# Sketch and Project: Randomized Iterative Methods for Linear Systems and Inverting Matrices


THE UNIVERSITY *of* EDINBURGH

Robert Mansel Gower

Supervisor:
Dr. Peter Richtárik

Thesis submitted for the degree of Doctor of Philosophy

School of Mathematics

The University of Edinburgh

2016

# Abstract

Probabilistic ideas and tools have recently begun to permeate into several fields where they had traditionally not played a major role, including fields such as numerical linear algebra and optimization. One of the key ways in which these ideas influence these fields is via the development and analysis of randomized algorithms for solving standard and new problems of these fields. Such methods are typically easier to analyze, and often lead to faster and/or more scalable and versatile methods in practice.

This thesis explores the design and analysis of new randomized iterative methods for solving linear systems and inverting matrices. The methods are based on a novel sketch-and-project framework. By sketching we mean, to start with a difficult problem and then randomly generate a simple problem that contains all the solutions of the original problem. After sketching the problem, we calculate the next iterate by projecting our current iterate onto the solution space of the sketched problem.

The starting point for this thesis is the development of an archetype randomized method for solving linear systems. Our method has six different but equivalent interpretations: sketch-and-project, constrain-and-approximate, random intersect, random linear solve, random update and random fixed point. By varying its two parameters – a positive definite matrix (defining geometry), and a random matrix (sampled in an i.i.d. fashion in each iteration) – we recover a comprehensive array of well known algorithms as special cases, including the randomized Kaczmarz method, randomized Newton method, randomized coordinate descent method and random Gaussian pursuit. We also naturally obtain variants of all these methods using blocks and importance sampling. However, our method allows for a much wider selection of these two parameters, which leads to a number of new specific methods. We prove exponential convergence of the expected norm of the error in a single theorem, from which existing complexity results for known variants can be obtained. However, we also give an exact formula for the evolution of the expected iterates, which allows us to give lower bounds on the convergence rate.

We then extend our problem to that of finding the projection of given vector onto the solution space of a linear system. For this we develop a new randomized iterative algorithm: *stochastic dual ascent (SDA)*. The method is dual in nature, and iteratively solves the dual of the projection problem. The dual problem is a non-strongly concave quadratic maximization problem without constraints. In each iteration of SDA, a dual variable is updated by a carefully chosen point in a subspace spanned by the columns of a random matrix drawn independently from a fixed distribution. The distribution plays the role of a parameter of the method. Our complexity results hold for a wide family of distributions of random matrices, which opens the possibility to fine-tune the stochasticity of the method to particular applications. We prove that primal iterates

associated with the dual process converge to the projection exponentially fast in expectation, and give a formula and an insightful lower bound for the convergence rate. We also prove that the same rate applies to dual function values, primal function values and the duality gap. Unlike traditional iterative methods, SDA converges under virtually no additional assumptions on the system (e.g., rank, diagonal dominance) beyond consistency. In fact, our lower bound improves as the rank of the system matrix drops. By mapping our dual algorithm to a primal process, we uncover that the SDA method is the dual method with respect to the sketch-and-project method from the previous chapter. Thus our new more general convergence results for SDA carry over to the sketch-and-project method and all its specializations (randomized Kaczmarz, randomized coordinate descent...etc). When our method specializes to a known algorithm, we either recover the best known rates, or improve upon them. Finally, we show that the framework can be applied to the distributed average consensus problem to obtain an array of new algorithms. The randomized gossip algorithm arises as a special case.

In the final chapter, we extend our method for solving linear system to inverting matrices, and develop a family of methods with specialized variants that maintain symmetry or positive definiteness of the iterates. All the methods in the family converge globally and exponentially, with explicit rates. In special cases, we obtain stochastic block variants of several quasi-Newton updates, including bad Broyden (BB), good Broyden (GB), Powell-symmetric-Broyden (PSB), Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS). Ours are the first stochastic versions of these updates shown to converge to an inverse of a fixed matrix. Through a dual viewpoint we uncover a fundamental link between quasi-Newton updates and approximate inverse preconditioning. Further, we develop an adaptive variant of the randomized block BFGS (AdaRBFGS), where we modify the distribution underlying the stochasticity of the method throughout the iterative process to achieve faster convergence. By inverting several matrices from varied applications, we demonstrate that AdaRBFGS is highly competitive when compared to the well established Newton-Schulz and approximate preconditioning methods. In particular, on large-scale problems our method outperforms the standard methods by orders of magnitude. The development of efficient methods for estimating the inverse of very large matrices is a much needed tool for preconditioning and variable metric methods in the big data era.

# Lay Summary

This thesis explores the design and analysis of methods (algorithms) for solving two common problems: solving linear systems of equations and inverting matrices. Many engineering and quantitative tasks require the solution of one of these two problems. In particular, the need to solve linear systems of equations is ubiquitous in essentially all quantitative areas of human endeavour, including industry and science. Specifically, linear systems are a central problem in numerical linear algebra, and play an important role in computer science, mathematical computing, optimization, signal processing, engineering, numerical analysis, computer vision, machine learning, and many other fields. This thesis proposes new methods for solving large dimensional linear systems and inverting large matrices that use tools and ideas from probability.

The advent of large dimensional linear systems of equations, based on big data sets, poses a challenge. On these large linear systems, the traditional methods for solving linear systems can take an exorbitant amount of time. To address this issue we propose a new class of randomized methods that are capable of quickly obtaining approximate solutions. This thesis lays the foundational work of this new class of randomized methods for solving linear systems and inverting matrices. The main contributions are providing a framework to design and analyze new and existing methods for solving linear systems. In particular, our framework unites many existing methods. For inverting matrices we also provide a framework for designing and analysing methods, but moreover, using this framework we design a highly competitive method for computing an approximate inverse of truly large scale positive definite matrices. Our new method often outperforms previously known methods by several orders of magnitude on large scale matrices.

# Author's Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, either in whole or in part, in any previous application for a degree. Except where otherwise acknowledged, the work presented is entirely my own. During the course of the PhD program I co-authored six papers [50, 47, 48, 51, 49, 52], all of which I was the first author. This thesis is based on three of these papers, with the table below indicating which chapters are on which papers. I confirm that I contributed to all the results within the papers on which this thesis is based.

| Chapter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Paper | [51, 49, 52] | [51] | [49] | [52] |

Robert Mansel Gower

29th of February 2016

# Acknowledgments

I am immensely grateful to my supervisor Dr. Peter Richtárik. Peter has been a mentor to me on all fronts of being a researcher. From the very process of developing novel research directions, to clear and elegant mathematical writing, delivering the best presentations, and the many facets of applying for a job academia. Peter is my role model for being a researcher and supervisor. Furthermore, his attempts at besting me at table tennis were also admirable.

I would like to thank Prof. Jacek Gondzio for numerous research discussions, support and collaboration. Many thanks to my second supervisor Dr. Andreas Grothey for accompanying the progress of the PhD through the years and career discussions. I am grateful to my examination committee, Prof. Nicholas J. Higham and Prof. Ben Leimkuhler for their many detailed pointers, suggestions for improvement and for their time and insightful questions.

I am indebted to the school of Mathematics of the University of Edinburgh, for not only providing me with a wonderful work environment, funding yearly trips to the Firbush sports center, but, most of all, for directly funding my PhD. I am most grateful to the Dr. Laura Wisewell Travel scholarship for funding my conference travels expenses in 2013 and 2015. The chance to participate in international conferences, with all the experts in my area in one place, was invaluable.

One of the many highlights of my PhD years was a chance encounter Prof. Donald Goldfarb at the Optimization and Big Data 2015 Workshop in Edinburgh. There we discovered that we both have impeccable taste in research projects, and had independently arrived at the same extension to Don's hailed BFGS method. Prof. Donald Goldfarb has since been an inspiration to me, a great enthusiastic collaborator, and warm person in general. Thanks also to Don for having me over at Columbia University, I look forward to our continued work together.

My brother, Dr. Artur Gower, has been a constant support throughout my PhD studies and throughout my life (he even ventured into the world just before me to check that the coast was clear). Being a year ahead of me on a PhD program in Ireland, Art has given me the heads-up on how to write good science, applying for funding, and finally how to land a job in science. Art went so far as to read draft's of my papers, proposals and co-authored a paper with me. For his ricochet advice, originally directed toward my brother Artur but then finding its way to me, I would like to thank Prof. Michel Destrade.

Thanks to my many friends in Edinburgh, without whom life in Edinburgh would have been as grey as the weather and stone that envelopes it. In particular, to my cohort (in order of their appearance) Pablo Gonzalez Brevis, Kimon Fountoulakis, Tim Schultz, Hanyi Chen, Jakub Konečný, Dominik Csiba and Nicolas Loizou I extend a heartfelt thanks for many discussions and their friendship. I would specially like to thank my current and past flatmates Tarek Alabbas and Clara Vergez. Thanks Tarek for your friendship through all the years and being my fellow

breakfast musketeer. The only reason I made it to work at any reasonable time was down to Clara banging on my bedroom door early in the morning for breakfast. Clara you have been an amazing flatmate, friend, and salsa co-star; thanks bro-ster.

Furthermore, I want to thank my mother Elza Maria, who made my existence possible, and for her unconditional and immeasurable love and support.

Finally, many thanks to Jess (aka leao fofinha), for her loving support, companionship, being my sous-chef, deputy nutritionist and just being awesome in general.


It takes some Nerv to do a PhD

# Contents

# List of Symbols

| | |
|---|---|
| $\langle x, y\rangle$ | The standard Euclidean inner product of $x, y \in \mathbb{R}^n$, $\langle x, y\rangle = \sum_{i=1}^n x_i y_i$ |
| $\|x\|_2$ | The standard Euclidean norm of $x \in \mathbb{R}^n$, $\|x\|_2 = \sqrt{\langle x, x\rangle}$ |
| $\|x\|_B$ | The norm defined by symmetric positive definite $B \in \mathbb{R}^{n\times n}$, $\|x\|_B = \sqrt{\langle Bx, x\rangle}$ |
| $e^i$ | The $i$th coordinate vector in $\mathbb{R}^m$ |
| $f_i$ | The $i$th coordinate vector in $\mathbb{R}^n$ |
| $\dim(V)$ | The dimension of $V$ where $V$ is a subspace |
| $\mathbf{Rank}\,(M)$ | The rank of $M$, where $M$ is a matrix |
| $\mathbf{Null}\,(M)$ | The nullspace of $M$, $\mathbf{Null}\,(M) = \{x \,\vert\, Mx = 0\}$ |
| $\mathbf{Range}\,(M)$ | The rangespace of $M$, e.g. if $M \in \mathbb{R}^{m\times n}$ then $\mathbf{Range}\,(M) = \{Mx \,\vert\, x \in \mathbb{R}^n\}$ |
| $\lambda_{\max}(M)$ | The largest eigenvalue of $M$ |
| $\lambda_{\min}(M)$ | The smallest eigenvalue of $M$ |
| $\lambda^+_{\min}(M)$ | The smallest nonzero eigenvalue of $M$ (assuming that $M$ is a nonzero matrix) |
| $M^\dagger$ | The Moore-Penrose pseudoinverse of $M$. |
| $\mathbf{Tr}\,(M)$ | The trace of $M$, e.g. if $M \in \mathbb{R}^{n\times n}$ then $\mathbf{Tr}\,(M) = \sum_{i=1}^n M_{ii}$ |
| $\|M\|_F$ | The Frobenius norm of $M$, $\|M\|_F = \sqrt{\mathbf{Tr}\,(M^\top M)}$ |
| $\|M\|_2$ | The spectral norm of $M$, $\|M\|_2 = \max_{\|v\|_2=1} \|Mv\|_2 = \sqrt{\lambda_{\max}(M^\top M)}$ |
| $\|M\|_B$ | $\stackrel{\text{def}}{=} \|B^{1/2} M B^{-1/2}\|_2 = \max_{\|v\|_B=1} \|Mv\|_B$ |
| $\|M\|_B^*$ | $\stackrel{\text{def}}{=} \|B^{1/2} M B^{1/2}\|_2$ |

### Special Matrices

| | | |
|---|---|---|
| $A$ | $\stackrel{\text{def}}{=}$ | The $m \times n$ real system matrix (In Chapter 4 we use $m = n$) |
| $S$ | $\stackrel{\text{def}}{=}$ | A random $m \times q$ matrix |
| $B$ | $\stackrel{\text{def}}{=}$ | A symmetric positive definite $n \times n$ matrix |
| $Z$ | $\stackrel{\text{def}}{=}$ | $A^\top S (S^\top A B^{-1} A^\top S)^\dagger S^\top A$ |
| $H$ | $\stackrel{\text{def}}{=}$ | $\mathbf{E}_{S\sim\mathcal{D}}\left[S\left(S^\top A B^{-1} A^\top S\right)^\dagger S^\top\right]$ |

CHAPTER 1

# Introduction

*We can only see a short distance ahead, but we can see plenty there that needs to be done.*

Alan Turing

## 1.1 Introduction: What's to Come

This thesis explores the design and analysis of new randomized iterative methods for solving linear systems and inverting matrices. All the methods presented in this thesis are globally and linearly convergent. Consequently, the methods are well suited to quickly obtain approximate solutions.

The methods are based on a novel sketch-and-project framework. By sketching we mean, to start with a difficult problem and then randomly generate a simple problem that contains all the solutions of the original problem. For instance, consider the linear system

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. We suppose throughout the thesis that there exists a solution $x^* \in \mathbb{R}^n$ to the linear system, that is, the linear system is *consistent*. Let $S \in \mathbb{R}^{m\times q}$ be a random matrix with the same number of rows as $A$ but far fewer columns ($q \ll n$.) The resulting *sketched* linear system is given by

$$S^\top Ax = S^\top b,$$

which has a relatively small number of rows, and is thus easier to solve. There has been a concerted effort into designing the distribution of $S$ with the property that the solution set of the sketched linear system is close to the solution set of the original linear system with high probability, particularly so for solving linear systems that arise from the least-squares problem [96, 77, 32]. Determining an $S$ with such a property can be difficult and often depends on properties of $A$ that are expensive to compute. Here we take a different approach and use sketching combined with a projection process. We apply our sketch-and-project technique not only to solve linear systems, but also to find the projection of a vector onto the solution space of a linear system and to invert matrices (by sketching, for example, the inverse equation $AX = I$).

Throughout the entirety of this thesis, we assume that we can access the system matrix $A$ through matrix-vector products. Thus every element $A_{ij}$ of the system matrix may not be explicitly available. The methods presented here are designed with this restriction in mind and are thus compatible with the setting where $A$ can only be accessed as an operator.

In the remainder of this section we give a summary of each chapter of this thesis. The detailed proofs and careful deductions of any claims made here are left to the chapters.

**Chapter 2: Linear Systems with Sketch-and-Project.** In Chapter 2 we develop an iterative process that gradually refines an approximate solution to (1.1) using a sequence of sketching matrices, as opposed to the one shot sketching method. To describe our method, let $x^k \in \mathbb{R}^n$ be our current estimate of the solution to (1.1). We obtain an improved estimate by projecting $x^k$ onto the solution space of a sketched system

$$x^{k+1} = \arg\min_x \|x^k - x\|_B^2 \quad \text{subject to} \quad S^\top A x = S^\top b, \tag{1.2}$$

where $S$ is drawn in each iteration independently from a pre-specified distribution, $B$ is a positive definite matrix and $\|x\|_B^2 \overset{\text{def}}{=} \langle Bx, x\rangle$. This iterative process has a closed form solution given by

$$x^{k+1} = x^k - B^{-1}A^\top S(S^\top A B^{-1} A^\top S)^\dagger S^\top (Ax^k - b), \tag{1.3}$$

where $\dagger$ denotes the (Moore-Penrose) pseudoinverse. Using the closed form expression for the update (1.3) we show that the iterates converge if $A$ has full column rank and under mild assumptions on the distribution of $S$. In particular, the convergence analysis will depend heavily on the following random matrix

$$Z \overset{\text{def}}{=} A^\top S(S^\top A B^{-1} A^\top S)^\dagger S^\top A,$$

which governs the iterative process (1.3). Indeed, we will show that when $A$ has full column rank and for any $x_0 \in \mathbb{R}^n$, the iterates (1.3) converge to the unique solution $x^* \in \mathbb{R}^n$ of the linear system exponentially fast according to

$$\mathbf{E}\left[\|x^k - x^*\|_B^2\right] \leq \rho^k \ \cdot \ \|x^0 - x^*\|_B^2, \tag{1.4}$$

where

$$\rho \overset{\text{def}}{=} 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}).$$

By $B^{-1/2}$ we denote the unique positive definite square root of $B^{-1}$. Therefore $B^{-1/2}B^{-1/2} = B^{-1}$. We use $\mathbf{E}\left[\cdot\right]$ to denote the expectation operator. For instance, $\mathbf{E}\left[Z\right]$ is the expected value of $Z$. Since $Z$ is a function of $S$, which is the only random component of $Z$, we have that $\mathbf{E}\left[Z\right]$ is an expectation taken over the distribution of $S$. Because of the importance that $Z$ plays in this thesis, later in this chapter in Section 1.3, we prove several properties of $Z$.

In Section 2.6 we design a discrete distribution for $S$ that yields easily interpretable convergence rates in terms of a scaled condition number. Furthermore, we show that by choosing $B$ and the distribution of $S$ appropriately we recover a comprehensive array of well known algorithms as special cases, such as the randomized Kaczmarz method [125] and randomized Coordinate Descent [68], demonstrating the expressive power of the framework.

Having established convergence rates for each method defined by $(B, S)$, we explore questions such as: *what distribution of $S$ yields the optimal convergence rates*? We determine that the optimal distribution of $S$, chosen from a family of discrete distributions, is the solution to a particular semidefinite program. This result determines, for instance, the optimal distribution for selecting the rows of the linear system in the randomized Kaczmarz method. Furthermore, the optimized randomized Kaczmarz method is shown to converge significantly faster than the randomized Kaczmarz method using the standard distribution for $S$.

Our framework also allows for $S$ to have a continuous distribution, and to give an insight into the possibilities, we present three methods based on a Gaussian sketching matrix $S$ and give convergence rates for each. We then conclude the chapter with further numeric experiments that compare the different methods presented throughout the chapter.

**Chapter 3: Stochastic Dual Ascent for Finding the Projection of a Vector onto a Linear System.** In Chapter 3 we consider the more general problem of finding the projection of a given vector $c \in \mathbb{R}^n$ onto the solution space of a linear system, that is

$$\min_{x \in \mathbb{R}^n} \quad \tfrac{1}{2}\|x - c\|_B^2 \quad \text{subject to} \quad Ax = b. \tag{1.5}$$

To solve this projection problem we develop a new randomized iterative algorithm: *stochastic dual ascent (SDA)*. The method is dual in nature, and iteratively solves the dual of the projection problem (1.5). The dual problem is a non-strongly concave quadratic maximization problem without constraints given by

$$\max_{y \in \mathbb{R}^m} \quad (b - Ac)^\top y - \tfrac{1}{2}\|A^\top y\|_{B^{-1}}^2. \tag{1.6}$$

Each iterate $y^{k+1} \in \mathbb{R}^m$ of the SDA method is carefully chosen from a random affine space that passes through the previous iterate

$$y^{k+1} \in y^k + \mathbf{Range}\,(S)\,, \tag{1.7}$$

where $S$ is a random matrix drawn independently from a fixed distribution. Specifically, $y^{k+1}$ is the point with least norm that maximizes the dual objective in (1.6) constrained within the random affine space (1.7).

By mapping our dual iterates (1.7) to primal iterates, we uncover that the SDA method is a dual version of the sketch-and-project method (1.3). We then proceed to strengthen our convergence results established in the previous chapter. First, we do away with the assumption that $A$ has full column rank and consider any nonzero matrix $A$ and consistent linear system. In this more general setting, we prove that for $x_0 \in c + \mathbf{Range}\left(B^{-1}A^\top\right)$ the primal iterates converge to the solution of (1.5) exponentially fast in expectation according to (1.4) with a convergence rate of

$$\rho = 1 - \lambda_{\min}^+(B^{-1/2}A^T H A B^{-1/2}),$$

where

$$H = \mathbf{E}\left[S(S^\top A B^{-1} A^\top S)^\dagger S^\top\right], \tag{1.8}$$

and $\lambda_{\min}^+(B^{-1/2}A^T H A B^{-1/2})$ denotes the smallest nonzero eigenvalue of $B^{-1/2}A^T H A B^{-1/2}$.

The only condition for this convergence to hold is that $H$ be nonsingular. We completely characterize the discrete distributions of $S$ for which $H$ is nonsingular in Section 3.4.1. This shows, for instance, that the Kaczmarz method converges so long as the system matrix has no zero rows. Thus assuming that $A$ has full column rank, as is required in the convergence theorems in Chapter 2, is an unnecessary assumption for proving convergence of the sketch-and-project method. But the full column rank assumption makes the proofs of convergence simpler and thus, for pedagogic reasons, we have presented the proofs assuming that $A$ has full column rank earlier on in Chapter 2. We present further improvements in the convergence analysis and give a tight and insightful lower bound for the convergence rate that depends on the rank of $A$.

We also prove that the same rate of convergence $\rho$ governs the convergence of the dual function values, primal function values and the duality gap. Unlike traditional iterative methods, SDA converges under virtually no additional assumptions on the system (e.g., rank, diagonal dominance) beyond consistency. In fact, our lower bound improves as the rank of the system matrix drops. When our method specializes to a known algorithm, we either recover the best known rates, or improve upon them. Finally, we show that the framework can be applied to the distributed average consensus problem to obtain an array of new algorithms. The randomized gossip algorithm arises as a special case [13, 90].

**Chapter 4: Randomized Matrix Inversion.** In Chapter 4 we extend our method for solving linear systems to inverting matrices, and develop a family of methods with a specialized variant which maintains symmetry or positive definiteness of the iterates.

The initial insight into our matrix inversion methods comes from the simple observation that for a nonsingular matrix $A \in \mathbb{R}^{n\times n}$ the inverse is the solution in $X$ to one of the *inverse equations*

$$AX = I \quad \text{or} \quad XA = I.$$

Our method for inverting $A$ calculates the new iterate $X_{k+1} \in \mathbb{R}^{n\times n}$ by projecting the previous iterate $X_k \in \mathbb{R}^{n\times n}$ onto the solution space of a sketched version of one of the two inverse equations: either the *row* sketched variant $S^\top AX = S^\top$ or the *column* sketched variant $XAS = S$, where $S \in \mathbb{R}^{n\times q}$ is a random matrix drawn from a fixed distribution at each iteration. For example, using the column sketched variant a new iterate is calculated by solving

$$X_{k+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \|X_k - X\|^2_{F(B)} \quad \text{subject to} \quad XAS = S, \tag{1.9}$$

where the norm is the weighted Frobenius norm. When $A$ is symmetric, it can be advantageous to maintain symmetries in the iterates $X_k$. We propose a sketch-and-project method that maintains symmetry in the iterates by imposing symmetry as a constraint in

$$X_{k+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \|X_k - X\|^2_{F(B)} \quad \text{subject to} \quad XAS = S, \quad X = X^\top. \tag{1.10}$$

All the methods we present converge globally and linearly, with the same explicit rate of convergence $\rho = 1-\lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})$. In special cases, we obtain stochastic block variants of several quasi-Newton updates, including the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update. Ours are the first stochastic quasi-Newton updates shown to converge to an inverse of a fixed matrix. Through a dual viewpoint we uncover a fundamental link between quasi-Newton

updates and approximate inverse preconditioning methods, which results in a new interpretation of the quasi-Newton methods. For instance, the BFGS update is the solution in $X$ to

$$X_{k+1} = \arg_X \min_{X\in\mathbb{R}^{n\times n}, Y\in\mathbb{R}^{n\times q}} \frac{1}{2}\|X - A^{-1}\|^2_{F(A)} \quad \text{subject to} \quad X = X_k + SY^\top + YS^\top,$$

for a particular (deterministic) choice of $S$. This shows that the BFGS udpate can be interpreted as a projection of $A^{-1}$ onto a space of symmetric matrices.

With explicit convergence rates for each method characterized by the distribution of $S$, we again raise the question of selecting a distribution of $S$ that results in a method with a faster convergence rate. Though different from the setting of solving a linear system, the goals of finding the inverse of $A$ and of designing a distribution of $S$ that results in an improved convergence rate are in synchrony. In particular, for many choices of $B$, it will transpire that the covariance of $S$ should be an estimate of the inverse of $A$. One way to interpret this result is that $S$ should be chosen so that it not only sketches/compresses the inverse equations $AX = I$ or $XA = I$, but also, it should improve the condition number of these equations.

This reasoning leads us to develop an adaptive variant of a randomized block BFGS (AdaRBFGS), where the distribution of $S$ depends on $X_k$. By inverting several matrices from varied applications, we demonstrate that AdaRBFGS is highly competitive when compared with the well established Newton-Schulz [115] and the approximate inverse preconditioning methods [19, 112, 46, 6]. In particular, on large-scale problems our method outperforms the standard methods by orders of magnitude. Since the inspiration behind AdaRBFGS method comes from the desire to design an *optimal adaptive* distribution for $S$ by examining the convergence rate, this work also highlights the importance of developing algorithms with explicit convergence rates.

**Organization of Thesis** Chapters 2, 3 and 4 are largely based on the papers [51, 49] and [52], respectively. Excluding preliminary results in linear algebra presented in Section 1.3, each of these chapter is mostly self-contained, including the objective, contributions, definitions and notation.

## 1.2 Why Randomized Methods

Why use randomization in algorithmic design? We can answer this question, in practical terms, by measuring the advantages of using randomized algorithms as compared with existing deterministic methods. The advantages in using randomized methods include: often algorithms that are easier to analyze and implement, better convergence in terms of improved convergence rates or range of convergence (the randomized methods are almost always globally convergent), lower memory requirements, and more scalable and parallelizable methods in practice.

To substantiate these claims, we now compare the sketch-and-project randomized methods (1.2) for solving the linear system (1.1) to stationary methods and the Krylov methods. Note that all the iterative methods mentioned here benefit from using a preconditioner, so much so, they are often only used in conjunction with a preconditioner. But to compare the methods on a equal footing, we assume no preconditioning strategy has been applied.

### 1.2.1 A Case Study Comparing to Stationary Methods

Iterative methods that fit the simple form

$$x^{k+1} = Gx^{k-1} + c, \tag{1.11}$$

are known as Stationary Methods, where $G \in \mathbb{R}^{n\times n}$ is the *iteration matrix* and $c \in \mathbb{R}^n$ is the *bias term*. Here neither $G$ nor $c$ depend on the iteration count. Methods that fit the format (1.11) include the Jacobi method, the Gauss-Seidel method, the Successive Overrelaxation (SOR) method and the Symmetric Successive Overrelaxation (SSOR) method [3, 113].

The sketch-and-project methods presented in this thesis (1.3) would fit the format (1.11) if it were not for the fact that $G$ and $c$ in our methods are randomly generated, and thus, can differ from one iteration to the next.

Despite this difference, the sketch-and-project methods and the stationary methods share many similarities.

**Low memory requirements.** Stationary methods and (1.3) are both easy to implement and have low memory requirements. Often only the previous iterate $x^k$ needs to be stored to enable the calculation of the next iterate.

**Existence of convergence rates.** A stationary method only converges if the spectral radius of $G$ is less than one [113], that is, if $\rho(G) < 1$. Often $G$ is constructed by splitting the system matrix $A$, e.g., for square systems the Jacobi method iteration matrix is $G = I - D^{-1}(A - D)$ where $D = \text{diag}(A_{11}, \ldots, A_{nn})$. Thus for $\rho(G) < 1$ to hold one needs strong assumptions on the spectrum of $A$. Again using the Jacobi method as an example, if $A$ is strictly diagonally dominant, then $\rho(G) < 1$ holds. In contrast, the sketch-and-project methods converge for virtually any matrix $A$ with an explicit convergence rate, as we show in Chapter 3.

**Parallelizable.** Due to their simple recurrence relationship, the stationary methods lead to straight forward parallel and distributed variants, see Section 2.5 in [7]. Furthermore,

variants of stationary method for solving nonlinear systems have also been adapted to parallel architectures [8, 110]. The sketch-and-project methods are similar in this aspect, in that, they are amenable to parallel implementations. For instance, the coordinate descent is member of the sketch-and-project family, and various distributed and parallel variants of coordinate descent have been designed to solve optimization problems [14, 37, 107, 104].

### 1.2.2 A Case Study Comparing to Krylov Methods

The Krylov methods are a well studied and established class of iterative methods for solving linear systems. In fact, the sketch-and-project methods share a certain similarity to Krylov methods. This can be seen by using the following equivalent dual formulation of (1.2) given by

$$x^{k+1} \;=\; \arg\min_{x\in\mathbb{R}^n} \|x \;- x^*\|_B^2 \quad \text{subject to} \quad x \in x^k + \mathbf{Range}\left(B^{-1}A^\top S\right). \tag{1.12}$$

We refer to (1.12) as the *constrain and approximate* viewpoint, because a new iterate $x^{k+1}$ is selected as the best possible *approximation* to the solution $x^*$ *constrained* to a randomly generated affine space. Later in Section 2.3 we prove that (1.2) and (1.12) are equivalent.

Formulation (1.12) is similar to the framework often used to describe Krylov methods [71, Chapter 1], which is

$$x^{k+1} = \arg\min_{x\in\mathbb{R}^n} \|x - x^*\|_B^2 \quad \text{subject to} \quad x \in x^0 + \mathcal{K}_{k+1}, \tag{1.13}$$

where $\mathcal{K}_{k+1} \subset \mathbb{R}^n$ is a $(k+1)$–dimensional subspace. Note that the constraint $x \in x^0+\mathcal{K}_{k+1}$ is an affine space that passes through $x^0$, as opposed to passing through $x^k$ in the sketch-and-project formulation (1.12). The objective $\|x - x^*\|_B^2$ is a generalization of the residual, where $B = A^\top A$ is used to characterize minimal residual methods (MINRES and GMRES) [92, 113] and $B = A$ is used to describe the Conjugate Gradients (CG) method [58]. Progress from one iteration to the next is guaranteed by using expanding nested search spaces at each iteration, that is, $\mathcal{K}_k \subset \mathcal{K}_{k+1}$.

**Low memory requirements.** To make an efficient Krylov method, the problem (1.13) should not be solved from scratch, but rather, one should build upon the knowledge that the previous iterate $x^k$ is the minima restricted to $\mathcal{K}_k$, and calculate $x^{k+1}$ by updating $x^k$ in an inexpensive manner. This inexpensive update is typically achieved through a *short recurrence* update, that is, an update applied to $x^k$ of the following form

$$x^{k+1} = x^k + \sum_{i=1}^{s} p^{k+2-i},$$

using a small number $s \in \mathbb{N}$ of vectors $p^{k+2-s}, \ldots, p^{k+1}$. To arrive at a short recurrence, one needs to carefully design an orthonormal basis for the spaces $\mathcal{K}_k$. Such a short recurrence does not always exist. By the Faber-Manteuffel Theorem [35], the sufficient and necessary conditions for a short recurrence in a Krylov method are that the system matrix is a nonsingular normal matrix with respect to the geometry defined by the $B$ matrix. [1]

[1] Though one cannot guarantee the existence of a short recurrence for a singular matrix in general, there do exist short recurrences for particular singular matrices that arise from deflation techniques [41].

In contrast, the sketch-and-project methods automatically have a short recurrence by design, as can be seen through (1.3), where $x^{k+1}$ is calculated by adding a single random vector to $x^k$ (a very short recurrence). Instead of using "growing" search spaces to guarantee convergence, the sketch-and-project methods are guaranteed to converge when the distribution of $S$ is such that the search space in (1.12) "covers" the space of interest with a nonzero probability[2]. Not only can we guarantee convergence on even singular matrices but, our lower bounds indicate that convergence improves for lower rank matrices.

**Existence of convergence analysis.** As for convergence rates, only certain instantiations of the Krylov methods, such as the CG, MINRES and GMRES methods have well understood convergence rates. These three aforementioned methods are only applicable when the system matrix $A$ is symmetric positive definite, symmetric and nonsingular, respectively. Again in contrast, the sketch-and-project methods converge for virtually any matrix $A$ with an explicit convergence rate, as we show in Chapter 3.

**Improved convergence rate.** As an example of how the rate of convergence of a Krylov method compares against the rate of convergence of a sketch-and-project type method, let us consider the setting where $A$ is positive definite. In this setting the CG method (a Krylov method) and the Coordinate Descent ($CD$) method (a sketch-and-project method) are the methods of choice. The iteration complexity of the CG method is $O(\sqrt{\lambda_{\max}(A)/\lambda_{\min}(A)})$ while the iteration complexity of the CD method is $O(\mathbf{Tr}\,(A)\,/\lambda_{\min}(A))$ (proof in Section 2.4.4). Thus the CD method requires at least the square of the number of iterations that the CG method requires to reach the same relative accuracy (ignoring the expectation). But this iteration complexity needs to be counter balanced with the very low iteration cost of the CD method. That is, an iteration of the CD method costs $O(n)$ while an iteration of the CG method costs $O(n^2)$ (ignoring possible gains in efficiency due to sparsity for both methods). This is a significant difference in iteration cost when solving large dimensional linear systems. Strohmer and Vershynin [125] give examples of when this lower cost of the CD method[3] pays off for the higher iteration complexity. Furthermore, accelerated versions of the CD method [67, 73, 132, 37] improve the iteration complexity of the CD method to $O(\sqrt{\mathbf{Tr}\,(A)\,/\lambda_{\min}(A)})$ at the cost of only a constant increase in the iteration cost. Thus when taking iteration cost into account, the accelerated CD methods asymptomatically achieve a solution with higher accuracy than the the CG method for the same computational cost. We do not consider accelerated variants of the CD method, or the sketch-and-project method, in this thesis. For further insight into how accelerated sketch-and-project methods compare with the CG method, see [73] for a comparison to an accelerated randomized Kaczmarz method and [67] for a comparison to an accelerated CD method.

**Parallelizable.** Where the sketch-and-project methods truly diverge from Krylov methods is in how parallelizable they are. Though we do not address this in this thesis, methods based

[2] As shown later in Chapter 3, this condition is captured in the requirement that $H$ (1.8) be nonsingular.

[3] In the paper [125] the authors compare the iteration complexity and iteration cost of the randomized Kaczmarz method against the Conjugate Gradients method applied to the least-squares problem. But their comparisons hold verbatim for the CD method and the CG method applied to a positive definite linear system.

on (1.2) are easy to adapt to a parallel architecture. In contrast, adapting the Krylov methods towards parallel computing is challenging. This is, in part, because of the delicate orthogonality conditions that need to be enforced on the basis of the search spaces $\mathcal{K}_k$ in order to guarantee short recurrences and convergence. A few of the current strategies towards adapting the Krylov methods for a parallel implementations are the following.

The conjugate gradients method can be paired with domain decomposition methods for solving linear systems that result from discretizing PDE's for an overall successful distributed method [31]. But in this strategy the gains in parallelism come from the domain decomposition method and not from the conjugate gradient method *per se*.

Efforts towards designing a distributed or communication efficient variant of Krylov methods are focused on two fronts: exploring the parallelism in matrix-vector and vector-vector products performed in the the Krylov methods [2], and the so called $s$–step Krylov methods.

The $s$–step methods, such as the $s$–step conjugate gradients methods [20], re-order the computations of standard Krylov methods so that $s$ iterations can be performed simultaneously and in parallel. Though implementing fast and reliable $s$–step methods come with two challenges: achieving numerical stability is more challenging than traditional Krylov methods [2] and correctly identifying the communication bottlenecks is difficult as it depends on the nonzero structure of the system matrix [2].

The sketch and project methods, on the other hand, are relatively easy to adapt to a parallel setting. First, the methods (1.2) make use of shared memory parallelism through their *block* variants. By block variants we refer to the setting when $S$ has more than one column. Through experiments and complexity analysis we see that block variants converge much fast than their "single column" counterparts. Furthermore, the main computational bottleneck, calculating $S^\top A$, can be completely amortized using multi-thread matrix-matrix products. This feature has been explored in implementing parallel coordinate descent methods [107]. In a distributed computing setting, the independence of the sampling of the matrix $S$ allows for distributed implementations, which has already been explored again for coordinate descent methods [103, 135, 1].

Thus there are a number of clear advantages in using the sketch-and-project methods, as compared with Krylov methods. This discussion has only been a small window into the advantages of randomized methods for solving linear systems. But randomized methods are having a profound impact on other related fields such as optimization methods, in particular, in minimizing partially separable functions, randomized methods are considered the state-of-the-art due, in part, to their fast convergence. Also in numerical linear algebra, with new randomized methods for solving least squares that, according to the experiments in [1], outperform the long-standing benchmark LAPACK.

## 1.3 Tools of the Trade

In this section we present several lemmas concerning pseudoinverses and projections. Though these lemmas are elementary in nature we include them for completeness since they are called upon several times throughout the thesis.

### 1.3.1 Pseudoinverse

The pseudoinverse matrix was introduced by Moore [78] and Penrose [94] in their pioneering work, though our exposition and definition follows that of [30].

The following lemma is a standard result in linear algebra required in defining the pseudoinverse and projections.

**Lemma 1.** *For any matrix $W$ and symmetric positive definite matrix $G$,*

$$\mathbf{Null}\left(W\right) = \mathbf{Null}\left(W^\top G W\right) \tag{1.14}$$

*and*

$$\mathbf{Range}\left(W^\top\right) = \mathbf{Range}\left(W^\top G W\right). \tag{1.15}$$

*Proof.* In order to establish (1.14), it suffices to show the inclusion $\mathbf{Null}\left(W\right) \supseteq \mathbf{Null}\left(W^\top G W\right)$ since the reverse inclusion trivially holds. Letting $s \in \mathbf{Null}\left(W^\top G W\right)$, we see that $\|G^{1/2} W s\|^2 = 0$, which implies $G^{1/2} W s = 0$. Therefore, $s \in \mathbf{Null}\left(W\right)$. Finally, (1.15) follows from (1.14) by taking orthogonal complements. Indeed, $\mathbf{Range}\left(W^\top\right)$ is the orthogonal complement of $\mathbf{Null}\left(W\right)$ and $\mathbf{Range}\left(W^\top G W\right)$ is the orthogonal complement of $\mathbf{Null}\left(W^\top G W\right)$. □

The pseudoinverse is a matrix that shares many properties with the inverse matrix. Given a real matrix $M \in \mathbb{R}^{m \times n}$, when the nullspace of $M$ contains a nonzero vector then $M$, seen as a linear transformation

$$M : \mathbb{R}^n \mapsto \mathbf{Range}\left(M\right),$$

is not injective and thus $M$ has no inverse. But we can construct a *pseudoinverse* of $M$. The pseudoinverse is constructed by considering a restriction of $M$ that is injective and invertible, and then extending this restriction. Specifically, consider the restriction

$$M|_{\mathbf{Range}(M^\top)} : \mathbf{Range}\left(M^\top\right) \mapsto \mathbf{Range}\left(M\right).$$

Note that this restriction is defined on the orthogonal complement of the nullspace of $M$, and thus removes the "troublesome" subspace that prevents $M$ from being invertible. Indeed, this restriction is invertible since $\mathbf{Range}\left(M M^\top\right) = \mathbf{Range}\left(M\right)$ and $\mathbf{Null}\left(M M^\top\right) = \mathbf{Null}\left(M^\top\right)$ by Lemma 1, thus the restriction is surjective and injective. The following extension of the inverse of this restriction is what we call the pseudoinverse, see Definition 2.

**Definition 2.** *Let $M \in \mathbb{R}^{m \times n}$ be any real matrix. $M^\dagger \in \mathbb{R}^{n \times m}$ is said to be the pseudoinverse if*

*i)* $M^\dagger M x = x$ *for all* $x \in \mathbf{Range}\left(M^\top\right)$.

*ii)* $M^\dagger x = 0$ *for all* $x \in \mathbf{Null}\left(M^\top\right)$.

Item $i$) defines $M^\dagger$ on

$$\mathbf{Range}\left(MM^\top\right) \overset{(1.15)}{=} \mathbf{Range}\left(M\right),$$

and item $ii$) defines $M^\dagger$ on $\mathbf{Null}\left(M^\top\right)$. Thus the two items together define $M^\dagger$ uniquely over $\mathbf{Range}\left(M\right) \oplus \mathbf{Null}\left(M^\top\right) = \mathbb{R}^m$.

For the original, and equivalent, definition of pseudoinverse see [78] and Penrose [94]. Alternatively, for a definition of pseudoinverse based on the SVD decomposition see Section 5.2.2 in [44].

We now collect the properties of the pseudoinverse that we use through the thesis in a sequence of lemmas.

**Lemma 3.** $MM^\dagger M = M$

*Proof.* Let $z \in \mathbb{R}^n$ and consider the decomposition $z = y + x$ where $y \in \mathbf{Range}\left(M^\top\right)$ and $x \in \mathbf{Null}\left(M\right)$. By item i) of Definition 2 we have

$$MM^\dagger Mz = My = M(y + x) = Mz.$$

□

**Lemma 4.** *If $M$ is symmetric then $M^\dagger$ is symmetric.*

*Proof.* Let $z_1, z_2 \in \mathbb{R}^n$ and consider the decompositions $z_1 = My_1 + x_1$ and $z_2 = My_2 + x_2$ where $y_1, y_2 \in \mathbf{Range}\left(M\right)$ and $x_1, x_2 \in \mathbf{Null}\left(M\right)$, which by the symmetry of $M$ always exist. It follows that

$$\begin{aligned} z_1^\top (M^\dagger)^\top z_2 &= (M^\dagger z_1)^\top z_2 \\ &\overset{\text{Definition 2 item ii)}}{=} (M^\dagger M y_1)^\top z_2 \\ &\overset{\text{Definition 2 item i)}}{=} y_1^\top z_2 \\ &= y_1^\top M y_2. \end{aligned}$$

and

$$\begin{aligned} z_1^\top M^\dagger z_2 &\overset{\text{Definition 2 item ii)}}{=} z_1^\top M^\dagger M y_2 \\ &\overset{\text{Definition 2 item i)}}{=} z_1^\top y_2 \\ &= y_1^\top M y_2, \end{aligned}$$

thus $(M^\dagger)^\top = M^\dagger$. □

**Lemma 5.** *If $M$ is symmetric positive semidefinite then $M^\dagger$ is symmetric positive semidefinite.*

*Proof.* For any $z \in \mathbb{R}^n$, consider the decomposition $z = My + x$ where $y \in \mathbf{Range}\left(M\right)$ and $x \in \mathbf{Null}\left(M\right)$, which by the symmetry of $M$ always exists. It follows that

$$z^\top M^\dagger z \overset{\text{Definition 2 item ii)}}{=} y^\top M M^\dagger M y \overset{\text{Definition 2 item i)}}{=} y^\top M y \geq 0,$$

which shows that $M^\dagger$ is positive semidefinite. The symmetry of $M^\dagger$ follows from Lemma 4. □

**Lemma 6.** *The matrix $M^\dagger M$ projects orthogonally onto* $\mathbf{Range}\left(M^\top\right)$ *and along* $\mathbf{Null}\left(M\right)$.

*Proof.* Consider the orthogonal decomposition $z = y + x$ where $y \in \mathbf{Range}\left(M^\top\right)$ and $x \in \mathbf{Null}\left(M\right)$. Then

$$M^\dagger M z = M^\dagger M y = y,$$

where we used item ii) then item i) of Definition 2. The result now follows by observing that $\mathbf{Range}\left(M^\top\right)$ and $\mathbf{Null}\left(M\right)$ are orthogonal complements. □

**Lemma 7.** *Consider the consistent linear system $Mx = d$ where $M, x$ and $d$ are of conforming dimensions. It follows that*

$$M^\dagger d = \arg\min_x \|x\|_2^2 \quad \textit{subject to} \quad Mx = d. \tag{1.16}$$

*Proof.* As $d \in \mathbf{Range}\left(M\right)$ we have from Lemma 3 that $MM^\dagger d = d$. Using the change of variables $z = x - M^\dagger d$ in (1.16) gives

$$z^* \overset{\text{def}}{=} \arg\min_z \|z + M^\dagger d\|_2^2, \quad \text{subject to} \quad z \in \mathbf{Null}\left(M\right). \tag{1.17}$$

By Lemma 6 we have that $M^\dagger d \in \mathbf{Range}\left(M^\top\right) = \mathbf{Null}\left(M\right)^\perp$. Consequently

$$\|z + M^\dagger d\|_2^2 = \|z\|_2^2 + \|M^\dagger d\|_2^2 \geq \|z\|_2^2,$$

thus the minimum $z^*$ of (1.17) is achieved at $z^* = 0$, from which it follows that the minimum of (1.16) is achieved at $x = z^* + M^\dagger d = M^\dagger d$. □

### 1.3.2 Projection matrices

The following proposition is a variant of a standard result of linear algebra (which is often presented in the $B = I$ case). While the results are folklore and easy to establish, in the proof of our main theorems we need certain details which are hard to find in textbooks on linear algebra, and hence hard to refer to. For the benefit of the reader, we include the detailed statement and proof.

**Proposition 8** (Decomposition and Projection)**.** *Let $M \in \mathbb{R}^{m\times n}$ by a real matrix and $B \in \mathbb{R}^{n\times n}$ a symmetric positive definite matrix. Each $x \in \mathbb{R}^n$ can be decomposed in a unique way as $x = s(x) + t(x)$, where $s(x) \in \mathbf{Range}\left(B^{-1}M^\top\right)$ and $t(x) \in \mathbf{Null}\left(M\right)$. Moreover, the decomposition can be computed explicitly as*

$$s(x) = \arg\min_s \left\{\|x - s\|_B \ : \ s \in \mathbf{Range}\left(B^{-1}M^\top\right)\right\} = B^{-1}Z_M x \tag{1.18}$$

*and*

$$t(x) = \arg\min_t \left\{\|x - t\|_B \ : \ t \in \mathbf{Null}\left(M\right)\right\} = (I - B^{-1}Z_M)x, \tag{1.19}$$

*where*

$$Z_M \overset{def}{=} M^\top (MB^{-1}M^\top)^\dagger M. \tag{1.20}$$

*Hence, the matrix $B^{-1}Z_M$ is a projection in the $B$-norm onto* $\mathbf{Range}\left(B^{-1}M^\top\right)$*, and $I-B^{-1}Z_M$ is a projection in the $B$-norm onto* $\mathbf{Null}\,(M)$*. Moreover, for all $x \in \mathbb{R}^n$ we have $\|x\|_B^2 = \|s(x)\|_B^2 + \|t(x)\|_B^2$, with*

$$\|t(x)\|_B^2 = \|(I - B^{-1}Z_M)x\|_B^2 = x^\top(B - Z_M)x \tag{1.21}$$

*and*

$$\|s(x)\|_B^2 = \|B^{-1}Z_M x\|_B^2 = x^\top Z_M x. \tag{1.22}$$

*Finally,*

$$\mathbf{Rank}\,(M) = \mathbf{Tr}\left(B^{-1}Z_M\right). \tag{1.23}$$

*Proof.* Fix arbitrary $x \in \mathbb{R}^n$. We first establish existence of the decomposition. By Lemma 1 applied to $W = M^\top$ and $G = B^{-1}$ we know that there exists $u$ such that $Mx = MB^{-1}M^\top u$. Now let $s = B^{-1}M^\top u$ and $t = x - s$. Clearly, $s \in \mathbf{Range}\left(B^{-1}M^\top\right)$ and $t \in \mathbf{Null}\,(M)$. For uniqueness, consider two decompositions: $x = s_1 + t_1$ and $x = s_2 + t_2$. Let $u_1, u_2$ be vectors such that $s_i = B^{-1}M^\top u_i$, $i = 1, 2$. Then $MB^{-1}M^\top(u_1 - u_2) = 0$. Invoking Lemma 1 again, we see that $u_1 - u_2 \in \mathbf{Null}\left(M^\top\right)$, whence $s_1 = B^{-1}M^\top u_1 = B^{-1}M^\top u_2 = s_2$. Therefore, $t_1 = x - s_1 = x - s_2 = t_2$, establishing uniqueness.

