Understanding Limitation of Two Symmetrized Orders by Worst-case Complexity

Peijun Xiao * Zhisheng Xiao † Ruoyu Sun ‡

November 4, 2019

Abstract

It was recently found that the standard version of multi-block cyclic ADMM diverges. Interestingly,Gaussian Back Substitution ADMM (GBS-ADMM) and symmetric Gauss-Seidel ADMM (sGS-ADMM) do not have the divergence issue. Therefore, it seems that symmetrization can improve the performance of the classical cyclic order. In another recent work, cyclic CD (Coordinate Descent) was shown to be $O(n^2)$ times slower than randomized versions in the worst-case. A natural question arises: can the symmetrized orders achieve a faster convergence rate than the cyclic order, or even getting close to randomized versions? In this paper, we give a negative answer to this question. We show that both Gaussian Back Substitution and symmetric Gauss-Seidel order suffer from the same slow convergence issue as the cyclic order in the worst case. In particular, we prove that for unconstrained problems, they can be $\mathcal{O}(n^2)$ times slower than R-CD. For linearly constrained problems with quadratic objective, we empirically show the convergence speed of GBS-ADMM and sGS-ADMM can be roughly $\mathcal{O}(n^2)$ times slower than randomly permuted ADMM.

1 Introduction

1.1 Background

Block decomposition is a simple yet powerful idea for solving large-scale computational problems. This idea is the key component of several popular methods such as coordinate descent (CD), SGD (Stochastic Gradient Descent) and ADMM (Alternating Direction Method of Multipliers).

Coordinate Descent Methods. Coordinate descent algorithms (CD) are iterative methods which update some coordinates of the variable vector while fixing the other coordinates at each iteration. CD is a popular choice for solving large-scale optimization problems (see [1] for a survey), including glmnet package for LASSO [2], libsvm package for support vector machine (SVM) [3–5], tensor decomposition [6], resource allocation in wireless communications [7], to name a few.

One of the major design choices for CD methods is the update order. There are two classes: deterministic orders, and randomized orders. For deterministic update order, the most basic variant is cyclic CD (C-CD),

*Coordinated Science Laboratory, Department of ISE, University of Illinois at Urbana-Champaign, Urbana, IL (peijux2@illinois.edu).

†Computational and Applied Mathematics, University of Chicago (zxiao@uchicago.edu).

‡Coordinated Science Laboratory, Department of ISE, University of Illinois at Urbana-Champaign, Urbana, IL (ruoyus@illinois.edu).

and another popular one is symmetric Gauss-Seidel CD (a.k.a. double-sweep CD). Among randomized variants, a popular choice in academia is independently-randomized CD (called R-CD for short), and a more popular method in practice is randomly permuted CD (RP-CD).

While CD is observed to be much faster than gradient descent methods (GD), is there any theoretical evidence for this observation? The strongest evidence is that R-CD is shown to be 1 to n times faster than GD ten years ago [8, 9]. Since then, most researchers have focused on randomized variants of CD [10–19]. Compared to the long history of CD methods (dating back to Gauss), it seems mysterious why the theoretical advantage of CD methods have been long lacking until only a decade ago. A very recent result gave a partial answer to this mystery: the most classical Gauss-Seidel order is actually $\mathcal{O}(n)$ times slower than GD and $\mathcal{O}(n^2)$ times slower than R-CD in the worst case [20], thus it is not surprising that the advantage of CD had not been established before the theoretical invetigation of randomized variants [8, 9].

An open question raised in [20] is: does there exist a deterministic variant of CD that achieves similar convergence speed to R-CD? This question is interesting due to several reasons. First, the complexity of deterministic algorithms is theoretically important, partly because we only have access to pseudo-randomness instead of randomness in practice. Second, it is not always feasible or easy to randomly pick coordinates due to hardware constraints. Third, understanding deterministic CD may help us better understand other algorithms such as ADMM, as randomized versions for those methods can be difficult to analyze. The third reason is one of the major motivations for this paper, and we will elaborate below.

ADMM. To solve large-scale problems with linear constraints, a natural idea is to combine CD methods with augmented Lagrangian method to obtain the so-called alternating direction method of multipliers (ADMM) [21–24]. Unlike CD where any reasonable update order can lead to convergence [25], for multi-block ADMM, even the most basic cyclic version does not converge [26]. Small step-size versions of multi-block ADMM can be shown to converge with extra assumptions on the problem (see, e.g. [27–42]), but the lesson from CD methods is that the speed advantage of CD exactly comes from large stepsize, thus we are more interested in ADMM with large step-size (such as step-size 1).

We are only aware of three major variants of multi-block ADMM with stepsize 1 that are convergent in numerical experiments: Gaussian back substitution ADMM (GBS-ADMM) [43, 44], symmetric Gauss-Seidel ADMM (sGS-ADMM) [45, 46] and randomly permuted ADMM (RP-ADMM) [47]. The first two use deterministic orders, and the third uses a random order. RP-ADMM is quite appealing since random permutation order is arguably the most popular update order for CD and SGD in practice. However, the theoretical analysis of random permutation is notoriously difficult even for CD and SGD [47–52], and for RP-ADMM only the expected convergence for quadratic objective function is known [47, 53]. In contrast, GBS-ADMM enjoys strong theoretical guarantee as the convergence for separable convex objective with linear constraints is proved [43]. For sGS-ADMM, the convergence guarantee is proved for a sub-class of convex problems.

If our purpose is just to resolve the divergence issue of cyclic ADMM, then GBS-ADMM and sGS-ADMM both provide rather satisfactory (though not perfect) theoretical guarantee on the convergence. However, the major purpose of using block decomposition is to solve large-scale problems, thus the convergence speed is also very important. What can we say about the convergence speed of GBS-ADMM and sGS-ADMM? Are they provably faster the one-block version, just like R-CD? To understand GBS-ADMM and sGS-ADMM, we need to first understand the case when GBS and sGS update orders are applied to unconstrained problems.

1.2 Main Contributions

We mainly study the two symmetrized versions of Gauss-Seidel order: symmetric Gauss-Seidel rule and Gaussian Back Substitution rule. Both orders can be applied to CD and ADMM on certain types of problems. We are interested in these update orders for a few reasons. First, besides cyclic order, they are arguably the most popular deterministic update order, thus analyzing them would provide a better understanding of deterministic orders in block decomposition. In addition, they have received revived interest recently (e.g. [54]). Second, they are among the few update orders that can enable the convergence of multi-block ADMM [44, 45].

We will give simple examples that CD and ADMM with sGS and GBS rules converge very slowly in practice, and provide rigorous proofs for the unconstrained examples. More specifically, the main contributions of this paper are summarized as below.

- We prove that for a certain class of examples, sGS-CD and GBS-CD converge at least $\mathcal{O}(n)$ times slower than GD and $\mathcal{O}(n^2)$ times slower than R-CD. Upper bounds of the two methods are also proved for quadratic problems, indicating that these gaps are tight. Therefore, these two symmetrized update orders are much slower than randomized order in the worst case.
- For constrained problems, to illustrate the slow speed of sGS-ADMM and GBS-ADMM in the worst case, we numerically study examples with non-zero linear constraints and zero or quadratic objective. Simulation results show that GBS-ADMM and sGS-ADMM are $\mathcal{O}(n)$ times slower than the single-block method ALM and roughly $\mathcal{O}(n^2)$ times slower than RP-ADMM.
- We propose a general framework of symmetrization, which improves the understanding of different update rules. In particular, we propose three basic symmetrization operations, and show that sGS order and GBS order can be obtained by combining these basic operations.

Table 1: Total complexity of sGS-CD, GBS-CD, C-CD and R-CD for solving a quadratic problem, with equal diagonal entries (ignoring a $\log 1/\epsilon$ factor). Here, κ_{CD} is a parameter determining the convergence speed of CD methods, explained in the footnote.

Algorithms	sGS-CD	GBS-CD	C-CD	R-CD
Upper bound	$n^4\kappa_{CD}$	$n^4\kappa_{CD}$	$n^4\kappa_{CD}$	$n^2\kappa_{CD}$
Lower bound	$\frac{1}{40}n^4\kappa_{CD}$	$\frac{1}{15}n^4\kappa_{CD}$	$\frac{1}{40}n^4\kappa_{CD}$	–

To help the readers understand the main theoretical results, in table 1 we compare our complexity bounds of sGS-CD and GBS-CD to C-CD [1] and R-CD in terms of κ_{CD}, a key parameter that determines the convergence speed of a block-decomposition method [2]. We ignore a $\log 1/\epsilon$ factor in the table. The upper bounds of sGS-CD and GBS-CD will be given in proposition 3.1 and proposition 3.3 respectively. The lower bounds of sGS-CD and GBS-CD will be given in theorem 3.1, and theorem 3.2 respectively.

1.3 Discussions

We remark that the worst-case slow convergence of an algorithm does not necessarily imply the slow convergence of an algorithm in practice. For instance, C-CD is shown to be up to $\mathcal{O}(n)$ times slower than GD

[1] The results of C-CD and R-CD are summarized in [20].

[2] In this table, $\kappa_{CD} = \frac{\lambda_{avg}(A)}{\lambda_{\min}(A)}$ where $\lambda_{avg}(A)$ and $\lambda_{\min}(A)$ are respectively the average of the eigenvalues and the minimal eigenvalue of the problem matrix A

in the worst case [20], but for almost all practical problems C-CD is faster than GD. It is an interesting open problem to explain the practical behavior of C-CD. Similarly, the slow convergence of algorithms with sGS and GBS orders does not imply that they are slow for practical problems, but knowing the worst-case performance provides better understanding of these algorithms.

Our results point out a significant gap: despite the extensive studies on ADMM, none of the existing variants of multi-block ADMM can inherit the advantage of R-CD: an improvement ratio of $O(1)$ to $\mathcal{O}(n)$ in convergence speed compared to the single-block method. Before our paper, it seems that sGS-ADMM and GBS-ADMM are the closest to this goal, but our results provide strong evidence that they are slow in the worst case. A remaining candidate for this goal is RP-ADMM, but this is a difficult task because even for RP-CD the precise convergence speed remains unproved. There are a few ways to fill in this theoretical gap: proposing a new method that achieves this goal, or proposing a new framework to justify the advantage of deterministic ADMM (which would justify the advantage of deterministic CD), or justifying the advantage of RP-ADMM. Each of these solutions would be rather non-trivial and quite interesting.

1.4 Notation, Terminology and Outline

Notation. Before we state the algorithms, we introduce the notation used in this paper. Throughout the paper, $A \in \mathbb{R}^{n\times n}$ is a symmetric positive semi-definite matrix. Let $\lambda_{\max}(A), \lambda_{\min}(A)$, and $\lambda_{\text{avg}}(A)$ respectively denote the maximum eigenvalue, minimum *non-zero* eigenvalue, and average eigenvalue of A respectively; sometimes we omit the argument A and just use $\lambda_{\max}$, $\lambda_{\min}$ and λ_{avg}. We denote the set of the eigenvalues of a matrix A as $\text{eig}(A)$ (allowing repeated elements if an eigenvalue has multiplicity larger than 1). The condition number of A is defined as $\kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$. We denote $\mathcal{R}(A) = \{Ax \mid x \in \mathbb{R}^n\}$ as the range space of A.

The less widely used notations are summarized below. An important notion is $\kappa_{CD} = \frac{\lambda_{avg}(A)}{\lambda_{\min}(A)}$, which is the key parameter that determines the convergence speed of a block-decomposition method. Further, we denote A_{ij} as the (i,j)-th entry of A and $L_i = A_{ii}$ as the i-th diagonal entry of A. Denote $L_{\max} = \max_i L_i$ and $L_{\min} = \min_i L_i$ as the maximum/minimum per-coordinate Lipschitz constant (i.e. maximum / minimum diagonal entry of A), and $L_{\text{avg}} = (\sum_{i=1}^n L_i)/n$ as the average of the diagonal entries of A (which is also the average of the eigenvalues of A). We use I to denote the identity matrix.

Terminology. Throughout the paper, we use "iteration" to denote one repeated procedure in an algorithm, and we use "pass" to denote one pass of all coordinates. In GD and C-CD, one iteration represents one iteration of updating all coordinates. In sGS-CD, one iteration represents one forward pass and one backward pass of updating all coordinates; in GBS-CD, one iteration represents one prediction step and one correction step on all coordiantes. Each iteration of GD (or C-CD) takes $\mathcal{O}(n^2)$ number of operations, and each iterations of sGS-CD (or GBS-CD) takes twice the number of operations of GD (or C-CD). As each iteration of different algorithms takes different numbers of operations, for a fair comparison, we focus on the total complexity of the algorithms.

Outline. The rest of the paper is organized as follows. In section 2, we present the algorithms discussed in the paper. In section 3, we present the upper bounds of the complexity of GBS-CD and sGS-CD and the lower bounds on the convergence rate of GBS-CD and sGS-CD. Proofs of these upper and lower bounds are given in the section 5 and section 6. In section 4, we discuss three symmetrization techniques for the iteration matrices. In section 7, we present some numerical results which show that for solving quadratic minimization or quadratic programming, there is also an $\mathcal{O}(n)$ gap. In section 8, we summarize the paper and discuss future research directions. Additional proofs of intermediate technical results are provided in

the appendix.

1.5 Overview of Techniques

In section 4, we introduce three symmetrization techniques: similar transformation, summation (or product) symmetrization and whole (or partial) symmetrization, which are useful in our proofs of the complexity.

For the first method sGS-CD, proving the upper bound of the convergence rate is similar to a standard proof of C-CD. More specifically, a standard method for analyzing C-CD is to prove sufficient descent. This proof is "iteration-independent" in the sense that the amount of decrease of each iteration is independent of other iterations, thus it also applies to sGS-CD. For the second method GBS-CD, proving the upper bound requires a different idea. The correction step is dependent on the prediction step, thus the technique of sufficient descent does not directly apply. Instead, we directly bound the spectral radius of the iteration matrix of GBS-CD. The details of these two proofs can be found in section 5.

To prove a lower bound of an algorithm, we need to analyze a worst-case example and calculate the spectral radius of the iteration matrix. In [20], computing the spectral radius for C-CD is rather straightforward since all eigenvalues can be written as the roots of a simple sparse polynomial. For sGS-CD and GBS-CD, the iteration matrices become more complicated and simple expressions of eigenvalues are difficult to obtain. We develop two different proof techniques to compute the spectral radius.

