrix} x = \begin{bmatrix} 3 \\ \frac{22}{3} \\ \frac{-51}{11} \\ \frac{-97}{17} \end{bmatrix}$. We are now able to solve this row by row and obtain that the solution is $x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$.

The computational complexity of this approach for an $n \times n$ matrix is $O(n^3)$. Since this algorithm computes the exact solution by design, there is no need to analyze the convergence rate. However, the cubic complexity makes the algorithm inefficient even for matrices of moderate size. Hence, other approaches are necessary.

Of note is that there exists a straightforward modification of this algorithm that allows us to find a solution to an underdetermined system, that is when the number of equations is smaller than the number of unknowns. Similar complexity analysis, in this case, yields $O(p^2 n)$ if there are $p$ equations and $n$ unknowns.

## 1.3 Gradient Descent

Note that the problem of computing the solution of $Ax = b$ can be cast as an optimization problem, that is we want to find $x$ that minimizes $F(x) = \|Ax - b\|_2^2 = x^\top (A^\top A)x - 2b^\top Ax + b^\top b$, where $\|v\|_2^2 = \sum_{i=1}^n v_i^2$ is the L2 norm of the vector.

There is a rich literature on minimizing functions. A common approach is to run an algorithm called gradient descent. Gradient descent is an iterative algorithm that moves in a negative direction of the gradient of the function to approach a minimum. If we initialize the algorithm at some $x(0)$, we can represent it symbolically as

$$x(t+1) = x(t) - \eta \nabla f(x(t))$$

where $\eta$ is the step to be tuned, and $x(0)$ is randomly initialized. This algorithm computes an arbitrarily good approximation to our solution.

Note that gradient descent is not guaranteed to find the optimum of an arbitrary function $f$. However, since our $F$ is a positive-definite quadratic form, it is well-known that gradient descent converges to a global minimum.

Observe that $\nabla F(x) = 2A^\top(Ax - b)$. Thus, per iteration of our algorithm, we perform $O(n^2)$ operations that come from matrix-vector multiplication. If the convergence rate of our algorithm is fast, then it is a considerable improvement over Gaussian Elimination. To analyze the convergence of this algorithm, we first define the condition number of the matrix:

**Definition 2.** *The condition number of matrix $M$, denoted by $\kappa(M)$ is given by $\kappa(M) = \frac{|\sigma_{\max}(M)|}{|\sigma_{\min}(M)|}$ where $\sigma_{\max}, \sigma_{\min}$ denote the maximum and minimum singular values (in absolute value) of matrix $M$.*

The convergence rate of gradient descent with the optimal step size is given by:

$$\rho_{\mathrm{GD}} = \frac{\kappa(A^\top A) - 1}{\kappa(A^\top A) + 1}.$$

Note that since $A^\top A$ is a positive-definite symmetric matrix, its singular values and eigenvalues are the same. Therefore, if we denote by $\lambda_{\max}, \lambda_{\min}$ the maximum and minimum eigenvalue of the matrix respectively, we obtain that $\kappa(A^\top A) = \frac{\lambda_{\max}(A^\top A)}{\lambda_{\min}(A^\top A)}$.

This example finishes our introduction. In the next section, we present different distributed approaches that exploit these ideas to achieve better running times.

# Chapter 2

# Distributed Algorithms

When working towards a distributed algorithm for solving linear systems, two approaches stand out: optimization-based and projection-based approaches.

Optimization-based approaches try to distribute a variant of gradient descent to reduce the expensive $O(n^2)$ computation of the gradient. The best-performing of these methods is the Distributed Heavy Ball Method.

Projection-based approaches try to initially find a solution to underdetermined systems $A_i x = b_i$, followed by a consensus algorithm that makes all the solutions of individual systems converge. The name "projection-based" comes from the fact that projecting onto the nullspaces of $A_i$s is usually a crucial part of the algorithm. The best-performing algorithm from this class is the Accelerated Projection-Based Consensus.

## 2.1 Distributed Heavy Ball Mathod

Initially introduced in [1], the Heavy Ball Method is a variation of gradient descent, where the iterative updates for minimizing a function $f$ are:

$$z(t+1) = \beta z(t) + \nabla f(x(t))$$

$$x(t+1) = x(t) - \alpha z(t+1)$$

where $\alpha, \beta$ are parameters of the algorithm to be tuned and $x(0)$ is randomly initialized. What allows us to distribute the computation is the fact that $F(x) = \|Ax - b\|_2^2$ can be written as $F(x) = \sum_{i=1}^{m} \|A_i x - b_i\|_2^2$. Using the linearity of the gradient we obtain that $\nabla F(x) = \sum_{i=1}^{m} 2A_i^\top (A_i x - b_i)$.

Hence, in step $t$, each machine computes the expression $2A_i^\top (A_i x - b_i)$ and sends the result to the taskmaster, which computes $x(t+1)$. As each $A_i \in \mathbb{R}^{p_i \times n}$, the complexity of calculating $2A_i^\top (A_i x - b_i)$ is $O(p_i n)$.

Next, the taskmaster needs to perform a linear combination of $m$ gradients. Although we can do it trivially in $O(nm)$ as each gradient is $n$-dimensional, we will utilize the fact that we are in a distributed setting to compute it component-wise distributively. Namely, each of the $m$ workers computes $O(\frac{n}{m})$ components of our gradient vector, where for each component we need $O(m)$ time. As this is done in parallel, the total complexity is $O(\frac{n}{m} \cdot m) = O(n)$, making the total cost per iteration $O(n \max_{1 \leq i \leq m} p_i)$, which is an improvement over $O(n^2)$ in a non-distributed setting.

It is proven in [2] that the convergence rate for optimally-tuned $\alpha, \beta$ of this algorithm is $\rho_{\text{D-HBM}} = \frac{\sqrt{\kappa(A^\top A)}-1}{\sqrt{\kappa(A^\top A)}+1}$, a further improvement over $\rho_{\text{GD}}$.

## 2.2 Accelerated Projection-Based Consensus

Initially introduced in [3], this algorithm starts by each machine $i$ finding an initial solution $x_i(0)$ to its underdetermined system $A_i x = b_i$ in $O(p_i^2 n)$ steps, using Gaussian Elimination or one of its variants. Furthermore, each machine computes $(A_i A_i^\top)^{-1}$ in $O(p_i^2 n + p_i^3) = O(np_i^2)$ time. Both of these operations are done only once at the start of the algorithm. Let $P_i = I - A_i^\top (A_i A_i^\top)^{-1} A_i$ be a projection matrix onto the nullspace of $A_i$. The taskmaster forms an initial estimate $\overline{x}(0) = \frac{1}{m} \sum_{i=1}^m x_i(t)$.

In the projection-based consensus step, each local worker computes $x_i(t+1) = x_i(t) + \gamma P_i(\overline{x}(t) - x_i(t))$, where and $\gamma$ is some momentum coefficient. For this step, we need to perform $O(np_i)$ computations coming from computing the term $P_i(\overline{x}(t) - x_i(t))$.

In the global averaging step, the taskmaster computes the estimate of $\overline{x}(t+1) = \frac{\eta}{m} \sum_{i=1}^m x_i(t+1) + (1-\eta)\overline{x}(t)$, where $\eta$ is a fixed momentum term, and $(1-\eta)$ is a memory term. Similarly to the Distributed Heavy Ball Method, this step is doable in $O(n)$ time. Therefore, the total cost per iteration is $O(n \max_{1 \le i \le m} p_i)$, the same as the cost of the Distributed Heavy Ball Method. Hence, the comparability (similarity, correlation, relation) between the two depends only on the algorithm convergence rate.

It is shown in [3] that the optimal convergence rate of the Accelerated Projection-Based Consensus depends on the condition number of

$$S = \sum_{i=1}^m A_i^\top (A_i A_i^\top)^{-1} A_i$$

such that $\rho_{\text{APC}} = \frac{\sqrt{\kappa(S)}-1}{\sqrt{\kappa(S)}+1}$.

## 2.3 Other Algorithms

In this section, we have briefly listed slower projection-based or optimization-based algorithms from the literature. On the projection-based side, the following are notable:

- The Block-Cimmino method, proposed in [4]–[6] has a convergence rate of $\rho_{\text{BC}} = 1 - \frac{2}{1+\kappa(S)}$.
- The algorithm proposed in [7] has a convergence rate of $\rho_{\text{MLM}} = 1 - \frac{1}{m}\lambda_{\min}(S)$.

As noted in [3], one can easily check that $\rho_{\text{Mou}} \geq \rho_{\text{BCM}} \geq \rho_{\text{APC}}$.

On the optimization-based side, the other algorithms are:

- Distributed Gradient Descent and its variants [8]–[10], with a convergence rate of $\rho_{\text{DGD}} = 1 - \frac{2}{1+\kappa(A^\top A)}$, as shown in [3].
- Distributed Nesterov's Accelerated Gradient Descent [11], with a convergence rate of $\rho_{\text{D-NAG}} = 1 - \frac{2}{\sqrt{3\kappa(A^\top A)+1}}$, as shown in [2].

Similarly, one easily checks that $\rho_{\text{DGD}} \geq \rho_{\text{D-NAG}} \geq \rho_{\text{D-HBM}}$. We can use the methods developed in this work to compare any two projection-based and optimization-based algorithms.