Note that $s = B^{-1}M^\top y$, where $y$ is any solution of the optimization problem

$$\min_y \tfrac{1}{2}\|x - B^{-1}M^\top y\|_B^2.$$

The first order necessary and sufficient optimality conditions are $Mx = MB^{-1}M^\top y$. In particular, we may choose $y$ to be the least norm solution of this system, which by Lemma 7 is given $y = (MB^{-1}M^\top)^\dagger Mx$, from which (1.18) follows. The variational formulation (1.19) can be established in a similar way, again via first order optimality conditions (note that the closed form formula (1.19) also directly follows from (1.18) and the fact that $t = x - s$).

Next, since $x = s + t$ and $s^\top Bt = 0$,

$$\|t\|_B^2 = (t+s)^\top Bt = x^\top Bt \overset{(1.19)}{=} x^\top B(I - B^{-1}Z_M)x = x^\top(B - Z_M)x \tag{1.24}$$

and

$$\|s\|_B^2 = \|x\|_B^2 - \|t\|_B^2 \overset{(1.24)}{=} x^\top Z_M x.$$

It only remains to establish (1.23). Since $B^{-1}Z_M$ projects onto $\mathbf{Range}\left(B^{-1}M^\top\right)$ and since the trace of a projection is equal to the dimension of the space they project onto, we have $\mathbf{Tr}\left(B^{-1}Z_M\right) = \dim(\mathbf{Range}\left(B^{-1}M^\top\right)) = \dim(\mathbf{Range}\left(M^\top\right)) = \mathbf{Rank}\,(M)$. □

All the iterative methods presented in the thesis are based on projections. In particular, the projections that govern most of the iterative methods here are constructed from the matrix

$$Z \overset{\text{def}}{=} Z_{S^\top A} \overset{(1.20)}{=} A^\top S(S^\top AB^{-1}A^\top S)^\dagger S^\top A. \tag{1.25}$$

We now collect in the following lemma several properties pertaining to $Z$ that are repeatedly used throughout the thesis.

**Lemma 9.** *The matrix $Z$ defined in* (1.25) *is symmetric positive semidefinite. Furthermore $B^{-1}Z$ is a projection with respect to the $B$–norm such that*

$$\mathbf{Range}\left(B^{-1}Zx\right) = \mathbf{Range}\left(B^{-1}A^\top S\right) \quad \textit{and} \quad \mathbf{Range}\left(I - B^{-1}Z\right) = \mathbf{Null}\left(S^\top A\right), \tag{1.26}$$

*and $B^{-1/2}ZB^{-1/2}$ is a projection with respect to the standard Euclidean geometry, consequently*

$$\|(I - B^{-1/2}ZB^{-1/2})x\|_2^2 = \left\langle (I - B^{-1/2}ZB^{-1/2})x, x \right\rangle, \quad \forall x \in \mathbb{R}^n, \tag{1.27}$$

*and*

$$\mathbf{Tr}\left(B^{-1/2}ZB^{-1/2}\right) = \mathbf{Rank}\left(A^\top S\right). \tag{1.28}$$

*Proof.* First note that $(B^{-1/2}A^\top S)^\top B^{-1/2}A^\top S = S^\top AB^{-1}A^\top S$ is symmetric positive semidefinite, and consequently by Lemma 5 the matrix $(S^\top AB^{-1}A^\top S)^\dagger$ is also symmetric positive semidefinite. Thus there exists $G$ such that $GG^\top = (S^\top AB^{-1}A^\top S)^\dagger$ and consequently $(A^\top SG)(A^\top SG)^\top = Z$ which proves that $Z$ is symmetric positive semidefinite.

By Lemma 8 (with $M = S^\top A$) we have that $B^{-1}Z$ is a projection, and (1.26) follows by (1.18) and (1.19). Again by Lemma 8 (with $M = S^\top AB^{-1/2}$ and $B = I$) we have that $B^{-1/2}ZB^{-1/2}$ projects orthogonally onto $\mathbf{Range}\left(B^{-1/2}A^\top S\right)$, whence (1.27) and (1.28) follow from (1.21) and (1.23), respectively. □

### 1.3.3 Random variables and the random matrix $S$

As explained in Section 1, the methods proposed in this thesis depend on a random matrix $S \in \mathbb{R}^{m\times q}$. Consequently the iterates of these methods are random variables. Throughout the thesis we make little to no assumption on the distribution of $S$, and unless explicitly stated, the reader should assume that $S$ is a random matrix in the most general sense. Here we formalize what is a random variable and what is a random matrix in the most general sense, that is, in the probability measure sense. For the reader that is not familiar with probability spaces and measure theory, we suggest the book [130] as quick an enjoyable introduction.

To formalize the notion of a random variable we need the definition of a *probability space.* A probability space $(\Omega, \mathcal{F}, P)$ is defined by three objects:

1. The $\Omega$ is a given set known as the *sample space*. It contains all the possible *outcomes* (elements).

2. The $\mathcal{F}$ is a set of subsets of $\Omega$. Specifically, it is a $\sigma$–algebra over $\Omega$. It contains all the possible *events* (subsets) we would like to consider.

3. The $P$ is a function that maps from $\mathcal{F}$ to [0, 1]. That is, given an event $E \in \mathcal{F}$ it return the probability $P(E) \in [0, 1]$ of $E$ occurring. Moreover, $P$ is a probability measure and thus $P(\Omega) = 1, P(\emptyset) = 0$ and $P$ satisfies the countable additivity property [130].

Often one is not interested in the probability space itself, but in functions over this probability space called random variables. Consider the map $r : \Omega \to \mathbb{R}$. We say $r$ is a *random variable* when

$$\{\omega : r(\omega) \leq a\} \subset \mathcal{F} \qquad \forall a \in \mathbb{R}.$$

An equivalent statement is as follows: The function $r$ is a random variable if the inverse image of $r$ over the interval $(-\infty, a]$ is contained in $\mathcal{F}$ for every $a \in \mathbb{R}$.

A random matrix is simply a matrix valued map where each element is a random variable. That is, consider a map $S : \Omega \to \mathbb{R}^{m\times q}$ where $m, q \in \mathbb{N}$. We say that $S$ is a *random matrix* when each element of $S$ is a random variable. For brevity, and as is customary, we use $S \in \mathbb{R}^{m\times q}$ as a shorthand for $S(\omega) \in \mathbb{R}^{m\times q}$ for all $\omega \in \Omega$. A *random vector* is a random matrix that has only one column or one row. We now provide an example of a random matrix. Note that this example, and in fact all the examples in this thesis, are simple enough as to not require this formal probability measure context.

**Example:** Let $e_i \in \mathbb{R}^m$ be the $i$th coordinate vector. Let $S = e_i$ with probability $1/m$ for all $i \in \{1, \dots, m\}$. In other words, $P(S = e_i) = 1/m$ for all $i \in \{1, \dots, m\}$. We will now show that $S$ is a random matrix by constructing a suitable probability space. Let $\Omega = \{1, \dots, m\}$, let $\mathcal{F} = 2^\Omega$ be the power set of $\Omega$ and let $P : \mathcal{F} \to [0, 1]$ be any probability measure that satisfies $P(\{i\}) = 1/m$ for $i = 1, \dots, m$. Then the map defined by $S(i) = e_i$ is our desired random matrix.

### 1.3.4 Convergence of a random sequence

As the methods presented in the thesis depend on a random matrix $S$ the iterates of our methods are themselves random variables. To guarantee that the iterates converge to the desired solution we need to establish the convergence of a sequence of random variables. Throughout the thesis we use two notions of the convergence of random variables; the convergence of the norm of the expectation and the convergence of the expected norm, which we describe here.

Consider a sequence of random matrices $(Y^k)_k$ on $\mathbb{R}^{m\times q}$. Let $\langle \cdot, \cdot \rangle$ and $\|Y\|^2 = \langle Y, Y \rangle$ be an inner product and induced norm, respectively. We say that the norm of the expectation of $(Y^k)_k$ converges to zero with rate $\rho \in [0,\ 1)$ if

$$\|\mathbf{E}\left[Y^k\right]\| \leq \rho^k \|Y^0\|. \tag{1.29}$$

Furthermore, from (1.29) we see that $\mathbf{E}\left[Y^k\right] \to 0$, and thus the sequence converges in expectation to zero.

We say that the expected norm of $(Y^k)_k$ converges to zero with rate $\rho$ if

$$\mathbf{E}\left[\|Y^k\|^2\right] \leq \rho^k \|Y^0\|^2. \tag{1.30}$$

Note that the order of the expectation operator and the norm are now exchanged in relation to (1.29). The convergence of the expected norm implies the convergence of the norm of expectation, as we prove in Lemma 10. In this lemma we also show that the convergence (1.30) implies *convergence in probability*. We say that $Y^k$ converges in probability to zero if for every $\epsilon > 0$ we have that

$$\lim_{k\to\infty} \mathbb{P}\left(\|Y^k\|^2 \geq \epsilon \|Y^0\|^2\right) = 0. \tag{1.31}$$

**Lemma 10.** *The convergence of the expected norm* (1.30) *implies the convergence of the norm of the expectation* (1.29) *as can be seen through the equality*

$$\mathbf{E}\left[\left\|Y^k\right\|^2\right] = \left\|\mathbf{E}\left[Y^k\right]\right\|^2 + \mathbf{E}\left[\left\|Y^k - \mathbf{E}\left[Y^k\right]\right\|^2\right]. \tag{1.32}$$

*Furthermore, the convergence of the expected norm* (1.30) *implies convergence in probability.*

*Proof.* Let $(Y^k)_k$ be a sequence of random vectors that converges to zero according to (1.30). The convergence of the norm of expectation follows from the equality

$$\begin{aligned}\left\|\mathbf{E}\left[Y^k\right]\right\|^2 &= \left\|\mathbf{E}\left[Y^k\right]\right\|^2 + \mathbf{E}\left[\|Y^k\|^2\right] - \mathbf{E}\left[\|Y^k\|^2\right] \\ &= \mathbf{E}\left[\|Y^k\|^2\right] - \left(\mathbf{E}\left[\|Y^k\|^2\right] - 2\mathbf{E}\left[\langle Y^k, \mathbf{E}\left[Y^k\right]\rangle\right] + \left\|\mathbf{E}\left[Y^k\right]\right\|^2\right) \\ &= \mathbf{E}\left[\left\|Y^k\right\|^2\right] - \mathbf{E}\left[\left\|Y^k - \mathbf{E}\left[Y^k\right]\right\|^2\right]. \end{aligned} \tag{1.33}$$

Indeed, since $\mathbf{E}\left[\left\|Y^k - \mathbf{E}\left[Y^k\right]\right\|^2\right] \geq 0$ we have that

$$\left\|\mathbf{E}\left[Y^k\right]\right\|^2 \leq \mathbf{E}\left[\left\|Y^k\right\|^2\right] \overset{(1.30)}{\leq} \rho^k \|Y^0\|^2.$$

Consequently the norm of the expected error converges with rate $\sqrt{\rho}$. Finally, let $\epsilon > 0$. Using Markov's inequality we have

$$\mathbb{P}(\|Y^k\|^2 \geq \epsilon \|Y^0\|^2) \leq \frac{\mathbf{E}\left[\|Y^k\|^2\right]}{\epsilon \|Y^0\|^2} \overset{(1.30)}{\leq} \frac{\rho^k}{\epsilon}. \tag{1.34}$$

Thus $\mathbb{P}(\|Y^k\|^2 \geq \epsilon \|Y^0\|^2) \to 0$ as $k \to \infty$. □

In every chapter that follows, we will present several convergence results of random sequences. In particular, in Chapter 2 we prove the convergence of the expected norm (1.30) and the convergence of the norm of the expectation (1.29) of a sequence of random vectors $Y^k = x^k - x^*$. In Chapter 4 we prove analogous convergence results of a sequence of random matrices $Y^k = X_k - A^{-1}$. Thus Lemma 10 is important as it sheds light on how these two types of convergence (1.30) and (1.29) are related.

Convergence according to (1.30) is also commonly referred to as *linear convergence*. This is because, as explained in the next Section 1.3.5, the number of iterations required to reach a certain precision grows linearly and proportionally to $1/(1-\rho)$. Another common synonym to linear convergence, that we sometimes use here, is to say that $Y^k$ converges *exponentially fast* to zero. This is because the expected norm of $Y^k$ decreases according to the exponential function $\rho^k$.

Note that if $Y^k$ is a random vector defined on $\mathbb{R}^m$ with the standard Euclidean inner product then $\mathbf{E}\left[\left\|Y^k - \mathbf{E}\left[Y^k\right]\right\|_2^2\right] = \sum_{i=1}^m \mathbf{E}\left[(Y_i^k - \mathbf{E}\left[Y_i^k\right])^2\right] = \sum_{i=1}^m \mathbf{Var}(Y_i^k)$, where $z_i^k$ denotes the $i$th element of $Y^k$. Thus, in this case, the equality (1.32) also shows that if the norm of expectation converges and the variance of $Y_i^k$ converges to zero for $i = 1, \ldots, m$, then the expected norm converges.

### 1.3.5 Iteration complexity

Both types of convergence (1.29) and (1.30) can be recast as iteration complexity bounds using the following lemma. With this lemma we can stipulate an lower bound on how many iterations are required to bring the sequence within an $\epsilon > 0$ relative distance of its limit point.

**Lemma 11.** *Consider the sequence $(\alpha_k)_k \in \mathbb{R}_+$ of positive scalars that converges to zero according to*

$$\alpha_k \leq \rho^k \, \alpha_0, \tag{1.35}$$

*where $\rho \in [0, 1)$. For a given $1 > \epsilon > 0$ we have that*

$$k \geq \frac{1}{1-\rho} \log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad \alpha_k \leq \epsilon \, \alpha_0. \tag{1.36}$$

*Proof.* First note that if $\rho = 0$ the result follows trivially. Assuming $\rho \in (0,\, 1)$, rearranging (1.35) and applying the logarithm to both sides gives

$$\log\left(\frac{\alpha_0}{\alpha_k}\right) \geq k \log\left(\frac{1}{\rho}\right). \tag{1.37}$$

Now using that

$$\frac{1}{1-\rho} \log\left(\frac{1}{\rho}\right) \geq 1, \tag{1.38}$$

for all $\rho \in (0, 1)$ and assuming that

$$k \geq \frac{1}{1-\rho} \log\left(\frac{1}{\epsilon}\right), \tag{1.39}$$

we have that

$$\begin{aligned} \log\left(\frac{\alpha_0}{\alpha_k}\right) &\overset{(1.37)}{\geq} k \log\left(\frac{1}{\rho}\right) \\ &\overset{(1.39)}{\geq} \frac{1}{1-\rho} \log\left(\frac{1}{\rho}\right) \log\left(\frac{1}{\epsilon}\right) \\ &\overset{(1.38)}{\geq} \log\left(\frac{1}{\epsilon}\right) \end{aligned}$$

Applying exponentials to the above inequality gives (1.36). □

As an example of the use this lemma, consider the sequence of random vectors $(Y^k)_k$ for which the expected norm converges to zero according to (1.30). Then applying Lemma 11 with $\alpha_k = \mathbf{E}\left[\|Y^k\|^2\right]$ for a given $1 > \epsilon > 0$ states that

$$k \geq \frac{1}{1-\rho} \log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad \mathbf{E}\left[\|Y^k\|^2\right] \leq \epsilon \, \|Y^0\|^2.$$

To give further insight into the implications of the convergence of the expected norm, for a given $1 > \epsilon > 0$ consider the sequence $\alpha_0 = 1/\epsilon$ and $\alpha^k = \mathbf{P}\left(\|Y^k\|_2^2 \geq \epsilon \|Y^0\|_2^2\right)$ for $k \geq 1$. From (1.34) we know that this sequence converges according to $\alpha^k \leq \rho^k \alpha^0$. We can now use Lemma 11 to determine how many iterates are required so that $\|Y^k\|_2^2 \leq \epsilon \|Y^0\|_2^2$ with high probability. Indeed, let $\delta \in (0,\, 1)$ then by Lemma 11 we have that

$$k \geq \frac{1}{1-\rho} \log\left(\frac{1}{\epsilon\delta}\right) \quad \Rightarrow \quad \mathbf{P}\left(\|Y^k\|_2^2 \geq \epsilon \|Y^0\|_2^2\right) \leq \delta.$$

This shows that convergence of the expected norm is almost as good as linear convergence without the expectation, that is, one can guarantee the iterates are relatively close with a high probability at the cost of only an additional logarithmic growth in the number of iterates.

CHAPTER 2

# Randomized Iterative Methods for Linear Systems

*Guid gear comes in sma' bulk.*
Good things come in small sizes (like sketched linear systems!)

Scottish proverb.

## 2.1 Introduction

The need to solve linear systems of equations is ubiquitous in essentially all quantitative areas of human endeavour, including industry and science. Linear systems are a central problem in numerical linear algebra, and play an important role in computer science, mathematical computing, optimization, signal processing, engineering, numerical analysis, computer vision, machine learning, and many other fields. For instance, in the field of large scale optimization, there is a growing interest in inexact and approximate Newton-type methods for [28, 34, 4, 138, 129, 45], which can benefit from fast subroutines for calculating approximate solutions of linear systems. In machine learning, applications arise for the problem of finding optimal configurations in Gaussian Markov Random Fields [111], in graph-based semi-supervised learning and other graph-Laplacian problems [5], least-squares SVMs, Gaussian processes and more.

In a large scale setting, direct methods can suffer from two shortcomings. First, direct methods often require direct access to individual elements of the system matrix and thus need the system matrix to be stored on RAM. But the dimensions and density of the problem at hand maybe such that the system matrix does not fit on RAM. Second, the complexity of direct methods is of order $O(n^3)$ which can be prohibitively slow when $n$ is large.

While classical iterative methods are deterministic, recent breakthroughs suggest that randomization can play a powerful role in the design and analysis of efficient algorithms [125, 68, 82, 32, 139, 67, 76, 106] which are in many situations competitive or better than existing deterministic methods.

In this chapter we develop the sketch-and-project family of randomized methods for solving linear systems that are well suited to quickly calculating approximate solutions.

### 2.1.1 Background and related work

The literature on solving linear systems via iterative methods is vast and has a long history [63, 112]. For instance, the Kaczmarz method, in which one cycles through the rows of the system and each iteration is formed by projecting the current point to the hyperplane formed by the active row, dates back to the 30's [62]. The Kaczmarz method is just one example of an array of row-action methods for linear systems (and also, more generally, feasibility and optimization problems) which were studied in the second half of the 20th century [16].

Research into the Kaczmarz method was reignited in 2009 by Strohmer and Vershynin [125], who gave a brief and elegant proof that a randomized variant thereof enjoys an exponential error decay (also know as "linear convergence"). This has triggered much research into developing and analyzing randomized linear solvers.

It should be mentioned at this point that the randomized Kaczmarz (RK) method arises as a special case (when one considers quadratic objective functions) of the stochastic gradient descent (SGD) method for *convex optimization* which can be traced back to the seminal work of Robbins and Monro's on stochastic approximation [108]. Subsequently, intensive research went into studying various extensions of the SGD method. However, to the best of our knowledge, no complexity results with exponential error decay were established prior to the aforementioned work of Strohmer and Vershynin [125]. This is the reason behind our choice of [125] as the starting point of our discussion.

Motivated by the results of Strohmer and Vershynin [125], Leventhal and Lewis [68] utilize similar techniques to establish the first bounds for *randomized coordinate descent* methods for solving systems with positive definite matrices, and systems arising from least squares problems [68]. These bounds are similar to those for the RK method. This development was later picked up by the optimization and machine learning communities, and much progress has been made in generalizing these early results in countless ways to various structured convex optimization problems. For a brief up to date account of the development in this area, we refer the reader to [37, 98] and the references therein.

The RK method and its analysis have been further extended to the least-squares problem [82, 139] and the block setting [84, 85]. In [76] the authors extend the randomized coordinate descent and the RK methods to the problem of solving underdetermined systems. The authors of [76, 102] analyze side-by-side the randomized coordinate descent and RK method, for least-squares, using a convenient notation in order to point out their similarities. Our work takes the next step, by analyzing these, and many other methods, through a genuinely general analysis. Also in the spirit of unifying the analysis of different methods, in [91] the authors provide a unified analysis of iterative Schwarz methods and Kaczmarz methods.

The use of random Gaussian directions as search directions in zero-order (derivative-free) minimization algorithm was recently suggested [87]. Our Gaussian positive definite and Gaussian least-squares in Sections 2.7.3 and 2.7.2, respectively, are special cases of these zero order methods applied to linear systems. More recently, Gaussian directions have been combined with exact and inexact line-search into a single *random pursuit* framework [124], and further utilized within a randomized variable metric method [121, 122].

## 2.2 Contributions and Overview

Given a real matrix $A \in \mathbb{R}^{m\times n}$ and a real vector $b \in \mathbb{R}^m$, in this chapter we consider the linear system

$$Ax = b. \tag{2.1}$$

We shall assume throughout that the system is *consistent*: there exists $x^*$ for which $Ax^* = b$.

We now comment on the main contribution of this chapter.

*1. New method.* We develop a novel, fundamental, and surprisingly simple *randomized iterative method* for solving (2.1).

*2. Six equivalent formulations.* Our method allows for several seemingly different but nevertheless equivalent formulations. First, it can be seen as a *sketch-and-project* method, in which the system (2.1) is replaced by its *random sketch*, and then the current iterate is projected onto the solution space of the sketched system. We can also view it as a *constrain-and-approximate* method, where we constrain the next iterate to live in a particular random affine space passing through the current iterate, and then pick the point from this subspace which best approximates the optimal solution. Third, the method can be seen as an iterative solution of a sequence of random (and simpler) linear equations. The method also allows for a simple geometrical interpretation: the new iterate is defined as the unique intersection of two random affine spaces which are orthogonal complements. The fifth viewpoint gives a closed form formula for the *random update* which needs to be applied to the current iterate in order to arrive at the new one. Finally, the method can be seen as a *random fixed point iteration.*

*3. Special cases.* These multiple viewpoints enrich our interpretation of the method, and enable us to draw previously unknown links between several existing algorithms. Our algorithm has two parameters, an $n \times n$ positive definite matrix $B$ defining geometry of the space, and a random matrix $S$. Through combinations of these two parameters, in special cases our method recovers several well known algorithms. For instance, we recover the randomized Kaczmarz method of Strohmer and Vershyinin [125], randomized coordinate descent method of Leventhal and Lewis [68], random pursuit [87, 122, 121, 123] (with exact line search), and the stochastic Newton method recently proposed by Qu et al [101]. However, our method is more general, and leads to i) various generalizations and improvements of the aforementioned methods (e.g., block setup, importance sampling), and ii) completely new methods. Randomness enters our framework in a very general form, which allows us to obtain a *Gaussian Kaczmarz method*, *Gaussian descent*, and more.

*4. Complexity: general results.* When $A$ has full column rank, our framework allows us to determine the complexity of these methods using a single analysis. Our main results are summarized in Table 2.1, where $\{x^k\}$ are the iterates of our method, $Z$ is a random matrix dependent on the data matrix $A$, parameter matrix $B \in \mathbb{R}^{n\times n}$ and random parameter matrix $S \in \mathbb{R}^{m\times q}$, defined as

$$Z \overset{\text{def}}{=} A^\top S(S^\top AB^{-1}A^\top S)^\dagger S^\top A, \tag{2.2}$$

where $\dagger$ denotes the (Moore-Penrose) pseudoinverse. For the definition of pseudoinverse, see Section 1.3.1. Moreover, $\|x\|_B \overset{\text{def}}{=} \sqrt{\langle x, x\rangle_B}$, where $\langle x, y\rangle_B \overset{\text{def}}{=} x^\top By$, for all $x, y \in \mathbb{R}^n$.

As we shall see later, we will often consider setting $B = I$, $B = A$ (if $A$ is positive definite) or $B = A^\top A$ (if $A$ is of full column rank). In particular, we first show that the convergence rate $\rho$

| | |
|---|---|
| $\mathbf{E}\left[x^{k+1}-x^*\right] = (I - B^{-1}\mathbf{E}\left[Z\right])\,\mathbf{E}\left[x^k - x^*\right]$ | Theorem 15 |
| $\|\mathbf{E}\left[x^{k+1}-x^*\right]\|_B \leq \rho \;\cdot\; \|\mathbf{E}\left[x^k - x^*\right]\|_B$ | Theorem 15 |
| $\mathbf{E}\left[\|x^{k+1}-x^*\|_B^2\right] \leq \rho \;\cdot\; \mathbf{E}\left[\|x^k - x^*\|_B^2\right]$ | Theorem 16 |

**Table 2.1:** *Our main complexity results. The convergence rate is:* $\rho = 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})$.

is always bounded between zero and one. We also show that as soon as $\mathbf{E}\left[Z\right]$ is invertible (which can only happen if $A$ has full column rank, which then implies that $x^*$ is unique), we have $\rho < 1$, and the method converges. Besides establishing a bound involving the *expected norm of the error* (see the last line of Table 2.1), we also obtain bounds involving the *norm of the expected error* (second line of Table 2.1). Studying the expected sequence of iterates directly is very fruitful, as it allows us to establish an *exact characterization* of the evolution of the expected iterates (see the first line of Table 2.1) through a *linear fixed point iteration*.

Both of these theorems on the convergence of the method can be recast as iteration complexity bounds by using Lemma 11. For instance from Theorem 15 in Table 2.1 we observe that for a given $\epsilon > 0$ we have that

$$k \geq \frac{1}{1-\rho}\log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad \|\mathbf{E}\left[x^k - x^*\right]\|_B \leq \epsilon\|x^0 - x^*\|_B. \tag{2.3}$$

*5. Complexity: special cases.* Besides these generic results, which hold without any major restriction on the sampling matrix $S$ (in particular, it can be either discrete or continuous), we give a specialized result applicable to discrete sampling matrices $S$ (see Theorem 19). In the special cases for which rates are known, our analysis recovers the existing rates.

*6. Extensions.* Our approach opens up many avenues for further development and research. For instance, it is possible to extend the results to the case when $A$ is not necessarily of full column rank, which we do in Chapter 3. Furthermore, as our results hold for a wide range of distributions, new and efficient variants of the general method can be designed for problems of specific structure by fine-tuning the stochasticity to the structure. Similar ideas can be applied to design randomized iterative algorithms for finding the inverse of a very large matrix, which is the focus of Chapter 4.

## 2.3 One Algorithm in Six Disguises

Our method has *two parameters*: i) an $n \times n$ positive definite matrix $B$ which is used to define the $B$-inner product and the induced $B$-norm by

$$\langle x, y \rangle_B \overset{\text{def}}{=} \langle Bx, y \rangle, \qquad \|x\|_B \overset{\text{def}}{=} \sqrt{\langle x, x \rangle_B}, \tag{2.4}$$

where $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product, and ii) a random matrix $S \in \mathbb{R}^{m \times q}$, to be drawn in an i.i.d. fashion at each iteration. We stress that we do not restrict the number of columns of $S$; indeed, we even allow $q$ to vary (and hence, $q$ is a random variable).

### 2.3.1 Six viewpoints

Starting from $x^k \in \mathbb{R}^n$, our method draws a random matrix $S$ and uses it to generate a new point $x^{k+1} \in \mathbb{R}^n$. As proven at the end of this section, our iterative method can be formulated in *six seemingly different but equivalent ways:*

**1. Sketching Viewpoint: Sketch-and-Project.** $x^{k+1}$ is the nearest point to $x^k$ which solves a *sketched* version of the original linear system:

$$\boxed{x^{k+1} \quad = \quad \arg\min_{x \in \mathbb{R}^n} \|x - x^k\|_B^2 \quad \text{subject to} \quad S^\top A x = S^\top b} \tag{2.5}$$

This viewpoint arises very naturally. Indeed, since the original system (2.1) is assumed to be complicated, we replace it by a simpler system—a *random sketch* of the original system (2.1)—whose solution set $\{x \mid S^\top A x = S^\top b\}$ contains all solutions of the original system. However, this system will typically have many solutions, so in order to define a method, we need a way to select one of them. The idea is to try to preserve as much of the information learned so far as possible, as condensed in the current point $x^k$. Hence, we pick the solution which is closest to $x^k$.

**2. Optimization Viewpoint: Constrain-and-Approximate.** $x^{k+1}$ is the best approximation of $x^*$ in a random space passing through $x^k$:

$$\boxed{x^{k+1} \; = \; \arg\min_{x \in \mathbb{R}^n} \|x \; - x^*\|_B^2 \quad \text{subject to} \quad x = x^k + B^{-1} A^\top S y, \quad y \text{ is free}} \tag{2.6}$$

The above step has the following interpretation. We choose a random affine space containing $x^k$, and constrain our method to choose the next iterate from this space. We then do as well as we can on this space; that is, we pick $x^{k+1}$ as the point which best approximates $x^*$. Note that $x^{k+1}$ does not depend on which solution $x^*$ is used in (2.6) (this can be best seen by considering the geometric viewpoint, discussed next).

**3. Geometric viewpoint: Random Intersect.** $x^{k+1}$ is the (unique) intersection of two affine spaces:

$$\boxed{\{x^{k+1}\} \quad = \quad \left(x^* + \mathbf{Null}\left(S^\top A\right)\right) \quad \bigcap \quad \left(x^k + \mathbf{Range}\left(B^{-1} A^\top S\right)\right)} \tag{2.7}$$

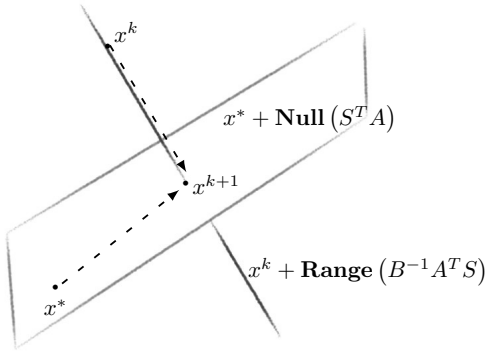



**Figure 2.1:** *The geometry of our algorithm. The next iterate, $x^{k+1}$, arises as the intersection of two random affine spaces: $x^k + \mathbf{Range}\left(B^{-1}A^\top S\right)$ and $x^* + \mathbf{Null}\left(S^\top A\right)$ (see (2.7)). The spaces are orthogonal complements of each other with respect to the $B$-inner product, and hence $x^{k+1}$ can equivalently be written as the projection, in the $B$-norm, of $x^k$ onto $x^* + \mathbf{Null}\left(S^\top A\right)$ (see (2.5)), or the projection of $x^*$ onto $x^k + \mathbf{Range}\left(B^{-1}A^\top S\right)$ (see (2.6)). The intersection $x^{k+1}$ can also be expressed as the solution of a system of linear equations (see (2.8)). Finally, the new error $x^{k+1} - x^*$ is the projection, with respect to the $B$-inner product, of the current error $x^k - x^*$ onto $\mathbf{Null}\left(S^\top A\right)$. This gives rise to a random fixed point formulation (see (2.11)).*

First, note that the first affine space above does not depend on the choice of $x^*$ from the set of optimal solutions of (2.1). A basic result of linear algebra says that the nullspace of an arbitrary matrix is the orthogonal complement of the range space of its transpose. Hence, whenever we have $h \in \mathbf{Null}\left(S^\top A\right)$ and $y \in \mathbb{R}^q$, where $q$ is the number of rows of $S$, then $\left\langle h, B^{-1}A^\top S y\right\rangle_B = \left\langle h, A^\top S y\right\rangle = 0$. It follows that the two spaces in (2.7) are orthogonal complements with respect to the $B$-inner product and as such, they intersect at a unique point (see Figure 2.1).

**4. Algebraic viewpoint: Random Linear Solve.** Note that $x^{k+1}$ is the (unique) solution (in $x$) of a linear system (with variables $x$ and $y$):

$$\boxed{x^{k+1} \quad = \quad \text{solution of} \quad S^\top A x = S^\top b, \quad x = x^k + B^{-1}A^\top S y} \tag{2.8}$$

This system is clearly equivalent to (2.7), and can alternatively be written as:

$$\begin{pmatrix} S^\top A & 0 \\ B & -A^\top S \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} S^\top b \\ Bx^k \end{pmatrix}. \tag{2.9}$$

Hence, our method reduces the solution of the (complicated) linear system (2.1) into a sequence of (hopefully simpler) random systems of the form (2.9).

**5. Algebraic viewpoint: Random Update.** By plugging the second equation in (2.8) into the first, we eliminate $x$ and obtain the system $(S^\top A B^{-1} A^\top S)y = S^\top(b - Ax^k)$. Note that for all solutions $y$ of this system we must have $x^{k+1} = x^k + B^{-1}A^\top S y$. In particular, we can choose the solution $y = y^k$ of minimal Euclidean norm, which by Lemma 7 is given by $y^k = (S^\top A B^{-1} A^\top S)^\dagger S^\top(b - Ax^k)$. This leads to an expression for $x^{k+1}$ with an explicit form of

the *random update* which must be applied to $x^k$ in order to obtain $x^{k+1}$:

$$\boxed{x^{k+1} = x^k - B^{-1}A^\top S(S^\top AB^{-1}A^\top S)^\dagger S^\top(Ax^k - b)} \tag{2.10}$$

In some sense, this form is the standard: it is customary for iterative techniques to be written in the form $x^{k+1} = x^k + d^k$, which is precisely what (2.10) does.

**6. Analytic viewpoint: Random Fixed Point.** Note that iteration (2.10) can be written as

$$\boxed{x^{k+1} - x^* \quad = \quad (I - B^{-1}Z)(x^k - x^*)} \tag{2.11}$$

where $Z$ is defined in (2.2) and where we used the fact that $Ax^* = b$. Matrix $Z$ plays a central role in our analysis, and can be used to construct explicit projection matrices of the two projections depicted in Figure 2.1.

The equivalence between these six viewpoints is formally captured in the next statement.

**Theorem 12** (Equivalence)**.** *The six viewpoints are equivalent: they all produce the same (unique) point $x^{k+1}$.*

*Proof.* The proof is simple, and follows directly from the above discussion. In particular, see the caption of Figure 2.1. □

### 2.3.2 Projection matrices

The explicit projection matrices of the projections depicted in Figure 2.1 can be constructed using the $Z$ matrix. Indeed, recall that $S$ is a $m \times q$ random matrix (with $q$ possibly being random), and that $A$ is an $m \times n$ matrix. Let us define the random quantity

$$d \overset{\text{def}}{=} \mathbf{Rank}\left(S^\top A\right) \tag{2.12}$$

and notice that $d \leq \min\{q, n\}$,

$$\dim\left(\mathbf{Range}\left(B^{-1}A^\top S\right)\right) = d, \qquad \text{and} \qquad \dim\left(\mathbf{Null}\left(S^\top A\right)\right) = n - d. \tag{2.13}$$

Recall that (1.26) shows that $B^{-1}Z$ is a projection onto $\mathbf{Range}\left(B^{-1}A^\top S\right)$ and along $\mathbf{Null}\left(S^\top A\right)$. This sheds additional light on Figure 2.1 as it gives explicit expressions for the associated projection matrices. This also shows that $I - B^{-1}Z$ is a projection and thus implies that $I - B^{-1}Z$ is a *contraction* with respect to the $B$-norm, which means that the random fixed point iteration (2.11) has only very little room not to work. While $I - B^{-1}Z$ is not a strict contraction, under some reasonably weak assumptions on $S$ it will be a strict contraction in expectation, which ensures convergence. We shall state these assumptions and develop the associated convergence theory for our method in Section 2.5 and Section 2.6.

## 2.4 Special Cases: Examples

In this section we briefly mention how by selecting the parameters $S$ and $B$ of our method we recover several existing methods. The list is by no means comprehensive and merely serves the purpose of an illustration of the flexibility of our algorithm. All the associated complexity results we present in this section, can be recovered from Theorem 19, presented later in Section 2.6.

### 2.4.1 The one step method

When $S$ is an $m \times m$ invertible matrix with probability one, then the system $S^\top Ax = S^\top b$ is equivalent to solving $Ax = b$, thus the solution to (2.5) must be $x^{k+1} = x^*$, independently of matrix $B$. Our convergence theorems also predict this one step behaviour, since $\rho = 0$ (see Table 2.1).

### 2.4.2 Random vector sketch

When $S = s \in \mathbb{R}^m$ is restricted to being a random column vector, then from (2.10) a step of our method is given by

$$x^{k+1} = x^k - \frac{s^\top (Ax^k - b)}{s^\top AB^{-1}A^\top s} B^{-1}A^\top s, \tag{2.14}$$

if $A^\top s \neq 0$ and $x^{k+1} = x^k$ otherwise. This is because the pseudoinverse of a scalar $\alpha \in \mathbb{R}$ is given by

$$\alpha^\dagger = \begin{cases} 1/\alpha & \text{if } \alpha \neq 0 \\ 0 & \text{if } \alpha = 0. \end{cases}$$

Next we describe several well known specializations of the random vector sketch and for brevity, we write the updates in the form of (2.14) and leave implicit that when the denominator is zero, no step is taken.

### 2.4.3 Randomized Kaczmarz

If we choose $S = e^i$ (unit coordinate vector in $\mathbb{R}^m$) and $B = I$ (the identity matrix), in view of (2.5) we obtain the method:

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^k\|_2^2 \quad \text{subject to} \quad A_{i:}x = b_i. \tag{2.15}$$

Using (2.10), these iterations can be calculated with

$$\boxed{x^{k+1} = x^k - \frac{A_{i:}x^k - b_i}{\|A_{i:}\|_2^2}(A_{i:})^\top} \tag{2.16}$$

**Complexity.** When $i$ is selected at random, this is the randomized Kaczmarz ($RK$) method [125]. A specific non-uniform probability distribution for $S$ yields simple and easily interpretable (but not necessarily optimal) complexity bound. In particular, by selecting $i$ with probability proportional to the magnitude of row $i$ of $A$, that is $p_i = \|A_{i:}\|_2^2/\|A\|_F^2$, it follows from

Theorem 19 that RK enjoys the following complexity bound:

$$\mathbf{E}\left[\|x^k - x^*\|_2^2\right] \leq \left(1 - \frac{\lambda_{\min}\left(A^\top A\right)}{\|A\|_F^2}\right)^k \|x^0 - x^*\|_2^2. \tag{2.17}$$

This result was first established by Strohmer and Vershynin [125]. We also provide new convergence results in Theorem 15, based on the convergence of the norm of the expected error. Theorem 15 applied to the RK method gives

$$\|\mathbf{E}\left[x^k - x^*\right]\|_2^2 \leq \left(1 - \frac{\lambda_{\min}\left(A^\top A\right)}{\|A\|_F^2}\right)^{2k} \|x^0 - x^*\|_2^2. \tag{2.18}$$

Now the convergence rate appears squared, which is a better rate, though, the expectation has moved inside the norm, which is a weaker form of convergence as proven in Lemma 10.

Analogous results for the convergence of the norm of the expected error holds for all the methods we present, though we only illustrate this with the RK method.

**Re-interpretation as SGD with exact line search.** Using the "Constrain and Approximate" formulation (2.6), randomized Kaczmarz method can also be written as

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^*\|_2^2 \quad \text{subject to} \quad x = x^k + y(A_{i:})^\top, \quad y \in \mathbb{R},$$

with probability $p_i$. Writing the least squares function $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ as

$$f(x) = \sum_{i=1}^m p_i f_i(x), \qquad f_i(x) = \frac{1}{2p_i}(A_{i:}x - b_i)^2,$$

we see that the random vector $\nabla f_i(x) = \frac{1}{p_i}(A_{i:}x - b_i)(A_{i:})^\top$ is an unbiased estimator of the gradient of $f$ at $x$. That is, $\mathbf{E}\left[\nabla f_i(x)\right] = \nabla f(x)$. Notice that RK takes a step in the direction $-\nabla f_i(x)$. This is true even when $A_{i:}x - b_i = 0$, in which case, the RK does not take any step. Hence, RK takes a step in the direction of the negative stochastic gradient. This means that it is equivalent to the Stochastic Gradient Descent (SGD) method. However, the stepsize choice is very special: RK chooses the stepsize which leads to the point which is closest to $x^*$ in the Euclidean norm.

Later in Section 3.4.2 in Chapter 3 we give yet another interpretation of the RK method, namely, that the RK method is the equivalent to applying the randomized coordinate descent method to the dual of the least-norm problem.

### 2.4.4 Randomized Coordinate Descent: positive definite case

If $A$ is symmetric positive definite, then we can choose $B = A$ and $S = e^i$ in (2.5), which results in

$$x^{k+1} \overset{\text{def}}{=} \arg\min_{x \in \mathbb{R}^n} \|x - x^k\|_A^2 \quad \text{subject to} \quad (A_{i:})^\top x = b_i, \tag{2.19}$$

where we used the symmetry of $A$ to get $(e^i)^\top A = A_{i:} = (A_{:i})^\top$. The solution to the above, given by (2.10), is

$$\boxed{x^{k+1} = x^k - \frac{(A_{i:})^\top x^k - b_i}{A_{ii}} e^i} \tag{2.20}$$

**Complexity.** When $i$ is chosen randomly, this is the *Randomized CD* method (CD-pd). Applying Theorem 19, we see the probability distribution $p_i = A_{ii}/\mathbf{Tr}\,(A)$ results in a convergence with

$$\mathbf{E}\left[\|x^k - x^*\|_A^2\right] \leq \left(1 - \frac{\lambda_{\min}\,(A)}{\mathbf{Tr}\,(A)}\right)^k \|x^0 - x^*\|_A^2. \tag{2.21}$$

This result was first established by Leventhal and Lewis [68].

**Interpretation.** Using the Constrain-and-Approximate formulation (2.6), this method can be interpreted as

$$x^{k+1} = \arg\min \|x - x^*\|_A^2 \quad \text{subject to} \quad x = x^k + ye^i, \quad y \in \mathbb{R}, \tag{2.22}$$

with probability $p_i$. Using the identity $Ax^* = b$, it is easy to check that the function $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ satisfies: $\|x - x^*\|_A^2 = 2f(x) + b^\top x^*$. Therefore, (2.22) is equivalent to

$$x^{k+1} = \arg\min f(x) \quad \text{subject to} \quad x = x^k + ye^i, \quad y \in \mathbb{R}. \tag{2.23}$$

The iterates (2.20) can also be written as

$$x^{k+1} = x^k - \frac{1}{L_i}\nabla_i f(x^k) e^i,$$

where $L_i = A_{ii}$ is the Lipschitz constant of the gradient of $f$ corresponding to coordinate $i$ and $\nabla_i f(x^k)$ is the $i$th partial derivative of $f$ at $x^k$.

### 2.4.5 Randomized block Kaczmarz

Our framework also extends to new block formulations of the randomized Kaczmarz method. Let $R$ be a random subset of $[m]$ and let $S = I_{:R}$ be a column concatenation of the columns of the $m \times m$ identity matrix $I$ indexed by $R$. Further, let $B = I$. Then (2.5) specializes to

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^k\|_2^2 \quad \text{subject to} \quad A_{R:}x = b_R.$$

In view of (2.10), this can be equivalently written as

$$\boxed{x^{k+1} = x^k - (A_{R:})^\top (A_{R:}(A_{R:})^\top)^\dagger (A_{R:}x^k - b_R)} \tag{2.24}$$

**Complexity.** From Theorem 16 we obtain the following new complexity result:

$$\mathbf{E}\left[\|x^k - x^*\|_2^2\right] \leq \left(1 - \lambda_{\min}\left(\mathbf{E}\left[(A_{R:})^\top (A_{R:}(A_{R:})^\top)^\dagger A_{R:}\right]\right)\right)^k \|x^0 - x^*\|_2^2.$$

To obtain a more meaningful convergence rate, we would need to bound the smallest eigenvalue of $\mathbf{E}\left[(A_{R:})^\top (A_{R:}(A_{R:})^\top)^\dagger A_{R:}\right]$. This has been done in [84, 85] when the image of $R$ defines a row paving of $A$. Our framework paves the way for analysing the convergence of new block methods for a large set of possible random subsets $R$, including, for example, overlapping partitions.

### 2.4.6 Randomized Newton: positive definite case

If $A$ is symmetric positive definite, then we can choose $B = A$ and $S = I_{:C}$, a column concatenation of the columns of $I$ indexed by $C$, which is a random subset of $\{1, \dots, n\}$. In view of (2.5), this results in

$$x^{k+1} \overset{\text{def}}{=} \arg\min_{x \in \mathbb{R}^n} \|x - x^k\|_A^2 \quad \text{subject to} \quad (A_{:C})^\top x = b_C. \tag{2.25}$$

In view of (2.10), we can equivalently write the method as

$$\boxed{x^{k+1} \quad = \quad x^k - I_{:C}((I_{:C})^\top A I_{:C})^{-1}(I_{:C})^\top (Ax^k - b)} \tag{2.26}$$

**Complexity.** Clearly, iteration (2.26) is well defined as long as $C$ is nonempty with probability 1. Such $C$ is referred to in [101] as a "non-vacuous" sampling. From Theorem 16 we obtain the following convergence rate:

$$\begin{aligned}\mathbf{E}\left[\|x^k - x^*\|_A^2\right] &\leq \rho^k \|x^0 - x^*\|_A^2 \\ &= \left(1 - \lambda_{\min}\left(\mathbf{E}\left[I_{:C}((I_{:C})^\top A I_{:C})^{-1}(I_{:C})^\top A\right]\right)\right)^k \|x^0 - x^*\|_A^2.\end{aligned} \tag{2.27}$$

The convergence rate of this particular method was first established and studied in [101]. Moreover, it was shown in [101] that $\rho < 1$ if one additionally assumes that the probability that $i \in C$ is positive for each column $i \in \{1, \dots, n\}$, i.e., that $C$ is a "proper" sampling.

**Interpretation.** Using formulation (2.6), and in view of the equivalence between $f(x)$ and $\|x - x^*\|_A^2$ discussed in Section 2.4.4, the Randomized Newton method can be equivalently written as

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad x = x^k + I_{:C}\, y, \quad y \in \mathbb{R}^{|C|}.$$

The next iterate is determined by advancing from the previous iterate over a subset of coordinates such that $f$ is minimized. Hence, an exact line search is performed in a random $|C|$ dimensional subspace.

Method (2.26) was first studied by Qu et al [101], and referred therein as "Method 1", or *Randomized Newton Method*. The name comes from the observation that the method inverts random principal submatrices of $A$ and that in the special case when $C = \{1, \dots, n\}$ with probability 1, it specializes to the Newton method (which in this case converges in a single step). The expression $\rho$ defining the convergence rate of this method is rather involved and it is not immediately obvious what is gained by performing a search in a higher dimensional subspace ($|C| > 1$) rather than in the one-dimensional subspaces ($|C| = 1$), as is standard in the optimization literature. Let us write $\rho = 1 - \sigma_\tau$ in the case when the $C$ is chosen to be a subset

of $\{1,\dots,n\}$ of size $\tau$, uniformly at random. In view of Lemma 11, the method takes $\tilde{O}(1/\sigma_\tau)$ iterations to converge, where the tilde notation suppresses logarithmic terms. It was shown in [101] that $1/\sigma_\tau \leq 1/(\tau\sigma_1)$. That is, one can expect to obtain at least *superlinear speedup* in $\tau$ — this is what is gained by moving to blocks / higher dimensional subspaces. For further details and additional properties of the method we refer the reader to [101].

### 2.4.7 Randomized Coordinate Descent: least-squares version

By choosing $S = Ae^i =: A_{:i}$ as the $i$th column of $A$ and $B = A^\top A$, the resulting iterates (2.6) are given by

$$x^{k+1} = \arg\min_{x\in\mathbb{R}^n} \|Ax-b\|_2^2 \quad \text{subject to} \quad x = x^k + y\,e^i, \quad y\in\mathbb{R}. \tag{2.28}$$

When $i$ is selected at random, this is the Randomized Coordinate Descent method (*CD-LS*) applied to the least-squares problem: $\min_x \|Ax-b\|_2^2$. Using (2.10), these iterations can be calculated with

$$\boxed{x^{k+1} = x^k - \frac{(A_{:i})^\top(Ax^k-b)}{\|A_{:i}\|_2^2}e^i} \tag{2.29}$$

**Complexity.** Applying Theorem 19, we see that by selecting $i$ with probability proportional to magnitude of column $i$ of $A$, that is $p_i = \|A_{:i}\|_2^2/\|A\|_F^2$, results in a convergence with

$$\mathbf{E}\left[\|x^k-x^*\|_{A^\top A}^2\right] \leq \rho^k \|x^0-x^*\|_{A^\top A}^2 = \left(1-\frac{\lambda_{\min}\left(A^\top A\right)}{\|A\|_F^2}\right)^k \|x^0-x^*\|_{A^\top A}^2. \tag{2.30}$$

This result was first established by Leventhal and Lewis [68].