For sGS-CD, the iteration matrix is the product of a forward-pass matrix and a backward-pass matrix, making it complicated to compute eigenvalues. Recall that sGS-CD is a symmetrization for C-CD from algorithmic perspective, but its iteration matrix $(I - \Gamma^{-T}Q)(I - \Gamma^{-1}Q)$ is not symmetric. A fundamental question is: can we interpret sGS-CD as a symmetrization of C-CD from the perspective of iteration matrices? We use the following trick: we express sGS-CD as alternating projections, then the iteration matrix of sGS-CD becomes a symmetrization of the C-CD iteration matrix. This builds a link with C-CD method and makes the computation of spectral radius fesible. Details of the proof can be found in section 6.2.

The analysis of GBS-CD is quite different from that for sGS-CD. The connection between its iteration matrix and C-CD iteration matrix is quite weak. Our approach is to decompose the iteration matrix into several terms, compute the eigenvalues of each of them and eventually derive the lower bound of the convergence rate. This proof has an extra advantage. The worst-case example used in [20] and this paper has a hyper-parameter c which takes values in $(0, 1)$. The lower bound of [20] is only proved for c asymptotically approaching 1, while our lower bound on GBS-CD is tight for any $c \in (0, 1)$.

2 Algorithms

For simplicity, throughout the paper, we only discuss the case that the block size is 1.

We first briefly discuss the main ideas of the two update orders sGS and GBS. When solving quadratic problems by CD, the first method sGS-CD consists of a forward Gauss-Seidel iteration and a backward Gauss-Seidel iteration. The second method GBS-CD consists of a prediction step and a correction step, where the prediction step is a forward Gauss-Seidel iteration, and in the correction step the new iterate is obtained by utilizing the changes in the iterates after performing the prediction step. sGS-ADMM [45] and GBS-ADMM [43] are ADMM algorithms based on sGS and GBS update rules, and they are used to solve linearly constrained problems. The first method sGS-ADMM [45] iteratively updates the primal variables by sGS order and then update the dual variable. The second method GBS-ADMM [43] first performs the prediction step on the primal variable and a update on the dual variable, then performs a correction step on

the primal variable. More details are given below.

2.1 Symmetric Gauss-Seidel Order

When applying CD to unconstrained problems, we focus on solving convex quadratic functions in the rest the paper. sGS-CD, also called Aitken's double sweep method [55], is presented in algorithm 1 to solve a convex quadratic problem:

$$\min_{x\in\mathbb{R}^n} \quad f(x) = \frac{1}{2}\|Ax-b\|^2 \tag{1}$$

In each iteration, sGS-CD performs a forward pass and a backward pass. In the forward pass, we update the coordinates in the order $(1,2,\ldots,n)$. In the backward pass, we update the coordinates in the reverse order $(n-1,\ldots,1)$. We call an update rule which updates the coordinates in the order of $(1,2,\ldots,n,n-1,\ldots,1)$ as the **symmetric Gauss-Seidel (sGS) update rule**.

Algorithm 1 n-block symmetric Gauss-Seidel Coordinate Descent (sGS-CD)

To solve eq. (1), denote $f(x) = \frac{1}{2}||Ax-b||^2$.
Initialization: $x_i^0 \in \mathbb{R}^{d_i\times 1}, i = 1,\ldots,n$.
Iteration k $(k = 0,1,2,\ldots)$:
Forward Pass:

$$x_1^{k+\frac{1}{2}} \in \underset{x_1}{\operatorname{argmin}} f\left(x_1, x_2^{k-1}, x_3^{k-1}, \ldots x_n^{k-1}\right) \tag{2}$$

$$x_2^{k+\frac{1}{2}} \in \underset{x_2}{\operatorname{argmin}} f\left(x_1^{k+\frac{1}{2}}, x_2, x_3^{k-1}, \ldots x_n^{k-1}\right) \tag{3}$$

$$\ldots \tag{4}$$

$$x_n^{k+\frac{1}{2}} \in \underset{x_n}{\operatorname{argmin}} f\left(x_1^{k+\frac{1}{2}}, x_2^{k+\frac{1}{2}}, x_3^{k+\frac{1}{2}}, \ldots x_n\right) \tag{5}$$

Backward Pass:

$$x_n^{k+1} = x_n^{k+\frac{1}{2}} \tag{6}$$

$$x_{n-1}^{k+1} \in \underset{x_{n-1}}{\operatorname{argmin}} f\left(x_1^{k+\frac{1}{2}}, x_2^{k+\frac{1}{2}}, \ldots, x_{n-2}^{k+\frac{1}{2}}, x_{n-1}, x_n^{k+1}\right) \tag{7}$$

$$\ldots \tag{8}$$

$$x_1^{k+1} \in \underset{x_1}{\operatorname{argmin}} f\left(x_1, x_2^{k+1}, x_3^{k+1}, \ldots x_n^{k+1}\right) \tag{9}$$

Remark 2.1. *In the definition of sGS-CD, the n-th coordinate is not updated in the backward pass. If we update n-th coordinate in the backward pass, i.e change the line* $x_n^{k+1} = x_n^{k+\frac{1}{2}}$ *in algorithm 1 into*

$$x_n^{k+1} \in \underset{x_n}{\operatorname{argmin}} f\left(x_1^{k+\frac{1}{2}}, x_2^{k+\frac{1}{2}}, \ldots, x_{n-1}^{k+\frac{1}{2}}, x_n\right),$$

then the value x_n^{k+1} *is still as the same as* $x_n^{k+\frac{1}{2}}$*, since* x_n^{k+1} *is optimal given current values of other coordinates.*

To solve a linearly constrained problem eq. (10),

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x_1,\cdots,x_n) \\ \text{s.t.} \quad & \sum_i A_i x_i = b, \end{aligned} \tag{10}$$

we consider the augmented Lagrangian function

$$\mathcal{L}(x_1,\ldots,x_n;\lambda) = f(x_1,\ldots,x_n) - \lambda^T(\sum_i A_i x_i - b) + \frac{\beta}{2}\|\sum_i A_i x_i - b\|^2. \tag{11}$$

Algorithm 2 n-block symmetric Gauss-Seidel ADMM (sGS-ADMM)

To solve eq. (10), define $\mathcal{L}(x_1,\ldots,x_n;\lambda)$ as in eq. (11).
Initialization: $x_i^0 \in \mathbb{R}^{d_i\times 1}, i=1,\ldots,n;\ \lambda^0\in\mathbb{R}^{n\times 1}$. Let $\sigma>0$ and $\beta>0$. Choose $T_i \succcurlyeq 0$ for $i=1,\cdots,n$.
Iteration k ($k=0,1,2,\ldots$):

Forward Pass:

$$x_1^{k+\frac{1}{2}} \in \operatorname*{argmin}_{x_1} \mathcal{L}\left(x_1, x_2^{k-1}, x_3^{k-1}, \ldots x_n^{k-1}; \lambda^k\right) + \frac{\sigma}{2}\|x_1 - x_1^{k+\frac{1}{2}}\|^2_{T_1} \tag{12}$$

$$x_2^{k+\frac{1}{2}} \in \operatorname*{argmin}_{x_2} \mathcal{L}\left(x_1^{k+\frac{1}{2}}, x_2, x_3^{k-1}, \ldots x_n^{k-1}; \lambda^k\right) + \frac{\sigma}{2}\|x_2 - x_2^{k+\frac{1}{2}}\|^2_{T_2} \tag{13}$$

$$\cdots \tag{14}$$

$$x_n^{k+\frac{1}{2}} \in \operatorname*{argmin}_{x_n} \mathcal{L}\left(x_1^{k+\frac{1}{2}}, x_2^{k+\frac{1}{2}}, x_3^{k+\frac{1}{2}}, \ldots x_n; \lambda^k\right) + \frac{\sigma}{2}\|x_n - x_n^{k+\frac{1}{2}}\|^2_{T_n} \tag{15}$$

Backward Pass:

$$x_n^{k+1} = x_n^{k+\frac{1}{2}} \tag{16}$$

$$x_{n-1}^{k+1} \in \operatorname*{argmin}_{x_{n-1}} \mathcal{L}\left(x_1^{k+\frac{1}{2}}, x_2^{k+\frac{1}{2}}, \ldots, x_{n-1}, x_n^{k+1}; \lambda^k\right) + \frac{\sigma}{2}\|x_{n-1} - x_{n-1}^{k+1}\|^2_{T_{n-1}} \tag{17}$$

$$\cdots \tag{18}$$

$$x_1^{k+1} \in \operatorname*{argmin}_{x_1} \mathcal{L}\left(x_1, x_2^{k+1}, x_3^{k+1}, \ldots x_n^{k+1}; \lambda^k\right) + \frac{\sigma}{2}\|x_1 - x_1^{k+1}\|^2_{T_1} \tag{19}$$

Dual Update:

$$\lambda^{k+1} = \lambda^k - \beta(A_1 x_1^{k+1} + \cdots + A_n x_n^{k+1} - b). \tag{20}$$

The general sGS-ADMM is defined in algorithm 2 for solving the constrained problem eq. (10), where each T_i is a self-adjoint positive semidefinite linear operator that satisfies the conditions mentioned in [45]. As our goal is to understand the worst-case convergence rate, we consider a special setting that $T_i = 0,\ \forall i$, which for the unconstrained problems reduces to the sGS-CD algorithm 1. In the section of experiments, we will consider sGS-ADMM with $T_i = 0, \forall i$, for linearly constrained problems.

2.2 Gaussian Back Substitution Order

In this subsection, we again focus on solving the convex quadratic problem eq. (1) with a matrix A and a vector b.

Suppose Γ is the lower triangular matrix of $Q \triangleq A^T A$ (with the diagonal entries), i.e., $\Gamma_{ij} = Q_{ij}$, if $i \geq j$ and 0 otherwise. Let x^* to be the optimal solution of eq. (1). The prediction step of GBS-CD is a regular iteration of C-CD method, and can be written as (see Appendix A for explanation):

$$\tilde{x}^k - x^* = (I - \Gamma^{-1}Q)(x^k - x^*). \tag{21}$$

Define

$$B \triangleq \begin{bmatrix} 1 & 0 \\ 0 & \Gamma_{2:n}^{-T} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & D_{2:n} \end{bmatrix}, \tag{22}$$

where Γ^{-T} is the transpose of the inverse of Γ and D is the diagonal matrix of Q. $\Gamma_{2:n}^{-T}$ and $D_{2:n}$ are the sub-matrices obtained by excluding the first row and first column of $\Gamma_{2:n}^{-T}$ and $D_{2:n}$ respectively .

The correction update step of GBS-CD at $(k+1)$-th iteration is

$$x^{k+1} = x^k - B(x^k - \tilde{x}^k). \tag{23}$$

This step eq. (23) corrects the prediction of $\tilde{x}^k$ in eq. (21). The variables $(x_2, \cdots, x_n)$ are updated in the back substitution order in eq. (23), thus this update rule is called **Gaussian back substitution (GBS) update rule**.

Combing the prediction and correction step together, we obtain the update rule as the following:

$$x^{k+1} - x^* = (I - B\Gamma^{-1}Q)(x^k - x^*) \tag{24a}$$

Note that B defined in eq. (22) is quite close to Γ^{-T} when all the diagonal entries of Q are 1, thus the iteration matrix $I - B\Gamma^{-1}Q$ is quite close to $I - \Gamma^{-T}\Gamma^{-1}Q$. Compared to the iteration matrix of C-CD $I - \Gamma^{-1}Q$, GBS-CD provides a "product symmetrization" which replaces Γ by $\Gamma^T\Gamma$ (with a tiny difference).

When applying GBS update order to ADMM for solving linearly constrain problem eq. (10), the prediction step is a regular primal update iteration of Cyclic ADMM (C-ADMM). After the prediction step, GBS-ADMM updates the dual variable using the predicted primal variable $\tilde{x}^k$ as shown in eq. (36).

To derive the correction step of GBS-ADMM, we first denote $\Omega \triangleq A^T A$ where A is the linear constraint matrix in problem eq. (10), and Γ_Ω is the lower triangular matrix of Ω (with the diagonal entries).

We define the correction matrix F as

$$F \triangleq \begin{bmatrix} 1 & 0 \\ 0 & [\Gamma_\Omega^{-T}]_{2:n} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & [D_\Omega]_{2:n} \end{bmatrix}, \tag{30}$$

where Γ_Ω^{-T} is the transpose of the inverse of Γ_Ω and D_Ω is the diagonal matrix of Ω. $[\Gamma_\Omega^{-T}]_{2:n}$ and $[D_\Omega]_{2:n}$ are the sub-matrices by excluding the first row and first column of Γ_Ω^{-T} and D_Ω respectively.

The correction step of GBS-ADMM at iteration $k+1$ is

$$x^{k+1} = x^k - F(x^k - \tilde{x}^k). \tag{31}$$

Detailed GBS-ADMM algorithm is presented in algorithm 4.