## 2.4 Connections to Preconditioning

The similarity of expressions $\frac{\sqrt{\kappa(S)}-1}{\sqrt{\kappa(S)}+1}, \frac{\sqrt{\kappa(A^\top A)}-1}{\sqrt{\kappa(A^\top A)}+1}$ poses a question if there is a connection between the two algorithms. As shown in [3], we can find the connection through preconditioning. Namely, a solution to the system $Ax = b$ is also a solution to the system:

$$\left(\begin{bmatrix} (A_1A_1^\top)^{-\frac{1}{2}} & & \\ & \ddots & \\ & & (A_mA_m^\top)^{-\frac{1}{2}} \end{bmatrix} A\right) x = \begin{bmatrix} (A_1A_1^\top)^{-\frac{1}{2}} & & \\ & \ddots & \\ & & (A_mA_m^\top)^{-\frac{1}{2}} \end{bmatrix} b.$$

Notice that we preconditioned our system by $B = \begin{bmatrix} (A_1 A_1^\top)^{-\frac{1}{2}} & & \\ & \ddots & \\ & & (A_m A_m^\top)^{-\frac{1}{2}} \end{bmatrix}$.

However, if we solve this system using the Distributed Heavy Ball Method, we obtain that the convergence rate depends on $\kappa((BA)^\top (BA)) = \kappa(\sum_{i=1}^m A_i^\top (A_i A_i^\top)^{-1} A_i)$, which recovers the convergence rate of Accelerated Projection-Based Consensus.

## 2.5 A Summary of Contributions

As Accelerated Projection-Based Consensus and Distributed Heavy Ball Method have the same computational complexity per iteration and convergence rates that mirror one another, the question of determining which one converges faster comes down to comparing the condition numbers of $S, A^\top A$. Since $S$ is a symmetric matrix, and as it is the sum of projection matrices all of its eigenvalues are positive, it holds that $\kappa(S) = \frac{\lambda_{\max}(S)}{\lambda_{\min}(S)}$, similar to $\kappa(A^\top A) = \frac{\lambda_{\max}(A^\top A)}{\lambda_{\min}(A^\top A)}$. Due to different forms of matrices $S, A^\top A$, obtaining an immediate comparison between the two convergence rates is difficult, even though numerical experiments show the convergence rate of Accelerated Projection-Based Consensus provides a better result.

In this paper, we:

- Introduce a novel notion of data heterogeneity - a key component in comparing projection-based and optimization-based algorithms.
- Provide efficient algorithms for computing the relevant data heterogeneity metrics.
- Prove bounds on the condition numbers of $A^\top A, S$ using notions of data heterogeneity.
- Use our analysis to illustrate numerically or analytically which algorithm converges faster in some scenarios.
- Run numerical experiments that verify our theoretical findings.

# Chapter 3

# Data Heterogeneity

Initially defined in our paper [12], in this section, we present the data heterogeneity metrics used in our analysis. We first connect concepts of data heterogeneity with other concepts in numerical linear algebra.

## 3.1 Cross-Machine Angular Heterogeneity

We first define principal angles, an object of study in linear algebra and numerical algorithms. Principal angles help us quantify at what angle two arbitrary subspaces of $\mathbb{R}^n$ intersect. Our notions of data heterogeneity consist of a special case of principal angles, and following this nomenclature allows us to reference already existing rich literature on principal angles.

**Definition 3** (**Principal Angles**). *The principal angles $\alpha_1, ..., \alpha_p$ between $\mathcal{R}(A_i)$ and $\mathcal{R}(A_j)$ are recursively defined by*

$$\cos \alpha_k = \max_{u \in \mathcal{R}(A_i), v \in \mathcal{R}(A_j)} \frac{u^\top v}{\|u\| \|v\|} = \frac{u_k^\top v_k}{\|u_k\| \|v_k\|} \tag{3.1}$$
$$\textit{subject to: } u_k^\top u_j = 0, v_k^\top v_j = 0, j \in \{1, ..., k-1\}$$

*where $u_k, v_k$ are the vectors that achieve the maximum.*

We can easily check that $\cos\alpha_k$ is a non-increasing sequence. With this definition, we identify the following:

**Definition 4** (**Cosine Similarity/Smallest Principal Angle**). *For any two distinct machines $i, j \in \{1, ..., m\}$, let $\theta_{ij}$ denote the minimum angle between their local data spaces, i.e.,*

$$\theta_{ij} := \cos^{-1}\left(\max_{u\in\mathcal{R}(A_i), v\in\mathcal{R}(A_j)}\left(\frac{|u^\top v|}{\|u\|\|v\|}\right)\right). \tag{3.2}$$

*Then $\cos\theta_{ij}$ is the* cosine similarity *between the local data of machines $i$ and $j$, and $\theta_{ij}$ is the* smallest principal angle *between their local data spaces.*

Finally, we define the cross-machine angular heterogeneity:

**Definition 5** (**Cross-machine Angular Heterogeneity**). *The inverse cosine of the maximum of all pairwise cosine similarities, i.e.,*

$$\theta_H := \cos^{-1}\left(\max_{1\le i<j\le m}\cos\theta_{ij}\right),$$

*is the* cross-machine angular heterogeneity *of the network.*

To illustrate the usefulness of this definition, let us first solve a special case of $\theta_{\mathrm{H}} = \frac{\pi}{2}$, which we call total orthogonality. In this setting, all of the subspaces are orthogonal to each other.

**Theorem 1** (**Total Orthogonality**). *Suppose $\theta_H = \frac{\pi}{2}$, i.e., suppose $\mathcal{R}(A_i) \perp \mathcal{R}(A_j)$ for all $i, j \in \{1, ..., m\}$. Then we have $\kappa(S) = 1$, and hence, $\rho_{APC} = 0$. Equivalently, APC converges to $x^*$ in a single step.*

*Proof.* Note that we have $S = \sum_{i=1}^{m} A_i^\top (A_i A_i^\top)^{-1} A_i$ by the definitions of $S$. This implies that

$$
\begin{aligned}
SA^\top A &= \left(\sum_{i=1}^{m} A_i^\top (A_i A_i^\top)^{-1} A_i\right)\left(\sum_{i=1}^{m} A_i^\top A_i\right) \\
&= \sum_{i=1}^{m} A_i^\top A_i + \sum_{i,j,i\neq j} A_i^\top (A_i A_i^\top)^{-1} A_i A_j^\top A_j \\
&= A^\top A + \sum_{i,j,i\neq j} A_i^\top (A_i A_i^\top)^{-1} (A_i A_j^\top) A_j. \qquad (3.3)
\end{aligned}
$$

Now, let $i, j \in \{1, ..., m\}$ be generic machine indices, let $a_{i,k}^\top \in \mathbb{R}^n$ denote the $k$-th row of $A_i$ for all $k \in \{1, ..., p_i\}$, and observe that $a_{i,k} \in \mathcal{R}(A_i)$ for all $k \in \{1, ..., p_i\}$. Therefore, total orthogonality implies that $(A_i A_j^\top)_{kl} = (a_{i,k})^\top a_{j,\ell} = 0$ for all $k \in \{1, ..., p_i\}$ and $\ell \in \{1, ..., p_i\}$. Hence $A_i A_j^\top = O$, where $O$ is the matrix with every entry being equal to zero, for all $i, j \in \{1, ..., m\}$. It now follows from (3.3) that $S(A^\top A) = A^\top A$. Since $A$ is assumed to be invertible, so is $A^\top A$, and hence, $S = I$. Thus, $\kappa(S) = 1$. This means $\rho_{\text{APC}} = 0$, as required. □

We will investigate the case of a general $\theta_{\text{H}}$ later.

## 3.2 Local Angular Heterogeneity

To analyze the convergence rate to a full extent, we also introduce the notion of local angular heterogeneity. As we will see, the Accelerated-Projection-Based Consensus is successful because it efficiently circumvents many difficulties caused by low local heterogeneity with optimization-based methods.

**Definition 6** (**Local Angular Heterogeneity**). *The local angular heterogeneity $\phi_i$ of machine $i$ is the minimum angle between any two of its local coefficient vectors (i.e., the* *rows of $A_i$). Hence,*

$$\phi_i := \cos^{-1}\left(\max_{1\leq k<\ell\leq p_i}\left(\frac{|e_k^\top A_i A_i^\top e_\ell|}{\|A_i^\top e_k\|\|A_i^\top e_\ell\|}\right)\right).$$

It is useful to also define the $\phi_{\min} = \min_{1\leq i\leq m}\phi_i$.

## 3.3 Algorithms for Computing Heterogeneity

It is trivial that a distributed algorithm for computing local angular heterogeneity with a runtime of $O(\max_{1\leq i\leq m} p_i^2 n)$ is possible by making each machine compute all the possible pairs of angles, each of which consists of computing some norms and dot products. How to design an algorithm that computes cross-machine angular heterogeneity is less trivial because the minimum is computed over all elements of a linear subspace.

Thankfully, we can use previously known results [13] about principal angle computation:

**Theorem 2 (Principal Angle Computation [13]).** *Recall that a matrix $M \in \mathbb{R}^{n\times p}$ can be QR-decomposed as $M = Q_M R_M$ such that the columns of $Q_M$ form an orthonormal basis for the column space of $M$. Let $Q_i$ and $Q_j$ be the orthonormal bases obtained from the QR-decompositions of $A_i^\top$ and $A_j^\top$, respectively. Then, $\cos(\theta_{ij})$ is the largest singular value of $Q_i^\top Q_j$.*

The mentioned theorem lends itself to the following distributed algorithm for computing $\theta_{\mathrm{H}}$. First, each machine runs Gram-Schmidt in $O(np_i^2)$ time to compute the QR-factorization of $A_i^\top$, and sends the orthogonal part $Q_i$ to the taskmaster. Then, the taskmaster provides each machine with all $Q_1, ..., Q_m$. Finally, each machine $i$ computes the singular value decomposition of $Q_i^\top Q_j$ for all other $j \neq i$ in $O(mp_i^3)$ time, sends the smallest angle among all seen to the taskmaster, which picks the global minimum. In the case of $p_1 = ... = p_m = p$, the complexity of this algorithm is $O(mp^3) = O(np^2)$.

# Chapter 4

# Convergence Guarantees

In this section, we use the developed notions of data heterogeneity to establish upper and lower bounds on $\kappa(S)$, and lower bounds on $\kappa(A^\top A)$.