**Interpretation.** Using the Constrain-and-Approximate formulation (2.6), the CD-LS method can be interpreted as

$$x^{k+1} = \arg\min_{x\in\mathbb{R}^n} \|x-x^*\|_{A^\top A}^2 \quad \text{subject to} \quad x = x^k + ye^i, \quad y\in\mathbb{R}. \tag{2.31}$$

The CD-LS method selects a coordinate to advance from the previous iterate $x^k$, then performs an exact minimization of the least squares function over this line. This is equivalent to applying coordinate descent to the least squares problem $\min_{x\in\mathbb{R}^n} f(x) \overset{\text{def}}{=} \frac{1}{2}\|Ax-b\|_2^2$. The iterates (2.28) can be written as

$$x^{k+1} = x^k - \frac{1}{L_i}\nabla_i f(x^k)e^i,$$

where $L_i \overset{\text{def}}{=} \|A_{:i}\|_2^2$ is the Lipschitz constant of the gradient corresponding to coordinate $i$ and $\nabla_i f(x^k)$ is the $i$th partial derivative of $f$ at $x^k$.

## 2.5 Convergence: General Theory

We shall present two complexity theorems: we first study the convergence of $\|\mathbf{E}\left[x^k - x^*\right]\|_B$ , and then move on to analysing the convergence of $\mathbf{E}\left[\|x^k - x^*\|\right]_B$. Both theorems depend on the same convergence rate $\rho \in [0,\, 1]$, which we examine in the next section. In particular, we show that $\rho < 1$ if and only if $A$ has full column rank. Thus the convergence results in this section only prove that the method convergence when $A$ has full column rank. Later in Chapter 3 we extend these convergence results, and show that $A$ need not have full column rank.

### 2.5.1 The rate of convergence

All of our convergence theorems (see Table 2.1) depend on the convergence rate

$$\rho \overset{\text{def}}{=} 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}). \tag{2.32}$$

To show that the rate is meaningful, in Lemma 13 we prove that $0 \leq \rho \leq 1$. We also give an alternative expression and provide a meaningful lower bound for $\rho$.

**Lemma 13.** *The quantity $\rho$ defined in* (2.32) *satisfies:*

$$0 \leq 1 - \frac{\mathbf{E}\left[d\right]}{n} \leq \rho \leq 1, \tag{2.33}$$

*where $d = \mathbf{Rank}\left(S^\top A\right)$. Furthermore*

$$\rho = \|I - B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}\|_2. \tag{2.34}$$

*Proof.* Recall from Lemma 9 that $B^{-1/2}ZB^{-1/2}$ is a projection, whence the spectrum of $B^{-1/2}ZB^{-1/2}$ is contained in $\{0, 1\}$. Using this, combined with the fact that the mapping $A \mapsto \lambda_{\max}(A)$ is convex on the set of symmetric matrices and Jensen's inequality, we get

$$\lambda_{\max}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}) \leq \mathbf{E}\left[\lambda_{\max}(B^{-1/2}ZB^{-1/2})\right] \leq 1. \tag{2.35}$$

The inequality $\lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}) \geq 0$ can be shown analogously using convexity of the mapping $A \mapsto -\lambda_{\min}(A)$. Thus, $\lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}) \in [0, 1]$, which implies $0 \leq \rho \leq 1$. We now refine the lower bound. As the trace of a matrix is equal to the sum of its eigenvalues, we have

$$\mathbf{E}\left[\mathbf{Tr}\left(B^{-1/2}ZB^{-1/2}\right)\right] = \mathbf{Tr}\left(\mathbf{E}\left[B^{-1/2}ZB^{-1/2}\right]\right) \geq n\,\lambda_{\min}(\mathbf{E}\left[B^{-1/2}ZB^{-1/2}\right]). \tag{2.36}$$

From (1.28) we that have $\mathbf{Tr}\left(B^{-1/2}ZB^{-1/2}\right) = d$. Thus rewriting (2.36) gives $1 - \mathbf{E}\left[d\right]/n \leq \rho$. Finally, from the symmetry of $Z$ it follows that $(I - B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})$ is symmetric, and consequently

$$\begin{aligned}\|(I - B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})\|_2 &= \lambda_{\max}(I - B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}) \\ &= \left(1 - \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})\right) = \rho.\end{aligned}$$

□

The lower bound on $\rho$ in (2.33) has a natural interpretation which makes intuitive sense. We shall present it from the perspective of the Constrain-and-Approximate formulation (2.6). As the dimension ($d$) of the search space $B^{-1}A^\top S$ increases (see (2.13)), the lower bound on $\rho$ decreases, and a faster convergence is possible. For instance, when $S$ is restricted to being a random column vector, as it is in the RK (2.16), CD-LS (2.29) and CD-pd (2.21) methods, the convergence rate is bounded with $1 - 1/n \leq \rho$. Using Lemma 11, this translates into the simple iteration complexity bound of $k \geq n \log(1/\epsilon)$. On the other extreme, when the search space is large, then the lower bound is close to zero, allowing room for the method to be faster.

We now characterize circumstances under which $\rho$ is strictly smaller than one.

**Lemma 14.** *If $\mathbf{E}[Z]$ is invertible, then $\rho < 1$, $A$ has full column rank and $x^*$ is unique.*

*Proof.* Assume that $\mathbf{E}[Z]$ is invertible. First, this means that $B^{-1/2}\mathbf{E}[Z]B^{-1/2}$ is positive definite, which in view of (2.32) means that $\rho < 1$. If $A$ did not have full column rank, then there would be $0 \neq x \in \mathbb{R}^n$ such that $Ax = 0$. However, we then have $Zx = 0$ and also $\mathbf{E}[Z]x = 0$, contradicting the assumption that $\mathbf{E}[Z]$ is invertible. Finally, since $A$ has full column rank, $x^*$ must be unique (recall that we assume throughout this chapter that the system $Ax = b$ is consistent). □

### 2.5.2 Exact characterization and norm of expectation

We now state a theorem which exactly characterizes the evolution of the expected iterates through a linear fixed point iteration. As a consequence, we obtain a convergence result for the norm of the expected error. While we do not highlight this in the text, this theorem can be applied to all the particular instances of our general method we detail throughout this chapter.

For any $M \in \mathbb{R}^{n\times n}$ let us define

$$\|M\|_B \overset{\text{def}}{=} \max_{\|x\|_B=1} \|Mx\|_B. \tag{2.37}$$

**Theorem 15** (Norm of expectation)**.** *For every $x^* \in \mathbb{R}^n$ satisfying $Ax^* = b$ we have*

$$\mathbf{E}\left[x^{k+1} - x^*\right] = \left(I - B^{-1}\mathbf{E}[Z]\right)\mathbf{E}\left[x^k - x^*\right]. \tag{2.38}$$

*Moreover, the induced $B$-norm of the iteration matrix $I - B^{-1}\mathbf{E}[Z]$ is equal to $\rho$:*

$$\|I - B^{-1}\mathbf{E}[Z]\|_B = 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}[Z]B^{-1/2}) = \rho. \tag{2.39}$$

*Therefore,*

$$\|\mathbf{E}\left[x^k - x^*\right]\|_B \leq \rho^k \|x^0 - x^*\|_B. \tag{2.40}$$

*Proof.* Taking expectations conditioned on $x^k$ in (2.11), we get

$$\mathbf{E}\left[x^{k+1} - x^* \mid x^k\right] = (I - B^{-1}\mathbf{E}[Z])(x^k - x^*). \tag{2.41}$$

Taking expectation again gives

$$\begin{aligned}\mathbf{E}\left[x^{k+1}-x^*\right] &= \mathbf{E}\left[\mathbf{E}\left[x^{k+1}-x^* \mid x^k\right]\right] \\ &\overset{(2.41)}{=} \mathbf{E}\left[(I-B^{-1}\mathbf{E}\left[Z\right])(x^k-x^*)\right] \\ &= (I-B^{-1}\mathbf{E}\left[Z\right])\mathbf{E}\left[x^k-x^*\right],\end{aligned}$$

and thus (2.38) holds. The equivalence (2.39) follows by

$$\begin{aligned}\rho &\overset{(2.34)}{=} \|I-B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}\|_2 \\ &= \max_{\|v\|_2=1}\|(I-B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})v\|_2 \\ &\overset{(v=B^{1/2}w)}{=} \max_{\|B^{1/2}w\|_2=1}\|B^{1/2}B^{-1/2}(I-B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})B^{1/2}w\|_2 \\ &= \max_{\|w\|_B=1}\|(I-B^{-1}\mathbf{E}\left[Z\right])w\|_B = \|I-B^{-1}\mathbf{E}\left[Z\right]\|_B.\end{aligned}$$

For all $k$, define $r^k \overset{\text{def}}{=} B^{1/2}(x^k-x^*)$. Left multiplying (2.38) by $B^{1/2}$ gives

$$\mathbf{E}\left[r^{k+1}\right] = (I-B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})\mathbf{E}\left[r^k\right].$$

Applying the norms to both sides we obtain the estimate

$$\|\mathbf{E}\left[r^{k+1}\right]\|_2 \le \|I-B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}\|_2\,\|\mathbf{E}\left[r^k\right]\|_2. \tag{2.42}$$

The claim (2.40) now follows by observing (2.34), that $\|r_k\|_2 = \|x^k-x^*\|_B$ and unrolling the recurrence in (2.42).

□

### 2.5.3 Expectation of norm

We now turn to analysing the convergence of the expected norm of the error, for which we need the following technical lemma.

**Theorem 16** (Expectation of norm)**.** *If $\mathbf{E}\left[Z\right]$ is positive definite, then*

$$\mathbf{E}\left[\|x^k-x^*\|_B^2\right] \le \rho^k\|x^0-x^*\|_B^2, \tag{2.43}$$

*where $\rho<1$ is given in* (2.32)*.*

*Proof.* Let $r^k = B^{1/2}(x^k-x^*)B^{1/2}$. Taking expectation in (2.11) conditioned on $r^k$ gives

$$\begin{aligned}\mathbf{E}\left[\|r^{k+1}\|_2^2 \mid r^k\right] &\overset{(2.11)}{=} \mathbf{E}\left[\|(I-B^{-1/2}ZB^{-1/2})r^k\|_2^2 \mid r^k\right] \\ &\overset{(1.27)}{=} \left\langle(I-B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2})r^k, r^k\right\rangle \\ &\le \|I-B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}\|_2\,\|r^k\|_2^2.\end{aligned}$$

Using (2.34), taking expectation again and unrolling the recurrence gives the result. □

The convergence rate $\rho$ of the expected norm of the error is "worse" than the $\rho^2$ rate of convergence of the norm of the expected error in Theorem 15. This should not be misconstrued as Theorem 15 offering a "better" convergence rate than Theorem 16, because, as explained in Lemma 10, convergence of the expected norm of the error is a stronger type of convergence. More importantly, the exponent is not of any crucial importance; clearly, an exponent of 2 manifests itself only in halving the number of iterations (see Lemma 11).

## 2.6 Methods Based on Discrete Sampling

When $S$ has a discrete distribution, we can establish under reasonable assumptions when $\mathbf{E}[Z]$ is positive definite (Proposition 18), we can optimize the convergence rate in terms of the chosen probability distribution, and finally, determine a probability distribution for which the convergence rate is expressed in terms of the scaled condition number (Theorem 19).

**Assumption 17.** *The random matrix $S$ has a discrete distribution. In particular, $S = S_i \in \mathbb{R}^{m\times q_i}$ with probability $p_i > 0$, $\sum_{i=1}^r p_i = 1$, where $S_i^\top A$ has full row rank and $q_i \in \mathbb{N}$, for $i = 1,\dots,r$. Furthermore $\mathbf{S} \overset{def}{=} [S_1,\dots,S_r] \in \mathbb{R}^{m\times\sum_{i=1}^r q_i}$ is such that $A^\top\mathbf{S}$ has full row rank.*

For simplicity, sampling $S$ satisfying the above assumption will be called a *complete discrete sampling*. We now give an example of such a sampling. If $A$ has full column rank and each row of $A$ is not strictly zero, $S = e^i$ with probability $p_i = 1/n$, for $i = 1,\dots,n$, then $\mathbf{S} = I$ and $S$ is a complete discrete sampling. In fact, from any basis of $\mathbb{R}^n$ we can construct a complete discrete sampling in an analogous way.

When $S$ is a complete discrete sampling, then $S^\top A$ has full row rank and

$$(S^\top AB^{-1}A^\top S)^\dagger = (S^\top AB^{-1}A^\top S)^{-1}.$$

Therefore we replace the pseudoinverse in (2.10) and (2.11) by the inverse. Furthermore, using a complete discrete sampling guarantees convergence of the resulting method.

**Proposition 18.** *If $S$ is a complete discrete sampling, $\mathbf{E}[Z]$ is positive definite.*

*Proof.* Let

$$D \overset{\text{def}}{=} \operatorname{diag}\left(\sqrt{p_1}((S_1)^\top AB^{-1}A^\top S_1)^{-1/2},\dots,\sqrt{p_r}((S_r)^\top AB^{-1}A^\top S_r)^{-1/2}\right) \tag{2.44}$$

which is a block diagonal matrix, and is well defined and invertible as $S_i^\top A$ has full row rank for $i = 1,\dots,r$. Taking the expectation of $Z$ (2.2) gives

$$\begin{aligned}\mathbf{E}[Z] &= \sum_{i=1}^r A^\top S_i(S_i^\top AB^{-1}A^\top S_i)^{-1}S_i^\top Ap_i \\ &= A^\top\left(\sum_{i=1}^r S_i\sqrt{p_i}(S_i^\top AB^{-1}A^\top S_i)^{-1/2}(S_i^\top AB^{-1}A^\top S_i)^{-1/2}\sqrt{p_i}S_i^\top\right)A \\ &= \left(A^\top\mathbf{S}D\right)\left(D\mathbf{S}^\top A\right), \end{aligned}\tag{2.45}$$

which is positive definite because $A^\top\mathbf{S}$ has full row rank and $D$ is invertible. □

With $\mathbf{E}[Z]$ positive definite, we can apply the convergence Theorem 15 and 16, and the resulting method converges.

### 2.6.1 Optimal probabilities

We can choose the discrete probability distribution that optimizes the convergence rate. For this, according to Theorems 16 and 15 we need to find $p = (p_1,\dots,p_r)$ that maximizes the

minimal eigenvalue of $B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}$. Let $S$ be a complete discrete sampling and fix the sample matrices $S_1,\ldots,S_r$. Let us denote $Z=Z(p)$ as a function of $p=(p_1,\ldots,p_r)$. Then we can also think of the spectral radius as a function of $p$ where

$$\rho(p)=1-\lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z(p)\right]B^{-1/2}).$$

If we let $\Delta_r=\{p=(p_1,\ldots,p_r)\in\mathbb{R}^r\ :\ \sum_{i=1}^r p_i=1,\ p\geq 0\}$, the problem of minimizing the spectral radius (i.e., optimizing the convergence rate) can be written as

$$\rho^* \quad \stackrel{\text{def}}{=} \quad \min_{p\in\Delta_r}\rho(p) \quad = \quad 1-\max_{p\in\Delta_r}\lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z(p)\right]B^{-1/2}).$$

This can be cast as a convex optimization problem, by first re-writing

$$\begin{aligned}B^{-1/2}\mathbf{E}\left[Z(p)\right]B^{-1/2} &= \sum_{i=1}^r p_i\left(B^{-1/2}A^\top S_i(S_i^\top AB^{-1}A^\top S_i)^{-1}S_i^\top AB^{-1/2}\right)\\ &= \sum_{i=1}^r p_i\left(V_i(V_i^\top V_i)^{-1}V_i^\top\right),\end{aligned}$$

where $V_i=B^{-1/2}A^\top S_i$. Thus

$$\rho^* \quad = \quad 1-\max_{p\in\Delta_r}\lambda_{\min}\left(\sum_{i=1}^r p_iV_i(V_i^\top V_i)^{-1}V_i^\top\right). \tag{2.46}$$

To obtain $p$ that maximizes the smallest eigenvalue, we solve

$$\begin{aligned}\max_{p,t}\quad & t\\ \text{subject to}\quad & \sum_{i=1}^r p_i\left(V_i(V_i^\top V_i)^{-1}V_i^\top\right)\succeq t\cdot I,\\ & p\in\Delta_r.\end{aligned} \tag{2.47}$$

Despite (2.47) being a convex semi-definite program[1], which is apparently a harder problem than solving the original linear system, investing the time into solving (2.47) using a solver for convex conic programming such as `cvx` [53] can pay off, as we show in Section 2.8.4. Though for a practical method based on this, we would need to develop an approximate solution to (2.47) which can be efficiently calculated.

### 2.6.2 Convenient probabilities

Next we develop a choice of probability distribution that yields a convergence rate that is easy to interpret. This result is an extension of Strohmer and Vershynin's [125] non-uniform probability distribution for the Kaczmarz method. Our extension includes a wide range of methods, such as the randomized Kaczmarz, randomized coordinate descent, as well as their block variants.

[1] When preparing a revision of the paper on which this chapter is based, we have learned about the existence of prior work [23] where the authors have also characterized the probability distribution that optimizes the convergences rate of the RK method as the solution to an SDP.

However, it is more general, and covers many other possible particular algorithms, which arise by choosing a particular set of sample matrices $S_i$, for $i = 1, \dots, r$.

**Theorem 19.** *Let $S$ be a complete discrete sampling such that $S = S_i \in \mathbb{R}^m$ with probability*

$$p_i = \frac{\mathbf{Tr}\left(S_i^\top A B^{-1} A^\top S_i\right)}{\|B^{-1/2} A^\top \mathbf{S}\|_F^2}, \quad \text{for} \quad i = 1, \dots, r. \tag{2.48}$$

*Then the iterates* (2.10) *satisfy*

$$\mathbf{E}\left[\|x^k - x^*\|_B^2\right] \leq \rho_c^k \|x^0 - x^*\|_B^2, \tag{2.49}$$

*where*

$$\rho_c = 1 - \frac{\lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right)}{\|B^{-1/2} A^\top \mathbf{S}\|_F^2}. \tag{2.50}$$

*Proof.* Let $t_i = \mathbf{Tr}\left(S_i^\top A B^{-1} A^\top S_i\right)$, and with (2.48) in (2.44) we have

$$D^2 = \frac{1}{\|B^{-1/2} A^\top \mathbf{S}\|_F^2} \mathrm{diag}\left(t_1 (S_1^\top A B^{-1} A^\top S_1)^{-1}, \dots, t_r (S_r^\top A B^{-1} A^\top S_r)^{-1}\right),$$

thus

$$\lambda_{\min}(D^2) = \frac{1}{\|B^{-1/2} A^\top \mathbf{S}\|_F^2} \min_i \left\{ \frac{t_i}{\lambda_{\max}(S_i^\top A B^{-1} A^\top S_i)} \right\} \geq \frac{1}{\|B^{-1/2} A^\top \mathbf{S}\|_F^2}. \tag{2.51}$$

Applying the above in (2.45) gives

$$\begin{aligned} \lambda_{\min}\left(B^{-1/2} \mathbf{E}\left[Z\right] B^{-1/2}\right) &= \lambda_{\min}\left(B^{-1/2} A^\top \mathbf{S} D^2 \mathbf{S}^\top A B^{-1/2}\right) \\ &= \lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S} D^2\right) \\ &\geq \lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right) \lambda_{\min}(D^2) \\ &\geq \frac{\lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right)}{\|B^{-1/2} A^\top \mathbf{S}\|_F^2}, \end{aligned} \tag{2.52}$$

where in the first step we used the fact that for arbitrary matrices $B, C$ of appropriate sizes, $\lambda_{\min}(BC) = \lambda_{\min}(CB)$, and in the first inequality the fact that if $B, C \in \mathbb{R}^{n \times n}$ are positive definite, then $\lambda_{\min}(BC) \geq \lambda_{\min}(B)\lambda_{\min}(C)$. Finally

$$1 - \lambda_{\min}\left(B^{-1/2} \mathbf{E}\left[Z\right] B^{-1/2}\right) \leq 1 - \frac{\lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right)}{\|B^{-1/2} A^\top \mathbf{S}\|_F^2}. \tag{2.53}$$

The result (2.49) follows by applying Theorem 16. □

The convergence rate $\lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right) / \|B^{-1/2} A^\top \mathbf{S}\|_F^2$ is known as the scaled condition number, and naturally appears in other numerical schemes, such as matrix inversion [33, 29]. When $S_i = s_i \in \mathbb{R}^n$ is a column vector then

$$p_i = \left((s_i)^\top A B^{-1} A^\top s_i\right) / \|B^{-1/2} A^\top \mathbf{S}\|_F^2,$$

for $i = 1, \dots r$. In this case, the bound (2.51) is an equality and $D^2$ is a scaled identity, so (2.52) and consequently (2.53) are equalities. For block methods, it is different story, and there is much more slack in the inequality (2.53). So much so, the convergence rate (2.50) does not indicate any advantage of using a block method (contrary to numerical experiments). To see the advantage of a block method, we need to use the exact expression for $\lambda_{\min}(D^2)$ given in (2.51). Though this results in a somewhat harder to interpret convergence rate, one could use a so called *matrix paving* to explore this convergence rate, as was done for the block Kaczmarz method (see [85, 84] for more details).

By appropriately choosing $B$ and $S$, this theorem applied to RK method (2.15), the CD-LS method (2.28) and the CD-pd method (2.19), yields the convergence results (2.17), (2.30) and (2.21), respectively, for single column sampling or block methods alike.

This theorem also suggests a preconditioning strategy, in that, a faster convergence rate will be attained if $\mathbf{S}$ is an approximate inverse of $B^{-1/2}A^\top$. For instance, in the RK method where $B = I$, this suggests that an accelerated convergence can be attained if $S$ is a random sampling of the rows of a preconditioner (approximate inverse) of $A$.

## 2.7 Methods Based on Gaussian Sampling

In this section we shall describe variants of our method in the case when $S$ is a Gaussian vector with mean $0 \in \mathbb{R}^m$ and a positive definite covariance matrix $\Sigma \in \mathbb{R}^{m\times m}$. That is, $S = \zeta \sim N(0, \Sigma)$. This applied to (2.10) results in iterations of the form

$$\boxed{x^{k+1} = x^k - \frac{\zeta^\top (Ax^k - b)}{\zeta^\top AB^{-1}A^\top \zeta} B^{-1}A^\top \zeta} \tag{2.54}$$

Unlike the discrete methods in Section 2.4, to calculate an iteration of (2.54) we need to compute the product of a matrix with a dense vector $\zeta$. This significantly raises the cost of an iteration. Though in our numeric tests in Section 2.8, the faster convergence of the Gaussian method often pays off for their high iteration cost.

To analyze the complexity of the resulting method let $\xi \overset{\text{def}}{=} B^{-1/2}A^\top S$, which is also Gaussian, distributed as $\xi \sim N(0, \Omega)$, where $\Omega \overset{\text{def}}{=} B^{-1/2}A^\top \Sigma A B^{-1/2}$. In this section we assume *$A$ has full column rank*, so that $\Omega$ is always positive definite. The complexity of the method can be established through

$$\rho \quad = \quad 1 - \lambda_{\min}\left(\mathbf{E}\left[B^{-1/2}ZB^{-1/2}\right]\right) = 1 - \lambda_{\min}\left(\mathbf{E}\left[\frac{\xi\xi^\top}{\|\xi\|_2^2}\right]\right). \tag{2.55}$$

We can simplify the above by using the lower bound

$$\mathbf{E}\left[\frac{\xi\xi^\top}{\|\xi\|_2^2}\right] \succeq \frac{2}{\pi}\frac{\Omega}{\mathbf{Tr}\left(\Omega\right)},$$

which is proven in Lemma 20 in the Appendix of this chapter. Thus

$$1 - \frac{1}{n} \leq \rho \leq 1 - \frac{2}{\pi}\frac{\lambda_{\min}(\Omega)}{\mathbf{Tr}\left(\Omega\right)}, \tag{2.56}$$

where we used the general lower bound in (2.33). Lemma 20 also shows that $\mathbf{E}\left[\xi\xi^\top/\|\xi\|_2^2\right]$ is positive definite, thus Theorem 16 guarantees that the expected norm of the error of all Gaussian methods converges exponentially to zero. This bound is tight upto a constant factor. Indeed, if $A = I = \Sigma$ then $\xi \sim N(0, I)$ and $\mathbf{E}\left[\xi\xi^\top/\|\xi\|_2^2\right] = \frac{1}{n}I$, which yields

$$1 - \frac{1}{n} \leq \rho \leq 1 - \frac{2}{\pi}\cdot\frac{1}{n}.$$

When $n = 2$, then in Lemma 21 of the Appendix of this chapter we prove that

$$\mathbf{E}\left[\frac{\xi\xi^\top}{\|\xi\|_2^2}\right] = \frac{\Omega^{1/2}}{\mathbf{Tr}\left(\Omega^{1/2}\right)},$$

which yields a very favourable convergence rate.

### 2.7.1 Gaussian Kaczmarz

Let $B = I$ and choose $\Sigma = I$ so that $S = \eta \sim N(0, I)$. Then (2.54) has the form

$$\boxed{x^{k+1} = x^k - \frac{\eta^\top (Ax^k - b)}{\|A^\top \eta\|_2^2} A^\top \eta} \tag{2.57}$$

which we call the *Gaussian Kaczmarz* (GK) method, for it is the analogous method to the Randomized Kaczmarz method in the discrete setting. Using the formulation (2.6), for instance, the GK method can be interpreted as

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \|x - x^*\|^2 \quad \text{subject to} \quad x = x^k + A^\top \eta y, \quad y \in \mathbb{R}.$$

Thus at each iteration, a random normal Gaussian vector $\eta$ is drawn and a search direction is formed by $A^\top \eta$. Then, starting from the previous iterate $x^k$, an exact line search is performed over this search direction so that the euclidean distance from the optimal is minimized.

### 2.7.2 Gaussian least-squares

Let $B = A^\top A$ and choose $S \sim N(0, \Sigma)$ with $\Sigma = AA^\top$. It will be convenient to write $S = A\eta$, where $\eta \sim N(0, I)$. Then method (2.54) then has the form

$$\boxed{x^{k+1} = x^k - \frac{\eta^\top A^\top (Ax^k - b)}{\|A\eta\|_2^2} \eta} \tag{2.58}$$

which we call the *Gauss-LS* method. This method has a natural interpretation through formulation (2.6) as

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 \quad \text{subject to} \quad x = x^k + y\eta, \quad y \in \mathbb{R}.$$

That is, starting from $x^k$, we take a step in a random (Gaussian) direction, then perform an exact line search over this direction that minimizes the least squares error. Thus the Gauss-LS method is the same as applying the Random Pursuit method [121] with exact line search to the Least-squares function.

### 2.7.3 Gaussian positive definite

When $A$ is positive definite, we achieve an accelerated Gaussian method. Let $B = A$ and choose $S = \eta \sim N(0, I)$. Method (2.54) then has the form

$$\boxed{x^{k+1} = x^k - \frac{\eta^\top (Ax^k - b)}{\|\eta\|_A^2} \eta} \tag{2.59}$$

which we call the *Gauss-pd* method.

Using formulation (2.6), the method can be interpreted as

$$x^{k+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ f(x) \overset{\text{def}}{=} \tfrac{1}{2}x^\top Ax - b^\top x \right\} \quad \text{subject to} \quad x = x^k + y\eta, \quad y \in \mathbb{R}.$$

That is, starting from $x^k$, we take a step in a random (Gaussian) direction, then perform an exact line search over this direction. Thus the Gauss-pd method is equivalent to applying the Random Pursuit method [121] with exact line search to $f(x)$.

All the Gaussian methods can be extended to block versions. We illustrate this by designing a Block Gauss-pd method where $S \in \mathbb{R}^{n\times q}$ has i.i.d. Gaussian normal entries and $B = A$. This results in the iterates

$$x^{k+1} = x^k - S(S^\top AS)^{-1}S^\top(Ax^k - b). \tag{2.60}$$

## 2.8 Numerical Experiments

We perform some preliminary numeric tests on consistent overdetermined linear systems and positive definite systems. Everything was coded and run in MATLAB R2014b. Let $\kappa_2 = \|A\|\|A^\dagger\|$ be the 2−norm condition number. In comparing different methods for solving overdetermined systems, we use the relative residual $\|Ax^k - b\|_2/\|b\|_2$, while for positive definite systems we use $\|x^k - x^*\|_A/\|x^*\|_A$ as a relative residual measure. We run each method until the relative residual is below $10^{-4} = 0.01\%$ or until 400 seconds in time is exceeded. We test this relatively low precision of 0.01% because our applications of interest (e.g. ridge regression in machine learning) only require a low precision.

Note that when $A$ has full column rank, the convergence of the relative residual implies the convergence of norm error $\|x - x^*\|$. Indeed, this follows from

$$\|Ax^k - b\|_2^2 = \left\langle A^\top A(x^k - x^*), x^k - x^* \right\rangle \geq \lambda_{\min}(A^\top A)\|x^k - x^*\|_2^2.$$

Consequently bringing the relative residual $\|Ax^k - b\|_2/\|b\|_2$ below 0.01% implies that

$$\|x^k - x^*\|_2 \leq \frac{\|b\|_2}{\sqrt{\lambda_{\min}(A^\top A)}} 10^{-4}.$$

When the solution $x^*$ and the right hand $b$ were not supplied by the data, we generated them as follows. The solution was generated using $x^* =$`rand`$(n, 1)$, that is, each entry of $x^*$ is selected uniformly at random from the interval $[0, 1]$. The right hand side $b$ was set to $b = Ax^*$. As the starting point we used $x_0 = 0 \in \mathbb{R}^n$ in all experiments.

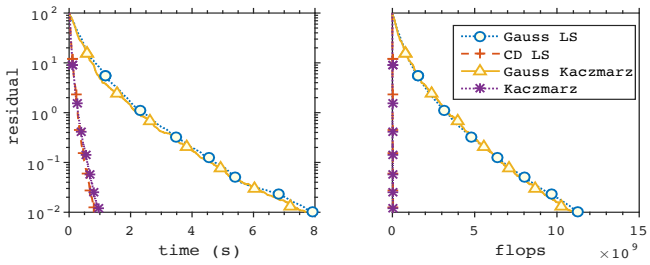
In each figure we plot the relative residual in percentage on the vertical axis, starting with 100%. For the horizontal axis, we use either wall-clock time measured using the `tic-toc` MATLAB function or the total number of floating point operations (*flops*). Specifically, we use an upper bound on the number of flops performed in each iteration as a proxy. For example, the number of flops required to compute the matrix-vector product $Av$ is bounded by $O(nnz(A))$ from above. This bound is tight when $v$ is dense. When $v$ is not dense, as is the case when $v = e_i$, we use an appropriately tight upper bound, such as $O(nnz(A_{i:}))$ when $v = e_i$.

In implementing the discrete sampling methods we used the convenient probability distributions (2.48).

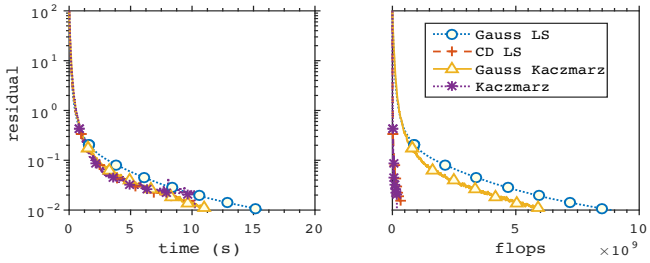
All tests were performed on a Desktop with 64bit quad-core Intel(R) Core(TM) i5-2400S CPU @2.50GHz with 6MB cache size with a Scientific Linux release 6.4 (Carbon) operating system.

### 2.8.1 Overdetermined linear systems

First we compare the methods Gauss-LS (2.58) , CD-LS (2.28) , Gauss-Kaczmarz (2.57) and RK (2.16) methods on synthetic linear systems generated with the matrix functions `rand` and `sprandn`, see Figure 2.2. The `rand`$(m, n)$ function returns a $m$-by-$n$ matrix where each entry is a random variable selected uniformly at random from the interval $[0, 1]$. The high iteration cost of the Gaussian methods resulted in poor performance on the dense problem generated using `rand` in Figure 2.2a. In Figure 2.2b we compare the methods on a sparse linear system




**(a)** *rand*



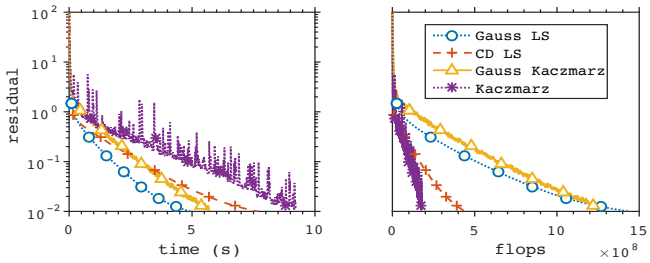

**(b)** *sprandn*

**Figure 2.2:** *The performance of the Gauss-LS, CD-LS, Gauss-Kaczmarz and RK methods on synthetic MATLAB generated problems (a)* `rand`$(n, m)$ *with* $(m; n) = (1000, 500)$ *(b)* `sprandn`$(m, n,$`density,rc`$)$ *with* $(m; n) = (1000, 500)$, `density`$= 1/\log(nm)$ *and* `rc`$= 1/\sqrt{mn}$. *In both experiments dense solutions were generated with* $x^* =$`rand`$(n, 1)$ *and* $b = Ax^*$.

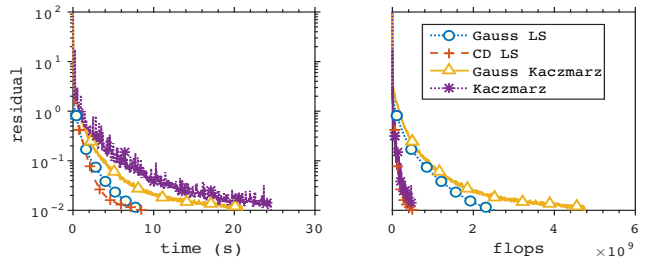
generated using the MATLAB sparse random matrix function `sprandn`$(m, n,$`density,rc`$)$, where `density` is the percentage of nonzero entries and `rc` is the reciprocal of the condition number. The `sprandn`$(m, n,$`density,rc`$)$ returns a random $m$-by-$n$ sparse matrix with approximately `density` $\times m \times n$ normally distributed nonzero entries. Please consult `http://uk.mathworks.com/help/matlab/ref/sprandn.html` for more details on `sprandn`. On this sparse problem the Gaussian methods are more efficient, and converge at a similar rate in time to the discrete sampling methods.

In Figure 2.3 we test two overdetermined linear systems taken from the the Matrix Market collection [11]. The collection also provides the right-hand side of the linear system. Both of these systems are very well conditioned, but do not have full column rank, thus Theorem 16 does not apply. The four methods have a similar performance on Figure 2.3a, while the Gauss-LS and CD-LS method converge faster on 2.3b as compared with the Gauss-Kaczmarz and Kaczmarz methods in terms of time taken. In terms of number of flops, the CD-LS and Kaczmarz method outperform the Gauss-LS and Gauss-Kaczmarz methods.

Finally, we test two problems, the `SUSY` problem and the `covtype.binary` problem, from the library of support vector machine problems LIBSVM [17]. These problems do not form consistent linear systems, thus only the Gauss-LS and CD-LS methods are applicable, see Figure 2.4. This is equivalent to applying the Gauss-pd and CD-pd to the least squares system $A^\top Ax = A^\top b$,

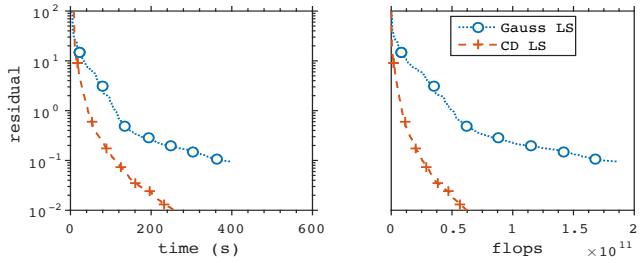



**(a)** *illc1033*

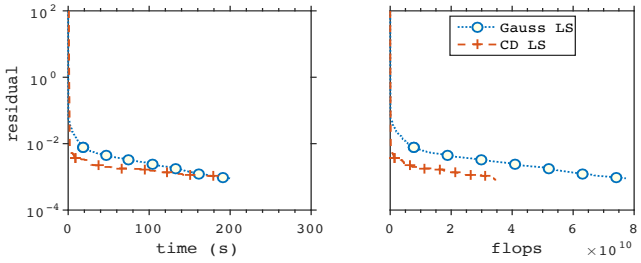



**(b)** *well1033*

**Figure 2.3:** *The performance of the Gauss-LS, CD-LS, Gauss-Kaczmarz and RK methods on linear systems (a)* `well1033` *where* $(m; n) = (1850, 750)$, $nnz = 8758$ *and* $\kappa_2 = 1.8$ *(b)* `illc1033` *where* $(m; n) = (1033; 320)$, $nnz = 4732$ *and* $\kappa_2 = 2.1$, *from the Matrix Market [11].*




**(a)** *SUSY*




**(b)** *covtype-libsvm-binary*

**Figure 2.4:** *The performance of Gauss-LS and CD-LS methods on two LIBSVM test problems: (a)* `SUSY`*:* $(m; n) = (5 \times 10^6; 18)$ *(b)* `covtype.binary`*:* $(m; n) = (581,012; 54)$.

which is always consistent.

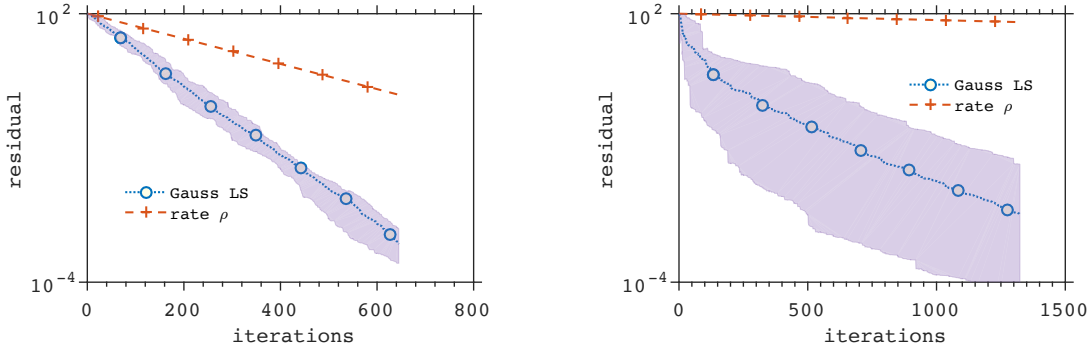
Despite the higher iteration cost of the Gaussian methods, their performance, in terms of the wall-clock time, is comparable to performance of the discrete methods when the system matrix is sparse.

### 2.8.2 Bound for Gaussian convergence

Now we compare the error over the number iterations of the Gauss-LS method to theoretical rate of convergence given by the bound (2.56). For the Gauss-LS method (2.56) becomes

$$1 - \frac{1}{n} \leq \rho \leq 1 - \frac{2}{\pi}\lambda_{\min}\left(\frac{A^\top A}{\|A\|_F^2}\right).$$

In Figures 2.5a and 2.5b we compare the empirical and theoretical bound on a random Gaussian matrix and the `liver-disorders` problem [17]. Furthermore, we ran the Gauss-LS method 100 times and plot as a shaded region the outcomes within the 95% and 5% quantiles. These tests indicate that the bound is tight for well conditioned problems, such as Figure 2.5a in which the system matrix has a condition number equal to 1.94. While in Figure 2.5b the system matrix has a condition number of 41.70 and there is some much more slack between the empirical convergence and the theoretical bound.




**(a)** $\texttt{rand}(n, m)$ **(b)** `liver-disorders`

**Figure 2.5:** *A comparison between the Gauss-LS method and the theoretical bound* $\rho_{theo} \overset{def}{=} 1 - \lambda_{\min}(A^\top A)/\|A\|_F^2$ *on (a)* $\texttt{rand}(n, m)$ *with* $(m; n) = (500, 50), \kappa_2 = 1.94$ *and a dense solution generated with* $x^* = \texttt{rand}(n, 1)$ *(b)* `liver-disorders` *with* $(m; n) = (345, 6)$ *and* $\kappa_2 = 41.70$.

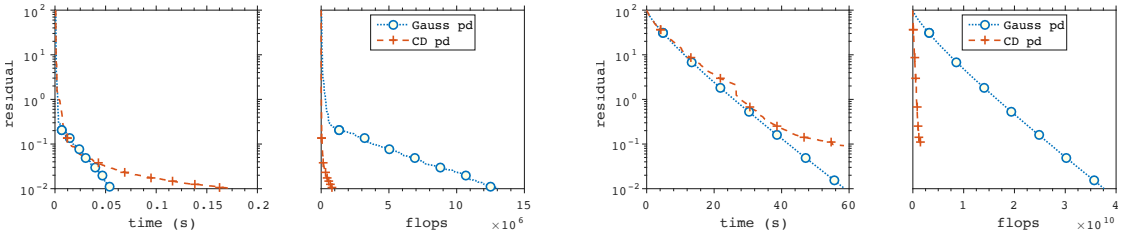
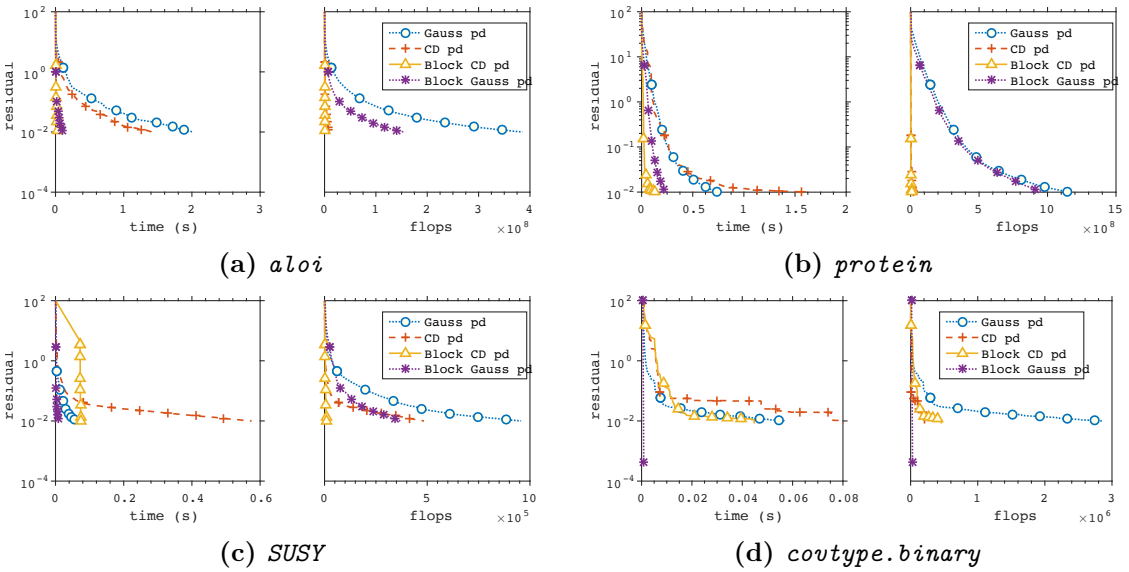



**Figure 2.6:** *Synthetic MATLAB generated problem. The Gaussian methods are more efficient on sparse matrices. LEFT: The* `Hilbert Matrix` *with* $n = 100$ *and condition number* $\|A\|\|A^{-1}\| \approx 0.001133 \times e^{349}$. *RIGHT: Sparse random matrix* $A =$ `sprandsym` *(*$n$, `density`, `rc`, `type`*) with* $n = 1000$, `density`$= 1/\log(n^2)$ *and* `rc` $= 1/n = 0.001$. *Dense solution generated with* $x^* =$`rand`$(n, 1)$.

### 2.8.3 Positive definite

First we compare the two methods Gauss-pd (2.59) and CD-pd (2.20) on synthetic data in Figure 2.6. Using the MATLAB function `hilbert` we generate the positive definite Hilbert matrix which has a very high condition number, see Figure 2.6(LEFT). Indeed, the $100 \times 100$ Hilbert matrix we tested has a condition number of approximately $0.001133 \times e^{349}$! Both methods converge slowly and, despite the dense system matrix, the Gauss-pd method has a similar performance to CD-pd. In Figure (2.6)(RIGHT) we compare the two methods on a system generated by the MATLAB function `sprandsym` ($m$, $n$, `density`, `rc`, `type`), where `density` is the percentage of nonzero entries, `rc` is the reciprocal of the condition number and `type=1` returns a positive definite matrix. The Gauss-pd and the CD-pd method have a similar performance in terms of wall clock time on this sparse problem.

To appraise the performance gain in using block variants, we perform tests using two block variants: the Randomized Newton method (2.25), which we will now refer to as the Block CD-pd method, and the Block Gauss-pd method (2.60). We set the block size to $q = \sqrt{n}$ in both methods. To solve the $q \times q$ system required in the block methods, we use MATLAB's built-in direct solver, sometimes referred to as "back-slash".




**(a)** *aloi* **(b)** *protein*

**(c)** *SUSY* **(d)** *covtype.binary*

**Figure 2.7:** *The performance of Gauss-pd and CD-pd methods on four ridge regression problems: (a)* `aloi`*:* $(m;n) = (108,000;128)$ *(b)* `protein`*:* $(m;n) = (17,766;357)$ *(c)* `SUSY`*:* $(m;n) = (5 \times 10^6;18)$ *(d)* `covtype.binary`*:* $(m;n) = (581,012;54)$.

Next we test the Newton system $\nabla^2 f(w_0)x = -\nabla f(w_0)$, arising from four ridge-regression problems of the form

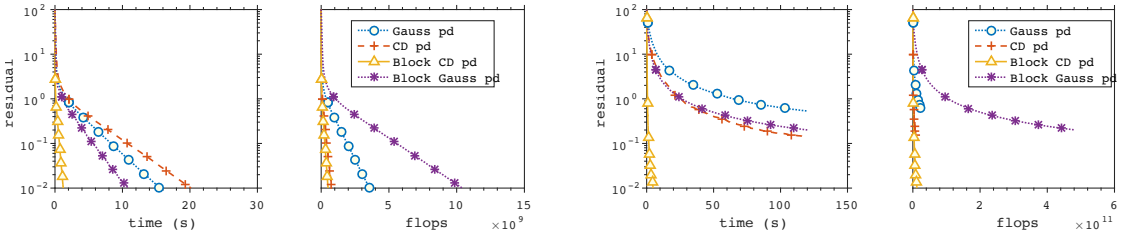
$$\min_{w\in\mathbb{R}^n} f(w) \stackrel{\text{def}}{=} \tfrac{1}{2}\|Aw-b\|_2^2 + \tfrac{\lambda}{2}\|w\|_2^2, \tag{2.61}$$

using data from LIBSVM [17]. In particular, we set $w_0 = 0$ and use $\lambda = 1$ as the regularization parameter, whence $\nabla f(w_0) = A^\top b$ and $\nabla^2 f(w_0) = A^\top A + I$.

In terms of wall clock time, the Gauss-pd method converged faster on all problems accept the `aloi` problem as compared with CD-pd. The two Block methods had a comparable performance on the `aloi` and the `protein` problem. The Block Gauss-pd method converged in one iteration on `covtype.binary` and was the fastest method on the `SUSY` problem.

We now compare the methods on two positive definite matrices from the Matrix Market collection [11], see Figure 2.8. The right-hand side was not supplied by the data set, and thus we generated $b$ using `rand(n,1)`. The Block CD-pd method converged much faster on both problems. The lower condition number ($\kappa_2 = 12$) of the `gr_30_30-rsa` problem resulted in fast convergence of all methods, see Figure 2.8a. While the high condition number ($\kappa_2 = 4.3 \cdot 10^4$) of the `bcsstk18` problem, resulted in a slow convergence for all methods, see Figure 2.8b.