Algorithm 3 n-block Gaussian Back Substitution Coordinate Descent (GBS-CD)

To solve eq. (1), denote $f(x) = \frac{1}{2}||Ax - b||^2$ and B is defined in eq. (22).
Initialization: $x_i^0 \in \mathbb{R}^{d_i \times 1}, i = 1, \ldots, n$.
Iteration k ($k = 0, 1, 2, \ldots$):
Prediction Step:

$$\tilde{x}_1 \in \underset{x_1}{\operatorname{argmin}} f\left(x_1, x_2^{k-1}, x_3^{k-1}, \ldots x_n^{k-1}\right) \tag{25}$$

$$\tilde{x}_2 \in \underset{x_2}{\operatorname{argmin}} f\left(\tilde{x}_1, x_2, x_3^{k-1}, \ldots x_n^{k-1}\right) \tag{26}$$

$$\ldots \tag{27}$$

$$\tilde{x}_n \in \underset{x_n}{\operatorname{argmin}} f\left(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \ldots x_n\right) \tag{28}$$

Correction Step:

$$x^{k+1} = x^k - B(x^k - \tilde{x}^k) \tag{29}$$

Remark 2.2. *Strictly speaking, GBS-CD (defined in algorithm 3) is not a special form of GBS-ADMM (defined in algorithm 4) to solve unconstrained problems. Although the correction matrix B eq.* (22) *and F eq.* (30) *for GBS-CD and GBS-ADMM are in the same form, they are different: B is constructed from $Q = A^T A$, where A appears in the quadratic objective of eq.* (1)*, but F is constructed from $\Omega = A^T A$ where A is the matrix of linear constraint in eq.* (10)*. Note that the original GBS-ADMM is only defined for a separable convex objective function. We presented a version for general convex objective, but as our goal is to reveal the limitation of GBS-ADMM, we do not need to consider the most general form, but only need to study some special forms (objective is zero or quadratic).*

If the objective function is quadratic as in eq. (1)*, we can define a new version of GBS-ADMM (which can be called obj-GBS-ADMM) as follows: for the correction step, we construct the matrix F from the matrix in the objective function instead of from the matrix in the linear constraint. GBS-CD defined in algorithm 3 can be viewed as a special form of this obj-GBS-ADMM for unconstrained quadratic problems.*

For theoretical analysis, we will prove that GBS-CD can be very slow. For GBS-ADMM, we will provide numerical experiments to show that it can be very slow for a bad example. The theoretical evidence for GBS-CD and the empirical evidence for GBS-ADMM together indicate that GBS-ADMM can be very slow in the worst case.

3 Main Results

Consider the quadratic minimization problem

$$\min_x f(x) \triangleq x^T Q x - 2b^T x,$$

where $Q \in \mathbb{R}^{n \times n}$ is symmetric positive semi-definite, $b \in \mathcal{R}(Q)$ and $Q_{ii} \neq 0, \ \forall i$.

We can assume $b \in \mathcal{R}(Q)$ since otherwise the minimum value of $\min_x x^T Q x - 2b^T x$ will be $-\infty$. We can assume $Q_{ii} \neq 0, \ \forall i$, since when some $Q_{ii} = 0$ all entries in the i-th row and the i-th column of Q should be

Algorithm 4 n-block Gaussian Back Substitution ADMM (GBS-ADMM)

To solve eq. (10), define $\mathcal{L}(x_1, \ldots, x_n; \lambda)$ as in eq. (11) and F as in eq. (30).
Initialization: $x_i^0 \in \mathbb{R}^{d_i \times 1}, i = 1, \ldots, n$.
Iteration k ($k = 0, 1, 2, \ldots$):
Prediction Step:

$$\tilde{x}_1 \in \underset{x_1}{\operatorname{argmin}} \mathcal{L}\left(x_1, x_2^{k-1}, x_3^{k-1}, \ldots x_n^{k-1}; \lambda^k\right) \tag{32}$$

$$\tilde{x}_2 \in \underset{x_2}{\operatorname{argmin}} \mathcal{L}\left(\tilde{x}_1, x_2, x_3^{k-1}, \ldots x_n^{k-1}; \lambda^k\right) \tag{33}$$

$$\ldots \tag{34}$$

$$\tilde{x}_n \in \underset{x_n}{\operatorname{argmin}} \mathcal{L}\left(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \ldots x_n; \lambda^k\right) \tag{35}$$

Dual Update:

$$\lambda^{k+1} = \lambda^k - \beta(A_1\tilde{x}_1 + \cdots + A_n\tilde{x}_n - b) \tag{36}$$

Correction Step:

$$x^{k+1} = x^k - F(x^k - \tilde{x}^k) \tag{37}$$

zero, which means that the i-th variable does not affect the objective and thus can be deleted. Recall that the maximum eigenvalue, minimum eigenvalue and average eigenvalue of Q are $\lambda_{\max}, \lambda_{\min}, \lambda_{\text{avg}}$ respectively, the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$, and $\kappa_{CD} = \lambda_{\text{avg}}/\lambda_{\min}$.

We first summarize the main results in table 2. The upper bounds will be given in proposition 3.1, proposition 3.2, proposition 3.3, and the lower bounds will be given in theorem 3.1 and theorem 3.2. In this table, we ignore the $\log 1/\epsilon$ factor, which is always necessary for an iterative algorithm to achieve error ϵ.

Table 2: Complexity of sGS-CD, GBS-CD, C-CD, R-CD, GD for equal-diagonal case (ignoring a $\log 1/\epsilon$ factor)

Algorithms	κ	κ_{CD}
sGS-CD Upper bound (proposition 3.1)	$n^3\kappa$	$n^4\kappa_{CD}$
sGS-CD Lower bound (theorem 3.1)	$\frac{1}{40}n^3\kappa$	$\frac{1}{40}n^4\kappa_{CD}$
GBS-CD Upper bound (proposition 3.3)	$n^3\kappa$	$n^4\kappa_{CD}$
GBS-CD Lower bound (theorem 3.2)	$\frac{1}{15}n^3\kappa$	$\frac{1}{15}n^4\kappa_{CD}$
GD	$n^2\kappa$	–
R-CD	–	$n^2\kappa_{CD}$
C-CD Upper Bound	$n^3\kappa$	$n^4\kappa_{CD}$
C-CD Lower Bound	$\frac{1}{40}n^3\kappa$	$\frac{1}{40}n^4\kappa_{CD}$

This table shows that the lower bounds match the upper bounds up to some constant factor. In addition, the table reveals the relations between the worst-case complexity of sGS-CD, GBS-CD, GD, R-CD and

C-CD.

The main implications of our results are the following:

- In the worst case, sGS-CD and GBS-CD are $\mathcal{O}(n)$ times slower than GD, and $\mathcal{O}(n^2)$ times slower than R-CD.
- sGS-CD and GBS-CD are as slow as C-CD up to a constant factor in the worst case.

Now we formally state the upper bounds and lower bounds on the convergence rate of sGS-CD and GBS-CD.

Proposition 3.1. *(Upper bound of sGS-CD) Consider the quadratic minimization problem* $\min_x f(x) \triangleq x^T A^T A x - 2b^T x$ *where* $A \in \mathbb{R}^{n\times n}$, $Q = A^T A$, $b \in \mathcal{R}(Q)$ *and* $Q_{ii} \neq 0,\ \forall i$. *For any* $x^0 \in \mathbb{R}^n$, *let* x^k *denotes the output of sGS-CD after* k *iterations, then*

$$f(x^{k+1}) - f^* \leq \left(\min\left\{1 - \frac{1}{n\kappa}\frac{L_{\min}}{L_{\text{avg}}}, 1 - \frac{L_{\min}}{L(2+\log n/\pi)^2}\frac{1}{\kappa}\right\}\right)^2 (f(x^k) - f^*). \tag{38}$$

Here, f^* *is the minimum value of the function* f.

Proposition 3.2. *(Upper bound of GBS-CD for relative iterates error) Consider the quadratic minimization problem* $\min_x f(x) \triangleq x^T A^T A x - 2b^T x$ *where* $A \in \mathbb{R}^{n\times n}$, $Q = A^T A$, $b \in \mathcal{R}(Q)$ *and* $Q_{ii} \neq 0,\ \forall i$. *For any* $x^0 \in \mathbb{R}^n$, *let* x^k *denotes the output of GBS-CD after* k *iterations, then*

$$\|x^{k+1} - x^*\| \leq \alpha\left(1 - \frac{1}{\kappa \cdot \min\left\{\sum_i L_i, (2+\frac{1}{\pi}\log n)^2 L\right\}}\right)^k \|x^0 - x^*\|. \tag{39}$$

Here, x^* *is the minimum of function* f. *The constant* α *is* $\frac{\|\Gamma^T\|}{\sqrt{\lambda_{\min}(\Gamma^T\Gamma)}}$ *where* Γ *is the lower triangular matrix of* Q.

Proposition 3.3. *(Upper bound of GBS-CD for objective error) Consider the quadratic minimization problem* $\min_x f(x) \triangleq x^T A^T A x - 2b^T x$ *where* $A \in \mathbb{R}^{n\times n}$, $Q = A^T A$, $b \in \mathcal{R}(Q)$ *and* $Q_{ii} \neq 0,\ \forall i$. *For any* $x^0 \in \mathbb{R}^n$, *let* x^k *denotes the output of GBS-CD after* k *iterations, then*

$$f(x^{k+1}) - f^* \leq \left(1 - \frac{1}{\kappa \cdot \min\left\{\sum_i L_i, (2+\frac{1}{\pi}\log n)^2 L\right\}}\right)^2 (f(x^k) - f^*) \tag{40}$$

Here, f^* *is the minimum value of the function* f.

Theorem 3.1. *(Lower bound of sGS-CD) For any initial point* $x^0 \in \mathbb{R}^n$, *any* $\delta \in (0, 1]$, *there exists a quadratic function* $f(x) = x^T A x - 2b^T x$ *such that*

$$f(x^k) - f^* \geq (1-\delta)\left(1 - \frac{4\pi^2}{n\kappa}\right)^{2k+2} (f(x^0) - f^*),\ \forall k, \tag{41}$$

where x^k *denotes the output of sGS-CD after* k *iterations,* f^* *is the minimum of the objective function* f.

Theorem 3.2. *(Lower bound of GBS-CD) For any initial point* $x^0 \in \mathbb{R}^n$, *for any* $\delta \in (0, 1]$, *there exists a quadratic function* $f(x) = x^T A x - 2b^T x$ *such that*

$$f(x^k) - f^* \geq (1-\delta)\left(1 - \frac{3\pi^2}{(12-\pi^2)n\kappa c}\right)^{2k+2} (f(x^0) - f^*),\ \forall k. \tag{42}$$

where f^* is the minimum of the objective function f and $c \in (0,1)$ is a constant defined for the quadratic function.

The proofs of proposition 3.1, proposition 3.2, proposition 3.3 will be given in section 5, and the proofs of theorem 3.1 and theorem 3.2 will be given in section 6. The overview of the proofs are presented before the formal proofs in section 5 and section 6.

Remark 1: The example we construct for theorem 3.1 and theorem 3.2 is simple: all diagonal entries of Q are 1 and all off-diagonal entries are c, where c takes values in $(0,1)$. This example has been studied in [20] to show the worst case complexity of C-CD and in [49] to show the good convergence behavior of RP-CD. More discussions of this example will be given in section 6.1.

Remark 2. Recall we use "iteration" to denote one repeated procedure in an algorithm such that each iterations of sGS-CD (or GBS-CD) takes twice the number of operations of GD (or C-CD). Since each iteration takes different numbers of operations among the algorithms, then for a fair comparison, our discussion focuses on the total complexity of the algorithms throughout this paper.

Now we discuss how to obtain table 2 from the above results. We say an algorithm has complexity $\tilde{\mathcal{O}}(g(n,\theta))$, if it takes $\mathcal{O}(g(n,\theta)\log(1/\epsilon))$ unit operations to achieve relative error ϵ. As we discussed before, each iteration of GD, C-CD and R-CD will take $\mathcal{O}(n^2)$ operations, and each iteration of sGS-CD and GBS-CD will take twice the number of operations of GD. Using the fact $-\ln(1-z) \geq -z, z \in (0,1)$, one can immediately show that to achieve $(1-1/u)^k \leq \epsilon$, one only needs $k \geq u\log(1/\epsilon)$ iterations. Thus we can transform the convergence rate to the number of iterations, then the total time complexity.

Consider the equal-diagonal case for now, then $\frac{L}{L_{\min}} = \frac{\lambda_{\max}}{\lambda_{\text{avg}}}$ and $\kappa_{\text{CD}} = \frac{L_{\text{avg}}}{\lambda_{\min}} = \frac{\lambda_{\text{avg}}}{\lambda_{\min}}$ and the upper bound of sGS-CD on convergence rate (eq. (38)) can be transformed to the following upper bound of complexity

$$\tilde{\mathcal{O}}\left(n^3\kappa\right) \quad \text{or} \quad \tilde{\mathcal{O}}\left(n^4\kappa_{\text{CD}}\right). \tag{43}$$

These two quantities are those entries of sGS-CD upper bound in table 2. Similarly, the other bounds on convergence rate in theorem 3.1, proposition 3.3 and theorem 3.2 can be transformed to corresponding bounds on the complexity, and they form the rest of table 2.

4 Symmetrization Rules

Both sGS and GBS update rules are motivated by symmetrizing Gauss-Seidel rule. What are their differences and relations? Can we understand them from a general framework, viewing them as special cases of general principles? This can help us understand and analyze these algorithms, and can potentially lead to the design of new algorithms.

We will focus on the perspective of iteration matrix, rather than algorithmic perspective. To illustrate the difference of these two perspectives, consider sGS rule. The "algorithmic symmetrization" means that the update order $1,2,\ldots,n$ is symmetrized to $1,2,\ldots,n,n-1,\ldots,1$; however, the iteration matrix of sGS-CD is not even symmetric, thus it is not clear what are the principles of symmetrization for sGS-CD from the perspective of iteration matrix. This section will answer this question.

In this section, we introduce three symmetrization rules that can be applied to any non-symmetric iteration matrix, and explain each of them with examples.

- (R1) similar transformation
- (R2) summation rule and product rule

- (R3) whole symmetrization vs partial symmetrization

(R1) similar transformation Recall that similar matrices share the same set of eigenvalues, thus if A or B is invertible, then matrix AB and BA have the same set of eigenvalues. This basic fact can be used to symmetrize a matrix. The simplest example is that BAA^T can be symmetrized to A^TBA. In many cases, this transformation is straightforward, but in certain cases this rule can be applied in a more complicated way (see below for the discussion on sGS-CD).

(R2) product rule and summation rule Given a matrix U, the summation rule means to symmetrize U by $U+U^T$, and the product rule means to symmetrize U by UU^T or U^TU. For this paper, we only deal with product rule, and we will briefly discuss how summation rule can be used.

(R3) partial symmetrization

If a matrix consists of several parts, and only some are non-symmetric, then to symmetrize the matrix we only need to symmetrize the non-symmetric parts, which we call "partial symmetrization" rule. Note that symmetrizing the whole matrix is also a special case of the partial symmetrization rule. For example, a "partial" of a matrix $A+BC$ can be A, BC, B, C and $A+BC$, and we can choose to symmetrize any part that is non-symmetric.

Example 1: For a matrix SA where A is symmetric and S is symmetric positive-semidifinite, we can apply R1 to symmetrize it to UAU^T where $S=U^TU$.