## 4.1 Upper Bounds on $\kappa(S)$

The first upper bound presented is a bound on $\kappa(S)$ using as a function of $\theta_{\mathrm{H}}$ (data heterogeneity) and $m$ (the number of machines). Even though the bound was originally introduced in [12], here we present a new and more elegant version of the proof of the bound.

**Theorem 3.** *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 $n$:*

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

*Proof.* For each $i \in \{1, ..., m\}$, recall $\tilde{P}_i := I - P_i$ denotes the orthogonal projector onto $\mathcal{R}(A_i)$. Using the property [14, Eq. (5.13.4)] of orthogonal projectors, we can show that $\tilde{P}_i = \sum_{j=1}^{p_i} v_j^i (v_j^i)^\top$, where $\{v_j^i : j \in \{1, ..., p_i\}\}$ form an orthonormal basis for $\mathcal{R}(A_i)$. It now follows from (2.2) that

$$S = \sum_{i=1}^{m} (I - P_i) = \sum_{i=1}^{m} \tilde{P}_i = \sum_{i=1}^{m} \sum_{j=1}^{p_i} v_j^i (v_j^i)^\top.$$

Equivalently, $S = VV^\top$, where

$$V := \begin{bmatrix} \uparrow & \uparrow & \dots & \uparrow & \uparrow & \dots & \uparrow & \dots & \uparrow \\ v_1^1 & v_2^1 & \dots & v_{p_1}^1 & v_1^2 & \dots & v_{p_2}^2 & \dots & v_{p_m}^m \\ \downarrow & \downarrow & \dots & \downarrow & \downarrow & \dots & \downarrow & \dots & \downarrow \end{bmatrix}.$$

It follows that

$$\kappa(S) = \kappa(VV^\top) \overset{(a)}{=} \frac{\lambda_{\max}(VV^\top)}{\lambda_{\min}(VV^\top)}, \tag{4.1}$$

where $(a)$ holds because the symmetry of $VV^\top$ implies that its eigenvalues equal its singular values. Thus, it suffices to bound the minimum and the maximum eigenvalues of $VV^\top$.

Since $V$ is a square matrix, we can use the singular value decomposition of $V$ to show that $V^\top V$ and $VV^\top$ have the same eigenvalues: the squares of the singular values of $V$. Hence, it suffices to bound the eigenvalues of $V^\top V$. To this end, we first define

$$V_i := \begin{bmatrix} \uparrow & \uparrow & \dots & \uparrow \\ v_1^i & v_2^i & \dots & v_{p_i}^i \\ \downarrow & \downarrow & \dots & \downarrow \end{bmatrix}$$

so that $V = \begin{bmatrix} V_1 & \dots & V_m \end{bmatrix}$. As $V^\top V$ is a symmetric matrix, we obtain

$$\lambda_{\max}(V^\top V) = \sup_{\|u\|=1} u^\top V^\top V u = \sup_{\|u\|=1} \|Vu\|^2.$$

Similarly,

$$\lambda_{\min}(V^\top V) = \inf_{\|u\|=1} u^\top V^\top V u = \inf_{\|u\|=1} \|Vu\|^2.$$

Hence, we analyze $f(u) := \|Vu\|^2$ over the unit sphere $\{u \in \mathbb{R}^n : \|u\| = 1\}$.

We write $u = \begin{bmatrix} u_1^\top & \ldots & u_m^\top \end{bmatrix}^\top$ such that $u_i \in \mathbb{R}^{p_i}$. This way, it is comfortable to block multiplication with $V = \begin{bmatrix} V_1 & \ldots & V_m \end{bmatrix}$. Therefore $f(u) = \|Vu\|^2 = \|\sum_{i=1}^m V_i u_i\|^2$ over the domain $\|u\| = 1$, $u \in \mathbb{R}^n$.

Consider the function $g(z_1, ..., z_m) = \|\sum_{i=1}^m z_i\|^2$ over the domain specified by $z_i \in \mathcal{R}(A_i)$ and $\sum_{i=1}^m \|z_i\|^2 = 1$. We will show that $\mathrm{Im}(f) = \mathrm{Im}(g)$, where Im denotes the image of a function.

We first show $\mathrm{Im}(f) \subseteq \mathrm{Im}(g)$. Take some $u$ in the domain of $f$. Note that the columns of $V_i$ form an orthonormal basis for $\mathcal{R}(A_i)$ by design. Therefore, we obtain that $\|V_i u_i\| = \|u_i\|$ and that $V_i u_i \in \mathcal{R}(A_i)$. Consequently, $V_1 u_1, ..., V_m u_m$ is in the domain of $g$. Moreover, $f(u) = \|\sum_{i=1}^m V_i u_i\|^2 = g(V_1 u_1, ..., V_m u_m)$. Therefore, $\mathrm{Im}(f) \subseteq \mathrm{Im}(g)$.

We now show $\mathrm{Im}(g) \subseteq \mathrm{Im}(f)$. Take some $z_1, ..., z_m$ in the domain of $g$. Note that $(v_1^i, ..., v_{p_i}^i)$ is an orthonormal basis for $\mathcal{R}(A_i)$, so there exist coefficients $u_{i1}, ..., u_{ip_i}$ such that $z_i = \sum_{j=1}^{p_i} u_{ij} v_j^i$. This can be written compactly as $z_i = V_i u_i$ where $u_i \in \mathbb{R}^{p_i}$ is a vector of coefficients. As above, we need to have $\|z_i\| = \|V_i u_i\| = \|u_i\|$ as columns of $V_i$ are orthonormal. Hence, if we define $u = \begin{bmatrix} u_1^\top & \ldots & u_m^\top \end{bmatrix}^\top$, we obtain that $\|u\|^2 = \sum_{i=1}^m \|z_i\|^2 = 1$, and $u \in \mathbb{R}^{p_1 + \ldots + p_m} = \mathbb{R}^n$, meaning that $u$ is in the domain of $f$. Finally, $g(z_1, ..., z_n) = \|\sum_{i=1}^m z_i\|^2 = \|\sum_{i=1}^m V_i u_i\|^2 = f(u)$, proving that $\mathrm{Im}(g) \subseteq \mathrm{Im}(f)$.

We conclude $\mathrm{Im}(g) = \mathrm{Im}(f)$, which implies

$$\lambda_{\max}(V^\top V) = \sup_{\|u\|=1} f(u) = \sup_{z_i \in \mathcal{R}(A_i), \sum_{i=1}^m \|z_i\|^2 = 1} g(z_1, ..., z_m)$$

and

$$\lambda_{\min}(V^\top V) = \inf_{\|u\|=1} f(u) = \inf_{z_i \in \mathcal{R}(A_i), \sum_{i=1}^m \|z_i\|^2 = 1} g(z_1, ..., z_m).$$

Therefore, it suffices to investigate $g(z_1, ..., z_m)$.

Observe that $g(z_1, ..., z_m) = \sum_{1\leq i,j\leq m} z_i^\top z_j = 1 + \sum_{1\leq i\neq j\leq m} z_i^\top z_j$. Recall $z_i \in \mathcal{R}(A_i)$ and the definition of $\cos\theta_{ij}, \cos\theta_{\mathrm{H}}$ to obtain $|z_i^\top z_j| \leq \|z_i\|\|z_j\|\cos\theta_{ij} \leq \|z_i\|\|z_j\|\cos\theta_{\mathrm{H}}$. Therefore, $g(z_1, ..., z_m) \leq 1 + \cos\theta_{\mathrm{H}}\left(\sum_{1\leq i\neq j\leq m}\|z_i\|\|z_j\|\right)$. Finally, note that $\|z_i\|\|z_j\| \leq \frac{\|z_i\|^2+\|z_j\|^2}{2}$ by the AM-GM inequality, yielding

$$g(z_1, ..., z_m) \leq 1 + \cos\theta_{\mathrm{H}}(m-1)\left(\sum_{i=1}^m \|z_i\|^2\right) = 1 + (m-1)\cos\theta_{\mathrm{H}}.$$

Analogously, we get that $g(z_1, ..., z_m) \geq 1 - (m-1)\cos\theta_{\mathrm{H}}$. Therefore:

$$\lambda_{\max}(V^\top V) = \sup_{\|u\|=1}\|Vu\|^2 = \sup_{\|u\|=1} f(u) =$$

$$\sup_{z_i\in\mathcal{R}(A_i), \sum_{i=1}^m \|z_i\|^2=1} g(z_1, ..., z_m) \leq 1 + (m-1)\cos\theta_{\mathrm{H}}.$$

And similarly:

$$\lambda_{\min}(V^\top V) = \inf_{\|u\|=1}\|Vu\|^2 = \inf_{\|u\|=1} f(u) =$$

$$\inf_{z_i\in\mathcal{R}(A_i), \sum_{i=1}^m \|z_i\|^2=1} g(z_1, ..., z_m) \geq 1 - (m-1)\cos\theta_{\mathrm{H}}.$$

Assuming that $1-(m-1)\cos\theta_{\mathrm{H}} > 0$ we can safely divide the two and obtain our result. □

Next is a bound on $\kappa(S)$ as a function of $n, p_i$ and $\theta_{\mathrm{H}}$.

**Theorem 4.** *Suppose that we are given an $n \times n$ system of equations and $m$ machines, and suppose there exists a machine with $p$ equations. Additionally, suppose $\theta_H$ satisfies $1 - \sqrt{p(n-p)\cos\theta_H} - (n-p)\cos\theta_H > 0$. Then, regardless of the number of machines $m$, we have*

$$\kappa(S) \leq \frac{1 + \sqrt{p(n-p)\cos\theta_H} + (n-p)\cos\theta_H}{1 - \sqrt{p(n-p)\cos\theta_H} - (n-p)\cos\theta_H}.$$

*Proof.* We begin the proof by defining the matrix $V$, as in 3.