Despite the clear advantage of using a block variant, applying a block method that uses a direct solver can be infeasible on very ill-conditioned problems. As an example, applying the Block CD-pd to the Hilbert system, and using MATLAB back-slash solver to solve the inner $q \times q$ systems, resulted in large numerical inaccuracies, and ultimately, prevented the method from converging. This occurred because the submatrices of the Hilbert matrix are also very ill-conditioned.




(a) `gr_30_30-rsa` (b) `bcsstk18`

**Figure 2.8:** *The performance of the Gauss-pd, CD-pd and the Block CD-pd methods on two linear systems from the MatrixMarket (a)* `gr_30_30-rsa` *with* $n = 900$, $nnz = 4322$ *(*`density`$= 0.53\%$*) and* $\kappa_2 = 12$. *(b)* `bcsstk18` *with* $n = 11948$, $nnz = 80519$ *(*`density`$= 0.1\%$*) and* $\kappa_2 = 4.3 \cdot 10^{10}$.

| data set | $\rho_c$ | $\rho^*$ | $1 - 1/n$ | optimized time(s) |
|---|---|---|---|---|
| `rand`(50,50) | $1 - 2 \cdot 10^{-6}$ | $1 - 3.05 \cdot 10^{-6}$ | $1 - 2.10^{-2}$ | 1.076 |
| `mushrooms-ridge` | $1 - 5.86 \cdot 10^{-6}$ | $1 - 7.15 \cdot 10^{-6}$ | $1 - 8.93 \cdot 10^{-3}$ | 4.632 |
| `aloi-ridge` | $1 - 2.17 \cdot 10^{-7}$ | $1 - 1.26 \cdot 10^{-4}$ | $1 - 7.81 \cdot 10^{-3}$ | 7.401 |
| `liver-disorders-ridge` | $1 - 5.16 \cdot 10^{-4}$ | $1 - 8.25 \cdot 10^{-3}$ | $1 - 1.67 \cdot 10^{-1}$ | 0.413 |
| `covtype.binary-ridge` | $1 - 7.57 \cdot 10^{-14}$ | $1 - 1.48 \cdot 10^{-6}$ | $1 - 1.85 \cdot 10^{-2}$ | 1.449 |

**Table 2.2:** *Optimizing the convergence rate for CD-pd.*

### 2.8.4 Comparison between optimized and convenient probabilities

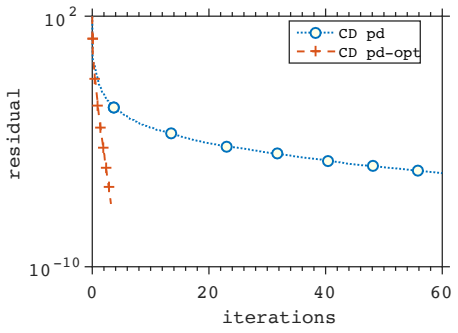
We compare the practical performance of using the convenient probabilities (2.48) against using the optimized probabilities by solving (2.47). We solved (2.47) using the disciplined convex programming solver `cvx` [53] for MATLAB.

In Table 2.2 we compare the different convergence rates for the CD-pd method, where $\rho_c$ is the convenient convergence rate (2.50), $\rho^*$ the optimized convergence rate, $(1 - 1/n)$ is the lower bound, and in the final "optimized time(s)" column the time taken to compute $\rho^*$. In Figure 2.9, we compare the empirical convergence of the CD-pd method when using the convenient probabilities (2.48) and CD-pd-opt, the CD-pd method with the optimized probabilities. We tested the two methods on four ridge regression problems and a synthetic positive definite system which is the square of a uniform random matrix: $A = \bar{A}^\top \bar{A}$ where $\bar{A} =$`rand`$(50)$.

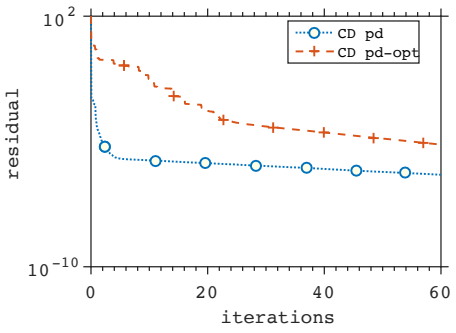
We ran each method for 60 seconds.

In most cases using the optimized probabilities results in a much faster convergence, see Figures 2.9a, 2.9c, 2.9d and 2.9e. In particular, the 7.401 seconds spent calculating the optimal probabilities for `aloi` paid off with a convergence that was 55 seconds faster. The `mushrooms` problem was insensitive to the choice of probabilities 2.9d. Finally despite $\rho^*$ being much less than $\rho_c$ on `covtype`, see Table 2.2, using optimized probabilities resulted in an initially slower method, though CD-pd-opt eventually catches up as CD-pd stagnates, see Figure 2.9b.

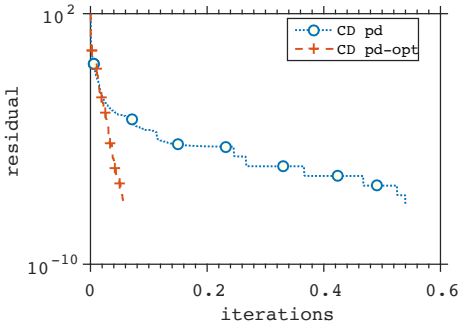
In Table 2.3 we compare the different convergence rates for the RK method. In Figure 2.10, we then compare the empirical convergence of the RK method when using the convenient probabilities (2.48) and RK-opt, the RK method with the optimized probabilities by solving (2.47). The rates $\rho^*$ and $\rho_c$ for the `rand`(500,100) problem are similar, and accordingly, both the convenient and optimized variant converge at a similar rate in practice, see Figure 2.10b. While the difference in the rates $\rho^*$ and $\rho_c$ for the `liver-disorders` is more pronounced, and in this case, the 0.83 seconds invested in obtaining the optimized probability distribution paid



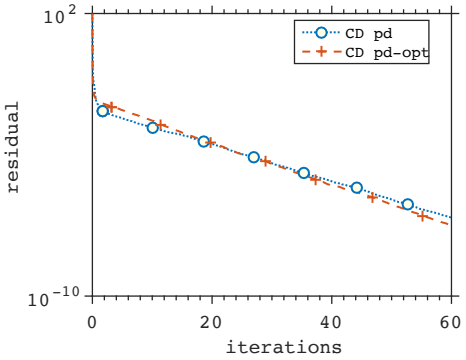

**(a)** *aloi*



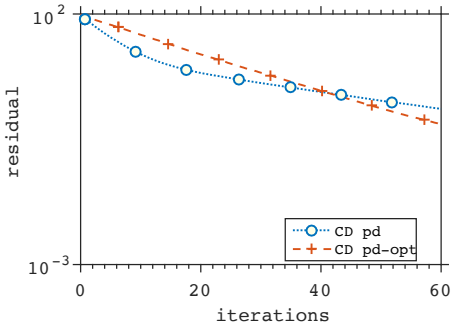

**(b)** *covtype.libsvm.binary*




**(c)** *liver-disorders-ridge*




**(d)** *mushrooms-ridge*




**(e)** $50X50$ *synthetic positive definite*

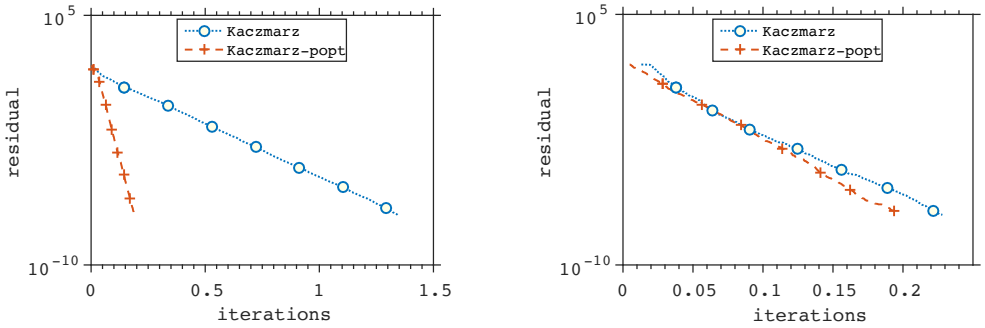
**Figure 2.9:** *The performance of CD-pd and optimized CD-pd methods on (a)* `aloi`*:* $(m;n) = (108,000;128)$ *(b)* `covtype.binary`*:* $(m;n) = (581,012;54)$ *(c)* `liver-disorders`*:* $(m;n) = (345,6)$ *(c)*`mushrooms`*:* $(m;n) = (8124,112)$ *(d)* $A = \bar{A}^\top \bar{A}$ *where* $\bar{A} =$`rand`$(50)$.

off in practice, as the optimized method converged 1.25 seconds before the RK method with the convenient probability distribution, see Figure 2.10a.

We conclude from these tests that the choice of the probability distribution can greatly affect the performance of the method. Hence, it is worthwhile to develop approximate solutions to (2.46).

| data set | $\rho_c$ | $\rho^*$ | $1-1/n$ | optimized time(s) |
|---|---|---|---|---|
| `rand`(500,100) | $1-3.37\cdot10^{-3}$ | $1-4.27\cdot10^{-3}$ | $1-1\cdot10^{-2}$ | 33.121 |
| `liver-disorders` | $1-5.16\cdot10^{-4}$ | $1-4.04\cdot10^{-3}$ | $1-1.67\cdot10^{-1}$ | 0.8316 |

**Table 2.3:** *Optimizing the convergence rate for randomized Kaczmarz.*




**(a)** *liver-disorders-popt-k* **(b)** *rand(500,100)*

**Figure 2.10:** *The performance of the Kaczmarz and optimized Kaczmarz methods on (a)* `liver-disorders`*:* $(m;n)=(345,6)$ *(b)* `rand`*(500,100)*

### 2.8.5 Conclusion of numeric experiments

We now summarize the findings of our numeric experiments.

- Consistently across our experiments, in terms of number of flops taken to reach a desired precision, the three discrete sampling methods CD-LS, CD-pd and Kaczmarz are the most efficient. That is, the Gaussian methods almost always require more flops to reach a solution with the same precision as their discrete sampling counterparts. This is due to the expensive matrix-vector product required by the Gaussian methods.
- In terms of wall-clock time, the Gaussian methods Guass-LS, Gauss-pd and Gauss Kaczmarz are competitive as compared to the discrete sampling methods. This occurred because MATLAB performs automatic multi-threading when calculating matrix-vector products, which was the bottleneck cost in the Gaussian methods. As our machine has four cores, this explains some of the difference observed when measuring performance in terms of number of flops and wall clock time.
- In terms of both time taken and flops, the block variants proved to be significantly more efficient.
- Using the optimized probabilities (2.47) for the discrete sampling methods RK and CD-pd can result in significant speed-ups, as compared to RK and CD-pd using the convenient probabilities (2.48). So much so, that the time spent solving the SDP (2.47) using `cvx` [53] often paid off. We can draw two interesting conclusions from this: (1) using convenient probabilities (2.48) is not the best choice, but simply, the choice that provides easily interpretable convergence rates, (2) it is worth further investigating the use of optimization probabilities, for instance, one should investigate it is possible to obtain affordable approximate solutions to (2.47).

## 2.9 Summary

In this chapter we presented a unifying framework for the randomized Kaczmarz method, randomized Newton method, randomized coordinate descent method and random Gaussian pursuit. Not only can we recover these methods by selecting appropriately the parameters $S$ and $B$, but also, we can analyze them and their block variants through a single Theorem 16. Furthermore, we obtain a new lower bound for all these methods in Theorem 15, and in the discrete case, recover all known convergence rates expressed in terms of the scaled condition number in Theorem 19.

Theorem 19 also suggests a preconditioning strategy. Developing preconditioning methods are important for reaching a higher precision solution on ill-conditioned problems. For as we have seen in the numerical experiments, the randomized methods struggle to bring the solution within $10^{-2}\%$ relative residual when the matrix is ill-conditioned.

This is also a framework on which randomized methods for linear systems can be designed. As an example, we have designed a new block variant of RK, a new Gaussian Kaczmarz method and a new Gaussian block method for positive definite systems. Furthermore, the flexibility of our framework and the general convergence Theorems 16 and 15 allows one to tailor the probability distribution of $S$ to a particular problem class. For instance, other continuous distributions such uniform, or other discrete distributions such Poisson might be more suited to a particular class of problems.

Numeric tests reveal that the new Gaussian methods designed for overdetermined systems are competitive on sparse problems, as compared with the Kaczmarz and CD-LS methods. The Gauss-pd also proved competitive as compared with CD-pd on all tests. Though, when applicable, the combined efficiency of using a direct solver and an iterative procedure, such as in Block CD-pd method, proved the most efficient.

The work opens up many possible future venues of research. Including investigating accelerated convergence rates through preconditioning strategies based on Theorem 19 or by obtaining approximate optimized probability distributions (2.47).

### Acknowledgments

I would like to thank Prof. Sandy Davie for useful discussions relating to Lemma 21, and Prof. Joel Tropp for help with formulating and proving Lemma 20.

## 2.10 Appendix: A Bound on the Expected Gaussian Projection Matrix

We now bound the covariance of a random Gaussian vector projected onto the sphere. This bound is used to study the complexity of Gaussian methods in Section 2.7.

**Lemma 20.** *Let $D \in \mathbb{R}^{n\times n}$ be a positive definite diagonal matrix, $U \in \mathbb{R}^{n\times n}$ an orthogonal matrix and $\Omega = UDU^\top$. If $u \sim N(0, D)$ and $\xi \sim N(0, \Omega)$ then*

$$\mathbf{E}\left[\frac{\xi\xi^\top}{\xi^\top\xi}\right] = U\mathbf{E}\left[\frac{uu^\top}{u^\top u}\right]U^\top, \tag{2.62}$$

*and*

$$\mathbf{E}\left[\frac{\xi\xi^\top}{\xi^\top\xi}\right] \succeq \frac{2}{\pi}\frac{\Omega}{\mathbf{Tr}\left(\Omega\right)}. \tag{2.63}$$

*Proof.* Let us write $S(\xi)$ for the random vector $\xi/\|\xi\|_2$ (if $\xi = 0$, we set $S(\xi) = 0$). Using this notation, we can write

$$\mathbf{E}\left[\xi(\xi^\top\xi)^{-1}\xi^\top\right] = \mathbf{E}\left[S(\xi)(S(\xi))^\top\right] = \mathbf{Cov}\left[S(\xi)\right],$$

where the last identity follows since $\mathbf{E}\left[S(\xi)\right] = 0$, which in turn holds as the Gaussian distribution is centrally symmetric. As $\xi = Uu$, note that

$$S(u) = \frac{U^\top\xi}{\|U^\top\xi\|_2} = \frac{U^\top\xi}{\|\xi\|_2} = U^\top S(\xi).$$

Left multiplying both sides by $U$ we obtain $US(u) = S(\xi)$, from which we obtain

$$\mathbf{Cov}\left[S(\xi)\right] = U\mathbf{Cov}\left[S(u)\right]U^\top,$$

which is equivalent to (2.62).

To prove[2] (2.63), note first that $M \overset{\text{def}}{=} \mathbf{E}\left[uu^\top/u^\top u\right]$ is a diagonal matrix. One can verify this by direct calculation (informally, this holds because the entries of $u$ are independent and centrally symmetric). The $i$th diagonal entry is given by

$$M_{ii} = \mathbf{E}\left[\frac{u_i^2}{\sum_{j=1}^n u_j^2}\right].$$

As the map $(x, y) \to x^2/y$ is convex on the positive orthant, we can apply Jensen's inequality, which gives

$$\mathbf{E}\left[\frac{u_i^2}{\sum_{j=1}^n u_j^2}\right] \geq \frac{\left(\mathbf{E}\left[|u_i|\right]\right)^2}{\sum_{j=1}^n \mathbf{E}\left[u_j^2\right]} = \frac{2}{\pi}\frac{D_{ii}}{\mathbf{Tr}\left(D\right)},$$

which concludes the proof. □

[2] A version of Lemma 20 was conjectured in the original draft of the paper on which this chapter is based. Prof. Joel Tropp provided this formulation and the remainder of this proof.

## 2.11 Appendix: Expected Gaussian Projection Matrix in 2D

**Lemma 21.** *Let $\xi \sim N(0,\Omega)$ and $\Omega \in \mathbb{R}^{2\times 2}$ be a positive definite matrix, then*

$$\mathbf{E}\left[\frac{\xi\xi^\top}{\xi^\top\xi}\right] = \frac{\Omega^{1/2}}{\mathbf{Tr}\left(\Omega^{1/2}\right)}. \tag{2.64}$$

*Proof.* Let $\Sigma = UDU^\top$ and $u \sim N(0,D)$. Given (2.62) it suffices to show that

$$\mathbf{Cov}\left[S(u)\right] = \frac{D^{1/2}}{\mathbf{Tr}\left(D^{1/2}\right)}, \tag{2.65}$$

which we will now prove.

Let $\sigma_x^2$ and $\sigma_y^2$ be the two diagonal elements of $D$. First, suppose that $\sigma_x = \sigma_y$. Then $u = \sigma_x\eta$ where $\eta \sim N(0,I)$ and

$$\mathbf{E}\left[\frac{uu^\top}{u^\top u}\right] = \frac{\sigma_x^2}{\sigma_x^2}\mathbf{E}\left[\frac{\eta\eta^\top}{\eta^\top\eta}\right] = \frac{1}{n}I = \frac{D^{1/2}}{\mathbf{Tr}\left(D^{1/2}\right)}.$$

Now suppose that $\sigma_x \neq \sigma_y$. We calculate the diagonal terms of the covariance matrix by integrating

$$\mathbf{E}\left[\frac{u_1^2}{u_1^2+u_2^2}\right] = \frac{1}{2\pi\sigma_x\sigma_y}\int_{\mathbb{R}^2}\frac{x^2}{x^2+y^2}e^{-\frac{1}{2}\left(x^2/\sigma_x^2+y^2/\sigma_y^2\right)}dxdy.$$

Using polar coordinates $x = R\cos(\theta)$ and $y = R\sin(\theta)$ we have

$$\int_{\mathbb{R}^2}\frac{x^2}{x^2+y^2}e^{-\frac{1}{2}\left(x^2/\sigma_x^2+y^2/\sigma_y^2\right)}dxdy = \int_0^{2\pi}\int_0^{\infty} R\cos^2(\theta)e^{-\frac{R^2}{2}C(\theta)}dRd\theta, \tag{2.66}$$

where $C(\theta) \overset{\text{def}}{=} \left(\cos(\theta)^2/\sigma_x^2 + \sin(\theta)^2/\sigma_y^2\right)$. Note that

$$\int_0^{\infty} Re^{-\frac{C(\theta)R^2}{2}}dR = -\frac{1}{C(\theta)}e^{-\frac{C(\theta)R^2}{2}}\Big|_0^{\infty} = \frac{1}{C(\theta)}. \tag{2.67}$$

This applied in (2.66) gives

$$\mathbf{E}\left[\frac{u_1^2}{u_1^2+u_2^2}\right] = \frac{1}{2\pi\sigma_x\sigma_y}\int_0^{2\pi}\frac{\cos^2(\theta)}{\cos(\theta)^2/\sigma_x^2+\sin(\theta)^2/\sigma_y^2}d\theta = \frac{b}{\pi}\int_0^{\pi}\frac{\cos^2(\theta)}{\cos^2(\theta)+b^2\sin^2(\theta)}d\theta,$$

where $b = \sigma_x/\sigma_y$. Multiplying the numerator and denominator of the integrand by $\sec^4(x)$ gives the integral

$$\mathbf{E}\left[\frac{u_1^2}{u_1^2+u_2^2}\right] = \frac{b}{\pi}\int_0^{\pi}\frac{\sec^2(\theta)}{\sec(\theta)^2\left(1+b^2\tan^2(\theta)\right)}d\theta.$$

Substituting $v = \tan(\theta)$ so that $v^2+1 = \sec^2(\theta)$, $dv = \sec^2(\theta)d\theta$ and using the partial fractions

$$\frac{1}{(v^2+1)(1+b^2v^2)} = \frac{1}{1-b^2}\left(\frac{1}{v^2+1} - \frac{b^2}{b^2v^2+1}\right),$$

gives the integral

$$\begin{aligned}\int \frac{dv}{(v^2+1)\,(1+b^2v^2)} &= \frac{1}{1-b^2}\left(\arctan(v) - b\arctan(bv)\right)\\ &= \frac{1}{1-b^2}\left(\theta - b\arctan(b\tan(\theta))\right). \end{aligned} \tag{2.68}$$

To apply the limits of integration, we must take care because of the singularity at $\theta = \pi/2$. For this, consider the limits

$$\lim_{\theta\to(\pi/2)^-} \arctan(b\tan(\theta)) = \frac{\pi}{2}, \qquad \lim_{\theta\to(\pi/2)^+} \arctan(b\tan(\theta)) = -\frac{\pi}{2}.$$

Using this to evaluate (2.68) on the limits of the interval $[0,\, \pi/2]$ gives

$$\lim_{t\to(\pi/2)^-} \frac{1}{1-b^2}\left(\theta - b\arctan(b\tan(\theta))\right)\Big|_0^t = \frac{1}{1-b^2}\frac{\pi}{2}(1-b) = \frac{\pi}{2(1+b)}.$$

Applying a similar argument for calculating the limits from $\pi/2^+$ to $\pi$, we find

$$\mathbf{E}\left[\frac{u_1^2}{u_1^2+u_2^2}\right] = \frac{2b}{\pi}\frac{\pi}{2(1+b)} = \frac{\sigma_x}{\sigma_y+\sigma_x}.$$

Repeating the same steps with $x$ swapped for $y$ we obtain the other diagonal element, which concludes the proof of (2.65). □

CHAPTER 3

# Stochastic Dual Ascent for Finding the Projection of a Vector onto a Linear System

*Dyfal donc a dyr y garreg.*
Tapping persistently breaks the stone.

Welsh proverb

## 3.1 Introduction

In this chapter we consider the more general problem of finding the projection of a given vector onto the solution space of a linear system. This projection problem includes the problem of determining the least norm solution of a linear system (when the given vector is the zero vector). To solve this projection problem, we develop a new randomized iterative algorithm—*stochastic dual ascent (SDA)*. The method is dual in nature: with the dual being a non-strongly concave quadratic maximization problem without constraints.

By mapping our dual iterates to primal iterates, we uncover that the SDA method is a dual version of the sketch-and-project method (1.3). We then proceed to strengthen our convergence results established in Chapter 2. First, we do away with the assumption that the system matrix has full column rank that was required to establish convergence through Theorem 15 and 16 and consider any matrix and consistent linear system. In this more general setting we show that the primal iterates still converge linearly with a convergence rate that is at least as small as the convergence rate established in Chapter 2.

Furthermore we give a formula and a tighter lower bound for the convergence rate. We also prove that the same rate of convergence applies to dual function values, primal function values and the duality gap. Unlike traditional iterative methods, SDA converges under virtually no additional assumptions on the system (e.g., rank, diagonal dominance) beyond consistency. In fact, our lower bound improves as the rank of the system matrix drops.

When our method specializes to a known algorithm, we either recover the best known rates, or improve upon them. Finally, we show that the framework can be applied to the

distributed average consensus problem to obtain an array of new algorithms. The randomized gossip algorithm arises as a special case [13, 90].

## 3.2 Contributions and Overview

### 3.2.1 The problem:

$$Ax = b, \tag{3.1}$$

where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. We shall only assume that the system is *consistent*, that is, that there exists $x^*$ for which $Ax^* = b$. Note that we make no assumptions on $n$ or $m$ and all the configurations $m < n$, $m = n$ and $m > n$ are allowed. While we assume the existence of a solution, we do not assume uniqueness. In situations with multiple solutions, one is often interested in finding a solution with specific properties. For instance, in compressed sensing and sparse optimization, one is interested in finding the least $\ell_1$-norm, or the least $\ell_0$-norm (sparsest) solution.

In this chapter we shall focus on the canonical problem of finding the solution of (3.1) closest, with respect to a Euclidean distance, to a given vector $c \in \mathbb{R}^n$:

$$\begin{array}{ll} \text{minimize} & P(x) \stackrel{\text{def}}{=} \frac{1}{2}\|x-c\|_B^2 \\ \text{subject to} & Ax = b \\ & x \in \mathbb{R}^n. \end{array} \tag{3.2}$$

where $B$ is an $n \times n$ symmetric positive definite matrix and $\|x\|_B \stackrel{\text{def}}{=} \sqrt{x^\top Bx}$. By $x^*$ we denote the (necessarily) unique solution of (3.2). Of key importance in this chapter is the *dual problem*[1] to (3.2), namely

$$\begin{array}{ll} \text{maximize} & D(y) \stackrel{\text{def}}{=} (b-Ac)^\top y - \frac{1}{2}\|A^\top y\|_{B^{-1}}^2 \\ \text{subject to} & y \in \mathbb{R}^m. \end{array} \tag{3.3}$$

Due to the consistency assumption, strong duality holds and we have $P(x^*) = D(y^*)$, where $y^*$ is any dual optimal solution.

### 3.2.2 A new family of stochastic optimization algorithms

We propose to solve (3.2) via a new method operating in the dual (3.3), which we call *stochastic dual ascent* (SDA). The iterates of SDA are of the form

$$y^{k+1} = y^k + S\lambda^k, \tag{3.4}$$

where $S$ is a random matrix with $m$ rows drawn in each iteration independently from a pre-specified distribution $\mathcal{D}$, which should be seen as a parameter of the method. In fact, by varying $\mathcal{D}$, SDA should be seen as a family of algorithms indexed by $\mathcal{D}$, the choice of which leads to specific algorithms in this family. By performing steps of the form (3.4), we are moving in the range space of the random matrix $S$. A key feature of SDA enabling us to prove strong convergence results despite the fact that the dual objective is in general not strongly concave is the way in which the "stepsize" parameter $\lambda^k$ is chosen: we choose $\lambda^k$ to be the *least-norm* vector for which $D(y^k + S\lambda)$

[1] Technically, this is both the Lagrangian and Fenchel dual of (3.2).

is maximized in $\lambda$. Plugging this $\lambda^k$ into (3.4), we obtain the SDA method:

$$\boxed{y^{k+1} = y^k + S\left(S^\top AB^{-1}A^\top S\right)^\dagger S^\top\left(b - A\left(c + B^{-1}A^\top y^k\right)\right)} \tag{3.5}$$

To the best of our knowledge, a randomized optimization algorithm with iterates of the *general* form (3.4) was not considered nor analyzed before. In the special case when $S$ is chosen to be a random unit coordinate vector, SDA specializes to the *randomized coordinate descent method*, first analyzed by Leventhal and Lewis [68]. In the special case when $S$ is chosen as a random column submatrix of the $m \times m$ identity matrix, SDA specializes to the *randomized Newton method* of Qu, Fercoq, Richtárik and Takáč [101].

With the dual iterates $\{y^k\}$ we associate a sequence of primal iterates $\{x^k\}$ as follows:

$$x^k \stackrel{\text{def}}{=} c + B^{-1}A^\top y^k. \tag{3.6}$$

In combination with (3.5), this yields the primal iterative process

$$\boxed{x^{k+1} = x^k - B^{-1}A^\top S\left(S^\top AB^{-1}A^\top S\right)^\dagger S^\top\left(Ax^k - b\right)} \tag{3.7}$$

Optimality conditions (see Section 3.3.1) imply that if $y^*$ is any dual optimal point, then $c + B^{-1}A^\top y^*$ is necessarily primal optimal and hence equal to $x^*$, the optimal solution of (3.2). Moreover, we have the following useful and insightful correspondence between the quality of the primal and dual iterates (see Proposition 27):

$$D(y^*) - D(y^k) = \tfrac{1}{2}\|x^k - x^*\|_B^2. \tag{3.8}$$

Hence, *dual convergence in function values is equivalent to primal convergence in iterates.*

This work belongs to a growing literature on randomized methods for various problems appearing in linear algebra, optimization and computer science. In particular, relevant methods include sketching algorithms, randomized Kaczmarz, stochastic gradient descent and their variants [125, 82, 32, 88, 139, 84, 102, 114, 126, 61, 106, 65, 134, 27, 66, 137, 85, 23, 83, 76, 51, 91, 73] and randomized coordinate and subspace type methods and their variants [68, 59, 116, 86, 132, 14, 105, 87, 107, 127, 80, 128, 81, 103, 39, 118, 36, 37, 67, 100, 38, 98, 99, 136, 101, 135, 117, 72, 22, 51].

### 3.2.3 The main results

We now describe two complexity theorems which form the core theoretical contribution of this chapter. The results hold for a wide family of distributions $\mathcal{D}$, which we describe next.

**Weak assumption on $\mathcal{D}$.** In our analysis, we only impose a very weak assumption on $\mathcal{D}$. In particular, we only assume that the $m \times m$ matrix

$$H \stackrel{\text{def}}{=} \mathbf{E}_{S\sim\mathcal{D}}\left[S\left(S^\top AB^{-1}A^\top S\right)^\dagger S^\top\right] \tag{3.9}$$

is well defined and nonsingular.[2] Hence, we do not assume that $S$ is chosen from any particular random matrix ensemble. This makes it possible for practitioners to choose the best distribution specific to a particular application.

We cast the first complexity result in terms of the primal iterates since solving (3.2) is our main focus in this work. Let $\mathbf{Range}(M), \mathbf{Rank}(M)$ and $\lambda_{\min}^{+}(M)$ denote the range space, rank and the smallest nonzero eigenvalue of $M$, respectively.

**Theorem 22** (**Convergence of primal iterates and of the residual).** *Assume that the matrix $H$, defined in* (3.9)*, is nonsingular. Fix arbitrary $x^0 \in \mathbb{R}^n$. The primal iterates $\{x^k\}$ produced by* (3.7) *converge linearly in expectation to $x^* + t$, where $x^*$ is the optimal solution of the primal problem* (3.2)*, and $t$ is the projection of $x^0 - c$ onto $\mathbf{Null}(A)$:*

$$t \overset{def}{=} \arg\min_{t'} \left\{ \|x^0 - c - t'\|_B \ : \ t' \in \mathbf{Null}(A) \right\}. \tag{3.10}$$

*In particular, for all $k \geq 0$ we have*

$$\textit{Primal iterates:} \qquad \mathbf{E}\left[\|x^k - x^* - t\|_B^2\right] \leq \rho^k \cdot \|x^0 - x^* - t\|_B^2, \tag{3.11}$$

$$\textit{Residual:} \qquad \mathbf{E}\left[\|Ax^k - b\|_B\right] \leq \rho^{k/2}\|A\|_B\|x^0 - x^* - t\|_B + \|At\|_B, \tag{3.12}$$

*where $\|A\|_B \overset{def}{=} \max\{\|Ax\|_B \ : \ \|x\|_B \leq 1\}$ and*

$$\rho \overset{def}{=} 1 - \lambda_{\min}^{+}\left(B^{-1/2}A^\top HAB^{-1/2}\right). \tag{3.13}$$

*Furthermore, the convergence rate is bounded by*

$$1 - \frac{\mathbf{E}\left[\mathbf{Rank}\left(S^\top A\right)\right]}{\mathbf{Rank}(A)} \leq \rho < 1. \tag{3.14}$$

As show shown in Section 2.4.3, if we let $S$ be a unit coordinate vector chosen at random, $B$ be the identity matrix and set $c = 0$, then (3.7) reduces to the *randomized Kaczmarz (RK)* method proposed and analyzed in a seminal work of Strohmer and Vershynin [125]. Theorem 22 implies that RK converges with an exponential rate so long as the system matrix has no zero rows (see Section 4.7). To the best of our knowledge, such a result was not previously established: current convergence results for RK assume that the system matrix is full rank [76, 102]. Not only do we show that the RK method converges to the least-norm solution for any consistent system, but we do so through a single all encompassing theorem covering a wide family of algorithms. Likewise, convergence of block variants of RK has only been established for full column rank [84, 85]. Block versions of RK can be obtained from our generic method by choosing $B = I$ and $c = 0$, as before, but letting $S$ to be a random column submatrix of the identity matrix. Again, our general complexity bound holds under no assumptions on $A$, as long as one can find $S$ such that $H$ becomes nonsingular.

The lower bound (3.14) says that for a singular system matrix, the number of steps required by SDA to reach an expected accuracy is at best inversely proportional to the rank of $A$. If $A$

[2] Note that from Lemma 5, the pseudo pseudoinverse of a symmetric positive semidefinite matrix is again symmetric and positive semidefinite. As a result, if the expectation defining $H$ is finite, $H$ is also symmetric and positive semidefinite. Hence, we could equivalently assume that $H$ be positive definite.

has row rank equal to one, for instance, then RK converges in one step (this is no surprise, given that RK projects onto the solution space of a single row, which in this case, is the solution space of the whole system). Our lower bound in this case becomes 0, and hence is tight.

While Theorem 22 is cast in terms of the primal iterates, if we assume that $x^0 = c + B^{-1}A^\top y^0$ for some $y^0 \in \mathbb{R}^m$, then an equivalent dual characterization follows by combining (3.6) and (3.8). In fact, in that case we can also establish the convergence of the primal function values and of the duality gap. *No such results were previously known.*

**Theorem 23** (**Convergence of function values).** *Assume that the matrix $H$, defined in* (3.9)*, is nonsingular. Fix arbitrary $y^0 \in \mathbb{R}^m$ and let $\{y^k\}$ be the SDA iterates produced by* (3.5)*. Further, let $\{x^k\}$ be the associated primal iterates, defined by* (3.6)*, $OPT \overset{def}{=} P(x^*) = D(y^*)$,*

$$U_0 \overset{def}{=} \tfrac{1}{2}\|x^0 - x^*\|_B^2 \overset{(3.8)}{=} OPT - D(y^0),$$

*and let $\rho$ be as in Theorem 22. Then for all $k \geq 0$ we have the following complexity bounds:*

$$\textit{Dual suboptimality:} \qquad \mathbf{E}\left[OPT - D(y^k)\right] \leq \rho^k U_0 \tag{3.15}$$

$$\textit{Primal suboptimality:} \qquad \mathbf{E}\left[P(x^k) - OPT\right] \leq \rho^k U_0 + 2\rho^{k/2}\sqrt{OPT \times U_0} \tag{3.16}$$

$$\textit{Duality gap:} \qquad \mathbf{E}\left[P(x^k) - D(y^k)\right] \leq 2\rho^k U_0 + 2\rho^{k/2}\sqrt{OPT \times U_0} \tag{3.17}$$

*In our analysis, no error bounds are necessary.*

### 3.2.4 Chapter outline

This chapter is structured as follows. Section 3.3 describes the algorithm in detail, both in its dual and primal form, and establishes several useful identities. In Section 4.7 we characterize discrete distributions for which our main assumption on $H$ is satisfied. We then specialize our method to several simple discrete distributions to better illustrate the results. We then show in Section 3.5 how SDA can be applied to design new randomized gossip algorithms. We also show that our framework can recover some standard methods. Theorem 22 is proved in Section 3.6 and Theorem 23 is proved in Section 3.7. In Section 3.8 we perform a simple experiment illustrating the convergence of the randomized Kaczmarz method on rank deficient linear systems. We then summarize in Section 3.9.

## 3.3 Stochastic Dual Ascent

By *stochastic dual ascent* (SDA) we refer to a randomized optimization method for solving the dual problem (3.3) performing iterations of the form

$$y^{k+1} = y^k + S\lambda^k, \tag{3.18}$$

where $S$ is a random matrix with $m$ rows drawn in each iteration independently from a prespecified distribution. We shall not fix the number of columns of $S$; in fact, we even allow for the number of columns to be random. By performing steps of the form (3.18), we are moving in the range space of the random matrix $S$, with $\lambda^k$ describing the precise linear combination of the columns used in computing the step. In particular, we shall choose $\lambda^k$ from the set

$$Q^k \overset{\text{def}}{=} \arg\max_{\lambda} D(y^k + S\lambda) \overset{(3.3)}{=} \arg\max_{\lambda} \left\{ (b - Ac)^\top (y^k + S\lambda) - \tfrac{1}{2} \left\| A^\top (y^k + S\lambda) \right\|^2_{B^{-1}} \right\}.$$

Since $D$ is bounded above (a consequence of weak duality), this set is nonempty. Since $D$ is a concave quadratic, $Q^k$ consists of all those vectors $\lambda$ for which the gradient of the mapping $\phi_k(\lambda) : \lambda \mapsto D(y^k + S\lambda)$ vanishes. This leads to the observation that $Q^k$ is the set of solutions of a random linear system:

$$Q^k = \left\{ \lambda \in \mathbb{R}^m \ : \ \left(S^\top A B^{-1} A^\top S\right) \lambda = S^\top \left(b - Ac - AB^{-1}A^\top y^k\right) \right\}.$$

If $S$ has a small number of columns, this is a small easy-to-solve system.

A key feature of our method enabling us to prove exponential error decay despite the lack of strong concavity is the way in which we choose $\lambda^k$ from $Q^k$. In SDA, $\lambda^k$ is chosen to be the least-norm element of $Q^k$,

$$\lambda^k \overset{\text{def}}{=} \arg\min_{\lambda \in Q^k} \|\lambda\|_2.$$

Using Lemma 7, the least-norm solution to the above is given by

$$\lambda^k = \left(S^\top A B^{-1} A^\top S\right)^\dagger S^\top \left(b - Ac - AB^{-1}A^\top y^k\right). \tag{3.19}$$

Note that if $S$ has only a few columns, then (3.19) requires projecting the origin onto a small linear system. The SDA algorithm is obtained by combining (3.18) with (3.19).

**Algorithm 1** Stochastic Dual Ascent (SDA)

1: **parameter:** $\mathcal{D}$ = distribution over random matrices
2: Choose $y^0 \in \mathbb{R}^m$ ▷ Initialization
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Sample an independent copy $S \sim \mathcal{D}$
5: $\lambda^k = \left(S^\top A B^{-1} A^\top S\right)^\dagger S^\top \left(b - Ac - AB^{-1}A^\top y^k\right)$
6: $y^{k+1} = y^k + S\lambda^k$ ▷ Update the dual variable

The method has one parameter: the distribution $\mathcal{D}$ from which the random matrices $S$ are drawn. Sometimes, one is interested in finding any solution of the system $Ax = b$, rather than

the particular solution described by the primal problem (3.2). In such situations, $B$ and $c$ could also be seen as parameters.

### 3.3.1 Optimality conditions

For any $x$ for which $Ax = b$ and for any $y$ we have

$$P(x) - D(y) \overset{(3.2)+(3.3)}{=} \tfrac{1}{2}\|x - c\|_B^2 + \tfrac{1}{2}\|A^\top y\|_{B^{-1}}^2 + (c - x)^\top A^\top y \geq 0,$$

where the inequality (weak duality) follows from the Fenchel-Young inequality[3]. As a result, we obtain the following necessary and sufficient optimality conditions, characterizing primal and dual optimal points.

**Proposition 24** (Optimality conditions)**.** *Vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ are optimal for the primal* (3.2) *and dual* (3.3) *problems respectively, if and only if they satisfy the following relation*

$$Ax = b, \qquad x = c + B^{-1}A^\top y. \tag{3.20}$$

In view of this, it will be useful to define a linear mapping from $\mathbb{R}^m$ to $\mathbb{R}^n$ as follows:

$$x(y) = c + B^{-1}A^\top y. \tag{3.21}$$

As an immediate corollary of Proposition 24 we observe that for any dual optimal $y^*$, the vector $x(y^*)$ must be primal optimal. Since the primal problem has a unique optimal solution, $x^*$, we must necessarily have

$$x^* = x(y^*) = c + B^{-1}A^\top y^*. \tag{3.22}$$

Another immediate corollary of Proposition 24 is the following characterization of dual optimality: $y$ is dual optimal if and only if

$$b - Ac = AB^{-1}A^\top y. \tag{3.23}$$

Hence, the set of dual optimal solutions is $\mathcal{Y}^* = (AB^{-1}A^\top)^\dagger(b - Ac) + \mathbf{Null}\left(AB^{-1}A^\top\right)$. Since, $\mathbf{Null}\left(AB^{-1}A^\top\right) = \mathbf{Null}\left(A^\top\right)$ (see Lemma 1), we have

$$\mathcal{Y}^* = \left(AB^{-1}A^\top\right)^\dagger (b - Ac) + \mathbf{Null}\left(A^\top\right).$$

Combining this with (3.22), we get

$$x^* = c + B^{-1}A^\top \left(AB^{-1}A^\top\right)^\dagger (b - Ac).$$

**Remark 25** (The dual is also a least-norm problem)**.** *Observe that:*

[3]Let $U$ be a vector space equipped with an inner product $\langle \cdot, \cdot \rangle : U \times U \to \mathbb{R}$. Given a function $f : U \to \mathbb{R}$, its convex (or Fenchel) conjugate $f^* : U \to \mathbb{R} \cup \{+\infty\}$ is defined by $f^*(v) = \sup_{u \in U} \langle u, v \rangle - f(u)$. A direct consequence of this is the Fenchel-Young inequality, which asserts that $f(u) + f^*(v) \geq \langle u, v \rangle$ for all $u$ and $v$. The inequality in the main text follows by choosing $f(u) = \frac{1}{2}\|u\|_B^2$ (and hence $f^*(v) = \frac{1}{2}\|v\|_{B^{-1}}^2$), $u = x - c$ and $v = A^\top y$. If $f$ is differentiable, then equality holds if and only if $v = \nabla f(u)$. In our case, this condition is $x = c + B^{-1}A^\top y$. This, together with primal feasibility, gives the optimality conditions (3.20). For more details on Fenchel duality, see [12].

1. *The particular dual optimal point* $y^* = (AB^{-1}A^\top)^\dagger(b - Ac)$ *is the solution of the following optimization problem:*

$$\min\left\{\tfrac{1}{2}\|y\|_2^2 \;:\; AB^{-1}A^\top y = b - Ac\right\}. \tag{3.24}$$

*Hence, this particular formulation of the dual problem has the same form as the primal problem: projection onto a linear system.*

2. *If* $A^\top A$ *is positive definite (which can only happen if* $A$ *is of full column rank, which means that* $Ax = b$ *has a unique solution and hence the primal objective function does not matter), and we choose* $B = A^\top A$*, then the dual constraint* (3.24) *becomes*

$$A(A^\top A)^{-1}A^\top y = b - Ac.$$

*This constraint has a geometric interpretation: we are seeking a vector* $y$ *whose orthogonal projection onto the column space of* $A$ *is equal to* $b - Ac$*. Hence the reformulated dual problem* (3.24) *is asking us to find the vector* $y$ *with this property having the least norm.*

### 3.3.2 Primal iterates associated with the dual iterates

With the sequence of dual iterates $\{y^k\}$ produced by SDA we can associate a sequence of primal iterates $\{x^k\}$ using the mapping (3.21):

$$x^k \overset{\text{def}}{=} x(y^k) = c + B^{-1}A^\top y^k. \tag{3.25}$$

This leads to the following *primal version of the SDA method*.

**Algorithm 2** Primal Version of Stochastic Dual Ascent (SDA-Primal)

**parameter:** $\mathcal{D}$ = distribution over random matrices
Choose $x^0 \in \mathbb{R}^n$ ▷ Initialization
**for** $k = 0, 1, 2, \ldots$ **do**
  Sample an independent copy $S \sim \mathcal{D}$
  $x^{k+1} = x^k - B^{-1}A^\top S\left(S^\top AB^{-1}A^\top S\right)^\dagger S^\top(Ax^k - b)$ ▷ Update the primal variable

**Remark 26.** *A couple of observations:*

1. Self-duality. *If* $A$ *is positive definite,* $c = 0$*, and if we choose* $B = A$*, then in view of* (3.25) *we have* $x^k = y^k$ *for all* $k$*, and hence Algorithms 1 and 2 coincide. In this case, Algorithm 2 can be described as* self-dual.

2. Space of iterates. *A direct consequence of the correspondence between the dual and primal iterates* (3.25) *is the following simple observation (a generalized version of this, which we prove later as Lemma 30, will be used in the proof of Theorem 22): Choose* $y^0 \in \mathbb{R}^m$ *and let* $x^0 = c + B^{-1}A^\top y^0$*. Then the iterates* $\{x^k\}$ *of Algorithm 2 are of the form* $x^k = c + B^{-1}A^\top y^k$ *for some* $y^k \in \mathbb{R}^m$*.*

3. Starting point. *While we have defined the primal iterates of Algorithm 2 via a linear transformation of the dual iterates—see* (3.25)*—we* can*, in principle, choose $x^0$ arbitrarily, thus breaking the primal-dual connection which helped us to define the method. In particular, we can choose $x^0$ in such a way that there does not exist $y^0$ for which $x^0 = c + B^{-1}A^\top y^0$. As is clear from Theorem 22, in this case the iterates $\{x^k\}$ will not converge to $x^*$, but to $x^* + t$, where $t$ is the projection of $x^0 - c$ onto the nullspace of $A$.*

It is now clear that the iterates of Algorithm 2 are the same as the iterates defined by the Random Update viewpoint (2.10) of the sketch-and-project method (2.5) in Chapter 2. Thus we have uncovered a hidden dual nature of the sketch-and-project method. It is this dual relationship that allows us to formulate and prove convergence of the dual function values and duality gap proven in Theorem 23. In particular the duality gap gives a means to measure the distance from the optimality. This certificate of convergence is standard in optimization methods, but seldom appears in the numerical linear algebra literature.

### 3.3.3 Relating the quality of the dual and primal iterates

The following simple but insightful result (mentioned in the introduction) relates the "quality" of a dual vector $y$ with that of its primal counterpart, $x(y)$. It says that the dual suboptimality of $y$ in terms of function values is equal to the primal suboptimality of $x(y)$ in terms of distance.

**Proposition 27.** *Let $y^*$ be any dual optimal point and $y \in \mathbb{R}^m$. Then*

$$D(y^*) - D(y) = \tfrac{1}{2}\|x(y^*) - x(y)\|_B^2.$$

*Proof.* Straightforward calculation shows that

$$\begin{aligned} D(y^*) - D(y) &\overset{(3.3)}{=} (b - Ac)^\top (y^* - y) - \tfrac{1}{2}(y^*)^\top AB^{-1}A^\top y^* + \tfrac{1}{2}y^\top AB^{-1}A^\top y \\ &\overset{(3.23)}{=} (y^*)^\top AB^{-1}A^\top (y^* - y) - \tfrac{1}{2}(y^*)^\top AB^{-1}A^\top y^* + \tfrac{1}{2}y^\top AB^{-1}A^\top y \\ &= \tfrac{1}{2}(y - y^*)^\top AB^{-1}A^\top (y - y^*) \\ &\overset{(3.21)}{=} \tfrac{1}{2}\|x(y) - x(y^*)\|_B^2. \end{aligned}$$

□

Applying this result to the sequence $\{(x^k, y^k)\}$ of dual iterates produced by SDA and their corresponding primal images, as defined in (3.25), we get the identity:

$$D(y^*) - D(y^k) = \tfrac{1}{2}\|x^k - x^*\|_B^2.$$

Therefore, *dual convergence in function values $D(y^k)$ is equivalent to primal convergence in iterates $x^k$*. Furthermore, a direct computation leads to the following formula for the *duality gap*:

$$P(x^k) - D(y^k) \overset{(3.25)}{=} (AB^{-1}A^\top y^k + Ac - b)^\top y^k = -(\nabla D(y^k))^\top y^k. \tag{3.26}$$

Note that computing the gap is significantly more expensive than the cost of a single iteration (in the interesting regime when the number of columns of $S$ is small). Hence, evaluation of the

duality gap should generally be avoided. If it is necessary to be certain about the quality of a solution however, the above formula will be useful. The gap should then be computed from time to time only, so that this extra work does not significantly slow down the iterative process.

## 3.4 Discrete Distributions

Both the SDA algorithm and its primal counterpart are generic in the sense that the distribution $\mathcal{D}$ is not specified beyond assuming that the matrix $H$ defined in (3.9) is well defined and nonsingular. In this section we shall first characterize finite discrete distributions for which $H$ is nonsingular. We then give a few examples of algorithms based on such distributions, and comment on our complexity results in more detail.