Example 2: For a matrix SA where A is non-symmetric and S is symmetric, we can apply R2 to get $(SA)(SA)^T$, or apply R2 partially to get A^TSA.

Example 3: For a matrix $S+A$ where S is symmetric and A is non-symmetric, we can apply R2 to get $(S+A)(S+A)^T$, or apply R2 partially to get $S+AA^T$.

Having understood these simple examples, we can proceed to analyze sGS-CD and GBS-CD . Recall that the iteration matrix of C-CD is

$$M_{\mathrm{C-CD}} = I - \Gamma^{-1}Q,$$

where $Q=A^TA$ is symemtric positive-semidefinite, and Γ^{-1} is non-symmetric.

The original iteration matrix of the second method GBS-CD is quite close to (not the same, but for simplicity, let us anlayze this version)

$$I - \Gamma^{-T}\Gamma^{-1}Q.$$

This is a partial symmetrization to $M_{\mathrm{C-CD}}$: pick the second term (partial rule for the sum), and and apply product symmetrization to the non-symmetric part Γ^{-1} (partial rule for the product).

The original iteration matrix of sGS-CD is quite close to (but not the same as)

$$J_1 = (I - \Gamma^{-T}Q)(I - \Gamma^{-1}Q).$$

This is not a symmetric matrix. We can re-write it as $J_1 = (I - \Gamma^{-T}A^TA)(I - \Gamma^{-1}A^TA)$, and apply R1 to get

$$\begin{aligned} J_2 &= AJ_1A^{-1} \\ &= (A - A\Gamma^{-T}A^TA)(A^{-1} - \Gamma^{-1}A^T) \\ &= (I - A\Gamma^{-T}A^T)AA^{-1}(I - A\Gamma^{-1}A^T) \\ &= (I - A\Gamma^{-T}A^T)(I - A\Gamma^{-1}A^T), \end{aligned} \tag{44}$$

which becomes a symmetric matrix now. Based on the above computation, we can build a sequence of symmetrization procedures that transform $M_{\mathrm{C-CD}}$ to J_1:

Step 1: Apply R1 to change $M_{\mathrm{C-CD}}$ to $I - A\Gamma^{-T}A^T$;

Step 2: Apply product symmetrization to get $J_2 = (I - A\Gamma^{-T}A^T)(I - A\Gamma^{-1}A^T)$;

Step 3: Apply R1 to get $J_1 = (I - \Gamma^{-T}A^TA)(I - \Gamma^{-1}A^TA) = (I - \Gamma^{-T}Q)(I - \Gamma^{-1}Q)$.

Now we obtain an interesting interpretation of the two methods. From the perspective of iteration matrix, the major difference between sGS-CD and GBS-CD (ignoring the similarity transformation) is: GBS-CD tries to symmetrize the whole matrix (in a somewhat complicated way), while sGS-CD only symmeterizes the non-symmetric part.

We can give one more example that is a special case of this general framework. Consider the following random-sGS order: at each iteration, with probabiliity 1/2, use the forward order $(1, 2, \ldots, n)$; with probability 1/2, use the backward order $(n, n-1, \ldots, 1)$. The (expected) iteration matrix of random-sGS-CD is $I - \frac{1}{2}(\Gamma^{-T} + \Gamma^{-1})Q$. This matrix can be obtained by applying partial summation-symmetrization rule to $M_{\mathrm{C-CD}}$ (applying summation rule to Γ^{-1}).

With this framework, we can have a better understanding of these algorithms, and can answer a few questions.

- *Question*: Why are there two different symmetrization methods sGS and GBS?

  Answer: Because the product symmetrization can be applied to either the whole matrix or a part.

- *Question*: Are sGS and GBS special, or outcomes of principled design?

  Answer: They are the natural outcomes of applying product symmetrization to the iteration matrix.

- *Question*: What is a principled procedure to design symmetrized algorithms?

  Answer: A possible two-step procedure is as follows: (i) choose a combination of symmetrization rules (product rule, summation rule or others) and a part of the iteration matrix, to obtain a symmetric matrix; (ii) apply similar transformation rule, to transform this matrix to a form that is related to an iterative method. The first step is relatively simple, as we can construct various symmetrized versions of M_{CD}, but not all of them can be related to a real iterative method. For instance, $I-(\Gamma^{-1}Q+Q\Gamma^{-T})$ is a symmetrizatino of M_{CD}, but seems not related to an algorithm. We suspect sGS-CD, GBS-CD and random-sGS-CD are three simplest methods obtained by the combination of our three symmetrization rules.

Besides the high-level understanding of the two methods, the above analysis has great impact on the proof techniques. First, symmetrizing GBS-CD in the above way is crucial for proving the upper bound of GBS-CD. Second, out of the four results (upper bound and lower bound of GBS-CD and sGS-CD respectively), two of them are quite simple: the lower bound of GBS-CD and the upper bound of sGS-CD.

5 Proof of Upper Bounds

5.1 Proof of proposition 3.1

Proof Outline for proposition 3.1 Recall we use "iteration" to denote one repeated procedure in an algorithm, and we use "pass" to denote one pass of all coordinates. sGS-CD consists of a forward pass and a backward pass in each iteration. We introduce a half-step $x^{k+\frac{1}{2}}$ to denote the output of sGS-CD

after performing the forward pass on x^k. We upper bound the convergence rate of the sequences defined by forward pass and backward pass separately, and the upper bound of sGS-CD can be obtained by combining these two bounds.

These two bounds are proved in a similar way, which can be divided into two steps. The first step is to relate the convergence rate with the spectral radius of a certain matrix. This step can be done from two different perspectives: the optimization perspective which views the forward (or backward) pass as inexact GD (with backward order for backward pass), and the matrix recursion perspective which directly writes the update in function value as a matrix recursion. In the second step, we estimate the spectral radius of the matrix mentioned in the first step.

Proof. We first assume Q is positive definite. Denote U to be the upper triangular matrix of Q, and Γ be the lower triangular matrix of Q. Let D_Q to be a diagonal matrix with entries Q_{ii}'s.

The update of sGS-CD consists of a forward pass and a backward pass. The forward pass updates coordinates in order $1, ..., n$, and the backward pass updates coordinates in order $n-1, ..., 1$. However, since the update on each coordinate is an exact minimization, then we can assume that the backward pass updates coordinates in order $\mathbf{n}, n-1, ..., 1$. Therefore, the iteration iteration matrix of sGS-CD is simply the product of update matrices of the forward and backward pass:

$$x^{k+1} = (I - U^{-1}Q)(I - \Gamma^{-1}Q)x^k$$

By defining $x^{k+1/2} = (I - \Gamma^{-1}Q)x^k$ as the "half step" update, we separate the effects of forward and backward pass. Using the techniques developed in [20], we can obtain an upper bound on the decreases of objective error from x^k to $x^{k+1/2}$ for forward pass (and from $x^{k+1/2}$ to x^{k+1} for backward pass). In particular, we can derive the following two inequalities from either optimization perspective or matrix recursion perspective, as indicated in [20]:

$$\frac{f(x^k) - f(x^*)}{f(x^k) - f(x^{k+\frac{1}{2}})} \leq \|D_Q^{-1/2}\Gamma^T Q^{-1}\Gamma D_Q^{-1/2}\| \triangleq c_1 \tag{45}$$

$$\frac{f(x^{k+1}) - f(x^*)}{f(x^{k+1}) - f(x^{k+\frac{1}{2}})} \leq \|D_Q^{-1/2}U^T Q^{-1}U D_Q^{-1/2}\| \triangleq c_2. \tag{46}$$

The proof of eq. (45) can be found in the proof of Claim B.1 in [20], and eq. (46) can be proved in exactly the same way. eq. (45) and eq. (46) imply

$$f\left(x^{k+\frac{1}{2}}\right) - f\left(x^*\right) \leq \left(1 - \frac{1}{c_1}\right)\left(f\left(x^k\right) - f\left(x^*\right)\right) \tag{47}$$

$$f\left(x^{k+1}\right) - f\left(x^*\right) \leq \left(1 - \frac{1}{c_2}\right)\left(f\left(x^{k+\frac{1}{2}}\right) - f\left(x^*\right)\right) \tag{48}$$

Combining (47) and (48), we obtain

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{1}{c_2}\right)\left(1 - \frac{1}{c_1}\right)\left(f(x^k) - f(x^*)\right) \tag{49}$$

Therefore, we can obtain an upper bound on the objective error convergence rate of GBS-CD by finding the upper bounds for c_1 and c_2. Notice that Q is symmetric, and thus we observe that c_1 equals to c_2 from their definitions in eq. (45) and eq. (46).

According to the fact that $\left\|D_Q^{-1/2} B D_Q^{-1/2}\right\| \leq \frac{1}{\min_i Q_{ii}}\|B\| = \frac{1}{L_{\min}}\|B\|$ for any positive definite matrix B, where $L_{\min} \triangleq \min_i Q_{ii}$, we have

$$c_1 = \left\|D_Q^{-1/2}\Gamma^T Q^{-1}\Gamma D_Q^{-1/2}\right\| \leq \frac{1}{L_{\min}}\left\|\Gamma^T Q^{-1}\Gamma\right\| \tag{50}$$

We then apply proposition B.1, which states that

$$\|\Gamma^T Q^{-1}\Gamma\| \leq \frac{1}{L_{\min}}\kappa \cdot \min\left\{\sum_i L_i, (2+\frac{1}{\pi}\log n)^2 L\right\} \tag{51}$$

Finally, combining (49), (50), (51), the fact that $c_1 = c_2$, and replacing $\sum_i L_i$ by nL_{avg}, we obtain the desired results in eq. (38):

$$f(x^{k+1}) - f^* \leq \left(\min\left\{1 - \frac{1}{n\kappa}\frac{L_{\min}}{L_{\text{avg}}}, 1 - \frac{L_{\min}}{L(2+\log n/\pi)^2}\frac{1}{\kappa}\right\}\right)^2 (f(x^k) - f^*).$$

If Q is positive semi-definite, then we replace Q^{-1} by $Q^\dagger$ which is the pseudo-inverse of Q. The rest of the analysis is similar to the proof in [20, Proposition 1] and omitted. □

5.2 Proof of proposition 3.2

Proof Outline for proposition 3.2:

A natural idea to derive the upper bound of the convergence rate of GBS-CD is to directly upper bound the spectral radius of the iteration matrix $M = I - B\Gamma^{-1}Q$. However, the computation is complicated because the iteration matrix is non-symmetric. We do not symmetrize M by applying the "product rule" on M, since the expression of $M^T M$ is still complicated.

To simplify the computation, we apply symmetrization techniques on M based on the following two observations. Firstly, we observe that we can replace the matrix B in the expression of M by Γ^{-T} and the eigenvalues of the new matrix $I - \Gamma^{-T}\Gamma^{-1}Q$ is unchanged. Secondly, we observe that we can symmetrize $I - \Gamma^{-T}\Gamma^{-1}Q$ by applying a similarity transformation to obtain $I - \Gamma^{-1}Q\Gamma^{-T}$, a symmetric matrix of which the spectral radius can be bounded. Using this bound of spectral radius, we can derive the upper bound of iterate convergence rate of GBS-CD.

Proof. **Step 1:** We simplify the analysis by assuming b equals to 0, and thus the optimal solution x^* equals to 0. The matrix recursion update of x^{k+1} is

$$x^{k+1} = (I - B\Gamma^{-1}Q)x^k.$$

Instead of directly bounding the spectral radius of the iteration matrix of x^k, we introduce a new sequence whose iteration matrix is close to a symmetric matrix, because it is easier to relate the convergence rate with spectral radius for symmetric matrix recursions. In particular, We introduce a new sequence y^k such that for every k, $y^k = B^{-1}x^k$, and the matrix recursion update of y^{k+1} is

$$\begin{aligned} y^{k+1} &= B^{-1}(I - B\Gamma^{-1}Q)x^k \\ &= (B^{-1} - \Gamma^{-1}Q)x^k \\ &= (I - \Gamma^{-1}QB)B^{-1}x^k \\ &= (I - \Gamma^{-1}QB)y^k \end{aligned}$$

Claim: we can replace B by Γ^{-T} in the last line.

According to lemma C.1, when $Q_{11} = 1$, the eigenvalues of $B^{-1}Q^{-1}\Gamma$ are the same as $\Gamma^T Q^{-1}\Gamma$. Hence, the eigenvalues of $\Gamma^{-1}QB$ are the same as $\Gamma^{-1}Q\Gamma^{-T}$, and therefore replacing B by Γ^{-T} in the iteration matrix of y^k will not affect the convergence of y^k. We note that the iteration matrix of y^k is similar to the iteration matrix of x^k.

Step 2: Notice that $I-\Gamma^{-1}Q\Gamma^{-T}$ is symmetric, and we can derive an upper bound of its spectral radius (and spectral norm) as below:

$$\begin{aligned}
&\rho(I-\Gamma^{-1}Q\Gamma^{-T})\\
=&1-\lambda_{\min}(\Gamma^{-1}Q\Gamma^{-T})\\
=&1-\frac{1}{\rho(\Gamma^T Q^{-1}\Gamma)}\\
=&1-\frac{1}{\|\Gamma^T Q^{-1}\Gamma\|}\\
\leq&1-\frac{1}{\kappa\cdot\min\left\{\sum_i L_i,(2+\frac{1}{\pi}\log n)^2 L\right\}}.
\end{aligned}$$

The second to last line follows from the fact that the spectral radius and the spectral norm of a symmetric matrix are the same. The last line is obtained by applying the upper bound of $\|\Gamma^T Q^{-1}\Gamma\|$ from proposition B.1.