Note that $V$ is a square matrix. We obtain its QR decomposition $V = QR$ using Gram-Schmidt orthogonalization. Note that the first $p_1$ entries of $V$ constitute an orthonormal basis, which implies that $Q$ is of the form $Q = \begin{bmatrix} v_1^1 & ... & v_{p_1}^1 & w_{p_1+1} & ... & w_n \end{bmatrix}$ where $w_{p_1+1}, \ldots, w_n \in \mathbb{R}^n$ and that $R$ is of the form

$$R = \begin{bmatrix} I_{p_1 \times p_1} & U_1 \\ 0_{(n-p_1) \times p_1} & U_2 \end{bmatrix}$$

where $U_1 \in \mathbb{R}^{p_1 \times (n-p_1)}$ and $U_2 \in \mathbb{R}^{(n-p_1) \times (n-p_1)}$. By expanding the product $QR$, we observe that

$$U_0 := \begin{bmatrix} \uparrow & ... & \uparrow & ... & \uparrow \\ v_1^2 & ... & v_{p_2}^2 & ... & v_{p_m}^m \\ \downarrow & ... & \downarrow & ... & \downarrow \end{bmatrix} = Q \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}.$$

We first comment on the component $U_1$ of $R$. As prescribed by Gram-Schmidt orthogonalization, we obtain $U_1$ by taking inner products of $v_1^2, ..., v_{p_m}^m$ with $q_1 = v_1^1, ..., q_{p_1} = v_{p_1}^1$. By the definition of $\theta_{1i}$, we have $|(v_k^1)^\top v_l^i| \leq \cos\theta_{1i}$, as $v_k^1$ and $v_l^i$ are in the orthonormal bases of $\mathcal{R}(A_1)$ and $\mathcal{R}(A_i)$, respectively. Therefore, $\max(|U_1|) \leq \max_{2 \leq i \leq m} \cos\theta_{1i}$ and $\max(|U_1^\top U_1|) \leq p_1 \max_{2 \leq i \leq m} \cos\theta_{1i}$. Next, we comment on the relationship between $U_1, U_2$. Observe that

$$\begin{bmatrix} I_{p_2 \times p_2} & M_1^2 & ... & M_{n-1}^2 \\ M_1^3 & I_{p_3 \times p_3} & ... & M_{n-1}^3 \\ & \ddots & & \\ M_1^m & M_2^m & ... & I_{p_m \times p_m} \end{bmatrix} = U_0^\top U_0 = U_1^\top U_1 + U_2^\top U_2. \tag{4.2}$$

We can write $U_1^\top U_1 + U_2^\top U_2 = I + L$, where $L$ packs all the off-diagonal entries. As $L$ is obtained by taking inner products between unit-norm basis vectors across different machines, using the same reasoning as above, we obtain $\max(|L|) \leq \max_{2 \leq i < j \leq m} \cos\theta_{ij}$. Furthermore, as both $U_1^\top U_1 + U_2^\top U_2$ and $I$ are Hermitian, it follows that $L$ is also Hermitian.

Given that $V$ is a square matrix, the eigenvalues of $S = VV^\top$ are the same as those of $V^\top V$. Also, since $V^\top V = R^\top Q^\top Q R = R^\top R$, it suffices to inspect the eigenvalues of

$$R^\top R = \begin{bmatrix} I_{p_1 \times p_1} & U_1 \\ U_1^\top & U_1^\top U_1 + U_2^\top U_2 \end{bmatrix} = \begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & L \end{bmatrix}.$$

As all of the matrices included in this sum are Hermitian, we can conclude by Weyl's inequality that $\lambda_{\min}(R^\top R) \geq \lambda_{\min}\left(\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}\right) + \lambda_{\min}\left(\begin{bmatrix} 0 & 0 \\ 0 & L \end{bmatrix}\right)$. Since $\max(|L|) \leq \max_{2 \leq i < j \leq m} \cos\theta_{ij}$ and $L$ has diagonal entries equal to 0, by Gerschgorin's circle theorem, we get that $\lambda_{\min}\left(\begin{bmatrix} 0 & 0 \\ 0 & L \end{bmatrix}\right) \geq -(n - p_1)\max_{2 \leq i < j \leq m} \cos\theta_{ij}$. We are left to consider the eigenvalues of $\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}$. Let $\lambda$ be an eigenvalue of $\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}$ that is not equal to 1. Considering that $\lambda \neq 1$, $(1-\lambda)I$ is invertible, using Shur's complement lemma on the characteristic polynomial yields

$$\det\left(\begin{bmatrix} I(1-\lambda) & U_1 \\ U_1^\top & I(1-\lambda) \end{bmatrix}\right) =$$
$$= \det\left(I(1-\lambda)\right)\det((1-\lambda)I - (1-\lambda)^{-1}U_1^\top U_1). \tag{4.3}$$

Setting the right-hand-side equal to zero and re-arranging, we obtain

$$\frac{(1-\lambda)^{p_1}}{(1-\lambda)^{n-p_1}}\det((1-\lambda)^2 I - U_1^\top U_1) = 0. \tag{4.4}$$

From equation (4.4) we obtain that eigenvalues $\lambda \neq 1$ of $\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}$ satisfy that $(1 -$

$\lambda)^2$ is an eigenvalue of $U_1^\top U_1$. Furthermore, every eigenvalue $\lambda'$ of $U_1^\top U_1$ satisfies $|\lambda'| \leq \|U_1^\top U_1\|_\infty$, as the spectral radius of a matrix is bounded by the infinity-norm. Recalling that $\max(|U_1^\top U_1|) \leq p_1 \max_{2\leq i\leq m} \cos\theta_{1i}$, we obtain $|\lambda'| \leq \|U_1^\top U_1\|_\infty \leq (n-p_1)p_1 \max_{2\leq i\leq m} \cos\theta_{1i}$. For brevity, define $\mathcal{I} = [1-\sqrt{(n-p_1)p_1 \max_{2\leq i\leq m} \cos\theta_{1i}}, 1+\sqrt{(n-p_1)p_1 \max_{2\leq i\leq m} \cos\theta_{1i}}]$. Hence, we obtain that $\lambda \in \mathcal{I}$. We conclude that every eigenvalue $\lambda \neq 1$ of $\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}$ is in the interval $\mathcal{I}$. Since $1 \in \mathcal{I}$, every eigenvalue of $\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}$ is in $\mathcal{I}$. Therefore, $\lambda_{\min}\left(\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}\right) \geq 1-\sqrt{(n-p_1)p_1 \max_{2\leq i\leq m} \cos\theta_{1i}}$. Finally, putting it all together:

$$\lambda_{\min}(VV^\top) = \lambda_{\min}(R^\top R) \geq$$
$$\lambda_{\min}\left(\begin{bmatrix} I & U_1 \\ U_1^\top & I \end{bmatrix}\right) + \lambda_{\min}\left(\begin{bmatrix} 0 & 0 \\ 0 & L \end{bmatrix}\right) \geq$$
$$1-\sqrt{(n-p_1)p_1 \max_{2\leq i\leq m} \cos\theta_{1i}} - (n-p_1) \max_{2\leq i<j\leq m} \cos\theta_{ij}.$$

Analogously,

$$\lambda_{\max}(VV^\top) = \lambda_{\max}(R^\top R) \leq$$
$$1+\sqrt{(n-p_1)p_1 \max_{2\leq i\leq m} \cos\theta_{1i}} + (n-p_1) \max_{2\leq i<j\leq m} \cos\theta_{ij}.$$

Assuming that $\theta_{\mathrm{H}}$ is sufficiently close to $\frac{\pi}{2}$ such that $1-\sqrt{p_1(n-p_1)\cos\theta_{\mathrm{H}}}-(n-p)\cos\theta_{\mathrm{H}} > 0$, we can safely divide the two and replace all the maxima of cosines with the global maximum - which is $\cos\theta_{\mathrm{H}}$ - to obtain our result. Finally, note that isolating the first machine and obtaining a result that includes $p_1$ was completely arbitrary, and the result holds for any machine.

$\square$

We now comment on the tightness of these bounds. Due to a more elegant proof, we focus on analyzing the tightness of the bound presented in 3. The critical observation is that we use the fact $|z_i^\top z_j| \leq \|z_i\|\|z_j\| \cos(\theta_{ij})$ both when we are upper and lower bounding the value of $g(z_1, ..., z_m)$. In the upper bound, we use that this fact implies that $z_i^\top z_j \leq \|z_i\|\|z_j\| \cos(\theta_{ij})$, and in the lower bound the fact that $\|z_i\|\|z_j\| \cos(\theta_{ij}) \geq -|z_i\|\|z_j\| \cos(\theta_{ij})$. Equality cannot hold in both of these simultaneously, and therefore one can conclude that the bound we end up with is unfortunately not tight. Therefore, this is a potential area of improvement and further research on our analysis.

## 4.2 Lower Bounds on $\boldsymbol{\kappa(A^\top A)}$

Here we present the lower bounds on $\kappa(A^\top A)$. Together with the upper bounds on $\kappa(S)$, this induces conditions on when APC has a faster convergence rate than the HBM.

**Theorem 5.** *Let $a_k^\top \in \mathbb{R}^{1\times n}$ denote the $k$-th row of $A$ for each $k \in \{1, ..., n\}$. We have*

$$\kappa(A) \geq \frac{\max_{k\in\{1,\dots,n\}} \|a_k\|}{\min_{\ell\in\{1,\dots,n\}} \left\{\|a_\ell\| \sin\theta_{min}^{(\ell)}\right\}}, \tag{4.5}$$

*where*

$$\theta_{min}^{(\ell)} := \cos^{-1} \max_{k\in\{1,\dots,n\}\setminus\{\ell\}} \left(\frac{|a_k^\top a_\ell|}{\|a_k\|\|a_\ell\|}\right)$$

*is the minimum angle between $a_\ell^\top$ and any other row of $A$.*

We omit the proof due to brevity, but it can be found in [15]. However, we do provide an outline of the proof containing the main ideas:

1. First, we show that $\kappa(A^\top A) = (\kappa(A))^2 = (\kappa(R))^2$, where $R$ is defined by the QR factorization $A = QR$.