### 3.4.1 Nonsingularity of $H$ for finite discrete distributions

For simplicity, we shall focus on *finite discrete* distributions $\mathcal{D}$. That is, we set $S = S_i$ with probability $p_i > 0$, where $S_1, \ldots, S_r$ are fixed matrices (each with $m$ rows). The next theorem gives a necessary and sufficient condition for the matrix $H$ defined in (3.9) to be nonsingular.

**Theorem 28.** *Let $\mathcal{D}$ be a finite discrete distribution, as described above. Then $H$ is nonsingular if and only if*

$$\mathbf{Range}\left([S_1 S_1^\top A, \cdots, S_r S_r^\top A]\right) = \mathbb{R}^m.$$

*Proof.* Let $K_i = S_i^\top A B^{-1/2}$. In view of the identity $\left(K_i K_i^\top\right)^\dagger = (K_i^\dagger)^\top K_i^\dagger$, we can write

$$H \overset{(3.9)}{=} \sum_{i=1}^r H_i,$$

where $H_i = p_i S_i (K_i^\dagger)^\top K_i^\dagger S_i^\top$. Since $H_i$ are symmetric positive semidefinite, so is $H$. Now, it is easy to check that $y^\top H_i y = 0$ if and only if $y \in \mathbf{Null}\,(H_i)$ (this holds for any symmetric positive semidefinite $H_i$). Hence, $y^\top H y = 0$ if and only if $y \in \cap_i \mathbf{Null}\,(H_i)$ and thus $H$ is positive definite if and only if

$$\bigcap_i \mathbf{Null}\,(H_i) = \{0\}. \tag{3.27}$$

In view of Lemma 1, $\mathbf{Null}\,(H_i) = \mathbf{Null}\left(\sqrt{p_i} K_i^\dagger S_i^\top\right) = \mathbf{Null}\left(K_i^\dagger S_i^\top\right)$. Now, $y \in \mathbf{Null}\left(K_i^\dagger S_i^\top\right)$ if and only of $S_i^\top y \in \mathbf{Null}\left(K_i^\dagger\right) = \mathbf{Null}\left(K_i^\top\right) = \mathbf{Null}\left(A^\top S_i\right)$. Hence, $\mathbf{Null}\,(H_i) = \mathbf{Null}\left(A^\top S_i S_i^\top\right)$, which means that (3.27) is equivalent to
$\mathbf{Null}\left([S_1 S_1^\top A, \cdots, S_r S_r^\top A]^\top\right) = \{0\}$. □

We have the following corollary.[4]

**Corollary 29.** *Assume that $S_i^\top A$ has full row rank for all $i$ and that $\mathbf{S} \overset{def}{=} [S_1, \ldots, S_r]$ is of full row rank. Then $H$ is nonsingular.*

We now give a few illustrative examples:

[4] We can also prove the corollary directly as follows: The first assumption implies that $S_i^\top A B^{-1} A^\top S_i$ is invertible for all $i$ and that $V \overset{\text{def}}{=} \text{Diag}\left(p_i^{1/2} (S_i^\top A B^{-1} A^\top S_i)^{-1/2}\right)$ is nonsingular. It remains to note that

$$H \overset{(3.9)}{=} \mathbf{E}\left[S\left(S^\top A B^{-1} A^\top S\right)^{-1} S^\top\right] = \sum_i p_i S_i \left(S_i^\top A B^{-1} A^\top S_i\right)^{-1} S_i^\top = \mathbf{S} V^2 \mathbf{S}^\top.$$

1. *Coordinate vectors.* Let $S_i = e_i$ ($i^{\text{th}}$ unit coordinate vector) for $i = 1, 2, \ldots, r = m$. In this case, $\mathbf{S} = [S_1, \ldots, S_m]$ is the identity matrix in $\mathbb{R}^m$, and $S_i^\top A$ has full row rank for all $i$ as long as the rows of $A$ are all nonzero. By Corollary 29, $H$ is positive definite.

2. *Submatrices of the identity matrix.* We can let $S$ be a random column submatrix of the $m \times m$ identity matrix $I$. There are $2^m - 1$ such potential submatrices, and we choose $1 \leq r \leq 2^m - 1$. As long as we choose $S_1, \ldots, S_r$ in such a way that each column of $I$ is represented in some matrix $S_i$, the matrix $\mathbf{S}$ will have full row rank. Furthermore, if $S_i^\top A$ has full row rank for all $i$, then by the above corollary, $H$ is nonsingular. Note that if the row rank of $A$ is $r$, then the matrices $S_i$ selected by the above process will necessarily have at most $r$ columns.

3. *Count sketch and Count-min sketch.* Many other "sketching" matrices $S$ can be employed within SDA, including the count sketch [18] and the count-min sketch [21]. In our context (recall that we sketch with the transpose of $S$), $S$ is a count-sketch matrix (resp. count-min sketch) if it is assembled from random columns of $[I, -I]$ (resp $I$), chosen uniformly with replacement, where $I$ is the $m \times m$ identity matrix.

### 3.4.2 Randomized Kaczmarz is the primal process associated with randomized coordinate ascent

Let $B = I$ (the identity matrix). The primal problem then becomes

$$\begin{array}{ll} \text{minimize} & P(x) \stackrel{\text{def}}{=} \frac{1}{2}\|x - c\|_2^2 \\ \text{subject to} & Ax = b \\ & x \in \mathbb{R}^n, \end{array}$$

and the dual problem is

$$\begin{array}{ll} \text{maximize} & D(y) \stackrel{\text{def}}{=} (b - Ac)^\top y - \frac{1}{2} y^\top A A^\top y \\ \text{subject to} & y \in \mathbb{R}^m. \end{array}$$

**Dual iterates.** Let us choose $S = e^i$ (unit coordinate vector in $\mathbb{R}^m$) with probability $p_i > 0$ (to be specified later). The SDA method (Algorithm 1) then takes the form

$$\boxed{y^{k+1} = y^k + \frac{b_i - A_i c - A_{i:} A^\top y^k}{\|A_{i:}\|_2^2} e_i} \tag{3.28}$$

This is the randomized coordinate ascent method applied to the dual problem. In the form popularized by Nesterov [86], it takes the form

$$y^{k+1} = y^k + \frac{e_i^\top \nabla D(y^k)}{L_i} e_i,$$

where $e_i^\top \nabla D(y^k)$ is the $i$th partial derivative of $D$ at $y^k$ and $L_i > 0$ is the Lipschitz constant of the $i$th partial derivative, i.e., constant for which the following inequality holds for all $\lambda \in \mathbb{R}$:

$$|e_i^\top \nabla D(y + \lambda e_i) - e_i^\top \nabla D(y)| \leq L_i |\lambda|. \tag{3.29}$$

It can be easily verified that (3.29) holds with $L_i = \|A_{i:}\|_2^2$ and that $e_i^\top \nabla D(y^k) = b_i - A_{i:}c - A_{i:}A^\top y^k$.

**Primal iterates.** The associated primal iterative process (Algorithm 2) takes the form

$$\boxed{x^{k+1} = x^k - \frac{A_{i:}x^k - b_i}{\|A_{i:}\|_2^2} A_{i:}^\top} \tag{3.30}$$

This is the randomized Kaczmarz method of Strohmer and Vershynin [125].

**The rate.** Let us now compute the rate $\rho$ as defined in (3.13). It will be convenient, but *not* optimal, to choose the probabilities according to Theorem 19, that is

$$p_i = \frac{\|A_{i:}\|_2^2}{\|A\|_F^2}, \tag{3.31}$$

where $\|\cdot\|_F$ denotes the Frobenius norm (we assume that $A$ does not contain any zero rows). Since

$$H \overset{(3.9)}{=} \mathbf{E}\left[S\left(S^\top A A^\top S\right)^\dagger S^\top\right] = \sum_{i=1}^m p_i \frac{e_i e_i^\top}{\|A_{i:}\|_2^2} \overset{(3.31)}{=} \frac{1}{\|A\|_F^2} I,$$

we have

$$\rho = 1 - \lambda_{\min}^+\left(A^\top H A\right) = 1 - \frac{\lambda_{\min}^+\left(A^\top A\right)}{\|A\|_F^2}. \tag{3.32}$$

Furthermore, if $r = \mathbf{Rank}\,(A)$, then in view of (3.14), the rate is bounded as

$$1 - \frac{1}{r} \leq \rho < 1.$$

Assume that $A$ is of rank $r = 1$ and let $A = uv^\top$. Then $A^\top A = (u^\top u)vv^\top$, and hence this matrix is also of rank 1. Therefore, $A^\top A$ has a single nonzero eigenvalue, which is equal to its trace. Therefore, $\lambda_{\min}^+(A^\top A) = \mathbf{Tr}\left(A^\top A\right) = \|A\|_F^2$ and hence $\rho = 0$. Note that the rate $\rho$ reaches its lower bound and the method converges in one step.

**Remarks.** For randomized coordinate ascent applied to (non-strongly) concave quadratics, rate (3.32) has been established by Leventhal and Lewis [68]. However, to the best of our knowledge, this is the first time this rate has also been established for the randomized Kaczmarz method. We do not only prove this, but show that this is because the iterates of the two methods are linked via a linear relationship. In the $c = 0, B = I$ case, and for row-normalized matrix $A$, this linear relationship between the two methods was recently independently observed by Wright [133]. While all linear complexity results for RK we are aware of require full rank assumptions, there exist nonstandard variants of RK which do not require such assumptions, one example being the

asynchronous parallel version of RK studied by Liu, Wright and Sridhar [74]. Finally, no results of the type (3.16) (primal suboptimality) and (3.17) (duality gap) previously existed for these methods in the literature.

### 3.4.3 Randomized block Kaczmarz is the primal process associated with randomized Newton

Let $B = I$, so that we have the same pair of primal dual problems as in Section 3.4.2.

**Dual iterates.** Let us now choose $S$ to be a random column submatrix of the $m \times m$ identity matrix $I$. That is, we choose a random subset $C \subset \{1, 2, \ldots, m\}$ and then let $S$ be the concatenation of columns $j \in C$ of $I$. We shall write $S = I_C$. Let $p_C$ be the probability that $S = I_C$. Assume that for each $j \in \{1, \ldots, m\}$ there exists $C$ with $j \in C$ such that $p_C > 0$. Such a random set is called *proper* [101].

The SDA method (Algorithm 1) then takes the form

$$\boxed{y^{k+1} = y^k + I_C \lambda^k} \tag{3.33}$$

where $\lambda^k$ is chosen so that the dual objective is maximized (see (3.19)). This is a variant of the *randomized Newton method* studied in [101]. By examining (3.19), we see that this method works by "inverting" randomized submatrices of the "Hessian" $AA^\top$. Indeed, $\lambda^k$ is in each iteration computed by solving a system with the matrix $I_C^\top AA^\top I_C$. This is the random submatrix of $AA^\top$ corresponding to rows and columns in $C$.

**Primal iterates.** In view of the equivalence between Algorithm 2 and the sketch-and-project method (2.5), the primal iterative process associated with the randomized Newton method has the form

$$\boxed{x^{k+1} = \arg\min_x \left\{ \|x - x^k\| \ : \ I_C^\top A x = I_C^\top b \right\}} \tag{3.34}$$

This method is a variant of the *randomized block Kaczmarz* method of Needell [84]. The method proceeds by projecting the last iterate $x^k$ onto a subsystem of $Ax = b$ formed by equations indexed by the set $C$.

**The rate.** Provided that $H$ is nonsingular, the shared rate of the randomized Newton and randomized block Kaczmarz methods is

$$\rho \overset{(3.13)}{=} 1 - \lambda_{\min}^+ \left( A^\top \mathbf{E} \left[ I_C \left( I_C^\top AA^\top I_C \right)^\dagger I_C^\top \right] A \right).$$

Qu et al [101] study the randomized Newton method for the problem of minimizing a smooth strongly convex function and prove linear convergence. In particular, they study the above rate in the case when $AA^\top$ is positive definite. Here we show that linear converges also holds for *weakly* convex quadratics (as long as $H$ is nonsingular).

### 3.4.4 Self-duality for positive definite $A$

If $A$ is positive definite, then we can choose $B = A$. As mentioned before, in this setting SDA is self-dual: $x^k = y^k$ for all $k$. The primal problem then becomes

$$\begin{aligned} \text{minimize} \quad & P(x) \stackrel{\text{def}}{=} \tfrac{1}{2} x^\top A x \\ \text{subject to} \quad & Ax = b \\ & x \in \mathbb{R}^n. \end{aligned}$$

and the dual problem becomes

$$\begin{aligned} \text{maximize} \quad & D(y) \stackrel{\text{def}}{=} b^\top y - \tfrac{1}{2} y^\top A y \\ \text{subject to} \quad & y \in \mathbb{R}^m. \end{aligned}$$

Note that the primal objective function does not play any role in determining the solution; indeed, the feasible set contains a single point only: $A^{-1}b$. However, it does affect the iterative process.

**Primal and dual iterates.** As before, let us choose $S = e^i$ (unit coordinate vector in $\mathbb{R}^m$) with probability $p_i > 0$, where the probabilities $p_i$ are arbitrary. Then both the primal and the dual iterates take the form

$$\boxed{y^{k+1} = y^k - \frac{A_{i:}y^k - b_i}{A_{ii}} e_i}$$

This is the randomized coordinate ascent method applied to the dual problem.

**The rate.** If we choose $p_i = A_{ii}/\mathbf{Tr}\,(A)$, then

$$H = \mathbf{E}\left[S\left(S^\top A S\right)^\dagger S^\top\right] = \frac{I}{\mathbf{Tr}\,(A)},$$

whence

$$\rho \stackrel{(3.13)}{=} 1 - \lambda_{\min}^+\left(A^{1/2} H A^{1/2}\right) = 1 - \frac{\lambda_{\min}(A)}{\mathbf{Tr}\,(A)}.$$

It is known that for this problem, randomized coordinate descent applied to the dual problem, with this choice of probabilities, converges with this rate[68].

## 3.5 Application: Randomized Gossip Algorithms

In this section we apply our method and results to the distributed consensus (averaging) problem.

Let $(V, E)$ be a connected network with $|V| = n$ nodes and $|E| = m$ edges, where each edge is an unordered pair $\{i, j\} \in E$ of distinct nodes. Node $i \in V$ stores a private value $c_i \in \mathbb{R}$. The goal of the *distributed consensus problem* is for the network to compute the average of these private values in a distributed fashion [13, 90]. This means that the exchange of information can only occur along the edges of the network.

The nodes may represent people in a social network, with edges representing friendship and private value representing certain private information, such as salary. The goal would be to compute the average salary via an iterative process where only friends are allowed to exchange information. The nodes may represent sensors in a wireless sensor network, with an edge between two sensors if they are close to each other so that they can communicate. Private values represent measurements of some quantity performed by the sensors, such as the temperature. The goal is for the network to compute the average temperature.

### 3.5.1 Consensus as a projection problem

We now show how one can model the consensus (averaging) problem in the form (3.2). Consider the projection problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x - c\|_2^2 \\ \text{subject to} & x_1 = x_2 = \cdots = x_n, \end{array} \tag{3.35}$$

and note that the optimal solution $x^*$ must necessarily satisfy

$$x_i^* = \bar{c} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} c_i,$$

for all $i$. There are many ways in which the constraint forcing all coordinates of $x$ to be equal can be represented in the form of a linear system $Ax = b$. Here are some examples:

1. *Each node is equal to all its neighbours.* Let the equations of the system $Ax = b$ correspond to constraints

   $$x_i = x_j,$$

   for $\{i, j\} \in E$. That is, we are enforcing all pairs of vertices joined by an edge to have the same value. Each edge $e \in E$ can be written in two ways: $e = \{i, j\}$ and $e = \{j, i\}$, where $i, j$ are the incident vertices. In order to avoid duplicating constraints, for each edge $e \in E$ we use $e = (i, j)$ to denote an arbitrary but fixed order of its incident vertices $i, j$. We then let $A \in \mathbb{R}^{m\times n}$ and $b = 0 \in \mathbb{R}^m$, where

   $$(A_{e:})^\top = f_i - f_j, \tag{3.36}$$

   and where $e = (i, j) \in E$, $f_i$ (resp. $f_j$) is the $i^{\text{th}}$ (resp. $j^{\text{th}}$) unit coordinate vector in $\mathbb{R}^n$. Note that the constraint $x_i = x_j$ is represented only once in the linear system. Further,

note that the matrix

$$L = A^\top A \tag{3.37}$$

is the *Laplacian* matrix of the graph $(V, E)$:

$$L_{ij} = \begin{cases} d_i & i = j \\ -1 & i \neq j,\ (i,j) \in E \\ 0 & \text{otherwise,} \end{cases}$$

where $d_i$ is the degree of node $i$.

2. *Each node is the average of its neighbours.* Let the equations of the system $Ax = b$ correspond to constraints

$$x_i = \frac{1}{d_i} \sum_{j \in N(i)} x_j,$$

for $i \in V$, where $N(i) \overset{\text{def}}{=} \{j \in V : \{i,j\} \in E\}$ is the set of neighbours of node $i$ and $d_i \overset{\text{def}}{=} |N(i)|$ is the degree of node $i$. That is, we require that the values stored at each node are equal to the average of the values of its neighbours. This corresponds to the choice $b = 0$ and

$$(A_{i:})^\top = f_i - \frac{1}{d_i} \sum_{j \in N(i)} f_j. \tag{3.38}$$

Note that $A \in \mathbb{R}^{n \times n}$.

3. *Spanning subgraph.* Let $(V, E')$ be any connected subgraph of $(V, E)$. For instance, we can choose a spanning tree. We can now apply any of the 2 models above to this new graph and either require $x_i = x_j$ for all $\{i,j\} \in E'$, or require the value $x_i$ to be equal to the average of the values $x_j$ for all neighbours $j$ of $i$ in $(V, E')$.

Clearly, the above list does not exhaust the ways in which the constraint $x_1 = \cdots = x_n$ can be modeled as a linear system. For instance, we could build the system from constraints such as $x_1 = x_2 + x_4 - x_3$, $x_1 = 5x_2 - 4x_7$ and so on.

Different representations of the constraint $x_1 = \cdots = x_n$, in combination with a choice of $\mathcal{D}$, will lead to a wide range of specific algorithms for the consensus problem (3.35). Some (but not all) of these algorithms will have the property that communication only happens along the edges of the network, and these are the ones we are interested in. The number of combinations is very vast. We will therefore only highlight two options, with the understanding that based on this, the interested reader can assemble other specific methods as needed.

### 3.5.2 Model 1: Each node is equal to its neighbours

Let $b = 0$ and $A$ be as in (3.36). Let the distribution $\mathcal{D}$ be defined by setting $S = e_i$ with probability $p_i > 0$, where $e_i$ is the $i^{\text{th}}$ unit coordinate vector in $\mathbb{R}^m$. We have $B = I$, which means that Algorithm 2 is the randomized Kaczmarz (RK) method (3.30) and Algorithm 1 is the randomized coordinate ascent method (3.28).

Let us take $y^0 = 0$ (which means that $x^0 = c$), so that in Theorem 22 we have $t = 0$, and hence $x^k \to x^*$. The particular choice of the starting point $x^0 = c$ in the primal process has

a very tangible meaning: for all $i$, node $i$ initially knows value $c_i$. The primal iterative process will dictate how the local values are modified in an iterative fashion so that eventually all nodes contain the optimal value $x_i^* = \bar{c}$.

**Primal method.** In view of (3.36), for each edge $e = (i, j) \in E$, we have $\|A_{e:}\|_2^2 = 2$ and $A_{e:}x^k = x_i^k - x_j^k$. Hence, if the edge $e$ is selected by the RK method, (3.30) takes the specific form

$$\boxed{x^{k+1} = x^k - \frac{x_i^k - x_j^k}{2}(f_i - f_j)} \tag{3.39}$$

From (3.39) we see that only the $i^{\text{th}}$ and $j^{\text{th}}$ coordinates of $x^k$ are updated, via

$$x_i^{k+1} = x_i^k - \frac{x_i^k - x_j^k}{2} = \frac{x_i^k + x_j^k}{2}$$

and

$$x_j^{k+1} = x_j^k + \frac{x_i^k - x_j^k}{2} = \frac{x_i^k + x_j^k}{2}.$$

Note that in each iteration of RK, a random edge is selected, and the nodes on this edge replace their local values by their average. This is a basic variant of the *randomized gossip* algorithm [13, 140].

**Invariance.** Let $f$ be the vector of all ones in $\mathbb{R}^n$ and notice that from (3.39) we obtain $f^\top x^{k+1} = f^\top x^k$ for all $k$. This means that for all $k \geq 0$ we have the invariance property:

$$\sum_{i=1}^n x_i^k = \sum_{i=1}^n c_i. \tag{3.40}$$

**Insights from the dual perspective.** We can now bring new insight into the randomized gossip algorithm by considering the dual iterative process. The dual method (3.28) maintains weights $y^k$ associated with the edges of $E$ via the process:

$$y^{k+1} = y^k - \frac{A_{e:}(c - A^\top y^k)}{2} e_e,$$

where $e$ is a randomly selected edge. Hence, only the weight of a single edge is updated in each iteration. At optimality, we have $x^* = c + A^\top y^*$. That is, for each $i$

$$\delta_i \overset{\text{def}}{=} \bar{c} - c_i = x_i^* - c_i = (A^\top y^*)_i = \sum_{e \in E} A_{ei} y_e^*,$$

where $\delta_i$ is the correction term which needs to be added to $c_i$ in order for node $i$ to contain the value $\bar{c}$. From the above we observe that these correction terms are maintained by the dual method as an inner product of the $i^{\text{th}}$ column of $A$ and $y^k$, with the optimal correction being $\delta_i = A_{:i}^\top y^*$.

**Rate.** Both Theorem 22 and Theorem 23 hold, and hence we automatically get several types of convergence for the randomized gossip method. In particular, to the best of our knowledge, no

primal-dual type of convergence exist in the literature. Equation (3.26) gives a stopping criterion certifying convergence via the duality gap, which is also new.

In view of (3.32) and (3.37), and since $\|A\|_F^2 = 2m$, the convergence rate appearing in all these complexity results is given by

$$\rho = 1 - \frac{\lambda_{\min}^+(L)}{2m},$$

where $L$ is the Laplacian of $(V, E)$. While it is know that the Laplacian is singular, the rate depends on the smallest nonzero eigenvalue. This means that the number of iterations needed to output an $\epsilon$-solution in expectation scales as $O\big(\big(2m/\lambda_{\min}^+(L)\big)\log(1/\epsilon)\big)$, i.e., linearly with the number of edges.

### 3.5.3 Model 2: Each node is equal to the average of its neighbours

Let $A$ be as in (3.38) and $b = 0$. Let the distribution $\mathcal{D}$ be defined by setting $S = f_i$ with probability $p_i > 0$, where $f_i$ is the $i^{\text{th}}$ unit coordinate vector in $\mathbb{R}^n$. Again, we have $B = I$, which means that Algorithm 2 is the randomized Kaczmarz (RK) method (3.30) and Algorithm 1 is the randomized coordinate ascent method (3.28). As before, we choose $y^0 = 0$, whence $x^0 = c$.

**Primal method.** Observe that $\|A_{i:}\|_2^2 = 1 + 1/d_i$. The RK method (3.30) applied to this formulation of the problem takes the form

$$\boxed{x^{k+1} = x^k - \frac{x_i^k - \frac{1}{d_i}\sum_{j\in N(i)} x_j^k}{1 + 1/d_i}\left(f_i - \frac{1}{d_i}\sum_{j\in N(i)} f_j\right)} \tag{3.41}$$

where $i$ is chosen at random. This means that only coordinates in $i \cup N(i)$ get updated in such an iteration, the others remain unchanged. For node $i$ (coordinate $i$), this update is

$$x_i^{k+1} = \frac{1}{d_i + 1}\left(x_i^k + \sum_{j\in N(i)} x_j^k\right). \tag{3.42}$$

That is, the updated value at node $i$ is the average of the values of its neighbours and the previous value at $i$. From (3.41) we see that the values at nodes $j \in N(i)$ get updated as follows:

$$x_j^{k+1} = x_j^k + \frac{1}{d_i + 1}\left(x_i^k - \frac{1}{d_i}\sum_{j'\in N(i)} x_{j'}^k\right). \tag{3.43}$$

**Invariance.** Let $f$ be the vector of all ones in $\mathbb{R}^n$ and notice that from (3.41) we obtain

$$f^\top x^{k+1} = f^\top x^k - \frac{x_i^k - \frac{1}{d_i}\sum_{j\in N(i)} x_j^k}{1 + 1/d_i}\left(1 - \frac{d_i}{d_i}\right) = f^\top x^k,$$

for all $k$. It follows that the method satisfies the invariance property (3.40).

**Rate.** The method converges with the rate $\rho$ given by (3.32), where $A$ is given by (3.38). If $(V, E)$ is a complete graph (i.e., $m = \frac{n(n-1)}{2}$), then $L = \frac{(n-1)^2}{n} A^\top A$ is the Laplacian. In this case, $\|A\|_F^2 = \mathbf{Tr}\left(A^\top A\right) = \frac{n}{(n-1)^2}\mathbf{Tr}\left(L\right) = \frac{n}{(n-1)^2}\sum_i d_i = \frac{n^2}{n-1}$ and hence

$$\rho \overset{(3.38)}{=} 1 - \frac{\lambda_{\min}^+(A^\top A)}{\|A\|_F^2} = 1 - \frac{\frac{n}{(n-1)^2}\lambda_{\min}^+(L)}{\frac{n^2}{n-1}} = 1 - \frac{\lambda_{\min}^+(L)}{2m}.$$

## 3.6 Proof of Theorem 22

In this section we prove Theorem 22. We proceed as follows: in Section 3.6.1 we characterize the space in which the iterates move, in Section 3.6.2 we establish a certain key technical inequality, in Section 3.6.3 we establish convergence of iterates, in Section 3.6.4 we derive a rate for the residual and finally, and in Section 3.6.5 we establish the lower bound on the convergence rate.

### 3.6.1 An error lemma

The following result describes the space in which the iterates move. It is an extension of the observation in item 2 of Remark 26 to the case when $x^0$ is chosen arbitrarily.

**Lemma 30.** *Let the assumptions of Theorem 22 hold. For all $k \geq 0$ there exists $w^k \in \mathbb{R}^m$ such that $x^k - x^* - t = B^{-1}A^\top w^k$.*

*Proof.* We proceed by induction. Since by definition, $t$ is the projection of $x^0 - c$ onto $\mathbf{Null}\,(A)$ (see (1.19)), applying Proposition 8 we know that $x^0 - c = s + t$, where $s = B^{-1}A^\top \hat{y}^0$ for some $\hat{y}^0 \in \mathbb{R}^m$. Moreover, in view of (3.22), we know that $x^* = c + B^{-1}A^\top y^*$, where $y^*$ is any dual optimal solution. Hence,

$$x^0 - x^* - t = B^{-1}A^\top(\hat{y}^0 - y^*).$$

Assuming the relationship holds for $k$, we have

$$\begin{aligned} x^{k+1} - x^* - t &\overset{\text{(Alg 2)}}{=} \left[x^k - B^{-1}A^\top S(S^\top AB^{-1}A^\top S)^\dagger S^\top(Ax^k - b)\right] - x^* - t \\ &= \left[x^* + t + B^{-1}A^\top w^k - B^{-1}A^\top S(S^\top AB^{-1}A^\top S)^\dagger S^\top(Ax^k - b)\right] - x^* - t \\ &= B^{-1}A^\top w^{k+1}, \end{aligned}$$

where $w^{k+1} = w^k - S(S^\top AB^{-1}A^\top S)^\dagger S^\top(Ax^k - b)$. □

### 3.6.2 A key inequality

The following inequality is of key importance in the proof of the main theorem.

**Lemma 31.** *Let $0 \neq W \in \mathbb{R}^{m\times n}$ and $G \in \mathbb{R}^{m\times m}$ be symmetric positive definite. Then the matrix $W^\top GW$ has a positive eigenvalue, and the following inequality holds for all $y \in \mathbb{R}^m$:*

$$y^\top WW^\top GWW^\top y \geq \lambda^+_{\min}(W^\top GW)\|W^\top y\|_2^2. \tag{3.44}$$

*Furthermore, this bound is tight.*

*Proof.* Fix arbitrary $y \in \mathbb{R}^m$. By Lemma 1, $W^\top y \in \mathbf{Range}\left(W^\top GW\right)$. Since $W$ is nonzero the positive semidefinite matrix $W^\top GW$ is also nonzero, and hence it has a positive eigenvalue. Hence, $\lambda^+_{\min}(W^\top GW)$ is well defined. Let $\lambda^+_{\min}(W^\top GW) = \lambda_1 \leq \cdots \leq \lambda_\tau$ be the positive eigenvalues of $W^\top GW$, with associated orthonormal eigenvectors $q_1, \ldots, q_\tau$. We thus have

$$W^\top GW = \sum_{i=1}^{\tau} \lambda_i q_i q_i^\top.$$

It is easy to see that these eigenvectors span $\mathbf{Range}\left(W^\top GW\right)$. Hence, we can write $W^\top y = \sum_{i=1}^{\tau}\alpha_i q_i$ and therefore

$$y^\top WW^\top GWW^\top y = \sum_{i=1}^{\tau}\lambda_i\alpha_i^2 \geq \lambda_1\sum_{i=1}^{\tau}\alpha_i^2 = \lambda_1\|W^\top y\|_2^2.$$

Furthermore this bound is tight, as can be seen by selecting $y$ so that $W^\top y = q_1$. □

### 3.6.3 Convergence of the iterates

Subtracting $x^* + t$ from both sides of the update step of Algorithm 2, and letting

$$Z \overset{\text{def}}{=} A^\top S(S^\top AB^{-1}A^\top S)^\dagger S^\top A, \tag{3.45}$$

we obtain the identity

$$x^{k+1} - (x^* + t) = (I - B^{-1}Z)(x^k - (x^* + t)), \tag{3.46}$$

where we used that $t \in \mathbf{Null}\,(A)$. Left multiplying (3.46) by $B^{1/2}$ we see that the residual

$$r^k \overset{\text{def}}{=} B^{1/2}(x^{k+1} - (x^* + t)),$$

satisfies the recurrence

$$r^{k+1} = (I - B^{-1/2}ZB^{-1/2})r^k. \tag{3.47}$$

In view of $\mathbf{E}\,[Z] = A^\top HA$, and taking norms and expectations (in $S$) on both sides of (3.47) gives

$$\begin{aligned}
\mathbf{E}\left[\|r^{k+1}\|_2^2 \,|\, r^k\right] &= \mathbf{E}\left[\|(I - B^{-1/2}ZB^{-1/2})r^k\|_2^2\right] \\
&\overset{(1.27)}{=} \mathbf{E}\left[(r^k)^\top(I - B^{-1/2}ZB^{-1/2})r^k\right] \\
&= \|r^k\|_2^2 - (r^k)^\top B^{-1/2}\mathbf{E}\,[Z]\,B^{-1/2}r^k \\
&= \|r^k\|_2^2 - (r^k)^\top B^{-1/2}A^\top HAB^{-1/2}r^k,
\end{aligned} \tag{3.48}$$

In view of Lemma 30, let $w^k \in \mathbb{R}^m$ be such that $r^k = B^{1/2}(B^{-1}A^\top w^k) = B^{-1/2}A^\top w^k$. Thus

$$\begin{aligned}
(r^k)^\top B^{-1/2}A^\top HAB^{-1/2}r^k &= (w^k)^\top AB^{-1}A^\top HAB^{-1}A^\top w^k \\
&\overset{(\text{Lemma 31})}{\geq} \lambda_{\min}^+(B^{-1/2}A^\top HAB^{-1/2})\cdot\|B^{-1/2}A^\top w^k\|_2^2 \\
&= (1-\rho)\cdot\|r^k\|_2^2,
\end{aligned} \tag{3.49}$$

where we applied Lemma 31 with $W = AB^{-1/2}$ and $G = H$, so that $W^\top GW = B^{-1/2}A^\top HAB^{-1/2}$. Substituting (3.49) into (3.48) gives $\mathbf{E}\left[\|r^{k+1}\|_2^2 \,|\, r^k\right] \leq \rho\cdot\|r^k\|_2^2$. Using the tower property of expectations, we obtain the recurrence

$$\mathbf{E}\left[\|r^{k+1}\|_2^2\right] \leq \rho\cdot\mathbf{E}\left[\|r^k\|_2^2\right].$$

To prove (3.11) it remains to unroll the recurrence.

### 3.6.4 Convergence of the residual

We now prove (3.12). Letting $V_k = \|x^k - x^* - t\|_B^2$, we have

$$\begin{aligned}
\mathbf{E}\left[\|Ax^k - b\|_B\right] &= \mathbf{E}\left[\|A(x^k - x^* - t) + At\|_B\right] \\
&\leq \mathbf{E}\left[\|A(x^k - x^* - t)\|_B\right] + \|At\|_B \\
&\leq \|A\|_B \mathbf{E}\left[\sqrt{V_k}\right] + \|At\|_B \\
&\leq \|A\|_B \sqrt{\mathbf{E}\left[V_k\right]} + \|At\|_B \\
&\overset{(3.11)}{\leq} \|A\|_B \sqrt{\rho^k V_0} + \|At\|_B,
\end{aligned}$$

where in the step preceding the last one we have used Jensen's inequality.

### 3.6.5 Proof of the lower bound

Now we prove (3.14). Using Lemma 1 with $G = H$ and $W = AB^{-1/2}$ gives

$$\mathbf{Range}\left(B^{-1/2}A^\top HAB^{-1/2}\right) = \mathbf{Range}\left(B^{-1/2}A^\top\right),$$

from which we deduce that

$$\begin{aligned}
\mathbf{Rank}\left(A\right) &= \dim\left(\mathbf{Range}\left(A^\top\right)\right) \\
&= \dim\left(\mathbf{Range}\left(B^{-1/2}A^\top\right)\right) \\
&= \dim\left(\mathbf{Range}\left(B^{-1/2}A^\top HAB^{-1/2}\right)\right) \\
&= \mathbf{Rank}\left(B^{-1/2}A^\top HAB^{-1/2}\right).
\end{aligned}$$

Hence, $\mathbf{Rank}\left(A\right)$ is equal to the number of nonzero eigenvalues of $B^{-1/2}A^\top HAB^{-1/2}$, from which we immediately obtain the bound

$$\mathbf{Tr}\left(B^{-1/2}A^\top HAB^{-1/2}\right) \quad \geq \quad \mathbf{Rank}\left(A\right)\, \lambda_{\min}^{+}(B^{-1/2}A^\top HAB^{-1/2}).$$

To conclude the proof, note that $\mathbf{E}\left[Z\right] = A^\top HA$ where $Z$ is defined in (3.45). In order to obtain (3.14), it only remains to combine the above inequality with

$$\mathbf{E}\left[\mathbf{Rank}\left(S^\top A\right)\right] \overset{(1.28)}{=} \mathbf{E}\left[\mathbf{Tr}\left(B^{-1/2}ZB^{-1/2}\right)\right] = \mathbf{Tr}\left(B^{-1/2}A^\top HAB^{-1/2}\right).$$

## 3.7 Proof of Theorem 23

In this section we prove Theorem 23. We dedicate a subsection to each of the three complexity bounds.

### 3.7.1 Dual suboptimality

Since $x^0 \in c + \mathbf{Range}\left(B^{-1}A^\top\right)$, we have $t = 0$ in Theorem 22, and hence (3.11) says that

$$\mathbf{E}\left[U_k\right] \leq \rho^k U_0. \tag{3.50}$$

It remains to apply Proposition 27, which says that $U_k = D(y^*) - D(y^k)$.

### 3.7.2 Primal suboptimality

Letting $U_k = \frac{1}{2}\|x^k - x^*\|_B^2$, we can write

$$\begin{aligned} P(x^k) - OPT &= \tfrac{1}{2}\|x^k - c\|_B^2 - \tfrac{1}{2}\|x^* - c\|_B^2 \\ &= \tfrac{1}{2}\|x^k - x^* + x^* - c\|_B^2 - \tfrac{1}{2}\|x^* - c\|_B^2 \\ &= \tfrac{1}{2}\|x^k - x^*\|_B^2 + (x^k - x^*)^\top B(x^* - c) \\ &\leq U_k + \|B^{1/2}(x^k - x^*)\|_2 \|B^{1/2}(x^* - c)\|_2 \\ &= U_k + \|x^k - x^*\|_B \|x^* - c\|_B \\ &= U_k + 2\sqrt{U_k}\sqrt{OPT}. \end{aligned} \tag{3.51}$$

By taking expectations on both sides of (3.51), and using Jensen's inequality, we obtain

$$\mathbf{E}\left[P(x^k) - OPT\right] \leq \mathbf{E}\left[U_k\right] + 2\sqrt{OPT}\sqrt{\mathbf{E}\left[U_k\right]} \overset{(3.50)}{\leq} \rho^k U_0 + 2\rho^{k/2}\sqrt{OPT \times U_0},$$

which establishes the bound on primal suboptimality (3.16).

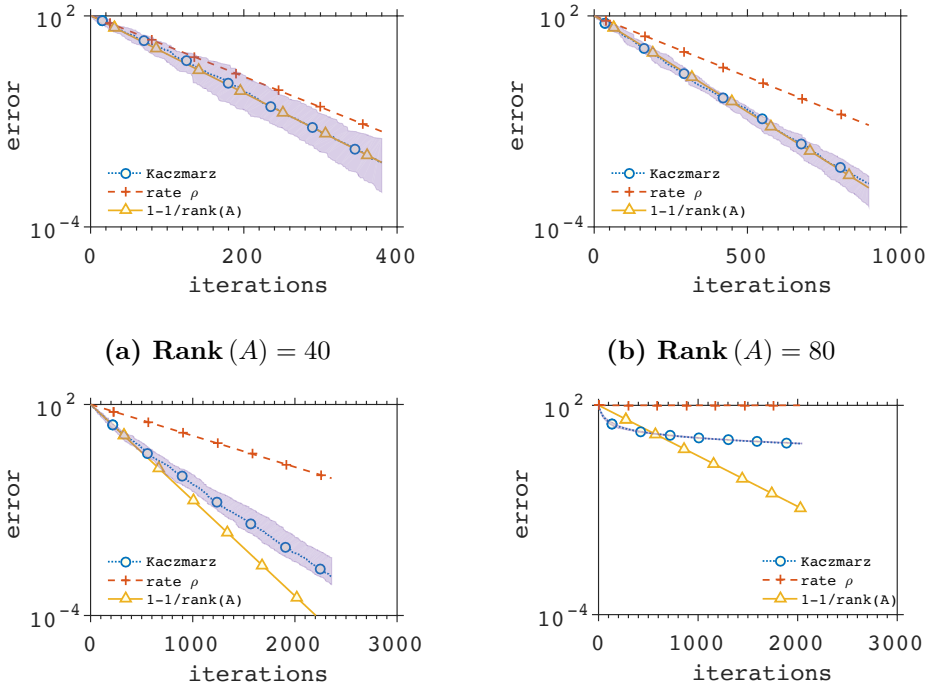
### 3.7.3 Duality gap

Having established rates for primal and dual suboptimality, the rate for the duality gap follows easily:

$$\begin{aligned} \mathbf{E}\left[P(x^k) - D(y^k)\right] &= \mathbf{E}\left[P(x^k) - OPT + OPT - D(y^k)\right] \\ &= \mathbf{E}\left[P(x^k) - OPT\right] + \mathbf{E}\left[OPT - D(y^k)\right] \\ &\overset{(3.15)+(3.16)}{=} 2\rho^k U_0 + 2\rho^{k/2}\sqrt{OPT \times U_0}. \end{aligned}$$

## 3.8 Numerical Experiments: Randomized Kaczmarz Method with Rank-Deficient System

To illustrate some of the novel aspects of our theory, we perform numerical experiments with the Randomized Kaczmarz method (3.30) (or equivalently the randomized coordinate ascent method applied to the dual problem (3.3)) and compare the empirical convergence to the convergence predicted by our theory. We test several randomly generated rank-deficient systems and compare the evolution of the empirical primal error $\|x^k - x^*\|_2^2/\|x^0 - x^*\|_2^2$ to the convergence dictated by the rate $\rho = 1 - \lambda_{\min}^+\left(A^\top A\right)/\|A\|_F^2$ given in (3.32) and the lower bound $1 - 1/\mathbf{Rank}\,(A) \leq \rho$. From Figure 3.1 we can see that the RK method converges despite the fact that the linear systems are rank deficient. While previous results do not guarantee that RK converges for rank-deficient matrices, our theory does as long as the system matrix has no zero rows. Furthermore, we observe in Figure 3.1 that the lower the rank of the system matrix, the faster the convergence of the RK method, and moreover, the closer the empirical convergence is to the convergence dictated by the rate $\rho$ and lower bound on $\rho$. In particular, on the low rank system in Figure 3.1a, the empirical convergence is very close to both the convergence dictated by $\rho$ and the lower bound. While on the full rank system in Figure 3.1d, the convergence dictated by $\rho$ and the lower bound on $\rho$ are no longer an accurate estimate of the empirical convergence.




**(a)** $\mathbf{Rank}\,(A) = 40$ **(b)** $\mathbf{Rank}\,(A) = 80$

**(c)** $\mathbf{Rank}\,(A) = 160$ **(d)** $\mathbf{Rank}\,(A) = 300$

**Figure 3.1:** *Synthetic MATLAB generated problems. Rank deficient matrix* $A = \sum_{i=1}^{\mathbf{Rank}(A)} \sigma_i u_i v_i^\top$ *where* $\sum_{i=1}^{300} \sigma_i u_i v_i^\top = \texttt{rand}(300, 300)$ *is an svd decomposition of a* $300 \times 300$ *uniform random matrix. We repeat each experiment ten times. The blue shaded region is the* 90% *percentile of relative error achieved in each iteration.*

## 3.9 Summary

We have developed a versatile and powerful algorithmic framework for solving linear systems: *stochastic dual ascent (SDA)*. The SDA method finds the projection of a given point, in a fixed but arbitrary Euclidean norm, onto the solution space of the system. Our method is dual in nature, but can also be described in terms of primal iterates via a simple affine transformation of the dual variables. Viewed as a dual method, SDA belongs to a novel class of randomized optimization algorithms: it updates the current iterate by adding the product of a random matrix, drawn independently from a fixed distribution, and a vector. The update is chosen as the best point lying in the random subspace spanned by the columns of this random matrix.

While SDA is the first method of this type, particular choices for the distribution of the random matrix lead to several known algorithms: randomized coordinate descent [68] and randomized Kaczmarz [125] correspond to a discrete distribution over the columns of the identity matrix, randomized Newton method [101] corresponds to a discrete distribution over column submatrices of the identity matrix, and Gaussian descent [121] corresponds to the case when the random matrix is a Gaussian vector.

We equip the method with several complexity results with the same rate of exponential decay in expectation (aka linear convergence) and establish a tight lower bound on the rate. In particular, we prove convergence of primal iterates, dual function values, primal function values, duality gap and of the residual. The method converges under very weak conditions beyond consistency of the linear system. In particular, no rank assumptions on the system matrix are needed. For instance, randomized Kaczmarz method converges linearly as long as the system matrix contains no zero rows.

Further, we show that SDA can be applied to the distributed (average) consensus problem. We recover a standard randomized gossip algorithm as a special case, and show that its complexity is proportional to the number of edges in the graph and inversely proportional to the smallest nonzero eigenvalue of the graph Laplacian. Moreover, we illustrate how our framework can be used to obtain new randomized algorithms for the distributed consensus problem.

Our framework extends to several other problems in optimization and numerical linear algebra. For instance, one can apply it to develop new stochastic algorithms for computing the inverse of a matrix and obtain state-of-the art performance for inverting matrices of huge sizes, which is the subject of the next chapter.

CHAPTER 4

# Randomized Matrix Inversion

## 4.1 Introduction

Here we extend our randomized methods for solving linear systems to methods for inverting matrices. Though there exists applications where one needs the explicit inverse of a matrix[1], inverting a matrix is seldom required. In contrast, calculating the approximate inverse of a matrix finds many applications. Most notably, calculating an approximate inverse finds applications in preconditioning [112] and, if the approximate inverse is guaranteed to be positive definite, then an iterative scheme for inverting a matrix can be used to design variable metric optimization methods. The methods we propose here converge globally and linearly to the inverse matrix and thus are well suited for quickly calculating approximate inverse matrices.

When only an approximate inverse is required, then iterative methods are the methods of choice, for they can terminate the iterative process when the desired accuracy is reached. This can be far more efficient than using an all-or-nothing direct method. Furthermore, iterative methods can make use of an initial estimate of the inverse when available.

The driving motivation of this work is the need to develop algorithms capable of computing the approximate inverse of very large matrices, where standard techniques take an excessive amount of time or simply fail. In particular, using the sketch-and-project technique we develop a family of randomized/stochastic methods for inverting a matrix with specialized variants maintaining symmetry or positive definiteness of the iterates. All methods in the family converge globally (i.e., from any starting point) and linearly (i.e., the error decays exponentially).

As special cases, we obtain stochastic block variants of several quasi-Newton updates, including bad Broyden (BB), good Broyden (GB), Powell-symmetric-Broyden (PSB), Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS). To the best of our knowledge, these are the first stochastic versions of quasi-Newton updates. Moreover, this is the first time that randomized quasi-Newton methods are shown to be iterative methods for inverting a matrix. We also offer a new interpretation of the quasi-Newton methods through a Lagrangian dual viewpoint. This new viewpoint uncovers a fundamental link between quasi-Newton updates and approximate inverse preconditioning.

We develop an adaptive variant of randomized block BFGS, in which we modify the distribution underlying the stochasticity of the method throughout the iterative process to achieve faster convergence. Through extensive numerical experiments with matrices arising from several

[1] for instance when one needs to store a Schur complement or a projection matrix

applications, we demonstrate that AdaRBFGS is highly competitive when compared with the well established Newton-Schulz and minimal residual methods. In particular, on large-scale problems our method outperforms the standard methods by orders of magnitude.

The development of efficient methods for estimating the inverse of very large matrices is a much needed tool for preconditioning and variable metric methods in the advent of the big data era.

### 4.1.1 Chapter outline

In Section 4.2 we summarize the main contributions of this chapter. In Section 4.3 we describe the quasi-Newton methods, which is the main inspiration of our methods. Subsequently, Section 4.4 describes two algorithms, each corresponding to a variant of the inverse equation, for inverting general square matrices. We also provide insightful dual viewpoints for both methods. In Section 4.5 we describe a method specialized to inverting symmetric matrices. Convergence in expectation is examined in Section 4.6, were we consider two types of convergence: the convergence of i) the expected norm of the error, and the convergence of ii) the norm of the expected error. In Section 4.7 we specialize our methods to discrete distributions, and comment on how one may construct a probability distribution leading to better complexity rates (i.e., importance sampling), and how to construct an adaptive probability distribution. In Section 4.8 we detail several instantiations of our family of methods, and their resulting convergence rates. We show how via the choice of the parameters of the method, we obtain *stochastic block variants* of several well known quasi Newton methods. We also describe the simultaneous randomized Kaczmarz method here. Section 4.9 is dedicated to the development of an adaptive variant of our randomized BFGS method, AdaRBFS, for inverting positive definite matrices. Finally, in Section 4.10 we show through numerical tests that AdaRBFGS significantly outperforms state-of-the-art iterative matrix inversion methods on large-scale matrices.