Since the iteration matrix of y^k is symmetric, the convergence rate of $\|y^k\|$ is determined by the spectral radius of the iteration matrix. Therefore, from the above inequalities about the spectral radius of $I-\Gamma^{-1}Q\Gamma^{-T}$, we obtain an upper bound on the convergence rate of y^k:

$$\|y^k\|\leq\left(1-\frac{1}{\kappa\cdot\min\left\{\sum_i L_i,(2+\frac{1}{\pi}\log n)^2 L\right\}}\right)^k\|y^0\| \tag{52}$$

To obtain the upper bound on the convergence rate of x^k, we square both sides of (52) and plug in $y^k = B^{-1}x^k$:

$$\begin{aligned}
&(x^k)^T B^{-T}B^{-1}x^k=\|B^{-1}x^k\|^2\\
\leq&\left(1-\frac{1}{\kappa\cdot\min\left\{\sum_i L_i,(2+\frac{1}{\pi}\log n)^2 L\right\}}\right)^{2k}\|B^{-1}x^0\|^2\\
\leq&\left(1-\frac{1}{\kappa\cdot\min\left\{\sum_i L_i,(2+\frac{1}{\pi}\log n)^2 L\right\}}\right)^{2k}\|B^{-1}\|^2\|x^0\|^2
\end{aligned}$$

Finally, according to the fact that for any positive semi-definite matrix A and vector z, $z^TAz\geq\lambda_{\min}(A)\|z\|^2$, we have

$$\|x^k\|\leq\left(1-\frac{1}{\kappa\cdot\min\left\{\sum_i L_i,(2+\frac{1}{\pi}\log n)^2 L\right\}}\right)^k\frac{\|B^{-1}\|\|x^0\|}{\sqrt{\lambda_{\min}(B^{-T}B^{-1})}}$$

If we denote $\alpha=\frac{\|B^{-1}\|\|x^0\|}{\sqrt{\lambda_{\min}(B^{-T}B^{-1})}}$, then we achieve the desired result. □

5.3 Proof of proposition 3.3

Proof Outline for proposition 3.3: In the proof of proposition 3.3, we first derive the iteration matrix of the objective error which is $M_f = I - AB\Gamma^{-1}A^T$. Although it is difficult to upper bound the spectral radius directly, we can obtain $M_s = I - A\Gamma^{-T}\Gamma^{-1}A^T$ without affecting the eigenvalues. We further observe that M_s is similar to $M_t = I - \Gamma^{-1}Q\Gamma^{-T}$, and then we can apply the the upper bound of $\rho(M_t)$ that is proved in section 5.2.

Proof. For simplicity, we assume $x^* = 0$.

We want to compute the convergence rate of

$$f(x^k) = (x^k)^T Q(x^k) = \|r^k\|^2,$$

where $r^k = Ax^k$ in which A satisfies $Q = A^T A$.

Given $x^{k+1} = (I - B\Gamma^{-1}Q)x^k$, we have:

$$\begin{aligned} r^{k+1} &= Ax^{k+1} = A(I - B\Gamma^{-1}Q)x^k \\ &= Ax^k - AB\Gamma^{-1}A^T(Ax^k) \quad (\text{by } Q = A^T A) \\ &= (I - AB\Gamma^{-1}A^T)r^k \end{aligned}$$

The iteration matrix of r^k is symmetric, if we can replace B by Γ^{-T}. Therefore, the convergence rate of r^k is determined by the spectral radius of the iteration matrix. To give an upper bound on the objective error convergence rate, we need to upper bound the spectral radius of $I - AB\Gamma^{-1}A^T$.

As discussed in section 4, for any invertible matrices U and V, UV is similar to VU, thus we have the following relation:

$$AB\Gamma^{-1}A^T \sim B\Gamma^{-1}A^T A = B\Gamma^{-1}Q \sim \Gamma^{-1}QB. \tag{53}$$

Moreover, by lemma C.1:

$$\text{eig}(\Gamma^{-1}QB) = \text{eig}(\Gamma^{-1}Q\Gamma^{-T}) \tag{54}$$

Combining eq. (53) and eq. (54), we have that

$$\rho(I - AB\Gamma^{-1}A^T) = \rho(I - \Gamma^{-1}Q\Gamma^{-T})$$

Therefore, we only need to have an upper bound on $\rho(I - \Gamma^{-1}Q\Gamma^{-T})$, which is given in section 5.2:

$$\rho(I - \Gamma^{-1}Q\Gamma^{-T}) \leq 1 - \frac{1}{\kappa \cdot \min\left\{\sum_i L_i, (2 + \frac{1}{\pi}\log n)^2 L\right\}}.$$

This implies

$$\|r^{k+1}\| \leq \left(1 - \frac{1}{\kappa \cdot \min\left\{\sum_i L_i, (2 + \frac{1}{\pi}\log(n))^2 L\right\}}\right)\|r^k\|$$

and equivalently,

$$f(x^{k+1}) - f^* \leq \left(1 - \frac{1}{\kappa \cdot \min\left\{\sum_i L_i, (2 + \frac{1}{\pi}\log(n))^2 L\right\}}\right)^2 (f(x^k) - f^*)$$

□

as desired.

6 Proof of Lower bounds

6.1 Worst-case example

In the proof of theorem 3.1 and theorem 3.2, we will consider the following quadratic problem eq. (55)

$$\min_{x\in\mathbb{R}^n} \quad x^T A^T A x \tag{55}$$

with a special case of matrix $A \in \mathbb{R}^{n\times n}$ such that

$$Q \triangleq A^T A = \begin{bmatrix} 1 & c & \dots & c \\ c & 1 & \dots & c \\ \vdots & \vdots & \ddots & \vdots \\ c & c & \dots & 1 \end{bmatrix} \tag{56}$$

for a given constant $c \in (0, 1)$.

Q is a positive definite matrix, with one eigenvalue $1 - c$ with multiplicity $n - 1$ and one eigenvalue $1 - c + cn$ with multiplicity 1. The condition number of Q is

$$\kappa = \frac{1 - c + cn}{1 - c}, \tag{57}$$

where κ approaches infinity when c approaches 1.

Remark 1: This instance has been analyzed in [20] for C-CD and [50] for RP-CD. Using this example (for $c \to 1$), [20] shows C-CD can be $\mathcal{O}(n^2)$ times slower than R-CD. Note that when analyzing sGS-CD, we also require $c \to 1$; but when analyzing GBS-CD, we allow any $c \in (0, 1)$).

Remark 2 (**Permutation Invariant**): The matrix Q is a permutation-invariant matrix, i.e. $P^T Q P = Q$ for any permutation matrix P. This implies that even if one randomly permutes the coordinates at the beginning and then apply sGS-CD or GBS-CD, the iterates do not change. Thus our lower bounds would still hold.

6.2 Proof of theorem 3.1

We first briefly discuss the proof ideas. It is difficult to lower bound the spectral radius of the iteration matrix directly, and we use a completely different path from the straightforward computation. An important observation is that C-CD for solving quadratic problems is equivalent to alternating projection method for solving linear systems, with iteration matrix $\hat{P} = P_n P_{n-1} \dots P_1$, where each P_i is a projection matrix. For sGS-CD, the iteration matrix can be written as

$$M_p = P_1 \cdots P_{n-1} P_n P_{n-1} \dots P_1,$$

which is a product symmetrization of $\hat{P}$. Thus, we can derive the lower bound of the convergence rate of sGS-CD based on that of C-CD.

Next we present a formal proof. We need to lower bound the objective error convergence rate of sGS-CD, using example eq. (55) introduced in section 6.1. In particular, we want to prove a lower bound of the convergence rate of $f(x^k) = (x^k)^T Q x^k = \|r^k\|^2$ where $r^k = Ax^k$ and A satisfies $A^T A = Q$. We introduce alternating projections and use them to find a simpler form of the iteration matrix of r^k.

Step 1: Alternating Projections. When solving the problem eq. (55) using coordinate descent, we have that the update rule for coordinate i is given by

$$x_i^+ = \frac{1}{A_i^T A_i} \left[A_i^T \left(-A_{-i} x_{-i} \right) \right], \tag{58}$$

where A_{-i} contains all columns of A except A_i, x_{-i} contains all elements of x except x_i and represents the current values, and x_i^+ represents the new value.

Since $r = Ax$, we can rewrite eq. (58) as

$$\begin{aligned} x_i^+ &= x_i - \frac{1}{A_i^T A_i} A_i^T r \\ r^+ &= r + A_i \left(x_i^+ - x_i \right). \end{aligned}$$

We define $P_i = I - (A_i^T A_i)^{-1} A_i A_i^T$ as the projection matrix that projects vectors onto the column space of A_i. The update rule for r is therefore

$$r^+ = \left(I - \frac{A_i A_i^T}{A_i^T A_i} \right) r = P_i r$$

We denote $r^{k+\frac{1}{2}}$ to be the value of r after the forward pass of sGS-CD at iteration $k+1$, and it can be expressed as

$$r^{k+\frac{1}{2}} = P_n P_{n-1} \dots P_1 r^k$$

Similarly, r^{k+1}, which is the value of r after the backward pass of sGS-CD at iteration $k+1$, can be expressed as

$$r^{k+1} = P_1 P_2 \dots P_{n-1} r^{k+\frac{1}{2}}.$$

Combining the forward pass and backward pass, we have matrix recursion for r^{k+1} of sGS-CD:

$$r^{k+1} = P_1 P_2 \dots P_{n-1} P_n P_{n-1} \dots P_1 r^k \tag{60}$$

Using the property of projection matrix $P_i = P_i P_i$, eq. (60) can be rewritten as

$$r^{k+1} = \left(P_n P_{n-1} \cdots P_1 \right)^T \left(P_n P_n P_{n-1} \cdots P_1 \right) r^k. \tag{61}$$

Note that a forward pass of sGS-CD is equivalent to one full iteration of C-CD. If we denote $\hat{P} = P_n P_{n-1} \cdots P_1$, then $\hat{r}^{k+1}$ of C-CD at iteration $k+1$ can be expressed as

$$\hat{r}^{k+1} = P_n P_{n-1} \cdots P_1 \hat{r}^k = \hat{P} \hat{r}^k.$$

The update of r^{k+1} for sGS-CD at iteration $k+1$ is

$$r^{k+1} = \hat{P}^T \hat{P} r^k. \tag{62}$$

In other words, eq. (62) shows that the iteration matrix of r^k for sGS-CD is a **"product symmetrization"** of the iteration matrix of $\hat{r}^k$ for C-CD. This implies we can apply results of the lower bound of C-CD.

Step 2: Apply the Lower Bound of C-CD.

Since we use exactly the same worst case example as in [20], [20, Theorem 3.1] can be interpreted as:

Theorem 6.1. *(Theorem 3.1 in [20], re-interpreted)*

When solving problem (55) *where A satisfies* (56) *using C-CD, for any initial point* $x^0 \in \mathbb{R}^n$*, any* $\delta \in (0, 1]$*, we have*

$$\frac{\|\hat{r}^k\|^2}{\|\hat{r}^0\|^2} \geq (1-\delta)\left(1-\frac{2\pi^2}{n\kappa}\right)^{2k+2} \tag{63}$$

where $\hat{r}^k = A\hat{x}^k$ *and* $\hat{x}^k$ *is the iterate in C-CD at iteration* k*.*

Note that eq. (63) implies

$$\|\hat{P}\| \geq (1-\delta)\left(1-\frac{2\pi^2}{n\kappa}\right), \tag{64}$$

because otherwise

$$\|\hat{r}^k\| \leq \|\hat{P}\|^{k+1}\|r^0\| < (1-\delta)\left(1-\frac{2\pi^2}{n\kappa}\right)^{k+1}\|r^0\|$$

which contradicts eq. (63).

Recall that for any matrix recursion $y^{k+1} = My^k$, if M is symmetric, then the convergence rate of $\|y^k\|$ is exactly the spectral radius of M. Since the iteration matrix of sGS-CD for r^k eq. (62) is symmetric, then the convergence rate of r^k for sGS-CD is exactly $\rho(\hat{P}^T\hat{P})$.

In addition, since

$$\rho(\hat{P}^T\hat{P}) = \|\hat{P}^T\hat{P}\| = \|\hat{P}\|^2,$$

then by (64), we have

$$\frac{\|r^{k+1}\|^2}{\|r^k\|^2} = \rho(\hat{P}^T\hat{P})^2 = \|\hat{P}\|^4 \geq \left((1-\delta)\left(1-\frac{2\pi^2}{n\kappa}\right)\right)^4.$$

Finally, we obtain the desired lower bound on objective error of sGS-CD:

$$\begin{aligned}
&\frac{\|r^k\|^2}{\|r_0\|^2} \\
\geq&(1-\delta)^2\left(1-\frac{2\pi^2}{n\kappa}\right)^{4k+4} \\
\geq&(1-2\delta)\left(1-\frac{4\pi^2}{n\kappa}\right)^{2k+2}
\end{aligned}$$

Remark: We do not directy analyze the iteration matrix $\left(I-U^{-1}Q\right)(I-\Gamma^{-1}Q)$ derived in section 5.1, since it is not even a symmetric matrix. The trick of this proof is to express the iteration matrix as the product of projection matrices (which requires considering the update of the residual). Using this form, we find an interesting fact that the iteration matrix of sGS-CD is a product-symmetrization of that of C-CD.

Recall that in Section 4, we mentioned that the iteration matrix of sGS-CD can be approximately written as $\left(I - A\Gamma^{-T}A^T\right)\left(I - A\Gamma^{-1}A^T\right)$. This matrix is the same as $P_1P_2\dots P_{n-1}P_nP_{n-1}\dots P_1$, because $I - A\Gamma^{-T}A^T = P_nP_{n-1}\dots P_1$ (the proof is skipped here). We present the proof based on the projection matrices because it is more intuitive than the derivation in Section 4. In fact, it is after deriving the form $P_1P_2\dots P_{n-1}P_nP_{n-1}\dots P_1$ that we realize that there has to be a similar form in the original domain, leading us to the derivation in Section 4.

6.3 Proof of theorem 3.2

We first briefly discuss the main proof ideas. We need to derive a lower bound on the the spectral radius of $I-\Gamma^{-1}Q\Gamma^{-T}$, which is equivalent to finding a lower bound on the spectral radius of $\Gamma^TQ^{-1}\Gamma$. Even though $\Gamma^TQ^{-1}\Gamma$ is symmetric, the eigenvalues of this matrix do not have a clear form. However, with our particular choice of Q, we can decompose the matrix into several terms and compute each of the spectral norms of these terms. Combining these bounds on spectral norms by triangle inequality leads to, somewhat luckily, a desired lower bound on the spectral radius of $\Gamma^TQ^{-1}\Gamma$.