2. Then, we exploit the fact that we can freely permute the rows of $A$ without changing the condition number: we arrange the rows of $A$ in the descending order of their norms.

We also relate $\kappa(R)$ to the diagonal entries of $R$ by using the relation $\kappa(R) = \|R\|\|R^{-1}\|$ and the fact that the eigenvalues of $R$ are its diagonal entries. As a result, we obtain $\kappa(R) \geq \frac{\max_{k\in\{1,\ldots,n\}} |r_{kk}|}{\min_{k\in\{1,\ldots,n\|} |r_{kk}|}$.

3. Next, we use the properties of Gram-Schmidt orthogonalization (a procedure integral to QR factorization) to determine the relationship between the entries of $R$ and the norms of the rows of $A$. We use the row order resulting from Step 1 along with Cauchy-Schwarz inequality to show that $\min_{k\in\{1,\ldots,n\}} r_{kk} \leq \min_{k\in\{1,\ldots,n\}} \left\{\|a_k\| \sin\theta_{\min}^{(k)}\right\}$.

4. Finally, we integrate the results of the previous three steps.

It follows that $\kappa(A^\top A) \geq \left(\frac{\max_{k\in\{1,\ldots,n\}} \|a_k\|}{\min_{\ell\in\{1,\ldots,n\}}\left\{\|a_\ell\| \sin\theta_{\min}^{(\ell)}\right\}}\right)^2$. The generality of this result and its minimal assumptions make it useful outside of the context of this work.

We discuss two important corollaries. First, it is easy to verify that (5) implies that:

$$\kappa(A^\top A) \geq \frac{\max_{k\in\{1,\ldots,n\}} \|a_k\|^2}{\min_{\ell\in\{1,\ldots,n\}} \|a_\ell\|^2}. \tag{4.6}$$

Thus, a row with an unusually large or small norm can cause the condition number of $A^\top A$ to be large.

If $\ell^* = \mathrm{argmin}_{\ell\in\{1,\ldots,n\}}\theta_{\min}^{(\ell)}$ is the row that makes the globally smallest angle, observe that $\min_{\ell\in\{1,\ldots n\}}\{\|a_\ell\| \sin\theta_{\min}^{(\ell)}\} \leq \|a_{\ell^*}\| \sin\theta_{\min}^{(\ell^*)}$. Furthermore, we need to have that $\theta_{\min}^{(\ell^*)} \leq \phi_{\min}$ as the global minimum $\theta_{\min}^{(\ell^*)}$ has to be upper bounded by any local minimum $\phi_i$. Using these two facts, we deduce from (4.5) that

$$\begin{aligned}
\kappa(A^\top A) &\geq \left(\frac{\max_{k\in\{1,\ldots,n\}} \|a_k\|}{\min_{\ell\in\{1,\ldots,n\}}\{\|a_\ell\| \sin\theta_{\min}^{(\ell)}\}}\right)^2 \\
&\geq \left(\frac{\max_{k\in\{1,\ldots,n\}} \|a_k\|}{\|a_{\ell^*}\| \sin\theta_{\min}^{(\ell^*)}}\right)^2 \\
&\geq \frac{1}{\sin^2\theta_{\min}^{(\ell^*)}} \geq \frac{1}{\sin^2\phi_{\min}}.
\end{aligned} \tag{4.7}$$

Therefore, it suffices to have just one machine with low local data heterogeneity for the condition number of $A^\top A$ to be large. We will comment on the tightness of bounds presented herein in section 4.4 of this chapter.

## 4.3 Lower Bounds on $\kappa(S)$

In this section, we present a lower bound on $\kappa(S)$. The proof utilizes the proofs of Theorem 5 and 3, presented in [15].

**Proposition 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 $m$.*

*Proof.* By definition, there are two machines $i, j \in [m]$ and two unit-norm vectors, $u \in \mathcal{R}(A_i)$ and $v \in \mathcal{R}(A_j)$, such that $u^\top v = \cos\theta_{\mathrm{H}}$. We now construct a matrix $V \in \mathbb{R}^{n\times n}$ as described in the proof of Theorem 3, and we set $v_1^i := u$ and $v_1^j := v$, which we can do without the loss of generality. Finally, we replace $A$ with $V^\top$ in Theorem 5 and use the fact that all the rows of $V^\top$ have the same norm to obtain the desired bound. □

## 4.4 Illustrative Examples

In this section, we discuss a few examples of matrices that help us answer the questions about the tightness and behavior of our bounds.

We first comment on the tightness of the lower bound on $\kappa(A^\top A)$. Consider the diagonal matrix $A = \mathrm{diag}(a_1, ..., a_n)$, and without loss of generality assume that $a_1 \geq a_2 \geq ... \geq a_n$. It can easily be confirmed that $A^\top A = \mathrm{diag}(a_1^2, ....a_n^2)$. Therefore, we obtain that $\kappa(A^\top A) = \frac{a_1^2}{a_n^2}$. On the other hand, observe that all the row vectors are orthogonal to each other, and therefore, $\sin\theta_{\min}^{(\ell)} = 1$ for every $\ell \in \{1, ..., n\}$. Consequently, our bound gives the same result of $\frac{a_1^2}{a_n^2}$. Also notable is the fact that, in this case, $\kappa(A^\top A)$ can be arbitrarily bad as we take the ratio of $\frac{a_1}{a_n}$ to be arbitrarily large. However, one can easily verify that $\theta_{\mathrm{H}} = \frac{\pi}{2}$

independent of the number of machines. That being the case, in light of 3, we obtain that $\kappa(S) = 1$ and that APC converges in a single step. Therefore, in this example, APC has arbitrarily better convergence guarantees than the Heavy Ball Method.

A straightforward generalization of the above example is if we take $A = \text{diag}(A_1, ..., A_m)$ to be a block diagonal matrix. In this case, if we distribute the equations to $m$ machines such that each machine gets only one entry from an $A_i$, we can similarly claim that $\theta_{\mathrm{H}} = \frac{\pi}{2}$, but we can make $\kappa(A^\top A) = \kappa(\text{diag}(A_1^\top A_1, ..., A_m^\top A_m))$ arbitrarily large.

Finally, let $J_n$ be the $n \times n$ matrix where every entry is 1. Consider $A = J_n - \varepsilon I_n$. For small values of $\varepsilon$, this is a matrix with little variation in its entries. Note that, if $a_i, a_j$ are distinct rows in $A$, we have $a_i^\top a_j = n - 2\varepsilon$, and $\|a_i\|_2 = \sqrt{n - 2\varepsilon + \varepsilon^2}$. Also, recall that $\theta_{ij}$ is defined as the minimum angle between any two vectors in $\mathcal{R}(A_i)$ and $\mathcal{R}(A_j)$. Therefore, any feasible angle is an upper bound on $\theta_{ij}$. Consequently, the angle made by $a_i \in \mathcal{R}(A_i)$ and $a_j \in \mathcal{R}(A_j)$ is $\cos^{-1}\left(\frac{|a_i^\top a_j|}{\|a_i\|\|a_j\|}\right)$. Using the inferences mentioned above, we conclude that

$$\theta_{ij} \leq \cos^{-1}\left(\frac{n - 2\varepsilon}{n - 2\varepsilon + \varepsilon^2}\right) = \cos^{-1}\left(1 - \frac{\varepsilon^2}{n - 2\varepsilon + \varepsilon^2}\right).$$

It now follows from the definition of $\theta_{\mathrm{H}}$ that $\theta_{\mathrm{H}} \leq \cos^{-1}(1 - \frac{\varepsilon^2}{n-2\varepsilon+\varepsilon^2})$. In other words, $\theta_{\mathrm{H}}$ can be chosen to be arbitrarily close to 0. Therefore, $\kappa(S) \geq \frac{1}{\sin^2 \theta_{\mathrm{H}}}$ is unbounded, making the convergence guarantees on APC arbitrarily bad. Finally, we have $\phi_{\min} = \cos^{-1}(1 - \frac{\varepsilon^2}{n-2\varepsilon+\varepsilon^2})$, meaning that $\phi_{\min}$ can be chosen to be arbitrarily close to 0. As a result, $\kappa(A^\top A) \geq \frac{1}{\sin^2 \phi_{\min}}$ is unbounded, thereby making the convergence rates of the Heavy Ball Method bad. Thus, when there is little to no variation in the entries of $A$, both algorithms perform poorly.

## 4.5 Comparison of Convergence Rates

We can observe that large data heterogeneity helps the convergence rates of the algorithms. Furthermore, the scale-invariant nature of the bound on APC makes it more likely to converge faster in many instances. We can use Theorem 3 and equation 4.7 to obtain a sufficient condition $\rho_{\text{APC}} < \rho_{\text{D-HBM}}$, which occurs if and only if $\kappa(S) \leq \kappa(A^\top A)$:

**Corollary 1.** *A sufficient condition for $\rho_{APC} < \rho_{D\text{-}HBM}$ is $\sin^2 \phi_{min} < \frac{1-(m-1)\cos\theta_H}{1+(m-1)\cos\theta_H}$, or equivalently,*

$$\frac{2}{1 + \frac{1}{(m-1)\cos\theta_H}} < \cos^2 \phi_{min}. \tag{4.8}$$

In [12], we use this inequality to examine two potential qualitative scenarios:

- Small $\phi_{\text{min}}$ and Large $\theta_{\text{H}}$. This is the case of high cross-machine heterogeneity accompanying low local heterogeneity. Hence, it 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. APC is likely to outperform the Heavy Ball Method.