### 4.1.2 Notation

Let $I$ denote the $n \times n$ identity matrix. Let

$$\langle X, Y \rangle_{F(B)} \stackrel{\text{def}}{=} \mathbf{Tr}\left(X^\top B Y B\right),$$

denote the weighted Frobenius inner product, where $X, Y \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ is a symmetric positive definite "weight" matrix. As the trace is invariant under cyclic permutations, a fact we use repeatedly throughout this chapter, we have

$$\|X\|^2_{F(B)} = \mathbf{Tr}\left(X^\top B X B\right) = \mathbf{Tr}\left(B^{1/2} X^\top B X B^{1/2}\right) = \|B^{1/2} X B^{1/2}\|_F^2, \tag{4.1}$$

where we have used the convention $F = F(I)$, since $\|\cdot\|_{F(I)}$ is the standard Frobenius norm. Let $\|\cdot\|_2$ denote the induced operator norm for square matrices defined via

$$\|Y\|_2 \stackrel{\text{def}}{=} \max_{\|v\|_2=1} \|Yv\|_2.$$

Finally, we define the weighted induced norm via

$$\|Y\|_B^* \stackrel{\text{def}}{=} \|B^{1/2} Y B^{1/2}\|_2.$$

### 4.1.3 Previous work

A widely used iterative method for inverting matrices is the Newton-Schulz method [115] introduced in 1933, and its variants which is still subject of ongoing research [70]. The drawback of the Newton-Schulz methods is that they do not converge for any initial estimate. Instead, an initial estimate $X_0$ such that $\|I - X_0 A\|_2 < 1$ is required to guarantee convergence. Though note that such a $X_0$ always exists[2]. The Newton-Schulz method enjoys local quadratic convergence. As has been observed before [93], and as we observe in our numerical experiments, the Newton-Schulz method can experience slow initial convergence before the asymptotic second order convergence rate sets in. This is contrast with the methods we present here that enjoy global linear convergence and a fast initial convergence.

Bingham [10] describes a method that uses the characteristic polynomial to recursively calculate the inverse, though it requires the calculating the coefficients of the polynomial when initiated, which is costly, and the method has fallen into disuse. Goldfarb [42] uses Broyden's method [15] for iteratively inverting matrices. Our methods include a stochastic variant of Broyden's method.

The approximate inverse preconditioning ($AIP$) methods [19, 112, 46, 6] calculate an approximate inverse by minimizing in $X \in \mathbb{R}^{n \times n}$ the residual $\|XA - I\|_F$ (Frobenius norm). They accomplish this by applying a number of iterations of the steepest descent or minimal residual method. A considerable drawback of the AIP methods, is that the approximate inverses are not guaranteed to be positive definite nor symmetric, even when $A$ is both. A solution to the lack of symmetry is to "symmetrize" the estimate between iterations, but then it is difficult to guarantee the quality of the new symmetric estimate. Another solution is to calculate directly a factored form $LL^\top = X$ and minimize in $L$ the residual $\|L^\top AL - I\|_F$. But now this residual is a non-convex function, and is thus difficult to minimize. A variant of our method naturally maintains symmetry of the iterates.

[2] Take for example $X_0 = \alpha A^T$ with $0 < \alpha < 2/\|A\|_2$.

## 4.2 Contributions and Overview

In this section we describe the main contributions of this chapter.

### 4.2.1 New algorithms

We develop a novel and surprisingly simple family of stochastic algorithms for inverting matrices. The problem of finding the inverse of an $n \times n$ invertible matrix $A$ can be characterized as finding the solution to either one of the two *inverse equations*[3] $AX = I$ or $XA = I$. Our methods make use of randomized sketching [96, 51, 95, 97] to reduce the dimension of the inverse equations in an iterative fashion. To the best of our knowledge, these are the first stochastic algorithms for inverting a matrix with global complexity rates.

In particular, our nonsymmetric method (Algorithm 3) is based on the inverse equation $AX = I$, and performs the *sketch-and-project* iteration

$$X_{k+1} = \arg\min_{X \in \mathbb{R}^{n\times n}} \tfrac{1}{2}\|X - X_k\|^2_{F(B)} \quad \text{subject to} \quad S^\top AX = S^\top, \tag{4.2}$$

where $S \in \mathbb{R}^{n\times q}$ is a random matrix drawn in an i.i.d. fashion from a fixed distribution $\mathcal{D}$, and $B \in \mathbb{R}^{n\times n}$ is symmetric positive definite. The distribution $\mathcal{D}$ and matrix $B$ are the parameters of the method. Note that if we choose $q \ll n$, the constraint in the projection problem (4.2) will be of a much smaller dimension than the original inverse equation, and hence the iteration (4.2) will become cheap.

In an analogous way, we design a method based on the inverse equation $XA = I$ (Algorithm 4). By adding the symmetry constraint $X = X^\top$ to (4.2), we obtain Algorithm 5—a specialized method for inverting symmetric matrices capable of maintaining symmetric iterates.

### 4.2.2 Dual formulation

Besides the *primal formulation* described in Section 4.2.1—*sketch-and-project*—we also provide *dual formulations* of all three methods (Algorithms 3, 4 and 5). For instance, the dual formulation of (4.2) is

$$X_{k+1} = \arg_X \min_{X \in \mathbb{R}^{n\times n}, Y \in \mathbb{R}^{n\times q}} \tfrac{1}{2}\|X_k - A^{-1}\|^2_{F(B)} \quad \text{subject to} \quad X = X_k + B^{-1}A^\top SY^\top. \tag{4.3}$$

We call the dual formulation *constrain-and-approximate* as one seeks to perform the best approximation of the inverse (with respect to the weighted Frobenius distance) while constraining the search to a random affine space of matrices passing through $X_k$. While the projection (4.3) cannot be performed directly since $A^{-1}$ is not known, it can be performed indirectly via the equivalent primal formulation (4.2).

[3] One may use other equations uniquely defining the inverse, such as $AXA = A$, but we do not explore these in this thesis.

| | |
|---|---|
| $\mathbf{E}\left[X_{k+1} - A^{-1}\right] = \left(I - B^{-1}\mathbf{E}\left[Z\right]\right)\mathbf{E}\left[X_{k+1} - A^{-1}\right]$ | Theorem 4.1 |
| $\left\|\mathbf{E}\left[X_{k+1} - A^{-1}\right]\right\|_B^* \leq \rho \ \cdot \ \left\|\mathbf{E}\left[X_{k+1} - A^{-1}\right]\right\|_B^*$ | Theorem 35 |
| $\mathbf{E}\left[\left\|X_{k+1} - A^{-1}\right\|_{F(B)}^2\right] \leq \rho \ \cdot \ \mathbf{E}\left[\left\|X_{k+1} - A^{-1}\right\|_{F(B)}^2\right]$ | Theorem 36 |

**Table 4.1:** *Our main complexity results.*

### 4.2.3 Quasi-Newton updates and approximate inverse preconditioning

As we will discuss in Section 4.3, through the lens of the sketch-and-project formulation, Algorithm 5 can be seen as *randomized block extension of the quasi-Newton updates* [15, 40, 43, 119]. We distinguish here between quasi-Newton methods, which are algorithms used in optimization, and quasi-Newton updates, which are the *matrix-update* rules used in the quasi-Newton methods. Standard quasi-Newton updates work with $q = 1$ ("block" refers to the choice $q > 1$) and $S$ chosen in a deterministic way, depending on the sequence of iterates of the underlying optimization problem. To the best of our knowledge, this is the first time stochastic versions of quasi-Newton updates were designed and analyzed. On the other hand, through the lens of the constrain-and-approximate formulation, our methods can be seen as *new variants of the approximate inverse preconditioning (*AIP*) methods* [19, 112, 46, 6]. Moreover, the equivalence between these two formulations reveals deep connections between what were before seen as distinct fields: the quasi-Newton and AIP literature. Our work also provides several new insights for *deterministic* quasi-Newton updates. For instance, the *bad Broyden update* [15, 56] is a particular best rank-1 update that minimizes the distance to the inverse of $A$ under the Frobenius norm. The *BFGS update* [15, 40, 43, 119] can be seen as a projection of $A^{-1}$ onto a space of rank-2 symmetric matrices. To the best of our knowledge, this has not been observed before.

### 4.2.4 Complexity

Our framework leads to global linear convergence (i.e., exponential decay) under very weak assumptions on $\mathcal{D}$. In particular, we provide an explicit convergence rate $\rho$ for the exponential decay of the norm of the expected error of the iterates (line 2 of Table 4.1) and the expected norm of the error (line 3 of Table 4.1), where $\rho$ is the same rate provided in Chapter 2, namely

$$\rho = 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}), \tag{4.4}$$

where

$$Z \overset{\text{def}}{=} A^\top S(S^\top AB^{-1}A^\top S)^{-1}SA^\top.$$

This sets our method apart from current methods for inverting matrices that lack global guarantees, such as Newton-Schulz, or the self-conditioning variants of the minimal residual method. By optimizing an upper bound on (4.4), we also obtain a new practical importance sampling. This should be contrasted with the optimized rate in Section 2.6.1 which results in a SDP, which is rarely practical to solve.

### 4.2.5 Adaptive randomized BFGS

We develop an additional highly efficient method—adaptive randomized BFGS (AdaRBFGS)—for calculating an approximate inverse of *positive definite matrices*. In extensive numeric tests in Section 4.10 we show that the AdaRBFGS method greatly outperforms the Newton-Schulz and approximate inverse preconditioning methods at obtaining an approximate inverse (with a relative precision of 99%). Furthermore, the AdaRBFGS method preserves positive definiteness, a quality not present in previous methods. Therefore, AdaRBFGS can be used to precondition positive definite systems and to design new variable-metric optimization methods. Since the inspiration behind this method comes from the desire to design an *optimal adaptive* distribution for $S$ by examining the complexity rate $\rho$, this work also highlights the importance of developing algorithms with explicit convergence rates.

### 4.2.6 Extensions

This work opens up many possible avenues for extensions. For instance, new efficient methods could be achieved by experimenting and analyzing through our framework with different sophisticated sketching matrices $S$, such as the Walsh-Hadamard matrix [75, 96]. Furthermore, our method produces low rank estimates of the inverse and can be adapted to calculate low rank estimates of any matrix. Our methods can be applied to singular matrices, in which case they converge to a particular pseudo-inverse.

Our results can be used to push forward work into stochastic variable metric methods. Such as the work by Leventhal and Lewis [69], where they present a randomized iterative method for estimating Hessian matrices that converge in expectation with known convergence rates for any initial estimate. Stich et al. [122] use Leventhal and Lewis' method to design a stochastic variable metric method for black-box minimization, with explicit convergence rates, and promising numeric results. We leave these and other extensions to future work.

## 4.3 Randomization of Quasi-Newton Updates

Our methods are inspired by, and in some cases can be considered to be, randomized block variants of the quasi-Newton updates. In this section we explain how our algorithms arise naturally from the quasi-Newton setting. Readers familiar with quasi-Newton methods may jump ahead to Section 4.3.3.

### 4.3.1 Quasi-Newton methods

A problem of fundamental interest in optimization is the unconstrained minimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{4.5}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is a sufficiently smooth function. Quasi-Newton (QN) methods, first proposed by Davidon in 1959 [24], are an extremely powerful and popular class of algorithms for solving this problem, especially in the regime of moderately large $n$. In each iteration of a QN method, one approximates the function locally around the current iterate $x_k$ by a quadratic of the form

$$f(x_k+s)\approx f(x_k)+(\nabla f(x_k))^\top s+\frac{1}{2}s^\top B_k s, \tag{4.6}$$

where $B_k$ is a suitably chosen approximation of the Hessian: $B_k\approx\nabla^2 f(x_k)$. After this, a direction $s_k$ is computed by minimizing the quadratic approximation in $s$, obtaining

$$s_k=-B_k^{-1}\nabla f(x_k), \tag{4.7}$$

if the matrix $B_k$ is invertible. The next iterate is then set to

$$x_{k+1}=x_k+h_k,\quad h_k=\alpha_k s_k,$$

for a suitable choice of stepsize $\alpha_k$, often chosen by a line-search procedure (i.e., by approximately minimizing $f(x_k+\alpha s_k)$ in $\alpha$).

Gradient descent arises as a special case of this process by choosing $B_k$ to be constant throughout the iterations. A popular choice is $B_k=LI$, where $I$ is the identity matrix and $L\in\mathbb{R}_+$ is the Lipschitz constant of the gradient of $f$. In such a case, the quadratic approximation (4.6) is a global upper bound on $f(x_k+s)$, which means that $f(x_k+s_k)$ is guaranteed to be at least as good (i.e., smaller or equal) as $f(x_k)$, leading to guaranteed descent. Newton's method also arises as a special case: by choosing $B_k=\nabla^2 f(x_k)$. These two algorithms are extreme cases on the opposite end of a spectrum. Gradient descent benefits from a trivial update rule for $B_k$ and from cheap iterations due to the fact that no linear systems need to be solved. However, curvature information is largely ignored, which slows down the practical convergence of the method. Newton's method utilizes the full curvature information contained in the Hessian, but requires the computation of the Hessian in each step, which is expensive for large $n$. QN methods aim to find a sweet spot on the continuum between these two extremes. In particular,

the QN methods choose $B_{k+1}$ to be a matrix for which the *secant equation* is satisfied:

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k). \tag{4.8}$$

The basic reasoning behind this requirement is the following: if $f$ is a convex quadratic then the Hessian matrix satisfies the secant equation for all pairs of vectors $x_{k+1}$ and $x_k$. If $f$ is not a quadratic, the reasoning is as follows. Using the fundamental theorem of calculus, we have that

$$\left(\int_0^1 \nabla^2 f(x_k + th_k)\, dt\right)(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k).$$

By selecting $B_{k+1}$ that satisfies the secant equation, we are enforcing that $B_{k+1}$ mimics the action of the integrated Hessian along the line segment joining $x_k$ and $x_{k+1}$. Unless $n = 1$, the secant equation (4.8) does not have a unique solution in $B_{k+1}$. All QN methods differ only in which particular solution is used. The formulas transforming $B_k$ to $B_{k+1}$ are called *QN updates*.

Since these matrices are used to compute the direction $s_k$ via (4.7), it is often more reasonable to instead maintain a sequence of inverses $X_k = B_k^{-1}$. By multiplying both sides of (4.8) by $X_{k+1}$, we arrive at the *secant equation for the inverse:*

$$X_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k. \tag{4.9}$$

The most popular classes of QN updates choose $X_{k+1}$ as the closest matrix to $X_k$, in a suitable norm (usually a weighted Frobenius norm with various weight matrices), subject to the secant equation, often with an explicit symmetry constraint:

$$X_{k+1} = \arg\min_{X \in \mathbb{R}^{n\times n}} \left\{ \|X - X_k\| \;:\; Xy_k = h_k,\; X = X^\top \right\}, \tag{4.10}$$

where $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$,

### 4.3.2 Quasi-Newton updates

Consider now problem (4.5) with the quadratic objective

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x + c, \tag{4.11}$$

where $A$ is an $n \times n$ symmetric positive definite matrix, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Granted, this is not a typical problem for which QN methods would be used by a practitioner. Indeed, the Hessian of $f$ does not change, and hence one *does not have to* track it. The problem can simply be solved by setting the gradient to zero, which leads to the system $Ax = b$, the solution being $x_* = A^{-1}b$. As solving a linear system is much simpler than computing the inverse $A^{-1}$, approximately tracking the (inverse) Hessian of $f$ along the path of the iterates $\{x_k\}$—the basic strategy of all QN methods—seems like too much effort for what is ultimately a much simpler problem.

However, and this is one of the main insights of this work, instead of viewing QN methods as optimization algorithms, we can alternatively interpret them as iterative algorithms producing a sequence of matrices, $\{B_k\}$ or $\{X_k\}$, hopefully converging to some matrix of interest. In particular, one would hope that if a QN method is applied to the quadratic problem (4.11), with

any symmetric positive definite initial guess $X_0$, then the sequence $\{X_k\}$ converges to $A^{-1}$.

For $f$ given by (4.11), the QN updates of the minimum distance variety given by (4.10) take the form

$$X_{k+1} = \arg\min_{X \in \mathbb{R}^{n\times n}} \left\{ \|X - X_k\| \ : \ XAh_k = h_k,\ X = X^\top \right\}. \tag{4.12}$$

### 4.3.3 Randomized quasi-Newton updates

While the motivation for our work comes from optimization, having arrived at the update (4.12), we can dispense of some of the implicit assumptions and propose and analyze a wider class of methods. In particular, in this chapter we analyze a large class of *randomized algorithms* of the type (4.12), where the vector $h_k$ is replaced by a random matrix $S$ and $A$ is *any* invertible, and not necessarily symmetric or positive definite matrix. This constitutes a randomized block extension of the QN updates.

## 4.4 Inverting Nonsymmetric Matrices

In this chapter we are concerned with the development and complexity analysis of a family of stochastic algorithms for computing the inverse of a nonsingular matrix $A \in \mathbb{R}^{n\times n}$. The starting point in the development of our methods is the simple observation that the inverse $A^{-1}$ is the (unique) solution of a linear matrix equation, which we shall refer to as *inverse equation*:

$$AX = I. \tag{4.13}$$

Alternatively, one can use the inverse equation $XA = I$ instead. Since (4.13) is difficult to solve directly, our approach is to iteratively solve a small randomly relaxed version of (4.13). That is, we choose a random matrix $S \in \mathbb{R}^{n\times q}$, with $q \ll n$, and instead solve the following *sketched inverse equation*:

$$S^\top AX = S^\top. \tag{4.14}$$

If we base the method on the second inverse equation, the sketched inverse equation $XAS = S$ should be used instead. Note that $A^{-1}$ satisfies (4.14). If $q \ll n$, the sketched inverse equation is of a much smaller dimension than the original inverse equation, and hence easier to solve. However, the equation will no longer have a unique solution and in order to design an algorithm, we need a way of picking a particular solution. Our algorithm defines $X_{k+1}$ to be the solution that is closest to the current iterate $X_k$ in a weighted Frobenius norm. This is repeated in an iterative fashion, each time drawing $S$ independently from a fixed distribution $\mathcal{D}$. The distribution $\mathcal{D}$ and the matrix $B$ can be seen as parameters of our method. The flexibility of being able to adjust $\mathcal{D}$ and $B$ is important: by varying these parameters we obtain various specific instantiations of the generic method, with varying properties and convergence rates. This gives the practitioner the flexibility to adjust the method to the structure of $A$, to the computing environment and so on.

### 4.4.1 Projection viewpoint: sketch-and-project

The next iterate $X_{k+1}$ is the nearest point to $X_k$ that satisfies a *sketched* version of the inverse equation:

$$\boxed{X_{k+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \frac{1}{2}\|X - X_k\|^2_{F(B)} \quad \text{subject to} \quad S^\top AX = S^\top} \tag{4.15}$$

In the special case when $S = I$, the only such matrix is the inverse itself, and (4.15) is not helpful. However, if $S$ is "simple", (4.15) will be easy to compute and the hope is that through a sequence of such steps, where the matrices $S$ are sampled in an i.i.d. fashion from some distribution, $X_k$ will converge to $A^{-1}$.

Alternatively, we can sketch the equation $XA = I$ and project onto $XAS = S$:

$$\boxed{X_{k+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \frac{1}{2}\|X - X_k\|^2_{F(B)} \quad \text{subject to} \quad XAS = S} \tag{4.16}$$

While the method (4.15) sketches the rows of $A$, the method (4.15) sketches the columns of $A$. Thus we refer to (4.15) as the row variant and to (4.16) as the column variant. The two

variants (4.15) and (4.16) both converge to the inverse of $A$, as will be established in Section 4.6.

If $A$ is singular, then the iterates of (4.16) converge to the left inverse, while the iterates of (4.15) converge to the right inverse, an observation we leave to future work.

### 4.4.2 Optimization viewpoint: constrain-and-approximate

The row sketch-and-project method can be cast in an apparently different yet equivalent viewpoint:

$$\boxed{X_{k+1} = \arg_X \min_{X\in\mathbb{R}^{n\times n}, Y\in\mathbb{R}^{n\times q}} \frac{1}{2}\|X - A^{-1}\|^2_{F(B)} \quad \text{subject to} \quad X = X_k + B^{-1}A^\top SY^\top} \tag{4.17}$$

In this viewpoint, at each iteration (4.17), we select a random affine space that passes through $X_k$. After that, we select the point in this space that is as close as possible to the inverse. This random search space is special in that, independently of the input pair $(B, S)$ we can efficiently compute the projection of $A^{-1}$ onto this space, without knowing $A^{-1}$ explicitly.

The column variant (4.16) also has an equivalent constrain-and-approximate formulation:

$$\boxed{X_{k+1} = \arg_X \min_{X\in\mathbb{R}^{n\times n}, Y\in\mathbb{R}^{n\times q}} \frac{1}{2}\|X - A^{-1}\|^2_{F(B)} \quad \text{subject to} \quad X = X_k + YS^\top A^\top B^{-1}} \tag{4.18}$$

These two variants (4.17) and (4.18) can be viewed as new variants of the approximate inverse preconditioner (AIP) methods [6, 46, 64, 60]. The AIP methods are a class of methods for computing approximate inverses of $A$ by minimizing $\|XA - I\|_F$ via iterative optimization algorithms. In particular, the AIP methods use variants of the steepest descent or a minimal residual method to minimize $\|XA - I\|_F$. The idea behind the AIP methods is to minimize the distance of $X$ from $A^{-1}$ in some sense. Our variants do just that, but under a weighted Frobenius norm. Furthermore, our methods project onto a randomly generated affine space instead of employing steepest descent of a minimal residual method.

### 4.4.3 Equivalence

We now prove that (4.15) and (4.16) are equivalent to (4.17) and (4.18), respectively, and give their explicit solution.

**Theorem 32.** *The viewpoints* (4.15) *and* (4.17) *are equivalent to* (4.16) *and* (4.18)*, respectively. Furthermore, if $S$ has full column rank, then the explicit solution to* (4.15) *is*

$$\boxed{X_{k+1} = X_k + B^{-1}A^\top S(S^\top AB^{-1}A^\top S)^{-1}S^\top(I - AX_k)} \tag{4.19}$$

*and the explicit solution to* (4.16) *is*

$$\boxed{X_{k+1} = X_k + (I - X_kA^\top)S(S^\top A^\top B^{-1}AS)^{-1}S^\top A^\top B^{-1}} \tag{4.20}$$

*Proof.* We will prove all the claims for the row variant, that is, we prove that (4.15) are (4.17) equivalent and that their solution is given by (4.19). The remaining claims, that (4.16) are (4.18) are equivalent and that their solution is given by (4.20), follow with analogous arguments.

It suffices to consider the case when $B = I$, as we can perform a change of variables to recover the solution for any $B$. Indeed, in view of (4.1), with the change of variables

$$\hat{X} \overset{\text{def}}{=} B^{1/2}XB^{1/2}, \quad \hat{X}_k \overset{\text{def}}{=} B^{1/2}X_kB^{1/2}, \quad \hat{A} \overset{\text{def}}{=} B^{-1/2}AB^{-1/2} \quad \text{and} \quad \hat{S} \overset{\text{def}}{=} B^{1/2}S, \tag{4.21}$$

(4.15) becomes

$$\min_{\hat{X}\in\mathbb{R}^{n\times n}} \frac{1}{2}\|\hat{X} - \hat{X}_k\|_F^2 \quad \text{subject to} \quad \hat{S}^\top\hat{A}\hat{X} = \hat{S}^\top. \tag{4.22}$$

Moreover, if we let $\hat{Y} = B^{1/2}Y$, then (4.17) becomes

$$\min_{\hat{X}\in\mathbb{R}^{n\times n},\hat{Y}\in\mathbb{R}^{n\times q}} \frac{1}{2}\|\hat{X} - \hat{A}^{-1}\|_F^2 \quad \text{subject to} \quad \hat{X} = \hat{X}_k + \hat{A}^\top\hat{S}\hat{Y}^\top. \tag{4.23}$$

By substituting the constraint in (4.23) into the objective function, then differentiating to find the stationary point, we obtain that

$$\hat{X} = \hat{X}_k + \hat{A}^\top\hat{S}(\hat{S}^\top\hat{A}\hat{A}^\top\hat{S})^{-1}\hat{S}^\top(I - \hat{A}\hat{X}_k), \tag{4.24}$$

is the solution to (4.23). After changing the variables back using (4.21), the update (4.24) becomes (4.44).

Now we prove the equivalence of (4.22) and (4.23) using Lagrangian duality. The sketch-and-project viewpoint (4.22) has a convex quadratic objective function with linear constraints, thus strong duality holds. Introducing Lagrangian multiplier $\hat{Y} \in \mathbb{R}^{n\times q}$, the Langrangian dual of (4.22) is given by

$$L(\hat{X}, \hat{Y}) = \frac{1}{2}\|\hat{X} - \hat{X}_k\|_F^2 - \left\langle \hat{Y}^\top, \hat{S}^\top\hat{A}(\hat{X} - \hat{A}^{-1})\right\rangle_F. \tag{4.25}$$

Clearly

$$(4.22) = \min_{X\in\mathbb{R}^{n\times n}} \max_{\hat{Y}\in\mathbb{R}^{n\times q}} L(\hat{X}, \hat{Y}).$$

We will now prove that

$$(4.23) = \max_{\hat{Y}\in\mathbb{R}^{n\times q}} \min_{X\in\mathbb{R}^{n\times n}} L(\hat{X}, \hat{Y}),$$

thus proving that (4.22) and (4.23) are equivalent by strong duality. Differentiating the Lagrangian in $\hat{X}$ and setting to zero gives

$$\hat{X} = \hat{X}_k + \hat{A}^\top\hat{S}\hat{Y}^\top. \tag{4.26}$$

Substituting back into (4.25) gives

$$\begin{aligned} L(\hat{X}, \hat{Y}) &= \frac{1}{2}\|\hat{A}^\top\hat{S}\hat{Y}^\top\|_F^2 - \left\langle \hat{A}^\top\hat{S}\hat{Y}^\top, \hat{X}_k + \hat{A}^\top\hat{S}\hat{Y}^\top - \hat{A}^{-1}\right\rangle_F \\ &= -\frac{1}{2}\|\hat{A}^\top\hat{S}\hat{Y}^\top\|_F^2 - \left\langle \hat{A}^\top\hat{S}\hat{Y}^\top, \hat{X} - \hat{A}^{-1}\right\rangle_F. \end{aligned}$$

**Algorithm 3** Stochastic Iterative Matrix Inversion (SIMI) – nonsymmetric row variant

**input:** invertible matrix $A \in \mathbb{R}^{n\times n}$
**parameters:** $\mathcal{D}$ = distribution over random matrices; positive definite matrix $B \in \mathbb{R}^{n\times n}$
**initialize:** arbitrary square matrix $X_0 \in \mathbb{R}^{n\times n}$
**for** $k = 0, 1, 2, \ldots$ **do**
    Sample an independent copy $S \sim \mathcal{D}$
    Compute $\Lambda = S(S^\top A B^{-1} A^\top S)^{-1} S^\top$
    $X_{k+1} = X_k + B^{-1}A^\top \Lambda (I - AX_k)$ ▷ This is equivalent to (4.15) and (4.17)
**output:** last iterate $X_k$

Adding $\pm\frac{1}{2}\|\hat{X}_k - \hat{A}^{-1}\|_F^2$ to the above gives

$$L(\hat{X}, \hat{Y}) = -\frac{1}{2}\|\hat{A}^\top \hat{S}\hat{Y}^\top + \hat{X}_k - \hat{A}^{-1}\|_F^2 + \frac{1}{2}\|\hat{X}_k - \hat{A}^{-1}\|_F^2.$$

Finally, substituting (4.26) into the above, minimizing in $\hat{X}$ then maximizing in $\hat{Y}$, and dispensing of the term $\frac{1}{2}\|\hat{X}_k - \hat{A}^{-1}\|_F^2$ as it does not depend on $\hat{Y}$ nor $\hat{X}$, we have that the dual problem is

$$\max_{\hat{Y}} \min_{\hat{X}} L(\hat{X}, \hat{Y}) = \min_{\hat{X},\hat{Y}} \frac{1}{2}\|\hat{X} - \hat{A}^{-1}\|_F^2 \quad \text{subject to} \quad \hat{X} = \hat{X}_k + \hat{A}^\top \hat{S}\hat{Y}^\top.$$

It now remains to change variables using (4.21) and set $Y = B^{-1/2}\hat{Y}$ to obtain (4.17). □

Based on Theorem 32, we can summarize the methods described in this section as Algorithm 3 and Algorithm 4.

**Algorithm 4** Stochastic Iterative Matrix Inversion (SIMI) – nonsymmetric column variant

**input:** invertible matrix $A \in \mathbb{R}^{n\times n}$
**parameters:** $\mathcal{D}$ = distribution over random matrices; positive definite matrix $B \in \mathbb{R}^{n\times n}$
**initialize:** arbitrary square matrix $X_0 \in \mathbb{R}^{n\times n}$
**for** $k = 0, 1, 2, \ldots$ **do**
    Sample an independent copy $S \sim \mathcal{D}$
    Compute $\Lambda = S(S^\top A^\top B^{-1} A S)^{-1} S^\top$
    $X_{k+1} = X_k + (I - X_k A^\top)\Lambda A^\top B^{-1}$ ▷ This is equivalent to (4.16) and (4.18)
**output:** last iterate $X_k$

The explicit formulas (4.19) and (4.20) for (4.15) and (4.16) allow us to efficiently implement these methods, and facilitate convergence analysis. In particular, we can now see that the convergence analysis of (4.20) will follow trivially from analyzing (4.19). This is because (4.19) and (4.20) differ only in terms of a transposition. That is, transposing (4.20) gives

$$X_{k+1}^\top = X_k^\top + B^{-1}AS(S^\top A^\top B^{-1} A S)^{-1} S^\top (I - A^\top X_k^\top),$$

which is the solution to the row variant of the sketch-and-project viewpoint but where the equation

$A^\top X^\top = I$ is sketched instead of $AX = I$. Thus, since the weighted Frobenius norm is invariant under transposition, it suffices to study the convergence of (4.19), then the convergence of (4.20) follows by simply swapping the role of $A$ for $A^\top$. We collect this observation is the following remark.

**Remark 33.** *The expression for the rate of convergence of Algorithm 4 is the same as the expression for the rate of convergence of Algorithm 3, but with every occurrence of $A$ swapped for $A^\top$.*

### 4.4.4 Relation to multiple linear systems

Any iterative method for solving linear systems can be applied to the $n$ linear systems that define the inverse through $AX = I$ to obtain an approximate inverse. Though not all methods for solving linear systems can be applied to solve these $n$ linear systems simultaneously, that is calculating each column of $X$ simultaneously, which is necessary for an efficient matrix inversion method.

The sketch-and-project methods we described in Chapters 2 and 3 can be easily and efficiently generalized to inverting a matrix, and the resulting method is equivalent to our row variant method (4.15) and (4.17). To show this, we perform the change of variables $\hat{X}_k = X_k B^{1/2}$, $\hat{A} = B^{-1/2}A$ and $\hat{S} = B^{1/2}S$ then (4.15) becomes

$$\hat{X}_{k+1} \overset{\text{def}}{=} X_{k+1}B^{1/2} = \arg\min_{\hat{X}\in\mathbb{R}^{n\times n}} \frac{1}{2}\|B^{1/2}(\hat{X} - \hat{X}_k)\|_F^2 \quad \text{subject to} \quad \hat{S}^\top \hat{A}\hat{X} = \hat{S}^\top.$$

The above is a separable problem and each column of $\hat{X}_{k+1}$ can be calculated separately. Let $\hat{x}_{k+1}^i$ be the $i$th column of $\hat{X}_{k+1}$ which can be calculated through

$$\hat{x}_{k+1}^i = \arg\min_{\hat{x}\in\mathbb{R}^n} \frac{1}{2}\|B^{1/2}(\hat{x} - \hat{x}_k^i)\|_2^2 \quad \text{subject to} \quad \hat{S}^\top \hat{A}\hat{x} = \hat{S}^\top e_i.$$

The above is exactly an iteration of the sketch-and-project method (2.5) applied to the system $\hat{A}\hat{x} = e_i$. Thus the convergence results established in [51] carry over to our row variant (4.15) and (4.17). In particular, the theory in [51] proves that the expected norm difference of each column of $B^{1/2}X_k$ converges to $B^{1/2}A^{-1}$ with rate $\rho$ as defined in (4.4). This equivalence breaks down when we impose additional matrix properties through constraints, such as symmetry.

## 4.5 Inverting Symmetric Matrices

When $A$ is symmetric, it may be useful to maintain symmetry in the iterates, in which case the nonsymmetric methods—Algorithms 3 and 4—have an issue, as they do not guarantee that the iterates are symmetric. However, we can modify (4.15) by adding a symmetry constraint. The resulting *symmetric* method naturally maintains symmetry in the iterates.

### 4.5.1 Projection viewpoint: sketch-and-project

The new iterate $X_{k+1}$ is the result of projecting $X_k$ onto the space of matrices that satisfy a sketched inverse equation and that are also symmetric, that is

$$\boxed{X_{k+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \frac{1}{2}\|X - X_k\|^2_{F(B)} \quad \text{subject to} \quad S^\top AX = S^\top, \quad X = X^\top} \tag{4.27}$$

See Figure 4.1 for an illustration of the symmetric update (4.27).

This viewpoint can be seen as a randomized block version of the quasi-Newton methods [43, 55], as detailed in Section 4.3. The flexibility in using a weighted norm is important for choosing a norm that better reflects the geometry of the problem. For instance, when $A$ is symmetric positive definite, it turns out that $B = A$ results in a good method. This added freedom of choosing an appropriate weighting matrix has proven very useful in the quasi-Newton literature, in particular, the highly successful BFGS method [15, 40, 43, 119] selects $B$ as an estimate of the Hessian matrix.

### 4.5.2 Optimization viewpoint: constrain-and-approximate

The viewpoint (4.27) also has an interesting dual viewpoint:

$$\boxed{X_{k+1} = \arg_X \min_{X\mathbb{R}^{n\times n}, Y\in\mathbb{R}^{n\times q}} \frac{1}{2}\|X - A^{-1}\|^2_{F(B)} \quad \text{subject to} \quad X = X_k + \frac{1}{2}(YS^\top AB^{-1} + B^{-1}A^\top SY^\top)} \tag{4.28}$$

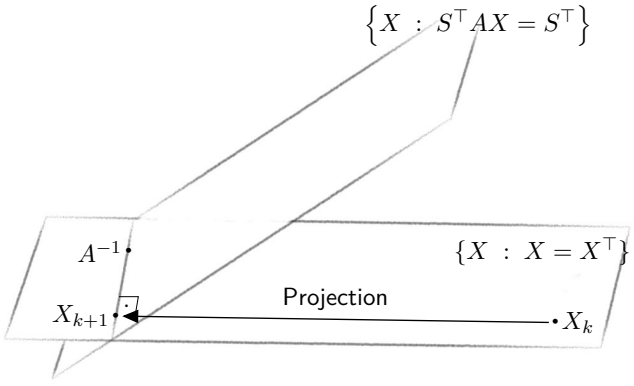



**Figure 4.1:** *The new estimate $X_{k+1}$ is obtained by projecting $X_k$ onto the affine space formed by intersecting $\{X \,:\, X = X^\top\}$ and $\left\{X \,:\, S^\top AX = S^\top\right\}$.*

The minimum is taken over matrices $X \in \mathbb{R}^{n\times n}$ and $Y \in \mathbb{R}^{n\times q}$. The next iterate $X_{k+1}$ is the best approximation to $A^{-1}$ restricted to a random affine space of symmetric matrices. Furthermore, (4.28) is a symmetric equivalent of (4.17); that is, the constraint in (4.28) is the result of projecting the constraint in (4.17) onto the space of symmetric matrices.

When $A$ is symmetric positive definite and we choose $B = A$ in (4.17) and (4.18), then

$$\|X - A^{-1}\|^2_{F(A)} = \mathbf{Tr}\left((X - A^{-1})A(X - A^{-1})A\right) = \|XA - I\|^2_F.$$

The above is exactly the objective function used in most approximate inverse preconditioners (AIP) [6, 46, 64, 60].

### 4.5.3 Equivalence

We now prove that the two viewpoints (4.27) and (4.28) are equivalent, and show their explicit solution.

**Theorem 34.** *If $A$ and $X_k$ are symmetric, then the viewpoints* (4.27) *and* (4.28) *are equivalent. That is, they define the same $X_{k+1}$. Furthermore, if $S$ has full column rank, then the explicit solution to* (4.27) *and* (4.28) *is*

$$\boxed{X_{k+1} = X_k - (X_kAS - S)\Lambda S^\top AB^{-1} + B^{-1}AS\Lambda(S^\top AX_k - S^\top)\left(AS\Lambda S^\top AB^{-1} - I\right)} \quad (4.29)$$

*where $\Lambda \overset{def}{=} (S^\top AB^{-1}AS)^{-1}$.*

*Proof.* We first prove the equivalence of (4.27) and (4.28) using Lagrangian duality. It suffices to prove the claim for $B = I$ as we did in the proof of Theorem 32, since using the change of variables (4.21) applied to (4.27) we have that (4.27) is equivalent to

$$\min_{\hat{X}\in\mathbb{R}^{n\times n}} \frac{1}{2}\|\hat{X} - \hat{X}_k\|^2_F \quad \text{subject to} \quad \hat{S}^\top\hat{A}\hat{X} = \hat{S}^\top, \quad \hat{X} = \hat{X}^\top. \quad (4.30)$$

Since (4.27) has a convex quadratic objective with linear constraints, strong duality holds. Thus we will derive a dual formulation for (4.30) then use the change of coordinates (4.21) to recover the solution to (4.27). Let $\hat{Y} \in \mathbb{R}^{n\times q}$ and $W \in \mathbb{R}^{n\times n}$ and consider the Lagrangian of (4.30) which is

$$L(\hat{X}, \hat{Y}, W) = \frac{1}{2}\|\hat{X} - \hat{X}_k\|^2_F - \left\langle \hat{Y}^\top, \hat{S}^\top\hat{A}(\hat{X} - \hat{A}^{-1})\right\rangle_F - \left\langle W, \hat{X} - \hat{X}^\top\right\rangle_F. \quad (4.31)$$

Differentiating in $\hat{X}$ and setting to zero gives

$$\hat{X} = \hat{X}_k + \hat{A}^\top\hat{S}\hat{Y}^\top + W - W^\top. \quad (4.32)$$

Applying the symmetry constraint $X = X^\top$ gives

$$W - W^\top = \frac{1}{2}\left(\hat{Y}\hat{S}^\top\hat{A} - \hat{A}^\top\hat{S}\hat{Y}^\top\right).$$

Substituting the above into (4.32) gives

$$\hat{X} = \hat{X}_k + \frac{1}{2}\left(\hat{Y}\hat{S}^\top\hat{A} + \hat{A}^\top\hat{S}\hat{Y}^\top\right). \quad (4.33)$$

Now let $\Theta = \frac{1}{2}(\hat{Y}\hat{S}^\top\hat{A} + \hat{A}^\top\hat{S}\hat{Y}^\top)$ and note that, since the matrix $\Theta + \hat{X}_k - \hat{A}^{-1}$ is symmetric, we get

$$\left\langle \hat{A}^\top\hat{S}\hat{Y}^\top, \Theta + \hat{X}_k - \hat{A}^{-1} \right\rangle_F = \left\langle \Theta, \Theta + \hat{X}_k - \hat{A}^{-1} \right\rangle_F. \tag{4.34}$$

Substituting (4.33) into (4.31) gives

$$\begin{aligned} L(\hat{X}, \hat{Y}, W) &= \frac{1}{2}\|\Theta\|_F^2 - \left\langle \hat{A}^\top\hat{S}\hat{Y}^\top, \Theta + \hat{X}_k - \hat{A}^{-1} \right\rangle_F \overset{(4.34)}{=} \frac{1}{2}\|\Theta\|_F^2 - \left\langle \Theta, \Theta + \hat{X}_k - \hat{A}^{-1} \right\rangle_F \\ &= -\frac{1}{2}\|\Theta\|_F^2 - \left\langle \Theta, \hat{X}_k - \hat{A}^{-1} \right\rangle_F. \end{aligned} \tag{4.35}$$

Adding $\pm\frac{1}{2}\|\hat{X}_k - \hat{A}^{-1}\|_F^2$ to (4.35) gives

$$L(\hat{X}, \hat{Y}, W) = -\frac{1}{2}\|\Theta + \hat{X}_k - \hat{A}^{-1}\|_F^2 + \frac{1}{2}\|\hat{X}_k - \hat{A}^{-1}\|_F^2.$$

Finally, using (4.33) and maximizing over $\hat{Y}$ then minimizing over $X$ gives the dual problem

$$\min_{\hat{X},\hat{Y}} \frac{1}{2}\|\hat{X} - \hat{A}^{-1}\|_F^2 \quad \text{subject to} \quad \hat{X} = \hat{X}_k + \frac{1}{2}(\hat{Y}\hat{S}^\top\hat{A} + \hat{A}^\top\hat{S}\hat{Y}^\top).$$

It now remains to change variables according to (4.21) and set $Y = B^{-1/2}\hat{Y}$.

It was recently shown in [48, Section 2] and [57, Section 4][4] that (4.29) is the solution to (4.27). But for completion, we now give a new simple proof.

From (4.33) we see that the solution is solely determined by $\hat{Y}\hat{S}^\top\hat{A}$, and thus we focus on obtaining this matrix. To simplify notation, let $\Gamma = \hat{A}^\top\hat{S}$ and let $\hat{Z} = \Gamma(\Gamma^\top\Gamma)^\dagger\Gamma^\top$. As $\hat{Z}$ is a projection matrix we have that $\hat{Z}^2 = \hat{Z}$ and $(I - \hat{Z})\hat{Z} = 0$, two properties we will use repeatedly.

Using the sketch constraint in (4.30) we have

$$\Gamma^\top\hat{X} = \hat{S}^\top\hat{A}\hat{X} = \hat{S}^\top = \Gamma^\top\hat{A}^{-1}, \tag{4.36}$$

therefore left multiplying (4.33) by $\Gamma^\top$ gives

$$\Gamma^\top\hat{X} = \Gamma^\top\hat{X}_k + \frac{1}{2}\Gamma^\top\left(\hat{Y}\Gamma^\top + \Gamma\hat{Y}^\top\right) \overset{(4.36)}{=} \Gamma^\top\hat{A}^{-1}. \tag{4.37}$$

Let $R = \hat{A}^{-1} - \hat{X}_k$ which is a symmetric matrix. Rearranging (4.37) gives

$$(\Gamma^\top\Gamma)\hat{Y}^\top = \Gamma^\top\left(2R - \hat{Y}\Gamma^\top\right).$$

The least norm solution of the above in term of $\hat{Y}^\top$ is given by

$$\hat{Y}^\top = (\Gamma^\top\Gamma)^\dagger\Gamma^\top\left(2R - \hat{Y}\Gamma^\top\right).$$

Left multiplying the above by $\Gamma$ gives

$$\Gamma\hat{Y}^\top = \Gamma(\Gamma^T\Gamma)^\dagger\Gamma^T\left(2R - \hat{Y}\Gamma^\top\right) = \hat{Z}\left(2R - \hat{Y}\Gamma^\top\right). \tag{4.38}$$

[4] To re-interpret methods for solving linear systems through Bayesian inference, Hennig constructs estimates of the inverse system matrix using the sampled action of a matrix taken during a linear solve [57].

This shows that $\Gamma\hat{Y}^\top$ is equal to a projection matrix times an unknown matrix that is

$$\Gamma\hat{Y}^\top = \hat{Z}\Psi = \hat{Z}\Psi\hat{Z} + \hat{Z}\Psi(I - \hat{Z}), \tag{4.39}$$

where $\Psi \in \mathbb{R}^{n\times n}$ is the unknown matrix. Note that in (4.39) we have decomposed the rows of $\hat{Z}\Psi$ into orthogonal components. Substituting (4.39) into (4.38) gives

$$\hat{Z}\Psi\hat{Z} + \hat{Z}\Psi(I - \hat{Z}) = \hat{Z}\left(2R - \hat{Z}\Psi^\top\hat{Z}\right), \tag{4.40}$$

where we used that $\hat{Z}(I - \hat{Z}) = 0$. Right multiplying (4.40) by $\hat{Z}$ and re-arranging gives

$$\hat{Z}(\Psi + \Psi^\top)\hat{Z} = 2\hat{Z}R\hat{Z}. \tag{4.41}$$

Right multiplying (4.40) by $I - \hat{Z}$ and re-arranging gives

$$\hat{Z}\Psi(I - \hat{Z}) = 2\hat{Z}R(I - \hat{Z}). \tag{4.42}$$

Finally, inserting (4.39) into (4.33) gives

$$\begin{aligned}
\hat{X} &= \hat{X}_k + \frac{1}{2}\left(\hat{Z}(\Psi + \Psi^\top)\hat{Z} + (I - \hat{Z})\Psi^\top\hat{Z} + \hat{Z}\Psi(I - \hat{Z})\right) \\
&\overset{(4.41)+(4.42)}{=} \hat{X}_k + \hat{Z}R\hat{Z} + \hat{Z}R(I - \hat{Z}) + (I - \hat{Z})R\hat{Z} \\
&= \hat{X}_k + R\hat{Z} + \hat{Z}R(I - \hat{Z}) \\
&= \hat{X}_k + (\hat{S} - \hat{X}_k\hat{A}\hat{S})\Lambda\hat{S}^\top\hat{A} + \hat{A}\hat{S}\Lambda(\hat{S}^\top - \hat{S}^\top\hat{A}\hat{X}_k)(I - \hat{A}\hat{S}\Lambda\hat{S}^\top\hat{A}),
\end{aligned}$$

where we used that $\Lambda = (\hat{S}^\top\hat{A}\hat{A}^\top\hat{S})^\dagger = (S^\top AB^{-1}A^\top S)^{-1}$. It now remains to use the change of variables (4.21) to obtain (4.29). □

**Algorithm 5** Stochastic Iterative Matrix Inversion (SIMI) – symmetric variant

**input:** symmetric invertible matrix $A \in \mathbb{R}^{n\times n}$
**parameters:** $\mathcal{D}$ = distribution over random matrices; symmetric positive definite $B \in \mathbb{R}^{n\times n}$
**initialize:** symmetric matrix $X_0 \in \mathbb{R}^{n\times n}$
**for** $k = 0, 1, 2, \ldots$ **do**
    Sample an independent copy $S \sim \mathcal{D}$
    Compute $\Lambda = S(S^\top AB^{-1}AS)^{-1}S^\top$
    Compute $\Theta = \Lambda AB^{-1}$
    Compute $M_k = X_kA - I$
    $X_{k+1} = X_k - M_k\Theta - (M_k\Theta)^\top + \Theta^\top(AX_kA - A)\Theta$ ▷ This is equivalent to (4.27) & (4.28)
**output:** last iterate $X_k$

## 4.6 Convergence

We now analyze the convergence of the *error*, $X_k - A^{-1}$, for iterates of Algorithms 3, 4 and 5. For the sake of economy of space, we only analyze Algorithms 3 and 5. Convergence of Algorithm 4 follows from convergence of Algorithm 3 by observing Remark 33.

The first analysis we present in Section 4.6.1 is concerned with the convergence of $\|\mathbf{E}\left[X_k - A^{-1}\right]\|^2$, that is, the *norm of the expected error*. We then analyze the convergence of $\mathbf{E}\left[\|X_k - A^{-1}\|\right]^2$, the *expected norm of the error*. The latter is a stronger type of convergence, as explained in Lemma 10.

The convergence of Algorithms 3 and 5 can be entirely characterized by studying the following random matrix

$$Z \overset{\text{def}}{=} A^\top S(S^\top AB^{-1}A^\top S)^{-1}S^\top A. \tag{4.43}$$

With this definition, the update step of Algorithm 3 can be re-written as a simple fixed point formula

$$X_{k+1} - A^{-1} = \left(I - B^{-1}Z\right)(X_k - A^{-1}). \tag{4.44}$$

We can also simplify the iterates of Algorithm 5 to

$$X_{k+1} - A^{-1} = \left(I - B^{-1}Z\right)(X_k - A^{-1})\left(I - ZB^{-1}\right). \tag{4.45}$$

Much like our convergence proofs in Section 2.5, the only stochastic component in our methods is contained in the matrix $Z$, and thus the convergence of the iterates will depend on the properties of $Z$ and its expected value $\mathbf{E}[Z]$. In particular, recall from Lemma 9 that $B^{-1}ZB^{-1}$ is an orthogonal projection.

### 4.6.1 Norm of the expected error

We start by proving that the norm of the expected error of the iterates of Algorithm 3 and Algorithm 5 converges to zero. The following theorem is remarkable in that we do not need to make any assumptions on the distribution $S$, except that $S$ has full column rank. Rather, the theorem pinpoints that convergence depends solely on the spectrum of $I - B^{1/2}\mathbf{E}[Z]B^{1/2}$.