Proof. As shown in section 5.3, the convergence rate for $\|r^k\|$ of GBS-CD is related to the spectral radius of $I - AB\Gamma^{-1}A^T$, the iteration matrix for r^k. It is furthered show that

$$\rho(I - AB\Gamma^{-1}A^T) = \rho(I - \Gamma^{-1}Q\Gamma^{-T}),$$

and therefore the goal is to find a lower bound for $\rho(I - \Gamma^{-1}Q\Gamma^{-T})$.

It is easy to see that finding a lower bound on $\rho(I-A\Gamma^{-T}\Gamma^{-1}A^T)$ is equivalent to finding an upper bound on $\lambda_{\min}(\Gamma^{-1}Q\Gamma^{-T})$. Further notice that finding a upper bound of the minimal eigenvalue of a symmetric matrix is equivalent to finding a lower bound of the largest eigenvalue of the inverse matrix. Therefore, it suffices to find a lower bound of $\rho(\Gamma^TQ^{-1}\Gamma) = \|\Gamma^TQ^{-1}\Gamma\|$.

The eigenvalues of $\Gamma^TQ^{-1}\Gamma$ do not have a clear form, so we decompose the matrix and analyze the spectral norm of the decomposition in proposition 6.1.

Proposition 6.1.

$$Q^{-1} = \frac{1}{\tilde{c}}I - \frac{c}{\tilde{c}(\tilde{c}+cn)}J, \tag{65}$$

where $\tilde{c} = 1 - c$ *and* $J = ee^T$.

proposition 6.1 can be proved simply by inspection (proof is provided in appendix D).

Denote $a = \frac{1}{\tilde{c}}$ and $b = \frac{c}{\tilde{c}(\tilde{c}+cn)}$, we have that

$$\begin{aligned}
&\|\Gamma^TQ^{-1}\Gamma\| \\
=&\|\Gamma^T(aI - bJ)\Gamma\| \\
=&\|a\Gamma^T\Gamma - b\Gamma^TJ\Gamma\| \\
\geq&\left|\|a\Gamma^T\Gamma\| - \|b\Gamma^TJ\Gamma\|\right| &\text{(66a)}\\
=&\left|\|a\Gamma\Gamma^T\| - \|b\Gamma^TJ\Gamma\|\right| \\
=&\left|a\|\Gamma\Gamma^T - Q + Q\| - \|b\Gamma^TJ\Gamma\|\right| \\
\geq&\left|a\left|\|\Gamma\Gamma^T - Q\| - \|Q\|\right| - b\|\Gamma^TJ\Gamma\|\right|, &\text{(66b)}
\end{aligned}$$

where the inequalities eq. (66a) and eq. (66b) follows from the reverse triangle inequality. We introduce the following propositions to compute the spectral norm of each term in eq. (66b). Proofs of these propositions are provided in the appendix D.

Proposition 6.2.

$$\|\Gamma\Gamma^T - Q\| \gtrapprox c^2 \frac{4n^2}{\pi^2}.$$

Proposition 6.3.

$$\|Q\| = 1 - c + cn.$$

Proposition 6.4.

$$\|\Gamma^T J \Gamma\| = n + cn(n-1) + \frac{c^2}{6} n(n-1)(2n-1).$$

We apply the propositions and plug in the definitions of a and b into eq. (66):

$$\begin{aligned}
&\|\Gamma^T Q^{-1} \Gamma\| \\
\geq& \left| a \left| \|\Gamma\Gamma^T - Q\| - \|Q\| \right| - b\|\Gamma^T J \Gamma\| \right| && \text{(67a)} \\
\geq& \left(a \left(\frac{4c^2n^2}{\pi^2} - (1 - c + cn) \right) - b \left(n + cn(n-1) + \frac{c^2}{6} n(n-1)(2n-1) \right) \right) && \text{(67b)} \\
=& \frac{1}{1-c} \frac{4c^2n^2}{\pi^2} - \frac{1-c+cn}{1-c} - \frac{c\left(n + cn(n-1) + \frac{c^2}{6} n(n-1)(2n-1) \right)}{(1-c)(1-c+cn)} \\
=& \frac{1}{1-c} \left(\left(\frac{4c^2}{\pi^2} - \frac{c^2}{3} \right) n^2 + \left(\frac{c^2}{2} - 2c \right) n - \left(2 - 2c + \frac{c^2}{6} \right) \right) && \text{(67c)}
\end{aligned}$$

In (67b), we remove the two absolute value signs in (67a) by assuming expressions inside the absolute value are non-negative. Note that $\frac{4c^2n^2}{\pi^2} - (1 - c + cn)$ is a quadratic function in n with positive qudratic coefficient, so for given c, we can solve for n such that the expression inside the absolute value is nonnegative. In particular, for any n that satisfies

$$n \geq \frac{\pi^2 \left(1 + \sqrt{1 + \frac{16(1-c)}{\pi^2}} \right)}{8c}, \tag{68}$$

$\frac{4c^2n^2}{\pi^2} - (1-c+cn)$ is nonnegative. The RHS of eq. (68) is approaching positive infinity when c is approaching 0, and it is approaching zero when c is approaching 1.

Assuming (68) holds, the expression in (67b), which can be re-written as eq. (67c), is also a quadratic function in n with positive qudratic coefficient. Therefore, for given c, if n satisfies

$$n \geq \frac{\left(2c - c^2\right) + \sqrt{\left(\frac{c^2}{2} - 2c\right)^2 - 4\left(\frac{4c^2}{\pi^2} - \frac{c^2}{3}\right)\left(2 - 2c + \frac{c^2}{6}\right)}}{\left(\frac{8c^2}{\pi^2} - \frac{2c^2}{3}\right)}, \tag{69}$$

we have that the expression on the RHS of eq. (67b) is non-negative. Although eq. (69) looks complicated, we observe that when c is relatively large, the RHS of eq. (69) is small. For example, when $c = 0.9$, the RHS of eq. (69) is around 20. Therefore, assuming eq. (68) and eq. (69) hold, we can remove both absolute value signs in (67a) and the inequality eq. (67b) holds.

Define the following constants

$$c_0 = \frac{4}{\pi^2} - \frac{1}{3} = \frac{12-\pi^2}{3\pi^2},$$
$$c_1 = \frac{c^2}{2} - 2c,$$
$$c_2 = 2 - 2c + \frac{c^2}{6}.$$

Plugging in $\kappa = \frac{1-c+cn}{1-c} = 1 + \frac{c}{1-c}n$ and $n = \frac{(1-c)\kappa - 1 + c}{c}$ in eq. (67c), we can extract an n from eq. (67c) and rewrite it as

$$n\left(\kappa c c_0 - c c_0 + \frac{c_1}{1-c} - \frac{c_2}{(1-c)n}\right) = \mathcal{O}(n\kappa c c_0). \tag{70}$$

Therefore, once we satisfy the condition eq. (68), we obtain

$$\|r^k\| \geq (1-\delta)\left(1 - \frac{3\pi^2}{(12-\pi^2)n\kappa c}\right)^{k+1}\|r^0\|,$$

or equivalently

$$f(x^k) - f^* \geq (1-\delta)\left(1 - \frac{3\pi^2}{(12-\pi^2)n\kappa c}\right)^{2k+2}(f(x^0) - f^*).$$

□

7 Experimental results

In this section, we provide some empirical results of the worst-case performance of CD and ADMM with sGS and GBS update rules by solving unconstrained quadratic problems and and linearly constrained problems. In section 7.1, we solve the quadratic problem eq. (55) with no constraints and show that sGS-CD and GBS-CD can be $\mathcal{O}(n)$ times slower than GD and $\mathcal{O}(n^2)$ times slower than R-CD, which match our theoretical analysis. In section 7.2, we solve a linearly constrained problem with zero objective (equivalent to solve a linear system) by sGS-ADMM and GBS-ADMM. The experiments show that sGS-ADMM and GBS-ADMM can be $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ than Augmented Lagrangian Method (ALM) and Randomly Permute ADMM (RP-ADMM) respectively. In other words, the experiments show that adding a constraint to the optimization problem will not prevent the slow convergence behavior of sGS-ADMM and GBS-ADMM. In section 7.3, we further provide empirical evidence that strong convexity of the objective cannot prevent the slow convergence behaviors, by considering an optimization problem with a strongly convex quadratic objective and linear constraints.

7.1 Solving Unconstrained Problem with Quadratic Objective

In this subsection, we evaluate the performance of GBS-CD, sGS-CD, C-CD (cyclic CD), R-CD (randomized CD), RP-CD (randomly permuted CD) and GD (gradient descent) for minimizing the quadratic problem defined in eq. (55). We use two metrics to compare the performance: the number of iterations to achieve certain accuracy, and the spectral radius.

In the first experiment, we initialize the solutions from the uniform distribution between 0 and 1. In table 3, each column presents the number of iterations needed for an algorithm to solve eq. (55) with various problem sizes n when $c = 0.8$ up to relative error accuracy 1e-8. To make the comparision of various methods more clear, we create table 4 based on table 3. The columns under "Ratio of GBS-CD" of table 4 are created by dividing the number of iterations for GD, C-CD, R-CD, and RP-CD in table 3 by that of GBS-CD. The resulting ratios can be interpreted as the acceleration ratios of these algorithms over GBS-CD. For instance, the entry 17360.0 in the fourth column of table 4 means that GBS-CD takes 17360 times more iterations than R-CD to solve eq. (55) when $c = 0.8$ and $n = 100$ (or simply put, GBS-CD is 17360 times slower than R-CD). Similarly, the columns under "Ratio of GBS-CD" of table 4 display the accelerationn ratios of various algorithms over GBS-CD.

Recall we have shown that GBS-CD can be $\mathcal{O}(n)$ times slower than GD in the worst case. In the second column of table 4, the ratios for $n = 100$ and $n = 600$ are 8.7 and 51.6 correspondingly. Note that $\frac{51.6}{8.7} = 5.88 \approx 6$, which matches our theoretical analysis. In the fourth column, we observe that the relative ratio is $\frac{17360.0}{499.3} = 34.8 \approx 36$, which also matches our theoretical analysis that GBS-CD is $\mathcal{O}(n^2)$ times slower than R-CD in the worst case. In general, table 4 shows that the experimental results match our theoretical analysis.

In the second experiment, we compare the spectral radius of the iteration matrices for various methods. In table 5, we present $1 - \rho(M)$ where M is the (expected) iteration matrix of GBS-CD, sGS-CD, C-CD, R-CD, RP-CD and GD for $c = 0.5, 0.8, 0.99$ and $n = 20, 100, 1000$. The first column shows the value of c in the matrix Q in eq. (55), and the next six columns present the values of $1 - \rho(M)$ for each algorithm. In the last four columns, we divide the values of $1 - \rho(M)$ of C-CD, R-CD, RP-CD and GD by that of GBS-CD to obtain ratios of the spectral radius of the (expected) update matrices. We omit the ratios of various algorithms over sGS-CD, since they are similar to the ones over GBS-CD. We observe that, as c approaches 1, the gap of the ratios between GD and GBS-CD grows in $\mathcal{O}(n)$ as n increases, and the gap of the ratios between RP-CD (or R-CD) and GBS-CD grows in $\mathcal{O}(n^2)$. The gap of the ratios between GBS-CD and C-CD is a constant 2 for different n. These observations match the comparison of the number of iterations in table 4.

Table 3: Comparison of the iterations of GBS-CD, sGS-CD, C-CD, R-CD, RP-CD and GD for solving eq. (55) when $c = 0.8$. The numbers represent the number of iterations needed to achieve relative error $1e - 8$.

n	**GBS-CD**	**sGS-CD**	**C-CD**	**GD**	**R-CD**	**RP-CD**
100	51431	51431	25749	5942	103	88
200	200184	200184	100377	11617	103	89
600	1735996	1735996	869123	33627	100	88

Table 4: Ratios of the iterations of GBS-CD and sGS-CD over the iterations of C-CD, R-CD, RP-CD and GD for solving eq. (55) when $c = 0.8$.

	Ratio of GBS-CD				Ratio of sGS-CD			
n	GD	C-CD	R-CD	RP-CD	GD	C-CD	R-CD	RP-CD
100	8.7	2.0	499.3	584.4	8.7	2.0	499.3	584.4
200	17.2	2.0	1943.5	2249.3	17.2	2.0	1943.5	2249.3
600	51.6	2.0	17360.0	19727.2	51.6	2.0	17360.0	19727.2

Table 5: Comparison of GBS-CD, sGS-CD, C-CD, R-CD, RP-CD and GD for solving eq. (55) when $c = 0.5, 0.8, 0.99$

c	1 - ρ(M)						Acceleration Ratio			
	GBS-CD	sGS-CD	GD	C-CD	R-CD	RP-CD	GD	C-CD	R-CD	RP-CD
n = 20										
0.5	4.2e-2	4.2e-2	4.8e-2	7.6e-2	4.0e-1	5.2e-1	1.1	1.8	9.3	12.1
0.8	7.4e-3	7.4e-3	1.2e-2	1.4e-2	1.8e-2	2.0e-1	1.6	1.9	2.4	26.8
0.99	2.5e-4	2.5e-4	5.0e-4	4.9e-4	1.0e-2	1.0e-2	2.0	2.0	40.0	41.2
n = 100										
0.5	1.9e-3	1.9e-3	9.9e-3	3.8e-3	3.9e-1	5.0e-1	5.1	2.0	202.1	259.1
0.8	3.0e-4	3.0e-4	2.5e-3	6.4e-4	1.8e-1	2.0e-1	8.1	2.1	586.3	651.5
0.99	1.0e-5	1.0e-5	1.0e-4	2.0e-5	1.0e-2	1.0e-2	10.0	2.0	1e3	1e3
n = 1000										
0.5	1.9e-5	1.9e-5	9.9e-4	3.9e-5	3.9e-1	5.0e-1	50.7	2.0	1.9e4	2.5e4
0.8	3.0e-6	3.0e-6	2.5e-4	6.2e-6	1.8e-1	2.0e-1	81.2	2.0	5.8e4	6.4e4
0.99	1.0e-7	1.0e-7	1.0e-5	2.0e-7	1.0e-2	1.0e-2	100.0	2.0	9.9e4	9.9e4

7.2 Zero Objective and Linear Constraints

In this subsection, we consider solving the constrained optimization problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & 0 \\ \text{s.t.} \quad & Qx = 0, \end{aligned} \tag{71}$$

where Q is defined by eq. (56). We provide the numerical results of GBS-ADMM, sGS-ADMM, RP-ADMM and ALM (Augmented Lagrangian Method) when solving the problem eq. (71) with $c = 0.5$ for the matrix Q. The number of iterations needed for each algorithm to obtain relative error 1e-5 are shown in table 6. As shown in the table, RP-ADMM is the most efficient algorithm among all algorithms, and ALM is the second most efficient algorithm. Most importantly, we observe that GBS-ADMM and sGS-ADMM are much slower than RP-ADMM and ALM. We also compare the ratios of the number of iterations of RP-ADMM and ALM over GBS-ADMM (and sGS-ADMM). The results are presented in table 7. As indicated in the table, GBS-ADMM and sGS-ADMM are approximately $\mathcal{O}(n^2)$ times slower than RP-ADMM and $\mathcal{O}(n)$ times slower than ALM in this example. More details are provided below.