- Large $\phi_{\text{min}}$ and Small $\theta_{\text{H}}$. This may happen in federated learning scenarios in which the local data, while diverse enough, are similar across different machines. This may lead to the value of $\phi_{\text{min}}$ being large. On the other hand, since the local datasets are similar to the global dataset, they are also similar to each other, which may result in a small $\theta_{\text{H}}$. In light of Proposition 1, it means that APC converges slowly in such a case. At the same time, however, highly diverse global data are likely to result in a large variation in the norms of $\{a_k : k \in \{1, ..., n\}\}$ (the rows of $A$), which leads to a large $\kappa(A^\top A)$ and, consequently, a poor convergence rate for the Heavy Ball Method.

We can utilize the developed theory to build upper and lower bounds on the convergence rate of any algorithm mentioned in 2. After doing some calculations, we obtain the results presented in Table 4.1, an adapted version of the results from [12]. We can employ those upper and lower bounds to establish similar guarantees on which an algorithm is expected to converge faster, as performed above.

Table 4.1: Summary of bounds on the optimal convergence rates of projection-based and gradient-based methods. MLM: Mou, Liu, and Morse [7]; BC: block Cimmino method [4]–[6]; APC: Accelerated Projection-Based Consensus [3]; DGD: Distributed Gradient Descent [10]; D-NAG: Distributed Nesterov's Accelerated Gradient Descent [11]; D-HBM: Distributed Heavy-Ball Method [1]

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

# Chapter 5

# Empirical Experiments

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

## 5.1 Dependence of $\rho_{\text{D-HBM}}$ and $\rho_{\text{APC}}$ on $m$

We follow the experiment as described in [12]. 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 the 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 $\rho_{\text{D-HBM}}$ (which is independent of the number of machines) and $\rho_{\text{APC}} = \rho_{\text{APC}}(m)$ for $m \in \{1, ..., n\}$.

Next, we repeat all of the above steps with $\mu = 1$ to examine the phenomenon of large-mean distributions leading to more pronounced differences between the convergence rates of APC and Heavy Ball Method, as described in [3]. Figure 5.1 plots the results of the experiment for $\mu \in \{0, 1\}$.

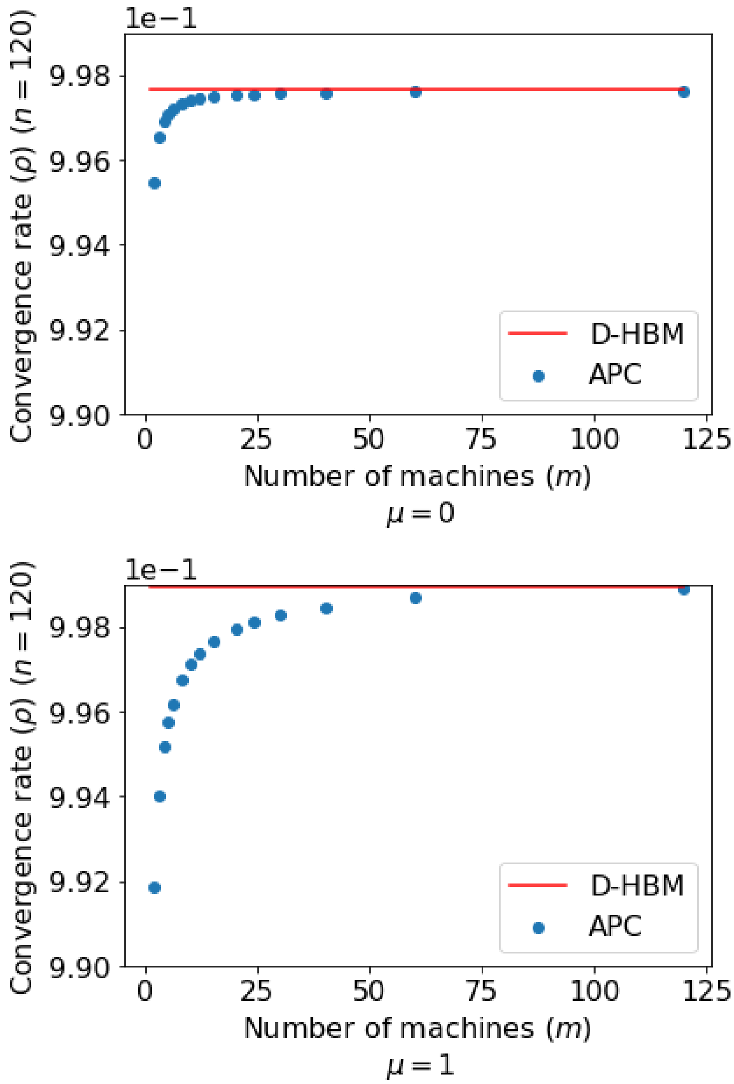


Figure 5.1: Variation of $\rho_{\text{APC}}$ and $\rho_{\text{D-HBM}}$ with $m$ for $n = 120$ and $\mu \in \{0, 1\}$.

We comment on the key inferences from the experiment.

- APC outperforms the Heavy Ball Method in both cases, which validates the conclusions drawn from our main results in chapter 4.
- As the number of machines increases, the optimal convergence rate of APC deteriorates and approaches that of the Heavy Ball Method, as expected. Namely, by increasing $m$, we also increase the number of local data spaces being packed into $\mathbb{R}^n$ (the universal

data space), possibly decreasing the angles between some of the local spaces. This ultimately 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 3.

- APC converges faster when the coefficient mean $\mu$ is increased. That is consistent with the findings of [3] and can be potentially explained by the distributional formula for $\theta_{\mathrm{H}}$ for non-central distributions presented in Section 10.6.4 in [16].

## 5.2 Dependence of $\rho_{\text{D-HBM}}$ and $\rho_{\text{APC}}$ on $n$

We follow the experiment as described in [12]. We retain the setup of the previous experiment, except that we now vary the number of equations (i.e., the size of the matrix $A$) and keep the number of machines fixed, drawing the entries of $A$ only from $\mathcal{N}(1, 1)$.

Note that $\rho_{\text{D-HBM}}$ depends on the matrix size $n \times n$ via $\kappa(A^\top A)$. Therefore, we expect this convergence rate to vary in our Monte-Carlo simulations. Figure 5.2 displays the results of the experiment for $m \in \{10, 20\}$.

We comment on the key inferences from the experiment.

- Both $\rho_{\text{APC}}$ and $\rho_{\text{D-HBM}}$ increase with $n$ because the inherent complexity of solving the system increases with the number of equations.

- The convergence rate of APC is remarkably insensitive to $n$ for large values of $n$, which can be explained with the help of Theorems 3 and 5. Namely, we know from Theorem 3 that $\kappa(S)$ is upper-bounded by a quantity that depends on $n$ only through the cross-machine angular heterogeneity $\theta_{\mathrm{H}}$. Given that the rows of $A$ are i.i.d. Gaussian random vectors, we do not expect $\theta_{\mathrm{H}}$ to decrease with $n$, which implies that $\kappa(S)$ (and hence $\rho_{\text{APC}}$) is bounded with respect to $n$.

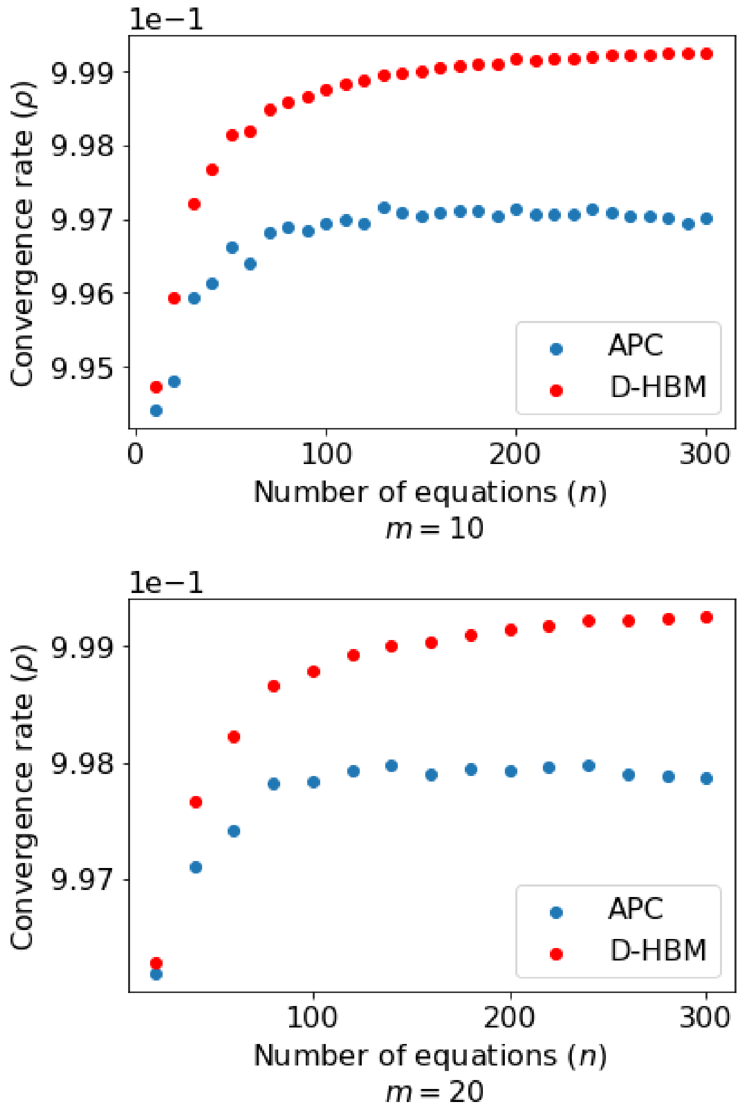


Figure 5.2: Variation of $\rho_{\text{APC}}$ and $\rho_{\text{D-HBM}}$ with $n$ for fixed $m$.