**Theorem 35.** *Let $S$ be a random matrix which has full column rank with probability 1 (so that $Z$ is well defined). Then the iterates $X_{k+1}$ of Algorithm 3 satisfy*

$$\mathbf{E}\left[X_{k+1} - A^{-1}\right] = (I - B^{-1}\mathbf{E}[Z])\mathbf{E}\left[X_k - A^{-1}\right]. \tag{4.46}$$

*Let $X_0 \in \mathbb{R}^{n\times n}$. If $X_k$ is calculated in either one of these two ways*

1. *Applying $k$ iterations of Algorithm 3,*
2. *Applying $k$ iterations of Algorithm 5 (assuming $A$ and $X_0$ are symmetric),*

*then $X_k$ converges to the inverse exponentially fast, according to*

$$\|\mathbf{E}\left[X_k - A^{-1}\right]\|_B^* \leq \rho^k \|X_0 - A^{-1}\|_B^*, \tag{4.47}$$

*where*

$$\rho \overset{def}{=} 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}). \tag{4.48}$$

*Moreover, we have the following lower and upper bounds on the convergence rate:*

$$0 \leq 1 - \frac{\mathbf{E}\left[q\right]}{n} \leq \rho \leq 1. \tag{4.49}$$

*Proof.* Let

$$R_k \overset{\text{def}}{=} B^{1/2}R_kB^{1/2} \quad \textit{and} \quad \hat{Z} \overset{\text{def}}{=} B^{-1/2}ZB^{-1/2}, \tag{4.50}$$

for all $k$. Thus $\hat{Z}$ is a projection matrix (see Lemma 9) and $\|R_k\|_2 = \|X_k - A^{-1}\|_B$. Left and right multiplying (4.44) by $B^{1/2}$ gives

$$R_{k+1} = (I - \hat{Z})R_k. \tag{4.51}$$

Taking expectation with respect to $S$ in (4.51) gives

$$\mathbf{E}\left[R_{k+1} \mid R_k\right] = (I - \mathbf{E}\left[\hat{Z}\right])R_k. \tag{4.52}$$

Taking full expectation in (4.51) and using the tower rule gives

$$\begin{aligned} \mathbf{E}\left[R_{k+1}\right] &= \mathbf{E}\left[\mathbf{E}\left[R_{k+1} \mid R_k\right]\right] \\ &\overset{(4.52)}{=} \mathbf{E}\left[(I - \mathbf{E}\left[\hat{Z}\right])R_k\right] \\ &= (I - \mathbf{E}\left[\hat{Z}\right])\mathbf{E}\left[R_k\right]. \end{aligned} \tag{4.53}$$

Applying the norm in (4.53) gives

$$\|\mathbf{E}\left[R_{k+1}\right]\|_2 \leq \|I - \mathbf{E}\left[\hat{Z}\right]\|_2\|\mathbf{E}\left[R_k\right]\|_2. \tag{4.54}$$

Furthermore

$$\begin{aligned} \|I - \mathbf{E}\left[\hat{Z}\right]\|_2 &= \lambda_{\max}\left(I - \mathbf{E}\left[\hat{Z}\right]\right) \\ &= 1 - \lambda_{\min}(\mathbf{E}\left[\hat{Z}\right]) \overset{(4.48)}{=} \rho, \end{aligned} \tag{4.55}$$

where we used to symmetry of $(I - \mathbf{E}\left[\hat{Z}\right])$ when passing from the operator norm to the spectral radius. Note that the symmetry of $\mathbf{E}\left[\hat{Z}\right]$ derives from the symmetry of $\hat{Z}$. It now remains to unroll the recurrence in (4.54) to get (4.47).

Now we analyze the iterates of Algorithm 5. Left and right multiplying (4.45) by $B^{1/2}$ we have

$$R_{k+1} = P(R_k) \overset{\text{def}}{=} \left(I - \hat{Z}\right) R_k \left(I - \hat{Z}\right). \tag{4.56}$$

Defining $\bar{P} : R \mapsto \mathbf{E}\left[P(R) \,|\, R_k\right]$, taking expectation in (4.56) conditioned on $R_k$, gives

$$\mathbf{E}\left[R_{k+1} \mid R_k\right] = \bar{P}(R_k).$$

As $\bar{P}$ is a linear operator, taking expectation again yields

$$\mathbf{E}\left[R_{k+1}\right] = \mathbf{E}\left[\bar{P}(R_k)\right] = \bar{P}(\mathbf{E}\left[R_k\right]). \tag{4.57}$$

Let $|||\bar{P}|||_2 \overset{\text{def}}{=} \max_{\|R\|_2=1}\|\bar{P}(R)\|_2$ be the operator induced norm. Applying norm in (4.57) gives

$$\begin{aligned} \|\mathbf{E}\left[X_{k+1} - A^{-1}\right]\|_B^* &= \|\mathbf{E}\left[R_{k+1}\right]\|_2 & (4.58)\\ &\leq |||\bar{P}|||_2\|\mathbf{E}\left[R_k\right]\|_2 \\ &= |||\bar{P}|||_2\|\mathbf{E}\left[X_k - A^{-1}\right]\|_B^*. & (4.59)\end{aligned}$$

Clearly, $P$ is a *positive linear map*, that is, it is linear and maps positive semi-definite matrices to positive semi-definite matrices. Thus, by Jensen's inequality, the map $\bar{P}$ is also a positive linear map. As every positive linear map attains its norm at the identity matrix (see Corollary 2.3.8 in [9]), we have that

$$\begin{aligned} |||\bar{P}|||_2 &= \|\bar{P}(I)\|_2 \\ &\overset{(4.56)}{=} \|\mathbf{E}\left[\left(I - \hat{Z}\right) I \left(I - \hat{Z}\right)\right]\|_2 \\ &\overset{\text{(Lemma 9)}}{=} \|\mathbf{E}\left[I - \hat{Z}\right]\|_2 \overset{(4.55)}{=} \rho. \end{aligned}$$

Inserting the above equivalence in (4.59), unrolling the recurrence and using the substitution (4.50) gives (4.47).

Finally (4.49) follows immediately from Lemma 13 as $A$ is invertible and $S$ has full column rank.

□

If $\rho = 1$, this theorem does not guarantee convergence. But when $\mathbf{E}\left[Z\right]$ is positive definite, as it will transpire in all practical variants of our method, some of which we describe in Section 4.8, the rate $\rho$ will be strictly less than one, and the norm of the expected error will converge to zero.

### 4.6.2 Expectation of the norm of the error

Now we consider the convergence of the expected norm of the error.

**Theorem 36.** *Let $S$ be a random matrix that has full column rank with probability 1 and such that $\mathbf{E}\left[Z\right]$ is positive definite, where $Z$ is defined in (4.43). Let $X_0 \in \mathbb{R}^{n\times n}$. If $X_k$ is calculated in either one of these two ways*

1. *Applying $k$ iterations of Algorithm 3,*
2. *Applying $k$ iterations of Algorithm 5 (assuming both $A$ and $X_0$ are symmetric matrices),*

*then $X_k$ converges to the inverse according to*

$$\mathbf{E}\left[\|X_k - A^{-1}\|_{F(B)}^2\right] \leq \rho^k\|X_0 - A^{-1}\|_{F(B)}^2. \tag{4.60}$$

*Proof.* First consider Algorithm 3, where $X_{k+1}$ is calculated by iteratively applying (4.44). Using again the substitution (4.50), then from (4.44) we have

$$R_{k+1} = \left(I - \hat{Z}\right) R_k. \tag{4.61}$$

From this we obtain

$$\begin{aligned} \|R_{k+1}\|_F^2 &\overset{(4.61)}{=} \|\left(I - \hat{Z}\right) R_k\|_F^2 \\ &= \mathbf{Tr}\left(\left(I - \hat{Z}\right)\left(I - \hat{Z}\right) R_k R_k^\top\right) \\ &\overset{\text{(Lemma 9)}}{=} \mathbf{Tr}\left(\left(I - \hat{Z}\right) R_k R_k^\top\right) \\ &= \|R_k\|_F^2 - \mathbf{Tr}\left(\hat{Z} R_k R_k^\top\right). \end{aligned} \tag{4.62}$$

Taking expectations, conditioned on $R_k$, we get

$$\mathbf{E}\left[\|R_{k+1}\|_F^2 \,|\, R_k\right] = \|R_k\|_F^2 - \mathbf{Tr}\left(\mathbf{E}\left[\hat{Z}\right] R_k R_k^\top\right).$$

Using that $\mathbf{Tr}\left(\mathbf{E}\left[\hat{Z}\right] R_k R_k^\top\right) \geq \lambda_{\min}\left(\mathbf{E}\left[\hat{Z}\right]\right) \mathbf{Tr}\left(R_k R_k^\top\right)$, which relies on the symmetry of $\mathbf{E}\left[\hat{Z}\right]$, we have that

$$\mathbf{E}\left[\|R_{k+1}\|_F^2 \,|\, R_k\right] \leq \left(1 - \lambda_{\min}\left(\mathbf{E}\left[\hat{Z}\right]\right)\right) \|R_k\|_F^2 = \rho \cdot \|R_k\|_F^2.$$

In order to arrive at (4.60), it now remains to take full expectation, unroll the recurrence and use the substitution (4.50) together with $\|R_k\|_F^2 \overset{(4.1)}{=} \|X_k - A^{-1}\|_{F(B)}^2$.

Now we assume that $A$ and $X_0$ are symmetric and $\{X_k\}$ are the iterates computed by Algorithm 5. Left and right multiplying (4.45) by $B^{1/2}$ we have

$$R_{k+1} = \left(I - \hat{Z}\right) R_k \left(I - \hat{Z}\right). \tag{4.63}$$

Taking norm we have

$$\begin{aligned} \|R_{k+1}\|_F^2 &\overset{\text{(Lemma 9)}}{=} \mathbf{Tr}\left(R_k \left(I - \hat{Z}\right) R_k \left(I - \hat{Z}\right)\right) \\ &= \mathbf{Tr}\left(R_k R_k \left(I - \hat{Z}\right)\right) - \mathbf{Tr}\left(R_k \hat{Z} R_k \left(I - \hat{Z}\right)\right) \\ &\leq \mathbf{Tr}\left(R_k R_k \left(I - \hat{Z}\right)\right), \end{aligned} \tag{4.64}$$

where in the last inequality we used that $I - \hat{Z}$ is an orthogonal projection and thus it is symmetric positive semi-definite, whence

$$\mathbf{Tr}\left(R_k \hat{Z} R_k \left(I - \hat{Z}\right)\right) = \mathbf{Tr}\left(\hat{Z}^{1/2} R_k \left(I - \hat{Z}\right) R_k \hat{Z}^{1/2}\right) \geq 0.$$

The remainder of the proof follows similar steps as those we used in the first part of the proof from (4.62) onwards.

□

Theorem 36 establishes that for all three methods, the expected norm of the error converges exponentially fast to zero. Moreover, the convergence rate $\rho$ is the same that appeared in Theorem 35, where we established the convergence of the norm of the expected error.

Using Lemma 11, both of the convergence results in Theorems 35 and 36 can be recast as iteration complexity bounds. For instance, for a given $0 < \epsilon < 1$, Theorem 35 combined with Lemma 11 (with $\alpha_k = (\|\mathbf{E}\left[X_k - A^{-1}\right]\|_B^*)^2$) gives

$$k \geq \left(\frac{1}{2}\right) \frac{1}{1-\rho} \log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad (\|\mathbf{E}\left[X_k - A^{-1}\right]\|_B^*)^2 \leq \epsilon (\|X_0 - A^{-1}\|_B^*)^2. \tag{4.65}$$

On the other hand, Theorem 36 combined with Lemma 11 (with $\alpha_k = \mathbf{E}[\|X_k - A^{-1}\|_{F(B)}^2]$) gives

$$k \geq \frac{1}{1-\rho} \log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad \mathbf{E}\left[\|X_k - A^{-1}\|_{F(B)}^2\right] \leq \epsilon \|X_0 - A^{-1}\|_{F(B)}^2. \tag{4.66}$$

To push the expected norm of the error below the $\epsilon$ tolerance (4.66), we require double the amount of iterates, as compared with bringing the norm of expected error below the same tolerance (4.65). This is because in Theorem 36 we determined that $\rho$ is the rate at which the expectation of the *squared* norm error converges, while in Theorem 35 we determined that $\rho$ is the rate at which the norm, without the square, of the expected error converges. Though it takes double the number of iterations to decrease the expectation of the norm error, as proven in Lemma 10, the former is a stronger form of convergence. Thus, Theorem 35 does not give a stronger result than Theorem 36, but rather, these theorems give qualitatively different results and ultimately enrich our understanding of the iterative process.

## 4.7 Discrete Random Matrices

We now consider the case of a discrete random matrix $S$. We show that when $S$ is a *complete discrete sampling*, then $\mathbf{E}[Z]$ is positive definite, and thus from Theorems 35 and 36 together with Remark 33, Algorithms 3, 4 and 5 converge.

**Definition 37** (Complete Discrete Sampling)**.** *The random matrix $S$ has a finite discrete distribution with $r$ outcomes. In particular, $S = S_i \in \mathbb{R}^{n\times q_i}$ with probability $p_i > 0$ for $i = 1,\dots,r$, where $S_i$ is of full column rank. We say that $S$ is a complete discrete sampling when $\mathbf{S} \overset{def}{=} [S_1,\dots,S_r] \in \mathbb{R}^{m\times\sum_{i=1}^{r} q_i}$ has full row rank.*

Since we consider $A$ to be invertible in this chapter, the above definition of complete discrete sampling is in synchrony with the definition presented in Chapter 2.

As an example of a complete discrete sampling, let $S = e_i$ (the $i$th unit coordinate vector in $\mathbb{R}^n$) with probability $p_i = 1/n$, for $i = 1,\dots,n$. Then $\mathbf{S}$, as defined in Definition 37, is equal to the identity matrix: $\mathbf{S} = I$. Consequently, $S$ is a complete discrete sampling. In fact, from any basis of $\mathbb{R}^n$ we could construct a complete discrete sampling in an analogous way.

Next we establish that when $S$ is discrete random matrix, that $S$ having a complete discrete distribution is a necessary and sufficient condition for $\mathbf{E}[Z]$ to be positive definite.

**Proposition 38.** *Let $S$ be a discrete random matrix with $r$ outcomes $S_r$ all of which have full column rank. The matrix $\mathbf{E}[Z]$ is positive definite if and only if $S$ is a complete discrete sampling. Furthermore*

$$\mathbf{E}[Z] = A^\top \mathbf{S} D^2 \mathbf{S}^\top A, \tag{4.67}$$

*where*

$$D \overset{def}{=} Diag\left(\sqrt{p_1}(S_1^\top A B^{-1} A^\top S_1)^{-1/2},\dots,\sqrt{p_r}(S_r^\top A B^{-1} A^\top S_r)^{-1/2}\right). \tag{4.68}$$

*Proof.* The equation (4.67) was established in Proposition 18. Since we assume that $S$ has full column rank with probability 1, the matrix $D$ is well defined and nonsingular. Given that $\mathbf{E}[Z]$ is positive semi-definite, we need only show that $\mathbf{Null}(\mathbf{E}[Z])$ contains only the zero vector if and only if $S$ is a complete discrete sampling. Let $v \in \mathbf{Null}(\mathbf{E}[Z])$ and $v \neq 0$, thus

$$0 = v^\top A^\top \mathbf{S} D^2 \mathbf{S}^\top A v = \|D\mathbf{S}^\top A v\|_2^2,$$

which shows that $\mathbf{S}^\top A v = 0$ and thus $v \in \mathbf{Null}\left(\mathbf{S}^\top A\right)$. As $A$ is nonsingular, it follows that $v = 0$ if and only if $\mathbf{S}^\top$ has full column rank. □

With a closed form expression for $\mathbf{E}[Z]$ we can optimize $\rho$ over the possible distributions of $S$ to yield a better convergence rate.

### 4.7.1 Optimizing an Upper Bound on the Convergence Rate

So far we have proven two different types of convergence for Algorithms 3, 4 and 5 in Theorems 35 and 36. Furthermore, both forms of convergence depend on the same convergence rate $\rho$ for which we have a closed form expression (4.48).

The availability of a closed form expression for the convergence rate opens up the possibility of designing particular distributions for $S$ optimizing the rate. In Section 2.6.1 we showed that

for a complete discrete sampling, computing the optimal probability distribution, assuming that the matrices $\{S_i\}_{i=1}^r$ are fixed, leads to a semi-definite program (SDP). Here we propose a more practical alternative: to optimize the following upper bound on the convergence rate:

$$\rho = 1 - \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}) \leq 1 - \frac{1}{\mathbf{Tr}\left(B^{1/2}(\mathbf{E}\left[Z\right])^{-1}B^{1/2}\right)} \stackrel{\text{def}}{=} \gamma.$$

To emphasize the dependence of $\gamma$ and $Z$ on the probability distribution $p = (p_1, \ldots, p_r) \in \mathbb{R}^r$, let us denote

$$\gamma(p) \stackrel{\text{def}}{=} 1 - \frac{1}{\mathbf{Tr}\left(B^{1/2}(\mathbf{E}\left[Z_p\right])^{-1}B^{1/2}\right)}, \tag{4.69}$$

where we have added a subscript to $Z$ to indicate that it is a function of $p$. We now minimize $\gamma(p)$ over the probability simplex:

$$\Delta_r \stackrel{\text{def}}{=} \left\{ p = (p_1, \ldots, p_r) \in \mathbb{R}^r \ : \ \sum_{i=1}^r p_i = 1, \ p \geq 0 \right\}.$$

**Theorem 39.** *Let $S$ be a complete discrete sampling and let $\overline{S}_i \in \mathbb{R}^{n \times q_i}$, for $i = 1, 2, \ldots, r$, be such that $\mathbf{S}^{-T} = [\overline{S}_1, \ldots, \overline{S}_r]$. Then*

$$\min_{p \in \Delta_r} \gamma(p) \quad = \quad 1 - \frac{1}{\left(\sum_{i=1}^r \|B^{-1/2}A^\top S_i \overline{S}_i^\top A^{-T}B^{1/2}\|_F\right)^2}. \tag{4.70}$$

*Proof.* In view of (4.69), minimizing $\gamma$ in $p$ is equivalent to minimizing $\mathbf{Tr}\left(B^{1/2}(\mathbf{E}\left[Z_p\right])^{-1}B^{1/2}\right)$ in $p$. Further, we have

$$\begin{aligned}
\mathbf{Tr}\left(B^{1/2}(\mathbf{E}\left[Z_p\right])^{-1}B^{1/2}\right) &\stackrel{(4.67)}{=} \mathbf{Tr}\left(B^{1/2}(A^\top \mathbf{S} D^2 \mathbf{S}^\top A)^{-1}B^{1/2}\right) \qquad (4.71)\\
&= \mathbf{Tr}\left(B^{1/2}A^{-1}\mathbf{S}^{-T}D^{-2}\mathbf{S}^{-1}A^{-T}B^{1/2}\right)\\
&\stackrel{(4.68)}{=} \sum_{i=1}^r \frac{1}{p_i}\mathbf{Tr}\left(B^{1/2}A^{-1}\overline{S}_i(S_i^\top A B^{-1}A^\top S_i)\overline{S}_i^\top A^{-T}B^{1/2}\right)\\
&= \sum_{i=1}^r \frac{1}{p_i}\|B^{-1/2}A^{-1}\overline{S}_i S_i^\top A B^{1/2}\|_F^2. \qquad (4.72)
\end{aligned}$$

Applying Lemma 42 in the Appendix, the optimal probabilities are given by

$$p_i = \frac{\|B^{-1/2}A^{-1}\overline{S}_i S_i^\top A B^{1/2}\|_F}{\sum_{j=1}^r \|B^{-1/2}A^{-1}\overline{S}_j S_j^\top A B^{1/2}\|_F}, \quad i = 1, 2, \ldots, r \tag{4.73}$$

Plugging this into (4.72) gives the result (4.70). □

Observe that in general, the optimal probabilities (4.73) cannot be calculated, since the formula involves the inverse of $A$, which is not known. However, if $A$ is symmetric positive definite, we can choose $B^{-1} = A^2$, which eliminates this issue. If $A$ is not symmetric positive definite, or if we do not wish to choose $B^{-1} = A^2$, we can approach the formula (4.73) as a recipe for a heuristic choice of the probabilities: we can use the iterates $\{X_k\}$ as a proxy for $A^{-1}$. With this setup, the resulting method is not guaranteed to converge by the theory developed in

this thesis. However, in practice one would expect it to work well. We have not done extensive experiments to test this, and leave this to future research. To illustrate, let us consider a concrete simple example. Choose $B = I$ and $S_i = e_i$ (the unit coordinate vector in $\mathbb{R}^n$). We have $\mathbf{S} = [e_1, \ldots, e_n] = I$, whence $\overline{S}_i = e_i$ for $i = 1, \ldots, r$. Plugging into (4.73), we obtain

$$p_i = \frac{\|X_k e_i e_i^\top A\|_F}{\sum_{j=1}^r \|X_k e_j e_j^\top A\|_F} = \frac{\|X_k e_i\|_2 \|e_i^\top A\|_2}{\sum_{j=1}^r \|X_k e_j\|_2 \|e_j^\top A\|_2}.$$

### 4.7.2 Adaptive Samplings

In Theorem 19 we determined a discrete probability distribution $\mathbf{P}(S = S_i) = p_i$ that yields a convergence rate $\rho$ that is easy to interpret. We will now make use of this convenient probability distribution and pose the question: Having decided on the probabilities $p_1, \ldots, p_r$, how should we choose the matrices $S_1, \ldots, S_r$ if we want $\rho$ to be as small as possible?

To answer this question, first we observe that the convergence rate in Theorem 19 is proportional to the scaled condition number defined by

$$\kappa_{2,F}(B^{-1/2}A^\top \mathbf{S}) \overset{\text{def}}{=} \|(B^{-1/2}A^\top \mathbf{S})^{-1}\|_2 \|B^{-1/2}A^\top \mathbf{S}\|_F = \sqrt{\frac{\mathbf{Tr}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right)}{\lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right)}} \geq \sqrt{n}. \quad (4.74)$$

That is, if we select the probabilities

$$p_i = \|B^{-1/2}A^\top S_i\|_F^2 \Big/ \|B^{-1/2}A^\top \mathbf{S}\|_F^2, \quad (4.75)$$

then Theorem 19 combined with (4.74) gives that the resulting convergence rate is

$$\rho = 1 - \frac{\mathbf{Tr}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right)}{\lambda_{\min}\left(\mathbf{S}^\top A B^{-1} A^\top \mathbf{S}\right)} = 1 - \frac{1}{\kappa_{2,F}^2(B^{-1/2}A^\top \mathbf{S})}. \quad (4.76)$$

Furthermore, following from Remark 33, we can determine a convergence rate for Algorithm 4 based on (4.76). That is, by merely transposing each occurrence of $A$ we have that, by selecting $S_i$ with probability

$$p_i = \|B^{-1/2}AS_i\|_F^2 \Big/ \|B^{-1/2}A\mathbf{S}\|_F^2, \quad (4.77)$$

then Algorithm 4 converges at the rate

$$\rho_2 = 1 - \frac{1}{\kappa_{2,F}^2(B^{-1/2}A\mathbf{S})}. \quad (4.78)$$

Since these rates improve as the condition number $\kappa_{2,F}^2(B^{-1/2}A^\top \mathbf{S})$ (or $\kappa_{2,F}^2(B^{-1/2}A\mathbf{S})$ for Algorithm 4) decreases, we should aim for matrices $S_1, \ldots, S_r$ that minimize the condition number. For instance, the lower bound in (4.74) is reached for $\mathbf{S} = (B^{-1/2}A^\top)^{-1} = A^{-T}B^{1/2}$. While we do not know $A^{-1}$, we can use our best current approximation of it, $X_k$, in its place. This leads to a method which *adapts* the probability distribution governing $S$ throughout the iterative process. This observation inspires a very efficient modification of Algorithm 5, which we call AdaRBFGS (Adaptive Randomized BFGS), and describe in Section 4.9.

Notice that, luckily and surprisingly, our twin goals of computing the inverse and optimizing

the convergence rate via the above adaptive trick are compatible. Indeed, we wish to find $A^{-1}$, whose knowledge gives us the optimal rate. This should be contrasted with the SDP approach mentioned earlier in Section 2.6.1: i) the SDP could potentially be harder than the inversion problem, and ii) having found the optimal probabilities $\{p_i\}$, we are still not guaranteed the optimal rate. Indeed, optimality is relative to the choice of the matrices $S_1, \dots, S_r$, which can be suboptimal.

**Remark 40** (Adaptive sampling)**.** *The convergence rate* (4.78) *suggests how one can select a sampling distribution for $S$ that would result in faster practical convergence. We now detail several practical choices for $B$ and indicate how to sample $S$. These suggestions require that the distribution of $S$ depends on the iterate $X_k$, and thus no longer fit into our framework. Nonetheless, we collect these suggestions here in the hope that others will wish to extend these ideas further, and as a demonstration of the utility of developing convergence rates.*

1. *If $B = I$, then Algorithm 3 converges at the rate $\rho = 1 - 1/\kappa_{2,F}^2(A^\top \mathbf{S})$, and hence $S$ should be chosen so that $\mathbf{S}$ is a preconditioner of $A^\top$. For example $\mathbf{S} = X_k^\top$, that is, $S$ should be a sampling of the rows of $X_k$.*

2. *If $B = I$, then Algorithm 4 converges at the rate $\rho = 1 - 1/\kappa_{2,F}^2(A\mathbf{S})$, and hence $S$ should be chosen so that $\mathbf{S}$ is a preconditioner of $A$. For example $\mathbf{S} = X_k$; that is, $S$ should be a sampling of the columns of $X_k$.*

3. *If $A$ is symmetric positive definite, we can choose $B = A$, in which case Algorithm 5 converges at the rate $\rho = 1 - 1/\kappa_{2,F}^2(A^{1/2}\mathbf{S})$. This rate suggests that $S$ should be chosen so that $\mathbf{S}$ is an approximation of $A^{-1/2}$. In Section 4.9 we develop this idea further, and design the AdaRBFGS algorithm.*

4. *If $B = A^\top A$, then Algorithm 3 can be efficiently implemented with $S = AV$, where $V$ is a complete discrete sampling. Furthermore $\rho = 1 - 1/\kappa_{2,F}^2(A\mathbf{V})$, where $\mathbf{V} \overset{def}{=} [V_1, \dots, V_r]$. This rate suggests that $V$ should be chosen so that $\mathbf{V}$ is a preconditioner of $A$. For example $\mathbf{V} = X_k$; that is, $V$ should be a sampling of the rows of $X_k$.*

5. *If $B = AA^\top$, then Algorithm 4 can be efficiently implemented with $S = A^\top V$, where $V$ is a complete discrete sampling. From* (4.78)*, the convergence rate of the resulting method is given by $1 - 1/\kappa_{2,F}^2(A^\top \mathbf{V})$. This rate suggests that $V$ should be chosen so that $\mathbf{V}$ is a preconditioner of $A^\top$. For example, $\mathbf{V} = X_k^\top$; that is, $V$ should be a sampling of the columns of $X_k$.*

6. *If $A$ is symmetric positive definite, we can choose $B = A^{-2}$, in which case Algorithm 5 can be efficiently implemented with $S = AV$. Furthermore $\rho = 1 - 1/\kappa_{2,F}^2(A\mathbf{V})$. This rate suggests that $V$ should be chosen so that $\mathbf{V}$ is a preconditioner of $A$. For example $\mathbf{V} = X_k$, that is, $V$ should be a sampling of the rows or the columns of $X_k$.*

## 4.8 Randomized Quasi-Newton Updates

Algorithms 3, 4 and 5 are in fact families of algorithms indexed by the two parameters: i) positive definite matrix $B$ and ii) distribution $\mathcal{D}$ (from which we pick random matrices $S$). This allows us to design a myriad of specific methods by varying these parameters. Here we highlight some of these possibilities, focusing on complete discrete distributions for $S$ so that convergence of the iterates is guaranteed through Theorems 35 and 36. We also compute the convergence rate $\rho$ for these special methods for the convenient probability distribution given by (4.75) and (4.77) (Theorem 19) so that the convergence rates (4.76) and (4.78) depend on a scaled condition number which is easy to interpret. We will also make some connections to existing quasi-Newton and Approximate Inverse Preconditioning methods. Table 4.2 provides a guide through this section.

| $A$ | $B^{-1}$ | $S$ | Inverse Equation | Randomized Update | Section |
|---|---|---|---|---|---|
| any | any | invertible | any | One Step | 4.8.1 |
| any | $I$ | $e_i$ | $AX = I$ | Simultaneous Kaczmarz (SK) | 4.8.2 |
| any | $I$ | vector | $XA = I$ | Bad Broyden (BB) | 4.8.3 |
| sym. | $I$ | vector | $AX = I, X = X^\top$ | Powell-Symmetric-Broyden (PSB) | 4.8.4 |
| any | $I$ | vector | $XA^{-1} = I$ | Good Broyden (GB) | 4.8.5 |
| sym. | $A^{-1} - X_k$ | vector | $AX = I$ or $XA = I$ | Symmetric Rank 1 (SR1) | 4.8.7 |
| s.p.d. | $A$ | vector | $XA^{-1} = I, X = X^\top$ | Davidon-Fletcher-Powell (DFP) | 4.8.8 |
| s.p.d. | $A^{-1}$ | vector | $AX = I, X = X^\top$ | Broyden-Fletcher-Goldfarb-Shanno (BFGS) | 4.8.9 |
| any | $(A^\top A)^{-1}$ | vector | $AX = I$ | Column | 4.8.10 |

**Table 4.2:** *Specific randomized updates for inverting matrices discussed in this section, obtained as special cases of our algorithms. First column: "sym" means "symmetric" and "s.p.d." means "symmetric positive definite". Block versions of all these updates are obtained by choosing $S$ as a matrix with more than one column (i.e., not as a vector).*

### 4.8.1 One Step Update

We have the freedom to select $S$ as almost any random matrix that has full column rank. This includes choosing $S$ to be a constant and invertible matrix, such as the identity matrix $I$, in which case $X_1$ must be equal to the inverse. Indeed, the sketch-and-project formulations of all our algorithms reveal that. For Algorithm 3, for example, the sketched system is $S^\top AX = S^\top$, which is equivalent to $AX = I$, which has as its unique solution $X = A^{-1}$. Hence, $X_1 = A^{-1}$, and we have convergence in one iteration/step. Through inspection of the complexity rate, we see that $B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2} = I$ and $\rho = \lambda_{\min}(B^{-1/2}\mathbf{E}\left[Z\right]B^{-1/2}) = 1$, thus this one step convergence is predicted in theory by Theorems 35 and 36.

### 4.8.2 Simultaneous randomized Kaczmarz update

Perhaps the most natural choice for the weighting matrix $B$ is the identity $B = I$. With this choice, Algorithm 3 is equivalent to applying the randomized Kaczmarz update simultaneously to the $n$ linear systems encoded in $AX = I$. To see this, note that the sketch-and-project

viewpoint (4.15) of Algorithm 3 is

$$X_{k+1} = \arg\min_{X \in \mathbb{R}^{n\times n}} \frac{1}{2}\|X - X_k\|_F^2 \quad \text{subject to} \quad S^\top A X = S^\top, \tag{4.79}$$

which, by (4.19), results in the explicit update

$$X_{k+1} = X_k + A^\top S(S^\top A A^\top S)^{-1} S^\top (I - A X_k). \tag{4.80}$$

If $S$ is a random coordinate vector, then (4.79) is equivalent to projecting the $j$th column of $X_k$ onto the solution space of $A_{i:}x = \delta_{ij}$, which is exactly an iteration of the randomized Kaczmarz update applied to solving $Ax = e_j$. In particular, if $S = e_i$ with probability $p_i = \|A_{i:}\|_2^2/\|A\|_F^2$ then according to (4.76), the rate of convergence of update (4.80) is given by

$$\mathbf{E}\left[\|X_k - A^{-1}\|_F^2\right] \leq \left(1 - \frac{1}{\kappa_{2,F}^2(A)}\right)^k \|X_0 - A^{-1}\|_F^2$$

where we used that $\kappa_{2,F}(A) = \kappa_{2,F}(A^\top)$. This is exactly the rate of convergence given by Strohmer and Vershynin in [125] for the randomized Kaczmarz method.

### 4.8.3 Randomized bad Broyden update

The update (4.80) can also be viewed as an adjoint form of the bad Broyden update [15, 56]. To see this, if we use Algorithm 4 with $B = I$, then the iterative process is

$$X_{k+1} = X_k + (I - X_k A)S(S^\top A^\top A S)^{-1} S^\top A^\top. \tag{4.81}$$

This update (4.81) is a randomized block form of the *bad Broyden update* [15, 56]. In the quasi-Newton setting, $S$ is not random, but rather the previous step direction $S = \delta \in \mathbb{R}^n$. Furthermore, if we rename $\gamma \overset{\text{def}}{=} AS \in \mathbb{R}^n$, then (4.81) becomes

$$X_{k+1} = X_k + \frac{\delta - X_k\gamma}{\|\gamma\|_2^2}\gamma^\top, \tag{4.82}$$

which is the standard way of writing the bad Broyden update [56]. The update (4.80) is an adjoint form of the bad Broyden in the sense that, if we transpose (4.80), then set $S = \delta$ and denote $\gamma = A^\top S$, we obtain the bad Broyden update, but applied to $X_k^\top$ instead.

From the constrain-and-approximate viewpoint (4.18) we give a new interpretation to the bad Broyden update, namely, the update (4.82) can be written as

$$X_{k+1} = \arg_X \min_{X \in \mathbb{R}^{n\times n},\ y\in\mathbb{R}^n} \frac{1}{2}\|X - A^{-1}\|_F^2 \quad \text{subject to} \quad X = X_k + y\gamma^\top.$$

Thus, *the bad Broyden update is the best rank-one update approximating the inverse.*

We can determine the rate at which our randomized variant of the BB update (4.81) converges by using (4.78). In particular, if $S = S_i$ with probability $p_i = \|AS_i\|_F^2 / \|A\mathbf{S}\|_F^2$,

then (4.87) converges with the rate

$$\mathbf{E}\left[\|X_k - A^{-1}\|_F^2\right] \leq \left(1 - \frac{1}{\kappa_{2,F}^2(A\mathbf{S})}\right)^k \|X_0 - A^{-1}\|_F^2.$$

### 4.8.4 Randomized Powell-Symmetric-Broyden update

If $A$ is symmetric and we use Algorithm 5 with $B = I$, the iterates are given by

$$\begin{aligned} X_{k+1} = X_k &+ AS^\top (S^\top A^2 S)^{-1} SA(X_k AS - S)\left((S^\top A^2 S)^{-1} S^\top A - I\right) \\ &- (X_k AS - S)(S^\top A^2 S)^{-1} S^\top A, \end{aligned} \tag{4.83}$$

which is a randomized block form of the Powell-Symmetric-Broyden update [48]. If $S = S_i$ with probability $p_i = \|AS_i\|_F^2 / \|A\mathbf{S}\|_F^2$, then according to (4.76), the iterates (4.83) and (4.80) converge according to

$$\mathbf{E}\left[\|X_k - A^{-1}\|_F^2\right] \leq \left(1 - \frac{1}{\kappa_{2,F}^2(A^\top\mathbf{S})}\right)^k \|X_0 - A^{-1}\|_F^2.$$

### 4.8.5 Randomized good Broyden update

Next we present a method that shares certain properties with Gaussian elimination and can be viewed as a randomized block variant of the good Broyden update [15, 56]. This method requires the following adaptation of Algorithm 4: instead of sketching the inverse equation, consider the update (4.84) that performs a column sketching of the equation $XA^{-1} = I$ by right multiplying with $Ae_i$, where $e_i$ is the $i$th coordinate vector. Projecting an iterate $X_k$ onto this sketched equation gives

$$X_{k+1} = \arg\min_{X \in \mathbb{R}^{n\times n}} \frac{1}{2}\|X - X_k\|_F^2 \quad \text{subject to} \quad Xe_i = Ae_i. \tag{4.84}$$

The iterates defined by the above are given by

$$X_{k+1} = X_k + (A - X_k)e_i e_i^\top. \tag{4.85}$$

Given that we are sketching and projecting onto the solution space of $XA^{-1} = I$, the iterates of this method converge to $A$. Therefore the inverse iterates $X_k^{-1}$ converge to $A^{-1}$. We can efficiently compute the inverse iterates by using the Woodbury formula [131] which gives

$$X_{k+1}^{-1} = X_k^{-1} - \frac{(X_k^{-1}A - I)e_i e_i^\top X_k^{-1}}{e_i^\top X_k^{-1} A e_i}. \tag{4.86}$$

This update (4.86) behaves like Gaussian elimination in the sense that, if $i$ is selected in a cyclic fashion, that is $i = k$ on the $k$th iteration, then from (4.85) it is clear that

$$X_{k+1}e_i = Ae_i, \quad \text{thus} \quad X_{k+1}^{-1}Ae_i = e_i, \quad \text{for } i = 1 \ldots k.$$

That is, on the $k$th iteration, the first $k$ columns of the matrix $X_{k+1}^{-1}A$ are equal to the first $k$ columns of the identity matrix. Consequently, $X_n = A$ and $X_n^{-1} = A^{-1}$. If instead, we select $i$ uniformly at random, then we can adapt (4.76) by swapping each occurrence of $A^\top$ for $A^{-1}$ and observing that $S_i = Ae_i$ thus $\mathbf{S} = A$. Consequently the iterates (4.85) converge to $A$ at a rate of

$$\rho = 1 - \kappa_{2,F}^2\left(A^{-1}A\right) = 1 - \frac{1}{n},$$

and thus the lower bound (4.49) is achieved and $X_k$ converges to $A$ according to

$$\mathbf{E}\left[\|X_k - A\|_F^2\right] \leq \left(1 - \frac{1}{n}\right)^k \|X_0 - A\|_F^2.$$

Despite this favourable convergence rate, this does not say anything about how fast $X_k^{-1}$ converges to $A^{-1}$. Therefore (4.86) is not an efficient method for calculating an approximate inverse. If we replace $e_i$ by a *step* direction $\delta_k \in \mathbb{R}^d$, then the update (4.86) is known as the *good Broyden* update [15, 56].

### 4.8.6 Approximate inverse preconditioning

When $A$ is symmetric positive definite, we can choose $B = A$, and Algorithm 3 is given by

$$X_{k+1} = X_k + S(S^\top AS)^{-1}S^\top(I - AX_k). \tag{4.87}$$

The constrain-and-approximate viewpoint (4.17) of this update is

$$X_{k+1} = \arg_X \min_{X\in\mathbb{R}^{n\times n}, Y\in\mathbb{R}^{n\times q}} \frac{1}{2}\|A^{1/2}XA^{1/2} - I\|_F^2 \quad \text{subject to} \quad X = X_k + SY^\top.$$

This viewpoint reveals that the update (4.87) is akin to the Approximate Inverse Preconditioning (*AIP*) methods [6, 46, 64, 60].

We can determine the rate a which (4.87) converges using (4.76). In particular, if $S = S_i$ with probability $p_i = \mathbf{Tr}\left(S_i^\top AS_i\right)/\mathbf{Tr}\left(\mathbf{S}^\top A\mathbf{S}\right)$, then (4.87) converges with rate

$$\rho \overset{(4.76)}{=} 1 - \frac{1}{\kappa_{2,F}^2(A^{1/2}\mathbf{S})} = 1 - \frac{\lambda_{\min}(\mathbf{S}^\top A\mathbf{S})}{\mathbf{Tr}\left(\mathbf{S}^\top A\mathbf{S}\right)}, \tag{4.88}$$

and according to

$$\mathbf{E}\left[\|A^{1/2}X_kA^{1/2} - I\|_F^2\right] \leq \left(1 - \frac{\lambda_{\min}(\mathbf{S}^\top A\mathbf{S})}{\mathbf{Tr}\left(\mathbf{S}^\top A\mathbf{S}\right)}\right)^k \|A^{1/2}X_0A^{1/2} - I\|_F^2.$$

This, as we will see in Section 4.8.9, is the same rate of convergence as a randomized variant of the BFGS method.

### 4.8.7 Randomized SR1

The Symmetric Rank-1 (SR1) update [25, 79] does not explicitly fit into our framework, and nor does it fit into the traditional quasi-Newton framework, since it requires a $B$ that is not positive

definite. Despite this, we present the update since it is still commonly used.

Before choosing $B$, note that, though our theory requires that $B$ be positive definite, Algorithms 3 and 4 can be defined with a matrix $B^{-1}$ even if $B$ does not exist! For instance, when $A$ is symmetric and $B^{-1} = A^{-1} - X_k$ then from (4.19) or (4.20) we get

$$X_{k+1} = X_k + (I - AX_k)^\top S(S^\top(A - AX_kA)S)^{-1}S^\top(I - AX_k). \tag{4.89}$$

This choice for $B$ presents problems, namely, the update (4.89) is not always well defined because it requires inverting $S^\top(A - AX_kA)S$ which is not necessarily invertible. To fix this, we should select the sketching matrix $S$ so that $S^\top(A - AX_kA)S$ is invertible. But this in turn means that $S$ will depend on $X_k$ and most likely cannot be sampled in an i.i.d fashion. Alternatively, we can use the pseudoinverse of $S^\top(A - AX_kA)S$ in place of the inverse.

Since $B$ is not positive definite, our convergence theory says nothing about this update.

### 4.8.8 Randomized DFP update

If $A$ is symmetric positive definite then we can choose $B = A^{-1}$. Furthermore, if we adapt the sketch-and-project formulation (4.15) to sketch the equation $XA^{-1} = I$ by right multiplying by $AS$, and additionally impose symmetry on the iterates, we arrive at the following update.

$$X_{k+1} = \arg\min_{X \in \mathbb{R}^{n\times n}} \frac{1}{2}\|X - X_k\|^2_{F(A^{-1})} \quad \text{subject to} \quad XS = AS, \quad X = X^\top. \tag{4.90}$$

The solution to the above is given by[5]

$$X_{k+1} = AS(S^\top AS)^{-1}S^\top A + \left(I - AS(S^\top AS)^{-1}S^\top\right)X_k\left(I - S(S^\top AS)^{-1}S^\top A\right). \tag{4.91}$$

Using the Woodbury formula [131], we find that

$$X^{-1}_{k+1} = X_k^{-1} + AS(S^\top AS)^{-1}S^\top A - X_k^{-1}S\left(S^\top X_k^{-1}S\right)^{-1}S^\top X_k^{-1}. \tag{4.92}$$

The update (4.92) is a randomized variant of the Davidon-Fletcher-Powell (DFP) update [24, 40]. We can adapt (4.76) to determine the rate at which $X_k$ converges to $A$ by swapping each occurrence of $A^\top$ for $A^{-1}$. Indeed, for example, let $S_i = Ae_i$ with probability $p_i = \lambda_{\min}(A)/\operatorname{\mathbf{Tr}}(A)$, then the iterates (4.85) converge to $A$ at a rate of

$$\mathbf{E}\left[\|X_k - A\|^2_{F(A^{-1})}\right] \leq \left(1 - \frac{\lambda_{\min}(A)}{\operatorname{\mathbf{Tr}}(A)}\right)^k \|X_0 - A\|^2_{F(A^{-1})}. \tag{4.93}$$

Thus $X_k$ converges to $A$ at a favourable rate. But this does not indicate at what rate does $X_k^{-1}$ converge to $A^{-1}$. This is in contrast to the randomized BFGS, which produces iterates that converge to $A^{-1}$ at this same favourable rate, as we show in the next section. This sheds new light on why BFGS update performs better than the DFP update.

[5] To arrive at this solution, one needs to swap the occurrences of $AS$ for $S$ and plug in $B = A^{-1}$ in (4.29). This is because, by swapping $AS$ for $S$ in (4.90) gives (4.27).

### 4.8.9 Randomized BFGS update

If $A$ is symmetric and positive definite, we can choose $B = A$ and apply Algorithm 5 to maintain symmetry of the iterates. The iterates are given by

$$X_{k+1} = S(S^\top AS)^{-1}S^\top + \left(I - S(S^\top AS)^{-1}S^\top A\right) X_k \left(I - AS(S^\top AS)^{-1}S^\top\right). \tag{4.94}$$

This is a block variant, see [48], of the BFGS update [15, 40, 43, 119]. The constrain-and-approximate viewpoint gives a new interpretation to the Block BFGS update. That is, from (4.27), the iterates (4.94) can be equivalently defined by

$$X_{k+1} = \arg_X \min_{X\in\mathbb{R}^{n\times n}, Y\in\mathbb{R}^{n\times q}} \frac{1}{2}\|XA - I\|_F^2 \quad \text{subject to} \quad X = X_k + SY^\top + YS^\top.$$

Thus the block BFGS update, and the standard BFGS update, can be seen as a method for calculating an approximate inverse subject to a particular symmetric affine space passing through $X_k$. This is a completely new way of interpreting the BFGS update.

If $p_i = \mathbf{Tr}\left(S_i^\top AS_i\right)/\mathbf{Tr}\left(\mathbf{S}A\mathbf{S}^\top\right)$, then according to (4.76), the update (4.94) converges according to

$$\mathbf{E}\left[\|X_kA - I\|_F^2\right] \le \left(1 - \frac{1}{\kappa_{2,F}^2(A^{1/2}\mathbf{S})}\right)^k \|X_0A - I\|_F^2. \tag{4.95}$$

A remarkable property of the update (4.94) is that it preserves positive definiteness of $A$. Indeed, assume that $X_k$ is positive definite and let $v \in \mathbb{R}^n$ and $P \overset{\text{def}}{=} S(S^\top AS)^{-1}S^\top$. Left and right multiplying (4.94) by $v^\top$ and $v$, respectively, gives

$$v^\top X_{k+1}v = v^\top Pv + v^\top (I - PA) X_k (I - AP) v \ge 0.$$

Thus $v^\top X_{k+1}v = 0$ implies that $Pv = 0$ and $(I - AP)v = 0$, which when combined gives $v = 0$. This proves that $X_{k+1}$ is positive definite. Thus the update (4.94) is particularly well suited for calculating the inverse of a positive definite matrices.

In Section (4.9), we detail an update designed to improve the convergence rate in (4.95). The result is a method that is able to invert large scale positive definite matrices orders of magnitude faster than the state-of-the-art.

### 4.8.10 Randomized Column update

We now describe an update that has no connection to any previous updates, yet the convergence rate we determine (4.98) is favourable, and comparable to all the other updates we develop.

For this update, we need to perform a linear transformation of the sampling matrices. For this, let $V$ be a complete discrete sampling where $V = V_i \in \mathbb{R}^{n\times q_i}$ with probability $p_i > 0$, for $i = 1, \ldots, r$. Let $\mathbf{V} = [V_1, \ldots, V_r]$. Let the sampling matrices be defined as $S_i = AV_i \in \mathbb{R}^{n\times q_i}$ for $i = 1, \ldots, r$. As $A$ is nonsingular, and $\mathbf{S} = A\mathbf{V}$, then $S$ is a complete discrete sampling. With these choices and $B = A^\top A$, the sketch-and-project viewpoint (4.15) is given by

$$X_{k+1} = \arg\min_{X\in\mathbb{R}^{n\times n}} \frac{1}{2}\|X - X_k\|_{F(A^\top A)}^2 \quad \text{subject to} \quad V_i^\top A^\top AX = V_i^\top A^\top.$$

The solution to the above are the iterates of Algorithm 3, which is given by

$$X_{k+1} = X_k + V_i(V_i^\top A^\top A V_i)^{-1} V_i^\top (A^\top - A^\top A X_k). \tag{4.96}$$

From the constrain-and-approximate viewpoint (4.17), this can be written as

$$X_{k+1} = \arg \min_{X \in \mathbb{R}^{n\times n}, Y \in \mathbb{R}^{n\times q}} \frac{1}{2}\|A(XA^\top - I)\|_F^2 \quad \text{subject to} \quad X = X_k + V_i Y^\top.$$

With these same parameter choices for $S$ and $B$, the iterates of Algorithm 5 are given by

$$\begin{aligned} X_{k+1} = X_k &+ V_i(V_i^\top A^2 V_i)^{-1} V_i^\top (AX_k - I)\left(A^2 V_i (V_i^\top A^2 V_i)^{-1} V_i^\top - I\right) \\ &- (X_k A - I) A V_i (V_i^\top A^2 V_i)^{-1} V_i^\top. \end{aligned} \tag{4.97}$$

If we choose $p_i = \|(AA^\top)^{-1/2} AA^\top V_i\|_F^2 / \|(AA^\top)^{-1/2} AA^\top \mathbf{V}\|_F^2 = \|A^\top V_i\|_F^2 / \|A^\top \mathbf{V}\|_F^2$, then according to (4.76), the iterates (4.96) and (4.97) converge exponentially in expectation to the inverse according to

$$\mathbf{E}\left[\|A(X_k A^\top - I)\|_F^2\right] \leq \left(1 - \frac{1}{\kappa_{2,F}^2(A\mathbf{V})}\right)^k \|A(X_0 A^\top - I)\|_F^2. \tag{4.98}$$

There also exists an analogous "row" variant of (4.96), which arises by using Algorithm 4, but we do not explore it here.