The order $\mathcal{O}(n^2)$ is estimated by the following simple method. Comparing $n = 50$ and $n = 100$, the acceleration ratios of RP-ADMM over sGS-ADMM are 24.2 and 96.8 respectively. Since $96.8/24.2 = 4 = 2^2$, we conjecture that the acceleration ratio grows quadratically. To further verify this conjecture, we found that from $n = 100$ to $n = 200$, the acceleration ratio grows by a factor of $391/96.8 \approx 4.04$; from $n = 200$ to $n = 400$, the acceleration ratio grows by a factor of $1528.7/391 \approx 3.91$. The constant factor can also be estimated: since when $n = 50$, we get an acceleration ratio 24.2, thus the general acceleration ratio of RP-ADMM over sGS-ADMM is about $n^2/100$.

The estimate for the acceleration ratio of RP-ADMM over GBS-ADMM is a bit more complicated. From $n = 50$ to 100, 200 and 400, the acceleration ratios grow by a factor of 3.45, 3.41 and 3.37 respectively. It seems that the general acceleration ratio of RP-ADMM over sGS-ADMM is about $\mathcal{O}(n^{1.8})$ (or maybe something like $\mathcal{O}(n^2/\log n)$). Nevertheless, it is possible that the worst-case ratio is indeed $\mathcal{O}(n^2)$, but due to various reasons (not enough accuracy, or the choice of initial point) this worst-case ratio is not fully reflected in the simulation. A more precise comparison is left to future work.

Table 6: Comparison of GBS-ADMM, sGS-ADMM, RP-ADMM, and ALM for solving eq. (71) when $c = 0.5$. The numbers represent the number of iterations to achieve relative error $1e-5$.

n	**GBS-ADMM**	**sGS-ADMM**	**RP-ADMM**	**ALM**
50	6064	10525	435	5886
100	37739	75922	785	18534
200	241343	551350	1410	57893
400	1504217	4153508	2717	166408

Table 7: Ratios of the iterations of GBS-ADMM and sGS-ADMM over the iterations of RP-ADMM, and ALM for solving eq. (71) when $c = 0.5$

	Ratio of GBS-ADMM		Ratio of sGS-ADMM	
n	RP-ADMM	ALM	RP-ADMM	ALM
50	13.9	1.0	24.2	1.8
100	48.0	2.0	96.8	4.1
200	164.1	4.2	391.0	9.5
400	553.6	9.0	1528.7	25.9

7.3 Convex Quadratic Objective and Linear Constraints

In section 7.2, we show that sGS-ADMM and GBS-ADMM have slow convergence speed when solving a problem with a zero objective and a linear constraint. Then, it is interesting to ask if a strongly convex objective will prevent the slow convergence of sGS-ADMM and GBS-ADMM. We provide a negative answer to this question by considering the following problem with a strongly convex quadratic objective and a linear constraint:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & x^T Q x \\ \text{s.t.} \quad & Qx = 0, \end{aligned} \tag{72}$$

where Q is defined by eq. (56), in which $c = 0.3$.

In table 8, we list the number of iterations needed for each algorithm to obtain relative error 1e-5. We observe that with strongly convex objective, GBS-ADMM and sGS-ADMM still converge much slower than RP-ADMM or ALM. We also compare the ratios of the number of iterations of RP-ADMM and ALM over GBS-ADMM (and sGS-ADMM) in table 9. We observe that GBS-ADMM and sGS-ADMM are about $\mathcal{O}(n^2)$ times slower than RP-ADMM and $\mathcal{O}(n)$ times slower than ALM in this example (again, more rigorously speaking, for GBS-ADMM, the acceleration ratio is a bit smaller than $\mathcal{O}(n^2)$ and larger than $\mathcal{O}(n^{1.8})$).

Table 8: Comparison of GBS-ADMM, sGS-ADMM, RP-ADMM, and ALM for solving eq. (72) where $c = 0.3$. The numbers represent the number of iterations to achieve relative error $1e-5$.

n	**GBS-ADMM**	**sGS-ADMM**	**RP-ADMM**	**ALM**
100	10709	13326	95	3696
200	54932	71877	126	8607
400	266344	391654	168	18044

Table 9: Ratios of the iterations of GBS-ADMM and sGS-ADMM over the iterations of RP-ADMM, and ALM for solving eq. (72) when $c = 0.3$

	Ratio of GBS-ADMM		Ratio of sGS-ADMM	
n	RP-ADMM	ALM	RP-ADMM	ALM
100	112.7	2.9	140.2	3.6
200	436.0	6.4	570.5	8.4
400	1585.4	14.7	2331.3	21.7

8 Conclusions

In this paper, we study the worst-case convergence rate of two symmeterized orders sGS and GBS for CD and ADMM. For ADMM, these two update orders are among the most popular variants, and also have strong convergence guarantee. In this work, we prove that for unconstrained problems, sGS-CD and GBS-CD are $\mathcal{O}(n^2)$ times slower than R-CD in the worst case. In addition, we show empirically that when solving quadratic problems with linear constraints, sGS-ADMM and GBS-ADMM can be roughly $\mathcal{O}(n^2)$ times slower than randomly permuted ADMM for a certain example. These results indicate that the symmetrization trick does not resolve the slow worst-case convergence speed of deterministic block-decomposition methods. Technically, we provide a unified framework of symmetrization, which includes sGS-CD and GBS-CD as special cases. This framework can help understand algorithms from the perspective of iteration matrices.

A Derivation of C-CD Iteration Matrix, for $n = 3$

For completeness, we illustrate how to derive the expression of $\tilde{x}^k$ in GBS-CD used in the GBS-CD definition of Section 2.2 (this derivation is same as the derivation of the update formula of C-CD for one iteration). We consider a simple case with $n = 3, d_i = 1, \forall i$, and let $a_i = A_i \in \mathbb{R}^{3\times 1}$ and define $Q = A^T A$; the case for general n is quite similar and omitted.

Recall the problem is to minimize a quadratic function:

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{2}\|Ax - b\|^2 \tag{73}$$

The update equations for the forward step can be written as

$$\begin{aligned} a_1^T(a_1\tilde{x_1}^k + a_2x_2^k + a_3x_3^k - b) = 0,\\ a_2^T(a_1\tilde{x_1}^k + a_2\tilde{x_2}^k + a_3x_3^k - b) = 0,\\ a_3^T(a_1\tilde{x_1}^k + a_2\tilde{x_1}^k + a_3\tilde{x_3}^k - b) = 0 \end{aligned}$$

Denote $\tilde{x}^k$ to be the updated iterate after the forward step in k-th iteration, then the above update equation becomes

$$\begin{bmatrix} a_1^Ta_1 & 0 & 0\\ a_2^Ta_1 & a_2^Ta_2 & 0\\ a_3^Ta_1 & a_3^Ta_2 & a_3^Ta_3 \end{bmatrix}\tilde{x}^k = \begin{bmatrix} 0 & -a_1^Ta_2 & -a_1^Ta_3\\ 0 & 0 & -a_2^Ta_3\\ 0 & 0 & 0 \end{bmatrix}x^k + A^Tb \tag{74}$$

Define

$$\Gamma \triangleq \begin{bmatrix} a_1^Ta_1 & 0 & 0\\ a_2^Ta_1 & a_2^Ta_2 & 0\\ a_3^Ta_1 & a_3^Ta_2 & a_3^Ta_3 \end{bmatrix}, \tag{75}$$

then we have $\Gamma\tilde{x}^k = (\Gamma - Q)x^k + A^Tb$. This leads to $\tilde{x}^k = (I - \Gamma^{-1}Q)x^k + \Gamma^{-1}[A^Tb]$. It is not hard to verify that the optimal solution $x^* = A^{-1}b$ satisfies $\Gamma^{-1}[A^Tb] = x^* - (I - \Gamma^{-1}Q)x^*$, thus $\tilde{x}^k - x^* = (I - \Gamma^{-1}Q)(x^k - x^*)$.

B Proposition of $\|\Gamma^T A^{-1}\Gamma\|$

Proposition B.1 (Claim B.2 in [20])**.** *Let A be a positive definite matrix with condition number κ and Γ is the lower triangular matrix of A. Then*

$$\|\Gamma^T A^{-1}\Gamma\| \le \kappa \cdot \min\left\{\sum_i L_i, (2 + \frac{1}{\pi}\log n)^2 L\right\}. \tag{76}$$

Proof. Denote

$$\Gamma_{\text{unit}} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0\\ 1 & 1 & 0 & \dots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & 1 & \dots & 0\\ 1 & 1 & 1 & \dots & 1 \end{bmatrix},$$

then $\Gamma = \Gamma_{\text{unit}} \circ A$, where $\circ$ denotes the Hadamard product. According to the classical result on the operator norm of the triangular truncation operator [56, Theorem 1], we have

$$\|\Gamma\| = \|\Gamma_{\text{unit}} \circ A\| \le (1 + \frac{1}{\pi} + \frac{1}{\pi}\log n)\|A\| \le (2 + \frac{1}{\pi}\log n)\|A\|.$$

Thus we have

$$\|\Gamma^T A^{-1}\Gamma\| \le \|\Gamma^T\Gamma\|\|A^{-1}\| = \|\Gamma\|^2 \frac{1}{\lambda_{\min}(A)} \tag{77}$$

$$\le (2+\frac{1}{\pi}\log n)^2 \frac{\|A\|^2}{\lambda_{\min}(A)} = (2+\frac{1}{\pi}\log n)^2 \kappa L, \tag{78}$$

which proves the second part of (76).

We can bound $\|\Gamma\|^2$ in another way (denote λ_i's as the eigenvalues of A):

$$\begin{aligned} \|\Gamma\|^2 \le \|\Gamma\|_F^2 = \frac{1}{2}(\|A\|_F^2 + \sum_i A_{ii}^2) = \frac{1}{2}\left(\sum_i \lambda_i^2 + \sum_i A_{ii}^2\right) \\ \le \frac{1}{2}\left((\sum_i \lambda_i)\lambda_{\max} + L_{\max}\sum_i A_{ii}\right) \overset{\text{(i)}}{=} \frac{1}{2}(L+L_{\max})\sum_i L_i \le L\sum_i L_i. \end{aligned} \tag{79}$$

where(i) is because $\sum_i \lambda_i = \mathrm{tr}(A) = \sum_i A_{ii}$ and $A_{ii} = L_i$. Thus

$$\|\Gamma^T A^{-1}\Gamma\| \le \|\Gamma\|^2 \frac{1}{\lambda_{\min}(A)} \overset{(79)}{\le} \frac{L}{\lambda_{\min}}\sum_i L_i = \kappa \sum_i L_i.$$

which proves the first part of (76). □

C lemma C.1

Lemma C.1. *Let a matrix $Q \in \mathbb{R}^{n\times n}$ and suppose $Q_{11} = 1$. Denote Γ to be the lower triangular of Q, and*

$$B \triangleq \begin{bmatrix} 1 & 0 \\ 0 & \Gamma_{2:n}^{-T} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & D_{2:n} \end{bmatrix}, \tag{80}$$

where Γ^{-T} is the transpose of the inverse of Γ and D is the diagonal matrix of Q, i.e. the diagonal entries of D is the same as Q and the off-diagonal entries are 0. $\Gamma_{2:n}^{-T}$ and $D_{2:n}$ are the sub-matrices by excluding the first row and first column of $\Gamma_{2:n}^{-T}$ and $D_{2:n}$ respectively .

The key step in the proof of lemma C.1 is to rewrite $B^{-1}Q^{-1}\Gamma$ and $\Gamma^T Q^{-1}\Gamma$ into four block matrices and compare the eigenvalues of the block matrices. If the eigenvalues are the same for the same block matrices of $B^{-1}Q^{-1}\Gamma$ and $\Gamma^T Q^{-1}\Gamma$, then the eigenvalues of the original matrices are the same by the connections between eigenvalues and trace of matrices.

Proof. WLOG, suppose all the diagonal entries of Q are 1. Denote $\mathrm{eig}(Z)$ as the set of eigenvalues of Z (allow repeated elements; e.g. if Z has eigenvalues 1.7 with multiplicity 2, then we define $\mathrm{eig}(Z) = \{1.7, 1.7\}$). We first notice the following simple fact.

Fact: For any square matrix M with the following block structure:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ \mathbf{0}_{n-1:n-1} & M_{22} \end{bmatrix}, \tag{81}$$

whrere M_{11}, M_{12} and M_{22} are submatrices of M with appropriate shapes, $\mathbf{0}_{n-1:n-1}$ is an $n-1$ by $n-1$ zero matrix, $\mathrm{eig}(M) = \mathrm{eig}(M_{11}) \cup \mathrm{eig}(M_{22})$.

The fact is simple to prove: the characteristic polynomial of M is $\det(\lambda I - M) = \det(\lambda I - M_{11}) \det(\lambda I - M_{22})$. Since $\det(\lambda I - Z) = \Pi_{\mu \in \text{eig}(Z)}(\lambda - \mu)$, we have $\text{eig}(M) = \text{eig}(M_{11}) \cup \text{eig}(M_{22})$.

Since Γ is the lower triangular part of Q, then the first column of $Q^{-1}\Gamma$ is equal to the first column of the identity matrix, i.e. a column vector in the form: $(1, 0, \cdots, 0)^T$. $Q^{-1}\Gamma$ can be represented in terms of its submatrices:

$$Q^{-1}\Gamma = \begin{bmatrix} 1 & \mathbf{b_{12}} \\ \mathbf{0} & \mathbf{b_{22}} \end{bmatrix}, \tag{82}$$

where $\mathbf{0}$ represents a column vector of all zeros in length of $n-1$.