## 5.3 Asymptotic Behavior of $\boldsymbol{\theta}_{\mathbf{H}}$

Consider the case when matrix $A$ is generated so that every entry is independent, identically distributed from $\mathcal{N}(0, 1)$, which is of interest to many scientific and engineering applications. Furthermore, assume that each machine has $p$ equations, that is, each $A_i \in \mathbb{R}^{p \times n}$. We will investigate the asymptotic behaviors of $\theta_{\text{H}}$ in such a setting.

We will utilize already rich existing literature on principal angles between random subspaces of $\mathbb{R}^n$. To do this, we first argue that for any two machines $i, j$, the row spaces

$\mathcal{R}(A_i), \mathcal{R}(A_j)$ which are generated using the process described above are independent and uniformly sampled from all $p$-dimensional subspaces of $\mathbb{R}^n$.

First, note that a linear subspace of $\mathbb{R}^n$ is determined by its basis elements. The basis elements of our two subspaces are vectors whose entries are i.i.d. samples from $\mathcal{N}(0,1)$. The independence of such subspaces is, therefore, trivial. Next, it is well known that the row space of a $p \times n$ random matrix whose elements are i.i.d. Gaussian is distributed according to a uniform distribution over $p$-dimensional subspaces of $\mathbb{R}^n$ [17]. The key idea of the argument is to utilize the rotational invariance of the multidimensional Gaussian. Hence, $\mathcal{R}(A_i), \mathcal{R}(A_j)$ are indeed uniform and independent samples of $p$-dimensional subspaces of $\mathbb{R}^n$.

We now go ahead and use the results of [18] [16] to investigate the principal angle between two uniformly randomly selected subspaces of $\mathbb{R}^n$, which in this case is $\theta_{ij}$.

The formula for the exact distribution of $\tan^{-2}\theta_{ij}$ is given in [16, Section 10.6.4], for the case of $\frac{n-2p-1}{2}$ being an integer:

$$\mathbb{P}(\tan^{-2}\theta_{ij} \leq x) = \left(1 - \frac{1}{1+x}\right)^{\frac{p^2}{2}} \sum_{k=0}^{\frac{p(n-2p-1)}{2}} \sum_{\kappa} \frac{\left(\frac{p}{2}\right)_\kappa C_\kappa(I_p)}{(1+x)^k k!}$$

where the second sum is over all the partitions $\kappa$ of $k$ that consist of $p$ entries, with the largest entry being at most $\frac{n-2p-1}{2}$, $(\frac{p}{2})_\kappa$ and $C_\kappa$ are the hypergeometric coefficients and zonal polynomial concerning partition $\kappa$, respectively.

As the exact formula is complex to work with, we turn to numerical estimates of the density for $\tan^{-2}\theta_{ij}$. We keep $p$ fixed and observe how the density changes as $n$ increases. The results can be found in Figure 5.3.

To get an estimate of the behavior of $\theta_{\mathrm{H}}$ given the densities of $\theta_{ij}$ for every pair $1 \leq i < j \leq m$, we employ the union bound to obtain $\mathbb{P}(\tan^{-2}\theta_{\mathrm{H}} > \varepsilon) = \mathbb{P}(\exists (i,j) : \tan^{-2}(\theta_{ij}) > \varepsilon) \leq \sum_{1 \leq i < j \leq m} \mathbb{P}(\tan^{-2}\theta_{ij} > \varepsilon) = \frac{n}{2p}\left(\frac{n}{p} - 1\right)(1 - \mathbb{P}(\tan^{-2}\theta_{ij} \leq \varepsilon)) = \frac{n}{2p}\left(\frac{n}{p} - 1\right)\mathbb{P}(\tan^{-2}\theta_{ij} > \varepsilon)$, where $\varepsilon > 0$ is an arbitrary number that represents how tight of a concentration we are interested in.

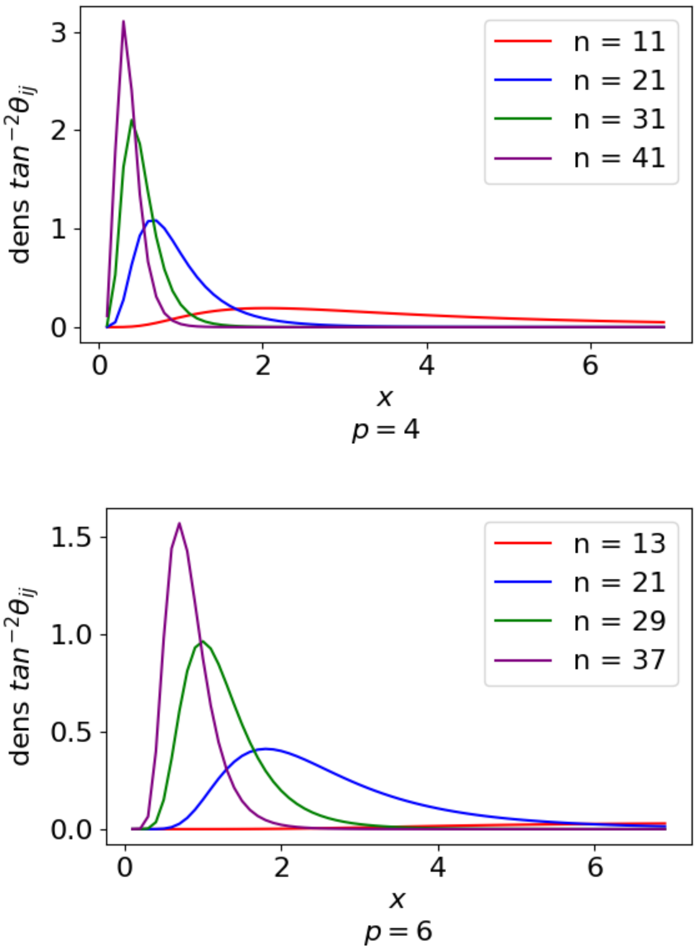


Figure 5.3: The behavior of the density of $\tan^{-2}\theta_{ij}$ under increasing $n$ when $p \in \{4, 6\}$

We therefore investigate the asymptotics of $\mathbb{P}(\tan^{-2}\theta_{ij} > \varepsilon)$ as $n \to \infty$, again by using a computational approach. Namely, we calculate the quantity with the distribution formula. The results can be found in Figure 5.4.

We comment on the key inferences from the experiment.

- Figure 5.3 suggests that as $n \to \infty$ and we keep $p$ fixed, the density of $\tan^{-2}\theta_{ij}$ concentrates around 0, meaning that $\theta_{ij} \to \frac{\pi}{2}$.

- Figure 5.4 suggests that $\mathbb{P}(\tan^{-2}\theta_{ij} > \varepsilon)$ decreases either exponentially or super-exponentially fast in the limit as $n \to \infty$. Recalling the union bound calculation, this implies that $\mathbb{P}(\tan^{-2}\theta_{\mathrm{H}} > \varepsilon) \to 0$ for any fixed $\varepsilon > 0$. Hence, as $n \to \infty, \theta_{\mathrm{H}} \to \frac{\pi}{2}$.

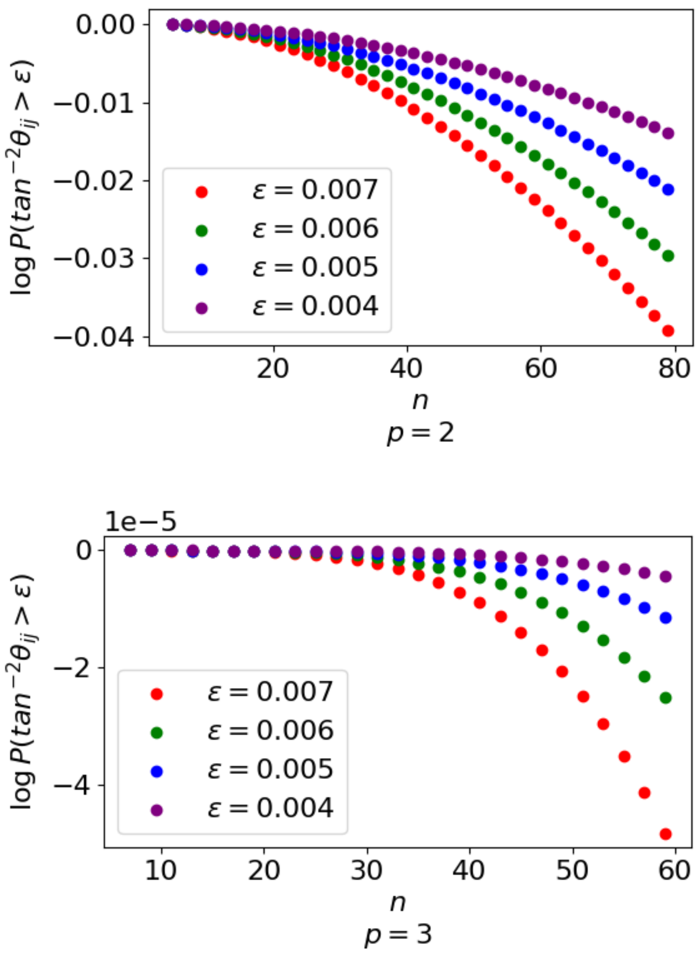


Figure 5.4: Asymptotics of $\mathbb{P}(\tan^{-2}\theta_{ij} > \varepsilon)$ under decreasing $\varepsilon$ when $p \in \{2, 3\}$

However, this convergence is slow and generally insufficient by itself to make definitive conclusions from Theorems 4, 3, since it follows from the required conditions that we must have $\mathbb{P}(\tan^{-2}\theta_{\mathrm{H}} > \frac{p^2}{n(n-2p)}) \to 0$.

## 5.4 High Heterogeneity Matrices

The last key inference from our previous experiment raises the question of what matrices satisfy the constraints of Theorem 3, namely that $\cos\theta_{\mathrm{H}} < \frac{1}{m-1}$. In section 4.4 we argued analytically that block diagonal matrices satisfy the requirements. In this section, we provide more general examples.

For this particular construction, let each machine get $p$ equations, and denote the local dataspace of the $i^{\text{th}}$ out of $m = \frac{n}{p}$ machines by $A_i = \left[A_{i1}, \ldots, A_{im}\right]$, where each $A_{ij} \in \mathbb{R}^{p \times p}$.

Let the entries of $A_{ij}$ for $j \neq i$ be independent and identically distributed (i.i.d.) as $\mathcal{N}(0, \sigma^2)$. Also, let the entries of $A_{ii}$ be generated i.i.d. according to $\mathcal{N}(0, 1)$. Using the algorithm for computing $\theta_{\text{H}}$ as described in 3.3, we can construct empirical distributions for $\theta_{\text{H}}$. Namely, we calculate $\theta_{\text{H}}$ for $T = 10^4$ matrice randomly generated as described in the example. The results are displayed in Figure 5.5 for various values of $\sigma$.

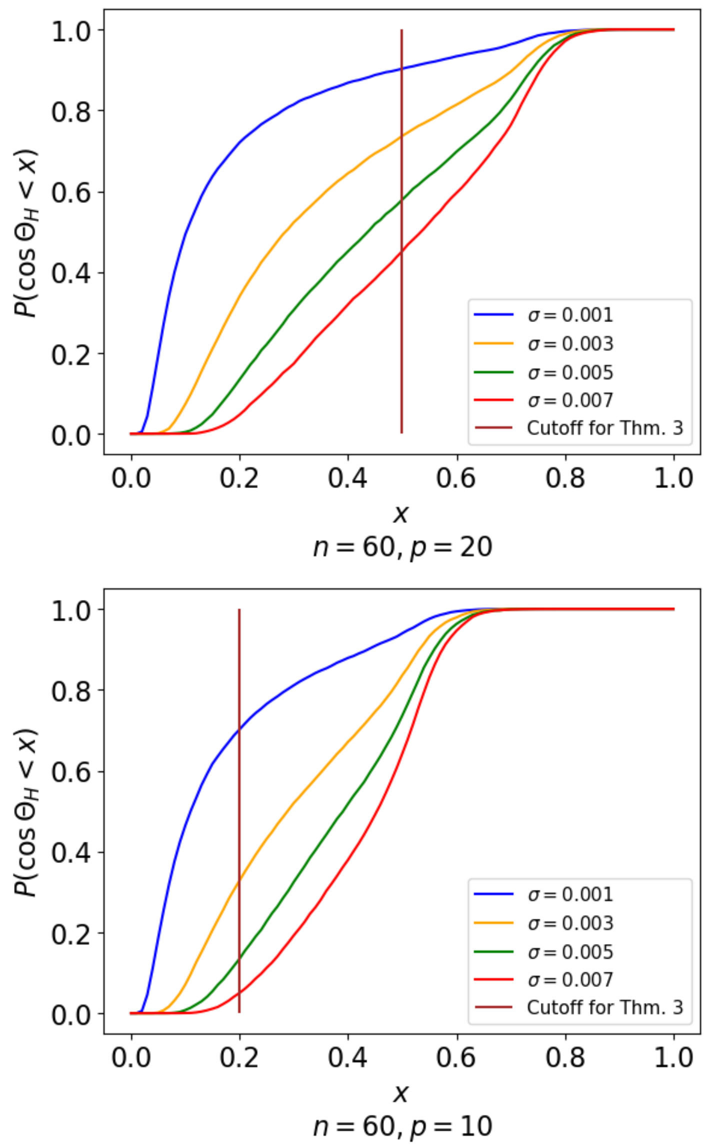


Figure 5.5: Empirical distribution of $\cos\theta_{\text{H}}$ under the generative process described in 5.4

In this section, we provided a construction of a family of matrices on which we can obtain good convergence guarantees. However, it is still an open question how to systematically construct these families, potentially without such a high source of heterogeneity as defined in the present case. Ideally, either the results about Gaussian matrices from 5.3 or the bound presented in Theorems 3, 4 can be strengthened, resulting in a conclusion that Gaussian matrices are high-heterogeneity matrices.

# Chapter 6

# Conclusion

When designing distributed algorithms for solving linear systems, one can take two very different approaches: optimization-based and projection-based algorithms. Optimization-based algorithms mostly rely on off-the-shelf optimization methods like the Heavy Ball Method. Projection-based methods more closely embody the approach of dividing the system into many underdetermined equations, finding a solution to them using familiar approaches such as Gaussian elimination, and finally guiding them to a consensus. The best-performing method of optimization-based algorithms is the Heavy Ball Method, while the best-performing method of the projection-based algorithms is the Accelerated Projection-Based Consensus.

The differences in design make any immediate comparison of the convergence rate of the two types of algorithms a daunting task, which is especially relevant when considering the empirical success of projection-based algorithms. To this end, we introduce a novel notion of data heterogeneity, which we use to give provable guarantees on the convergence rates of the algorithms. We exploit these guarantees to explain why projection-based algorithms are inclined to perform better in real-life scenarios.

We provide fast distributed algorithms for computing all of our heterogeneity metrics. Then, we empirically validate that our metrics behave well asymptotically and check if our theorems explain the numerical phenomena seen in our experiments.

A practitioner may or may not choose to use these algorithms to determine the precise notion of heterogeneity in the system. However, they might use vague heuristics on the heterogeneity of the data based on the sources it is collected from. For example, maybe we are collecting data with sensors, and at each moment in time, a subset of sensors dominates, essentially the scenario described in 5.4. To account for these qualitative heuristics, we conclude with a figure that summarizes our results in all possible qualitative scenarios:

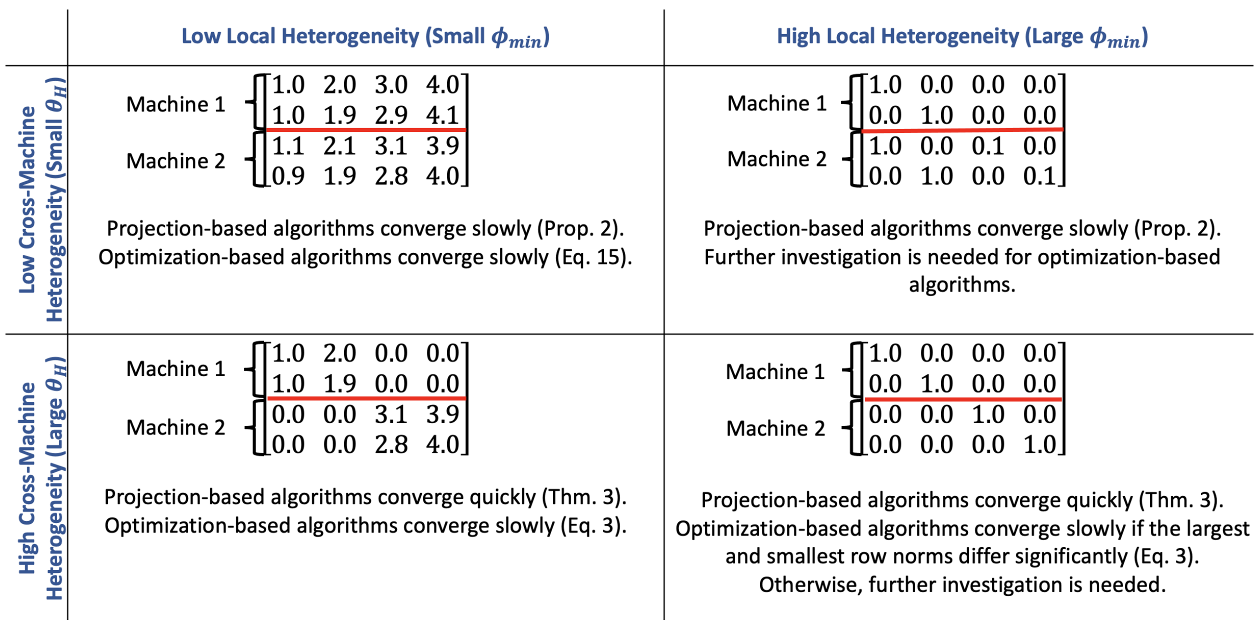


Figure 6.1: Performance of projection-based and optimization-based methods for different local and cross-machine heterogeneity regimes

# References


[1] 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.

[2] 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.

[3] 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.

[4] 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, A1248–A1269, 2015.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 8394–8399. DOI: 10.1109/CDC49753.2023.10383603.

[13] Å. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of Computation*, vol. 27, no. 123, pp. 579–594, 1973, ISSN: 00255718, 10886842. [Online]. Available: http://www.jstor.org/stable/2005662 (visited on 11/06/2023).

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

[15] 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*, 2024. arXiv: 2304.10640 [cs.DC].

[16] R. J. Muirhead, *Aspects of Multivariate Statistical Theory*. John Wiley & Sons, Inc, 1982.

[17] A. T. James, "Normal multivariate analysis and the orthogonal group," *The Annals of Mathematical Statistics*, vol. 25, no. 1, pp. 40–75, 1954, ISSN: 00034851. [Online]. Available: http://www.jstor.org/stable/2236512 (visited on 03/05/2024).

[18] P.-A. Absil, A. Edelman, and P. Koev, "On the largest principal angle between random subspaces," *Linear Algebra and its Applications*, vol. 414, no. 1, pp. 288–294, 2006, ISSN: 0024-3795. DOI: https://doi.org/10.1016/j.laa.2005.10.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0024379505004878.