We write B^{-1} and Γ^T into block matrices:

$$B^{-1} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \Gamma^T_{2:n} \end{bmatrix}, \quad \Gamma^T = \begin{bmatrix} 1 & \mathbf{\Gamma^T_{1,2:n}} \\ \mathbf{0} & \Gamma^T_{2:n} \end{bmatrix}, \tag{83}$$

where $\mathbf{0}^T$ is a row vector of all zeros in length of $n-1$. $\mathbf{\Gamma^T_{1,2:n}}$ is a row vector of length $n-1$ and its entries are the last $n-1$'s entries of the first row of Γ^T. $\Gamma^T_{2:n}$ is the submatrix of Γ^T which does not contain the first row and the last row of Γ^T. Note that we assume $Q_{11} = 1$, so the upper left block of Γ^T is 1.

Using eq. (83), we compute $B^{-1}Q^{-1}\Gamma$ and $\Gamma^T Q^{-1}\Gamma$ in the form eq. (81).

$$B^{-1}Q^{-1}\Gamma = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \Gamma^T_{2:n} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{b_{12}} \\ \mathbf{0} & \mathbf{b_{22}} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{b_{12}} \\ \mathbf{0} & \Gamma^T_{2:n}\mathbf{b_{22}} \end{bmatrix} \tag{84}$$

$$\Gamma^T Q^{-1}\Gamma = \begin{bmatrix} 1 & \mathbf{\Gamma^T_{1,2:n}} \\ \mathbf{0} & \Gamma^T_{2:n} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{b_{12}} \\ \mathbf{0} & \mathbf{b_{22}} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{b_{12}} + \mathbf{\Gamma^T_{1,2:n}}\mathbf{b_{22}} \\ \mathbf{0} & \Gamma^T_{2:n}\mathbf{b_{22}} \end{bmatrix} \tag{85}$$

Therefore, the diagonal blocks of $B^{-1}Q^{-1}$ and $\Gamma^T Q^{-1}\Gamma$ are the same, and thus, by the fact we introduced at the beginning of the proof, the eigenvalues of $B^{-1}Q^{-1}\Gamma$ and $\Gamma^T Q^{-1}\Gamma$ are the same. □

D Supplemental Proofs for theorem 3.2

D.1 Proof of proposition 6.1

Proof. Denote $\tilde{c} = 1 - c$ and $J = ee^T$, where e is a vector with 1 on every entry. Observe that

$$Q = cJ + \tilde{c}I. \tag{86}$$

Recall that by Sherman-Morrison formula [57], if a matrix is in the form of $W + uv^T$, where u, v are vectors and the matrix W is nonsingular, then the inverse of the matrix $W + uv^T$ is

$$(W + uv^T)^{-1} = W^{-1} - \frac{W^{-1}uv^T W^{-1}}{1 - v^T W^{-1} u}. \tag{87}$$

Since Q is invertible and $I = ee^T$, then we can apply the Sherman-Morrison formula to obtain the inverse of Q.

$$Q^{-1} = (cJ + \tilde{c}I)^{-1} = \frac{1}{\tilde{c}}I - \frac{c}{\tilde{c}(\tilde{c} + cn)}J. \tag{88}$$

□

D.2 Proof of proposition 6.3

Proof. By assumption of Q from eq. (56), we know Q is a positive definite matrix. With simple calculations, we know Q has one eigenvalue $1-c$ with multiplicity $n-1$ and one eigenvalue $1-c+cn$ with multiplicity 1. Hence, $\|Q\| = 1-c+cn$. □

D.3 Proof of proposition 6.2

Proof.

$$\Gamma\Gamma^T - Q = c^2 \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 2 & 2 & \dots & 2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2 & 3 & \dots & n-1 \end{bmatrix} \triangleq c^2 M \tag{89}$$

We apply proposition D.1, which is proved in [58], to approximate the spectral norm of $\Gamma\Gamma^T - Q$.

Proposition D.1 (Proposition 3.2 in [58])**.** *Let* $M = [m_{ij}] = [\min(i,j)]$*, define* $\theta_k := \frac{2k\pi}{2n+1}$*. Then* $\|M\| = (2+2\cos\theta_n)^{-1} \gtrapprox 4n^2/\pi^2$.

Observe that the submatrix $M_{2:n}$ is a special matrix which satisfies the definition $M_ij = \min(i,j)$ in proposition D.1. Since the entries on the first row and the first column of M are all zeros, then

$$\|\Gamma\Gamma^T - Q\| = c^2\|M\| = c^2 = \|M_{2:n}\|.$$

From the approximation of the spectral norm of $M_{2:n}$, we obtain an approximation of the spectral norm of $\Gamma\Gamma^T - Q$.

$$\|\Gamma\Gamma^T - Q\| \gtrapprox c^2 \frac{4n^2}{\pi^2}. \tag{90}$$

□

D.4 Proof of proposition 6.4

Proof. We observe that

$$\Gamma^T e = \begin{bmatrix} 1 & c & \cdots & c & c \\ 0 & 1 & \cdots & c & c \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & c \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+c(n-1) \\ 1+c(n-2) \\ \vdots \\ 1+c \\ 1 \end{bmatrix} \tag{91}$$

Based on the observation eq. (91), we can directly compute the spectral norm of $\Gamma^T JL$.

$$\begin{aligned}
\|\Gamma^T J \Gamma\| &= \|\Gamma^T e e^T \Gamma\| && \text{(92a)}\\
&= \|\Gamma^T e\|^2 && \text{(92b)}\\
&= \sum_{i=1}^{n} (1 + c(n-i))^2 && \text{(92c)}\\
&= \sum_{i=1}^{n} 1 + \sum_{i=1}^{n} 2c(n-i) + \sum_{i=1}^{n} c^2 (n-i)^2 && \text{(92d)}\\
&= n + cn(n-1) + \frac{c^2}{6} n(n-1)(2n-1). && \text{(92e)}
\end{aligned}$$

□

References

[1] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[2] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[3] J. Platt, "Fast training of support vector machines using sequential minimal optimization. advances in kernel methods—support vector learning (pp. 185–208)," *AJ, MIT Press, Cambridge, MA*, 1999.

[4] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear svm," in *Proceedings of the 25th international conference on Machine learning*, pp. 408–415, ACM, 2008.

[5] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[6] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[7] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[8] D. Leventhal and A. S. Lewis, "Randomized methods for linear constraints: convergence rates and conditioning," *Mathematics of Operations Research*, vol. 35, no. 3, pp. 641–654, 2010.

[9] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[10] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *Journal of Machine Learning Research*, vol. 14, no. Feb, pp. 567–599, 2013.

[11] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, no. 1-2, pp. 1–38, 2014.

[12] Z. Lu and L. Xiao, "On the complexity analysis of randomized block-coordinate descent methods," *Mathematical Programming*, vol. 152, no. 1-2, pp. 615–642, 2015.

[13] Q. Zheng, P. Richtarik, and T. Zhang, "Randomized dual coordinate ascent with arbitrary sampling," *arXiv preprint arXiv:1411.5873*, 2015.

[14] Q. Lin, Z. Lu, and L. Xiao, "An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 2244–2273, 2015.

[15] Y. Zhang and L. Xiao, "Stochastic primal-dual coordinate method for regularized empirical risk minimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2939–2980, 2017.

[16] O. Fercoq and P. Richtárik, "Accelerated, parallel, and proximal coordinate descent," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 1997–2023, 2015.

[17] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar, "An asynchronous parallel stochastic coordinate descent algorithm," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 285–322, 2015.

[18] A. Patrascu and I. Necoara, "Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization," *Journal of Global Optimization*, vol. 61, no. 1, pp. 19–46, 2015.

[19] C.-J. Hsieh, H.-F. Yu, and I. S. Dhillon, "Passcode: Parallel asynchronous stochastic dual co-ordinate descent.," in *ICML*, pp. 2370–2379, 2015.

[20] R. Sun and Y. Ye, "Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version," *arXiv preprint arXiv:1604.07130*, 2016.

[21] R. Glowinski and A. Marroco, "Approximation par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, vol. 9, no. R2, pp. 41–76, 1975.

[22] T. F. Chan and R. Glowinski, *Finite element approximation and iterative solution of a class of mildly non-linear elliptic equations*. Computer Science Department, Stanford University Stanford, 1978.

[23] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[24] 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.

[25] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.

[26] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of admm for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, vol. 155, no. 1-2, pp. 57–79, 2016.

[27] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *arXiv preprint arXiv:1208.3922*, 2012.

[28] D. Han and X. Yuan, "A note on the alternating direction method of multipliers," *Journal of Optimization Theory and Applications*, vol. 155, no. 1, pp. 227–238, 2012.

[29] C. Chen, Y. Shen, and Y. You, "On the convergence analysis of the alternating direction method of multipliers with three blocks," in *Abstract and Applied Analysis*, vol. 2013, Hindawi Publishing Corporation, 2013.

[30] B. He, H.-K. Xu, and X. Yuan, "On the proximal jacobian decomposition of alm for multiple-block separable convex minimization problems and its relationship to admm," 2013.

[31] B. He, L. Hou, and X. Yuan, "On full jacobian decomposition of the augmented lagrangian method for separable convex programming," *Preprint*, 2013.

[32] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o (1/k) convergence," *arXiv preprint arXiv:1312.3040*, 2013.

[33] T. Lin, S. Ma, and S. Zhang, "On the convergence rate of multi-block admm," *arXiv preprint arXiv:1408.4265*, vol. 229, 2014.

[34] M. Hong, T.-H. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z.-Q. Luo, "A block successive upper bound minimization method of multipliers for linearly constrained convex optimization," *arXiv preprint arXiv:1401.7079*, 2014.

[35] X. Cai, D. Han, and X. Yuan, "The direct extension of ADMM for three-block separable convex minimization models is convergent when one function is strongly convex," *Optimization Online*, 2014.

[36] D. Sun, K.-C. Toh, and L. Yang, "A convergent proximal alternating direction method of multipliers for conic programming with 4-block constraints," *arXiv preprint arXiv:1404.5378*, 2014.

[37] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the admm with multi-block variables," *arXiv preprint arXiv:1408.4266*, 2014.

[38] D. Han, X. Yuan, and W. Zhang, "An augmented lagrangian based parallel splitting method for separable convex minimization with applications to image processing," *Mathematics of Computation*, vol. 83, no. 289, pp. 2263–2291, 2014.

[39] X. Li, D. Sun, and K.-C. Toh, "A schur complement based semi-proximal admm for convex quadratic conic programming and extensions," *Mathematical Programming*, pp. 1–41, 2014.

[40] M. Li, D. Sun, and K.-C. Toh, "A convergent 3-block semi-proximal admm for convex minimization problems with one strongly convex block," *Asia-Pacific Journal of Operational Research*, p. 1550024, 2015.

[41] T. Lin, S. Ma, and S. Zhang, "Iteration complexity analysis of multi-block admm for a family of convex minimization without strong convexity," *arXiv preprint arXiv:1504.03087*, 2015.

[42] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block admm with o (1/k) convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

[43] B. He, M. Tao, and X. Yuan, "Alternating direction method with gaussian back substitution for separable convex programming," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 313–340, 2012.

[44] B. He, M. Tao, and X. Yuan, "Convergence rate and iteration complexity on the alternating direction method of multipliers with a substitution procedure for separable convex programming," *Math. Oper. Res., under revision*, vol. 2, 2012.

[45] X. Li, D. Sun, and K.-C. Toh, "A schur complement based semi-proximal admm for convex quadratic conic programming and extensions," *Mathematical Programming*, vol. 155, no. 1-2, pp. 333–373, 2016.

[46] L. Chen, D. Sun, and K.-C. Toh, "An efficient inexact symmetric gauss–seidel based majorized admm for high-dimensional convex composite conic programming," *Mathematical Programming*, vol. 161, no. 1-2, pp. 237–270, 2017.

[47] R. Sun, Z.-Q. Luo, and Y. Ye, "On the expected convergence of randomly permuted admm," *arXiv preprint arXiv:1503.06387*, 2015.

[48] B. Recht and C. Ré, "Beneath the valley of the noncommutative arithmetic-geometric mean inequality: conjectures, case-studies, and consequences," *arXiv preprint arXiv:1202.4184*, 2012.

[49] S. J. Wright and C.-P. Lee, "Analyzing random permutations for cyclic coordinate descent," *arXiv preprint arXiv:1706.00908*, 2017.

[50] C.-P. Lee and S. J. Wright, "Random permutations fix a worst case for cyclic coordinate descent," *IMA Journal of Numerical Analysis*, vol. 39, no. 3, pp. 1246–1275, 2018.

[51] M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo, "Why random reshuffling beats stochastic gradient descent," *arXiv preprint arXiv:1510.08560*, 2015.

[52] M. Gurbuzbalaban, A. Ozdaglar, N. D. Vanli, and S. J. Wright, "Randomness and permutations in coordinate descent methods," *arXiv preprint arXiv:1803.08200*, 2018.

[53] C. Chen, M. Li, X. Liu, and Y. Ye, "On the convergence of multi-block alternating direction method of multipliers and block coordinate descent method," *arXiv preprint arXiv:1508.00193*, 2015.

[54] X. Li, D. Sun, and K.-C. Toh, "A block symmetric gauss–seidel decomposition theorem for convex composite quadratic programming and its applications," *Mathematical Programming*, vol. 175, no. 1-2, pp. 395–418, 2019.

[55] C. A. Floudas and P. M. Pardalos, *Encyclopedia of optimization*, vol. 1. Springer Science & Business Media, 2001.

[56] J. R. Angelos, C. C. Cowen, and S. K. Narayan, "Triangular truncation and finding the norm of a hadamard multiplier," *Linear algebra and its applications*, vol. 170, pp. 117–135, 1992.

[57] J. Sherman and W. J. Morrison, "Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix," *The Annals of Mathematical Statistics*, vol. 21, pp. 124–127, mar 1950.

[58] S. Sra, "Explicit diagonalization of an anti-triangular cesaró matrix," 2014.