## 4.9 AdaRBFGS: Adaptive Randomized BFGS

All the updates we have developed thus far use a sketching matrix $S$ that is sampled in an i.i.d. fashion from a fixed distribution $\mathcal{D}$ at each iteration. In this section we assume that $A$ is symmetric positive definite, and propose AdaRBFGS: a variant of the RBFGS update, discussed in Section 4.8.9, which *adaptively* changes the distribution $\mathcal{D}$ throughout the iterative process. Due to this change, Theorems 35 and 36 are no longer applicable. Superior numerical efficiency of this update is verified through extensive numerical experiments in Section 4.10.

### 4.9.1 Motivation

We now motivate the design of this new update by examining the convergence rate (4.95) of the RBFGS iterates (4.94). Recall that in RBFGS we choose $B = A$ and $S = S_i$ with probability

$$p_i = \mathbf{Tr}\left(S_i^\top A S_i\right) / \mathbf{Tr}\left(\mathbf{S} A \mathbf{S}^\top\right), \quad i = 1, 2, \ldots, r, \tag{4.99}$$

where $S$ is a complete discrete sampling and $\mathbf{S} = [S_1, \ldots, S_r]$. The convergence rate is

$$\rho = 1 - \frac{1}{\kappa_{2,F}^2(A^{1/2}\mathbf{S})} \overset{(4.74)}{=} 1 - \frac{\lambda_{\min}(\mathbf{S}^\top A \mathbf{S})}{\mathbf{Tr}\left(\mathbf{S}^\top A \mathbf{S}\right)}.$$

Consider now the question of choosing the matrix $\mathbf{S}$ in such a way that $\rho$ is as small as possible. Note that the optimal choice is any $\mathbf{S}$ such that

$$\mathbf{S}^\top A \mathbf{S} = I.$$

Indeed, then $\rho = 1 - 1/n$, and the lower bound (4.74) is attained. For instance, the choice $\mathbf{S} = A^{-1/2}$ would be optimal. This means that in each iteration we would choose $S$ to be a random column (or random column submatrix) of $A^{-1/2}$. Clearly, this is not a feasible choice, as we do not know the inverse of $A$. In fact, it is $A^{-1}$ which we are trying to find! However, this leads to the following interesting observation: *the goals of finding the inverse of $A$ and of designing an optimal distribution $\mathcal{D}$ are in synchrony.*

### 4.9.2 The algorithm

While we do not know $A^{-1/2}$, we can use the information of the iterates $\{X_k\}$ themselves to construct a good *adaptive* sampling. Indeed, the iterates contain information about the inverse and hence we can use them to design a better sampling $S$. In order to do so, it will be useful to maintain a factored form of the iterates,

$$X_k = L_k L_k^\top, \tag{4.100}$$

where $L_k \in \mathbb{R}^{n \times n}$ is invertible. With this in place, let us choose $S$ to be a random column submatrix of $L_k$. In particular, let $C_1, C_2, \ldots, C_r$ be nonempty subsets that form a partition of

$\{1, 2, \ldots, n\}$, and at iteration $k$ choose

$$S = L_k I_{:C_i} \overset{\text{def}}{=} S_i, \tag{4.101}$$

with probability $p_i$ given by (4.99) for $i = 1, 2, \ldots, r$. For simplicity, assume that $C_1 = \{1, \ldots, c_1\}$, $C_2 = \{c_1 + 1, \ldots, c_2\}$ and so on, so that, by the definition of $\mathbf{S}$, we have

$$\mathbf{S} = [S_1, \ldots, S_r] = L_k. \tag{4.102}$$

Note that now both $\mathbf{S}$ and $p_i$ depend on $k$. The method described above satisfies the following recurrence.

**Theorem 41.** *Consider one step of the AdaRBFGS method described above. Then*

$$\mathbf{E}\left[\|X_{k+1} - A^{-1}\|^2_{F(A)} \,|\, X_k\right] \leq \left(1 - \frac{\lambda_{\min}(AX_k)}{\mathbf{Tr}\,(AX_k)}\right) \|X_k - A^{-1}\|^2_{F(A)}. \tag{4.103}$$

*Proof.* Using the same arguments as those in the proof of Theorem 36, we obtain

$$\mathbf{E}\left[\|X_{k+1} - A^{-1}\|^2_{F(A)} \,|\, X_k\right] \leq \left(1 - \lambda_{\min}\left(A^{-1/2}\mathbf{E}\left[Z \mid X_k\right] A^{-1/2}\right)\right) \|X_k - A^{-1}\|^2_{F(A)}, \tag{4.104}$$

where

$$Z \overset{(4.105)}{=} AS_i(S_i^\top AS_i)^{-1}S_i^\top A. \tag{4.105}$$

So, we only need to show that

$$\lambda_{\min}\left(A^{-1/2}\mathbf{E}\left[Z \mid X_k\right] A^{-1/2}\right) \geq \frac{\lambda_{\min}(AX_k)}{\mathbf{Tr}\,(AX_k)}.$$

Since $S$ is a complete discrete sampling, Proposition 38 applied to our setting says that

$$\mathbf{E}\left[Z \mid X_k\right] = A\mathbf{S}D^2\mathbf{S}^\top A, \tag{4.106}$$

where

$$D \overset{\text{def}}{=} \text{Diag}\left(\sqrt{p_1}(S_1^\top AS_1)^{-1/2}, \ldots, \sqrt{p_r}(S_r^\top AS_r)^{-1/2}\right). \tag{4.107}$$

We now have

$$\begin{aligned}
\lambda_{\min}\left(A^{-1/2}\mathbf{E}\left[Z \mid X_k\right] A^{-1/2}\right) &\overset{(4.106)+(4.102)}{\geq} \lambda_{\min}\left(A^{1/2}L_kL_k^\top A^{1/2}\right)\lambda_{\min}(D^2) \\
&\overset{(4.100)}{=} \frac{\lambda_{\min}(AX_k)}{\lambda_{\max}(D^{-2})} \\
&\overset{(4.107)}{=} \frac{\lambda_{\min}(AX_k)}{\max_i \lambda_{\max}(S_i^\top AS_i)/p_i} \\
&\geq \frac{\lambda_{\min}(AX_k)}{\max_i \mathbf{Tr}\left(S_i^\top AS_i\right)/p_i} \\
&\overset{(4.99)+(4.102)}{=} \frac{\lambda_{\min}(AX_k)}{\mathbf{Tr}\,(AX_k)},
\end{aligned}$$

where in the second equality we have used the fact that the largest eigenvalue of a block diagonal

matrix is equal to the maximum of the largest eigenvalues of the blocks. □

If $X_k$ converges to $A^{-1}$, then necessarily the one-step rate of AdaRBFGS proved in Theorem 41 asymptotically reaches the lower bound

$$\rho_k \overset{\text{def}}{=} 1 - \frac{\lambda_{\min}(AX_k)}{\mathbf{Tr}\,(AX_k)} \to 1 - \frac{1}{n}.$$

In other words, as long as this method works, the convergence rate gradually improves, and becomes asymptotically optimal and independent of the condition number. We leave a deeper analysis of this and other adaptive variants of the methods developed in this chapter to future work.

### 4.9.3 Implementation

To implement the AdaRBFGS update, we need to maintain the iterates $X_k$ in the factored form (4.100). Fortunately, a factored form of the update (4.94) was introduced in [54], which we shall now describe and adapt to our objective. Assuming that $X_k$ is symmetric positive definite such that $X_k = L_k L_k^\top$, we shall describe how to obtain a corresponding factorization of $X_{k+1}$. Letting $G_k = (S^\top L_k^{-T} L_k^{-1} S)^{1/2}$ and $R_k = (S^\top AS)^{-1/2}$, it can be verified through direct inspection [54] that $X_{k+1} = L_{k+1} L_{k+1}^\top$, where

$$L_{k+1} = L_k + SR_k \left(G_k^{-1} S^\top L_k^{-T} - R_k^\top S^\top AL_k\right). \tag{4.108}$$

If we instead of (4.101) consider the more general update $S = L_k \tilde{S}$, where $\tilde{S}$ is chosen in an i.i.d. fashion from some fixed distribution $\tilde{\mathcal{D}}$, then

$$L_{k+1} = L_k + L_k \tilde{S} R_k \left((\tilde{S}^\top \tilde{S})^{-1/2} \tilde{S}^\top - R_k^\top \tilde{S}^\top L_k^\top AL_k\right). \tag{4.109}$$

The above can now be implemented efficiently, see Algorithm 6.

**Algorithm 6** Adaptive Randomized BFGS (AdaRBFGS)

**input:** symmetric positive definite matrix $A$
**parameter:** $\tilde{\mathcal{D}}$ = distribution over random matrices with $n$ rows
**initialize:** pick invertible $L_0 \in \mathbb{R}^{n\times n}$
**for** $k = 0, 1, 2, \ldots$ **do**
  Sample an independent copy $\tilde{S} \sim \tilde{\mathcal{D}}$
  Compute $S = L_k \tilde{S}$ ▷ $S$ is sampled adaptively, as it depends on $k$
  Compute $R_k = (\tilde{S}^\top A\tilde{S})^{-1/2}$
  $L_{k+1} = L_k + SR_k \left((\tilde{S}^\top \tilde{S})^{-1/2} \tilde{S}^\top - R_k^\top S^\top AL_k\right)$ ▷ Update the factor
**output:** $X_k = L_k L_k^\top$

In Section 4.10 we test two variants based on (4.109). The first is the *AdaRBFGS_gauss* update, in which the entries of $\tilde{S}$ are standard Gaussian. The second is *AdaRBFGS_cols*, where $\tilde{S} = I_{:C_i}$, as described above, and $|C_i| = q$ for all $i$ for some $q$.

## 4.10 Numerical Experiments

Given the demand for approximate inverses of positive definite matrices in preconditioning and in variable metric methods in optimization, and the author's own interest in the aforementioned applications, we restrict our test to inverting positive definite matrices.

We test four iterative methods for inverting matrices. This rules out the all-or-nothing direct methods such as Gaussian elimination of LU based methods.

For our tests we use two variants of Algorithm 6: AdaRBFGS_gauss, where $\tilde{S} \in \mathbb{R}^{n\times q}$ is a normal Gaussian matrix, and AdaRBFGS_cols, where $\tilde{S}$ consists of a collection of $q$ distinct coordinate vectors in $\mathbb{R}^n$, selected uniformly at random. At each iteration the AdaRBFGS methods compute the inverse of a small matrix $S^\top AS$ of dimension $q \times q$. To invert this matrix we use MATLAB's inbuilt `inv` function, which uses $LU$ decomposition or Gaussian elimination, depending on the input. Either way, `inv` costs $O(q^3)$. For simplicity, we selected $q = \sqrt{n}$ in all our tests.

We compare our method to two well established and competitive methods, the *Newton-Schulz method* [115] and the global self-conditioned Minimal Residual (*MR*) method [19]. The Newton-Schulz method arises from applying the Newton-Raphson method to solve the equation $X^{-1} = A$, which gives

$$X_{k+1} = 2X_k - X_kAX_k. \tag{4.110}$$

The MR method was designed to calculate approximate inverses, and it does so by minimizing the norm of the residual along the preconditioned residual direction, that is

$$\|I - AX_{k+1}\|_F^2 = \min_{\alpha\in\mathbb{R}} \left\{ \|I - AX\|_F^2 \quad \text{subject to} \quad X = X_k + \alpha X_k(I - AX_k) \right\}, \tag{4.111}$$

see [112, chapter 10.5] for a didactic introduction to MR methods. The resulting iterates of the MR method are given by

$$X_{k+1} = X_k + \frac{\mathbf{Tr}\left(R_k^\top AX_kR_k\right)}{\mathbf{Tr}\left((AX_kR_k)^\top AX_kR_k\right)} X_kR_k, \tag{4.112}$$

where $R_k = I - AX_k$.

We perform two sets of tests. On the first set, we choose a different starting matrix for each method which is optimized, in some sense, for that method. We then compare the empirical convergence of each method, including the time taken to calculate $X_0$. In particular, the Newton-Schulz is only guaranteed to converge for an initial matrix $X_0$ such that $\rho(I - X_0A) < 1$. Indeed, the Newton-Schulz method did not converge in most of our experiments when $X_0$ was not carefully chosen according to this criteria. To remedy this, we choose $X_0 = 0.99 \cdot A^\top/\rho^2(A)$ for the Newton-Schulz method, so that $\rho(I - X_0A) < 1$ is satisfied. To compute $\rho(A)$ we used the inbuilt MATLAB function `normest` which is coded in C++. While for MR we followed the suggestion in [112] and used the projected identity for the initial matrix $X_0 = \left(\mathbf{Tr}\left(A\right)/\mathbf{Tr}\left(AA^\top\right)\right) \cdot I$. For our AdaRBFGS methods we simply used $X_0 = I$, as this worked well in practice.

In the second set of tests, which we relegate to the Appendix of this chapter, we compare the empirical convergence of the methods starting from the same matrix, namely the identity matrix $X_0 = I$.

We run each method until the relative residual $\|I - AX_k\|_F / \|I - AX_0\|_F$ is below $10^{-2}$. All experiments were performed and run in MATLAB R2014b. To appraise the performance of each method we plot the relative residual against time taken and against the number of floating point operations (*flops*).

### 4.10.1 Experiment 1: synthetic matrices

First we compare the four methods on synthetic matrices generated using the `rand` function as follows: $A = \bar{A}^\top \bar{A}$ where $\bar{A} =$`rand`$(n)$. The resulting matrix $A$ is positive definite with high probability. To appraise the difference in performance of the methods as the dimension of the problem grows, we tested for $n = 1000$, 2000 and 5000. As the dimension grows, only the two variants of the AdaRBFGS method are able to reach the $10^{-2}$ desired tolerance in a reasonable amount time and number of flops (see Figure 4.2).

### 4.10.2 Experiment 2: LIBSVM matrices

Next we invert the Hessian matrix $\nabla^2 f(x)$ of four ridge-regression problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) \stackrel{\text{def}}{=} \frac{1}{2}\|Ax - b\|_2^2 + \frac{\lambda}{2}\|x\|_2^2, \qquad \nabla^2 f(x) = A^\top A + \lambda I, \tag{4.113}$$

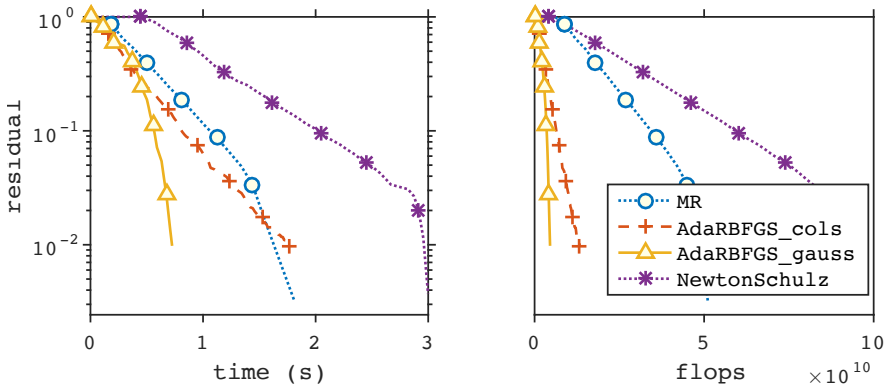
using data from LIBSVM [17], see Figure 4.3. We use $\lambda = 1$ as the regularization parameter. On the two problems of smaller dimension, `aloi` and `protein`, the four methods have a similar performance, and encounter the inverse in a few seconds. On the two larger problems, `gisette-scale` and `real-sim`, the two variants of AdaRBFGS significantly outperform the MR and the Newton-Schulz method.

### 4.10.3 Experiment 3: UF sparse matrices

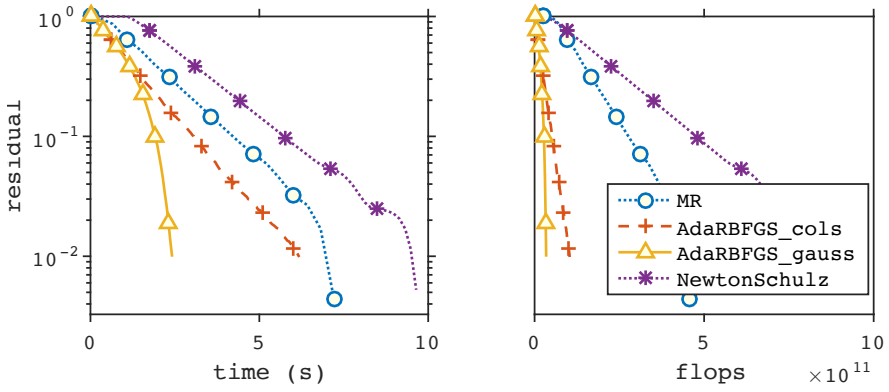
For our final batch of tests, we invert several sparse matrices from the Florida sparse matrix collection [26]. We have selected six problems from six different applications, so that the set of matrices display a varied sparsity pattern and structure, see Figures 4.4 and 4.5.

On the matrix `Bates/Chem97ZtZ` of moderate size, the four methods perform well, with the Newton-Schulz method converging first in time and AdaRBFGS_cols first in flops. While on the matrices of larger dimension, the two variants of AdaRBFGS converge much faster, often orders of magnitude before the MR and Newton-Schulz method reach the required precision.

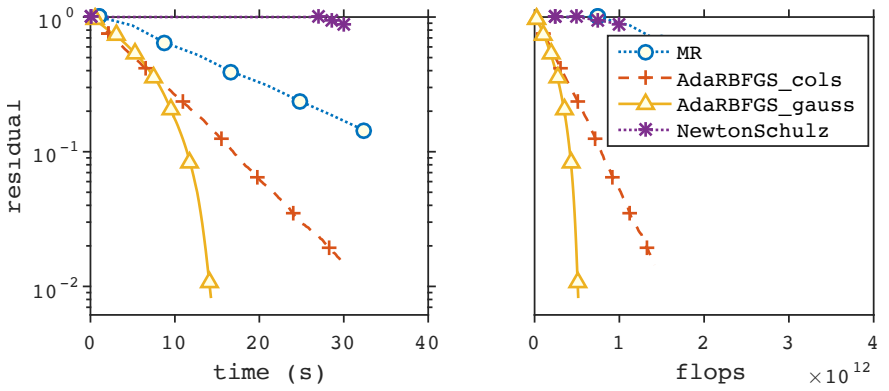
The significant difference between the performance of the methods on large scale problems can be, in part, explained by their iteration cost. The iterates of the Newton-Schulz and MR method compute $n \times n$ matrix-matrix products. While the cost of an iteration of the AdaRBFGS methods is dominated by the cost of a $n \times n$ matrix by $n \times q$ matrix product. As a result, and because we set $q = \sqrt{n}$, this is difference of $n^3$ to $n^{2+1/2}$ in iteration cost, which clearly shows on the larger dimensional instances.




**(a)** *rand with* $n = 10^4$

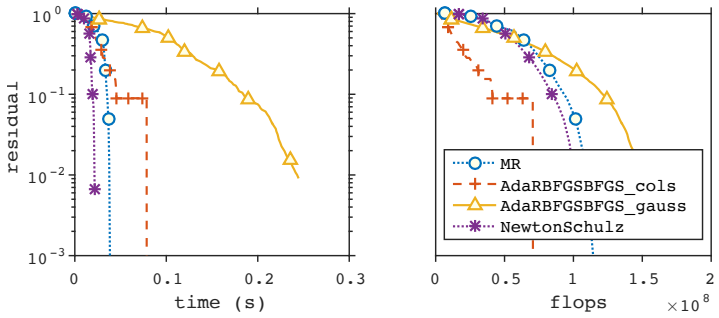



**(b)** *rand with* $n = 2 \cdot 10^4$

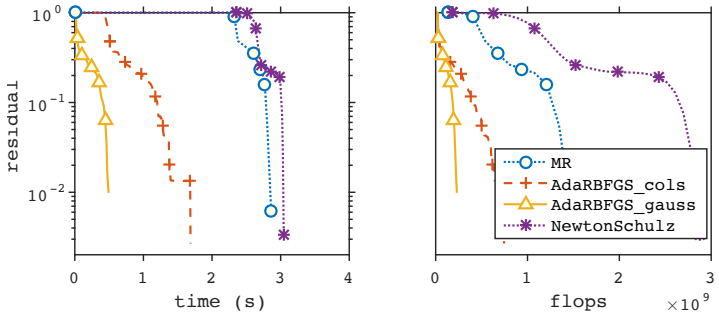



**(c)** *rand with* $n = 5 \cdot 10^4$

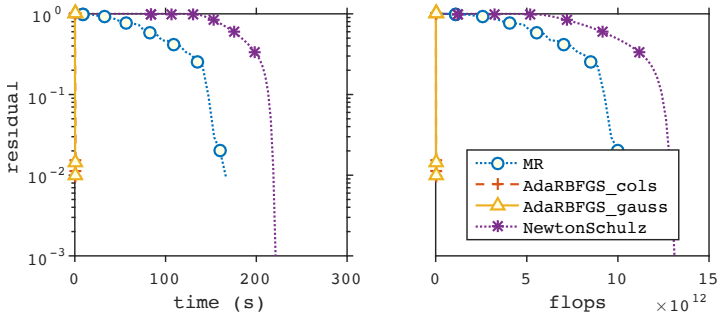
**Figure 4.2:** *Synthetic MATLAB generated problems. Uniform random matrix* $A = \bar{A}^\top \bar{A}$ *where* $\bar{A} =$*rand*$(n)$.



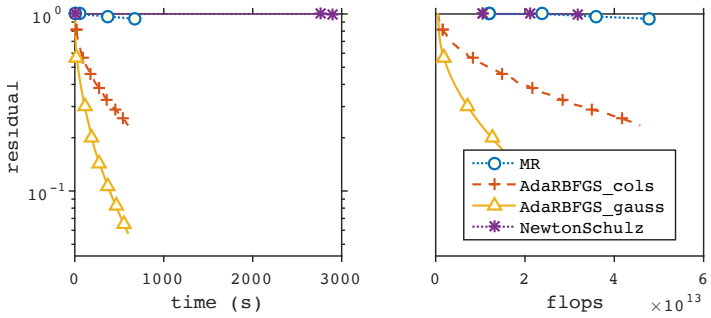

**(a)** *aloi*




**(b)** *protein*

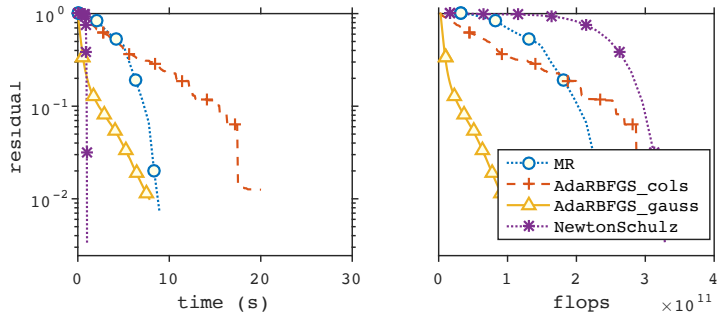



**(c)** *gisette_scale*

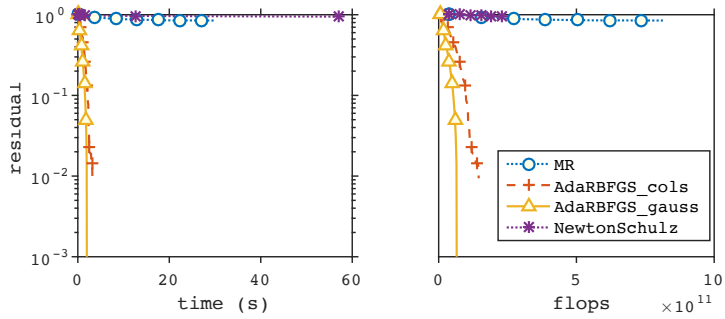



**(d)** *real_sim*

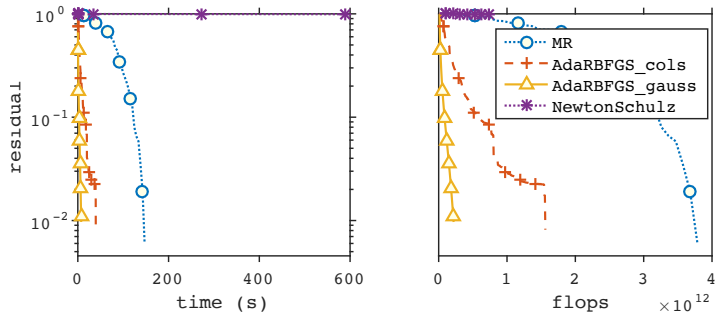
**Figure 4.3:** *The performance of Newton-Schulz, MR, AdaRBFGS_gauss and AdaRBFGS_cols methods on the Hessian matrix of four LIBSVM test problems: (a)* `aloi`*:* $(m;n) = (108,000;128)$ *(b)* `protein`*:* $(m;n) = (17,766;357)$ *(c)* `gisette_scale`*:* $(m;n) = (6000;5000)$ *(d)* `real-sim`*:* $(m;n) = (72,309;20,958)$.



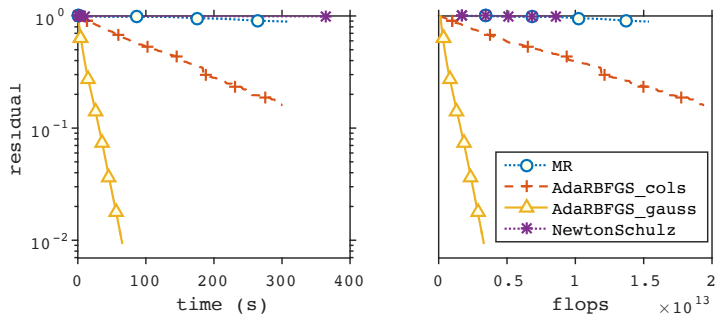

**(a)** *Bates/Chem97ZtZ*




**(b)** *FIDAP/ex9*

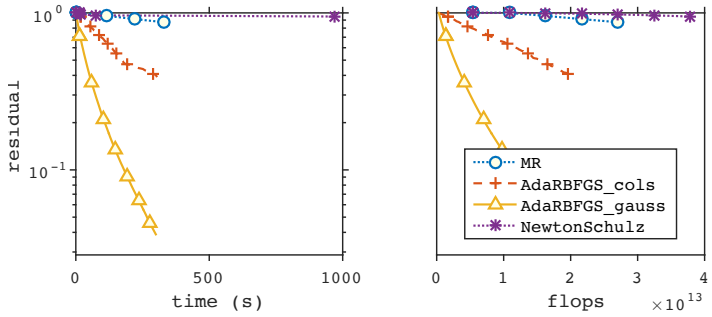



**(c)** *Nasa/nasa4704*

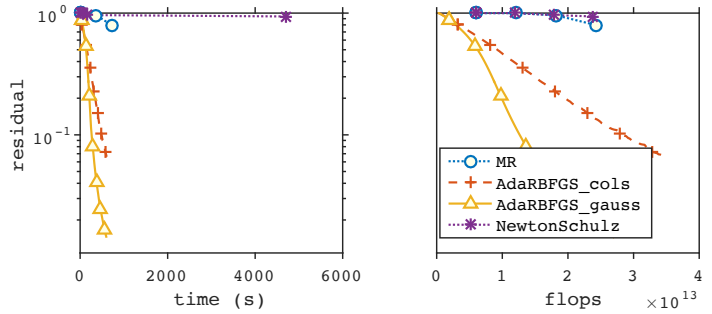



**(d)** *HB/bcsstk18*

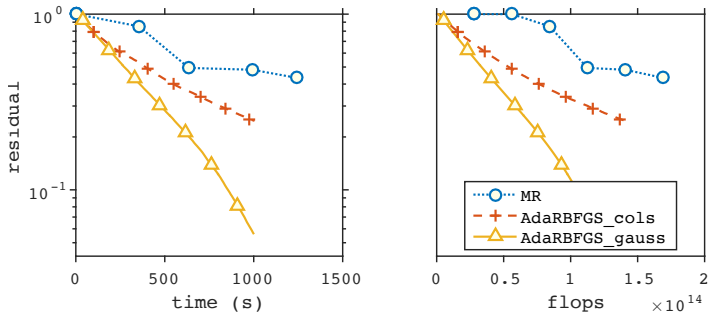
**Figure 4.4:** *The performance of Newton-Schulz, MR, AdaRBFGS_gauss and AdaRBFGS_cols on (a)* `Bates-Chem97ZtZ`*:* $n = 2\,541$*, (b)* `FIDAP/ex9`*:* $n = 3,363$*, (c)* `Nasa/nasa4704`*:* $n = 4,704$*, (d)* `HB/bcsstk18`*:* $n = 11,948$*.*




**(a)** *Pothen/bodyy4*




**(b)** *ND/nd6k*




**(c)** *GHS_psdef/wathen100*

**Figure 4.5:** *The performance of Newton-Schulz, MR, AdaRBFGS_gauss and AdaRBFGS_cols on (a)* `Pothen/bodyy4`*:* $n = 17,546$ *(b)* `ND/nd6k`*:* $n = 18,000$ *(c)* `GHS_psdef/wathen100`*:* $n = 30,401$.

### 4.10.4 Conclusion of numeric experiments

Through our extensive numeric experiments, it is clear that the AdaRBFGS methods are highly efficient at calculating an approximate inverse of a positive definite matrix. Furthermore, in many of these experiments the Newton-Schulz and MR method suffer from an initially slow convergence, particularly so on large-scale problems, see Figures 4.2c, 4.3d, 4.4b, 4.4c and 4.4d for example. But after a sufficient number of iteration, the asymptotic second order convergence rate of the Newton-Schulz and MR method sets in, see Figures 4.2a, 4.2b, 4.3a , 4.3b and 4.3c for example.

These experiments also indicate that the AdaRBFGS method enjoys super linear convergence as can be seen, for example, in Figure 4.2 the residual decreases superlinearly in both time and flops. Yet we have still to prove that the AdaRBFGS methods are even linearly convergent. Thus it is now an open question whether or not the AdaRBFGS convergent linearly or superlinearly.

## 4.11 Summary

We developed a family of stochastic methods for iteratively inverting matrices, with a specialized variant for asymmetric, symmetric and positive definite matrices. The methods have two dual viewpoints, a sketch-and-project viewpoint which is an extension of the least-change formulation of the quasi-Newton methods, and a constrain-and-approximate viewpoint which is related to the approximate inverse preconditioning (API) methods. The equivalence between these two viewpoints reveals a new connection between the quasi-Newton and the API methods, which were previously considered to be unrelated.

Under mild conditions, we prove convergence rates through two different perspectives, the convergence of the expected norm of the error, and the norm of the expected error. Our convergence theorems are general enough to accommodate discrete samplings and continuous samplings, though we only explore discrete sampling here in more detail.

For discrete samplings, we determine a probability distribution for which the convergence rates are equal to a scaled condition number, and thus are easily interpretable. Furthermore, for discrete sampling, we determining a practical optimized sampling distribution, that is obtained by minimizing an upper bound on the convergence rate. We develop new randomized block variants of the quasi-Newton updates, including the BFGS update, complete with convergence rates, and provide new insights into these methods using our dual viewpoint.

For positive definite matrices, we develop an Adaptive Randomized BFGS methods (AdaRBFGS), which in large-scale numerical experiments, prove to be orders of magnitude faster (in time and flops) then the self-conditioned minimal residual method and the Newton-Schulz method. In particular, only the AdaRBFGS methods are able to approximately invert the $20,958 \times 20,958$ ridge regression matrix based on the `real-sim` data set in reasonable time and flops.

This work opens up many possible venues for future work, including, developing methods that use continuous random sampling, implementing a limited memory approach akin to the LBFGS [89] method, which could maintain an operator that serves as an approximation to the inverse. As recently shown in [49], an analogous method applied to linear systems converges with virtually no assumptions on the system matrix. This can be extended to calculating the pseudoinverse matrix, something we leave for future work.

## 4.12 Appendix: Optimizing an Upper Bound on the Convergence Rate

**Lemma 42.** *Let $a_1, \dots, a_r$ be positive real numbers. Then*

$$\left[\frac{\sqrt{a_1}}{\sum_{i=1}^r \sqrt{a_i}}, \dots, \frac{\sqrt{a_n}}{\sum_{i=1}^r \sqrt{a_i}}\right] = \arg\min_{p \in \Delta_r} \sum_{i=1}^r \frac{a_i}{p_i}.$$

*Proof.* Incorporating the constraint $\sum_{i=1}^r p_i = 1$ into the Lagrangian we have

$$\min_{p \geq 0} \sum_{i=1}^r \frac{a_i}{p_i} + \mu \sum_{i=1}^r (p_i - 1),$$

where $\mu \in \mathbb{R}$. Differentiating in $p_i$ and setting to zero, then isolating $p_i$ gives

$$p_i = \sqrt{\frac{a_i}{\mu}}, \quad \text{for } i = 1, \dots r. \tag{4.114}$$

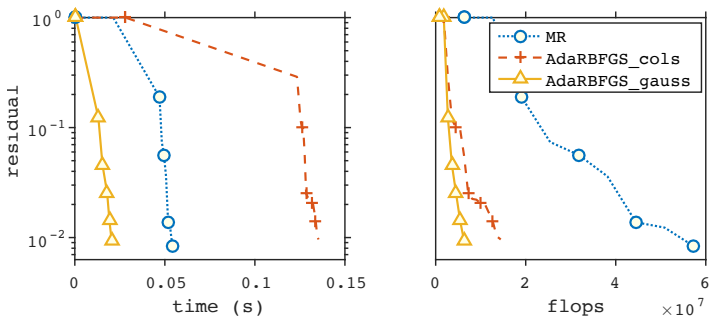
Summing over $i$ gives

$$1 = \sum_{i=1}^r \sqrt{\frac{a_i}{\mu}} \quad \Rightarrow \quad \mu = \left(\sum_{i=1}^r \sqrt{a_i}\right)^2.$$

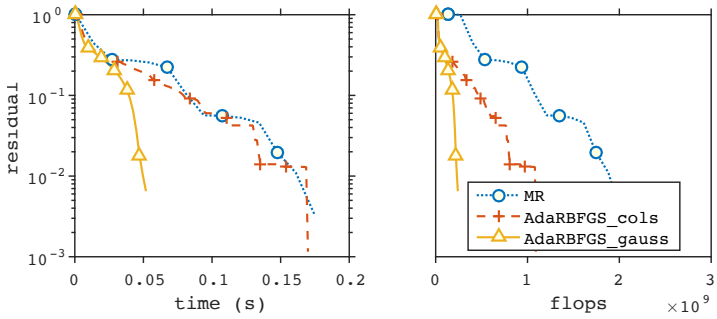
Inserting this back into (4.114) gives $p_i = \sqrt{a_i} / \sum_{i=1}^r \sqrt{a_i}$. □

## 4.13 Appendix: Numerical Experiments with the Same Starting Matrix

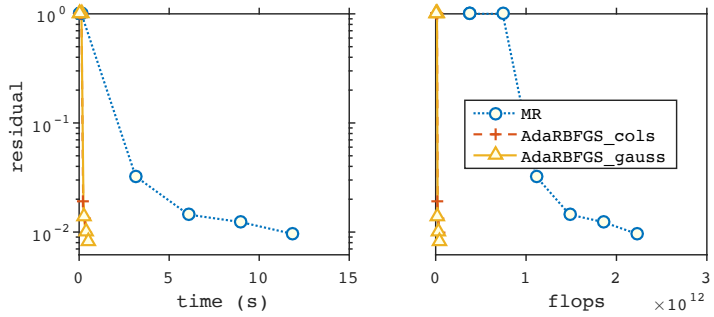
We now investigate the empirical convergence of the methods MR, AdaRBFGS_cols and AdaRGFBS_gauss when initiated with the same starting matrix $X_0 = I$, see Figures 4.6, 4.7 and 4.8. We did not include the Newton-Schultz method in these figures because it diverged on all experiments when initiated from $X_0 = I$. Again we observe that, as the dimension grows, only the two variants of the AdaRBFGS are capable of inverting the matrix to the desired $10^{-2}$ precision in a reasonable amount of time. Furthermore, the AdaRBFGS_gauss variant had the overall best best performance.



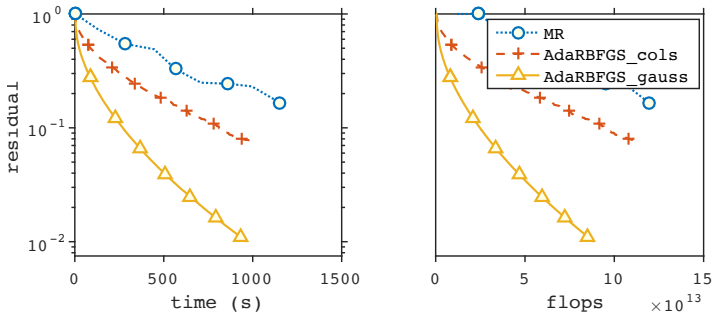

**(a)** *aloi*




**(b)** *protein*

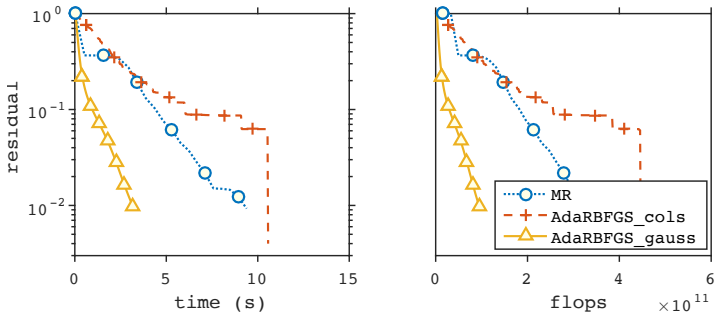



**(c)** *gisette_scale*

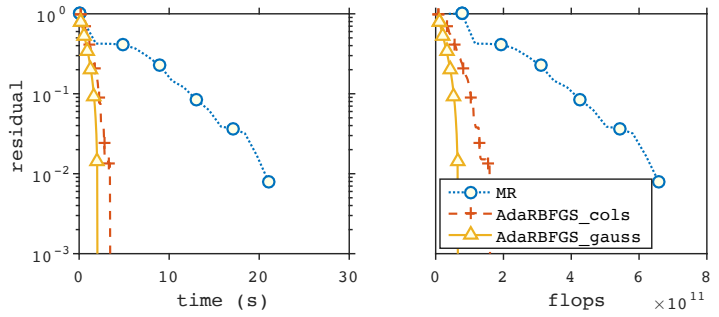



**(d)** *real_sim*

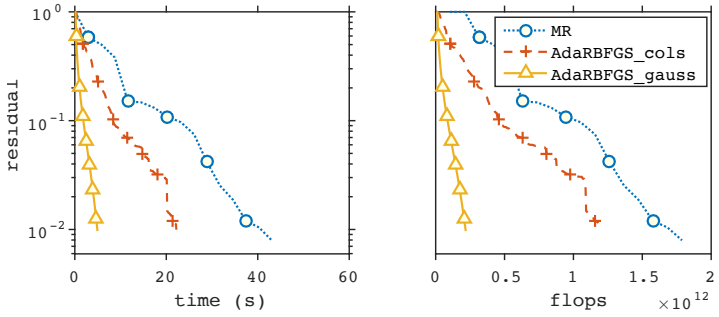
**Figure 4.6:** *The performance of Newton-Schulz, MR, AdaRBFGS_gauss and AdaRBFGS_cols methods on the Hessian matrix of four LIBSVM test problems: (a)* `aloi`*:* $(m;n) = (108,000;128)$ *(b)* `protein`*:* $(m;n) = (17,766;357)$ *(c)* `gisette_scale`*:* $(m;n) = (6000;5000)$ *(d)* `real-sim`*:* $(m;n) = (72,309;20,958)$*. The starting matrix* $X_0 = I$ *was used for all methods.*



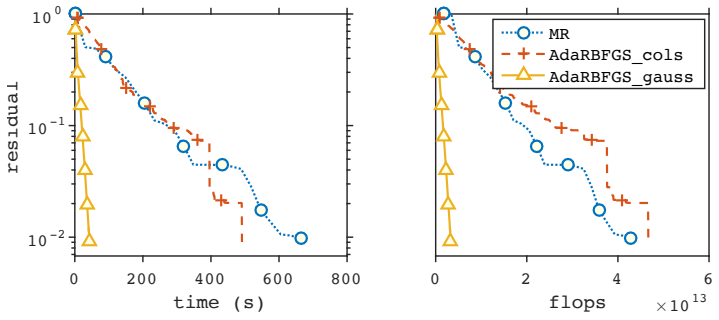

**(a)** *Bates/Chem97ZtZ*




**(b)** *FIDAP/ex9*

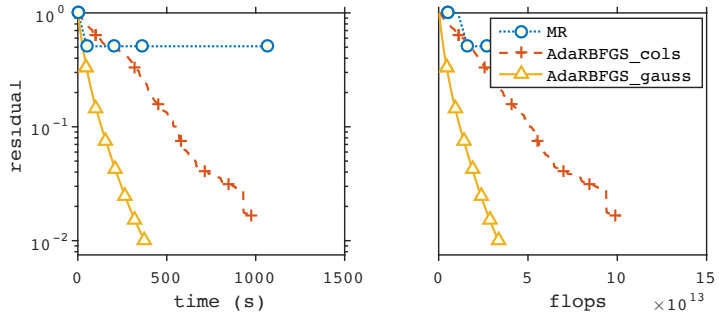



**(c)** *Nasa/nasa4704*

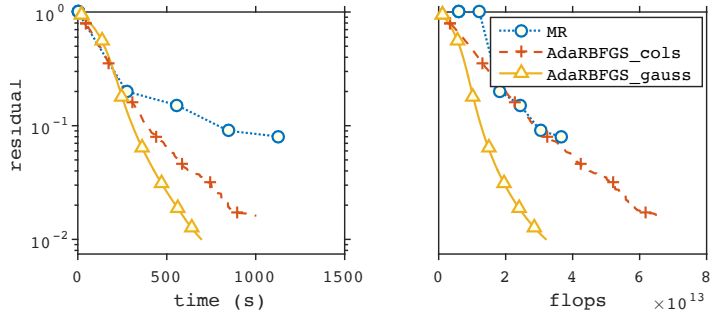



**(d)** *HB/bcsstk18*

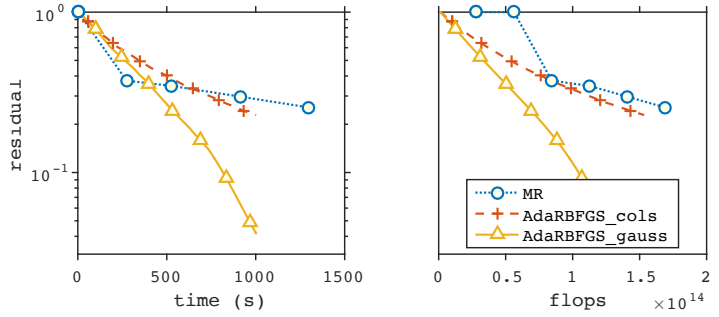
**Figure 4.7:** *The performance of Newton-Schulz, MR, AdaRBFGS_gauss and AdaRBFGS_cols on (a)* `Bates-Chem97ZtZ`*:* $n = 2\,541$*, (b)* `FIDAP/ex9`*:* $n = 3,363$*, (c)* `Nasa/nasa4704`*:* $n = 4,704$*, (d)* `HB/bcsstk18`*:* $n = 11,948$*. The starting matrix* $X_0 = I$ *was used for all methods.*




**(a)** *Pothen/bodyy4*




**(b)** *ND/nd6k*




**(c)** *GHS_psdef/wathen100*

**Figure 4.8:** *The performance of Newton-Schulz, MR, AdaRBFGS_gauss and AdaRBFGS_cols on (a)* `Pothen/bodyy4`*:* $n = 17,546$ *(b)* `ND/nd6k`*:* $n = 18,000$ *(c)* `GHS_psdef/wathen100`*:* $n = 30,401$*. The starting matrix* $X_0 = I$ *was used for all methods.*

CHAPTER 5

# Conclusion and Future Work

*Fuils an bairns soud never see things hauf duin.*
It needs powers of perception and mature judgement to visualise the result of an enterprise.

Scottish proverb.

This thesis laid the foundational work for a class of randomized methods for solving linear systems, equipt with convergence analysis, and general enough to accommodate several existing methods (Randomized Kaczmarz and Coordinate descent), but also allows for the design of completely new methods with continuous samplings, optimized samplings and block variants. Thus there is still much to explore in designing new randomized methods for solving linear systems. One particularly promising direction is to use new sophisticated sketching matrices $S$, such as the Walsh-Hadamard matrix [75, 96], to design new practical and efficient methods.

Using duality theory, we redeveloped the sketch-and-project method through a dual perspective which we refer to as the Stochastic Dual Ascent (SDA) method. This perspective allowed us to extend the application of the sketch-and-project methods to that of finding the projection of a given vector onto the solution space of a linear system. This extensions enables us to solve the distributed consensus problem and it reveals that a standard randomized gossip algorithm is a special case of the sketch-and-project methods.

Applying the same sketch and project principle to the inverse equations ($AX = I$ or $XA = I$), we developed new randomized methods for inverting nonsymmetric and symmetric matrices. These randomized methods can be viewed as randomized quasi-Newton updates. Through a dual perspective, we establish a new connection connection between the quasi-Newton methods and the approximate inverse preconditioning methods. Furthermore, we design a highly efficient method, AdaRBFGS, for calculating approximate inverses of positive definite matrices. This opens up many avenues for developing stochastic preconditioning and variable metric methods. Indeed, the AdaRBFGS method has already been used as the basis of a new stochastic variable metric method [109].

Perhaps the most exciting direction for future work is to extend the sketch-and-project framework to solve linear equations in other settings. For instance, our framework could be extended to solve linear equations in a general Euclidean place. This would open up several

new application areas including linear matrix equations such Sylvester equation, Lyapunov equation and more [120]. Finally, perhaps the sketch-and-project framework can be extended to solving linear equations defined by bounded linear operators between two Hilbert spaces. Such a development would lead to new randomized methods for solving differential and integral equations.

# Bibliography


[1] H. Avron, P. Maymounkov, and S. Toledo. "Blendenpik: supercharging LAPACK's least-squares solver". *SIAM Journal on Scientific Computing* 32 (3) (2010), pp. 1217–1236.

[2] G. Ballard, E. Carson, J. Demmel, M. Hoemmen, N. Knight, and O. Schwartz. "Communication lower bounds and optimal algorithms for numerical linear algebra". *Acta Numerica* 23 (May 2014), pp. 1–155.

[3] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1994.

[4] S. Bellavia. "An inexact interior point method". *Journal of Optimization Theory and Applications* 96 (1) (1998), pp. 109–121.

[5] Y. Bengio, O. Delalleau, and N. Le Roux. "Label propagation and quadratic criterion". In: *Semi-Supervised Learning*. Ed. by O. Chapelle, B. Schölkopf, and A. Zien. MIT Press, 2006, pp. 193–216.

[6] M. Benzi and M. Tůma. "Comparative study of sparse approximate inverse preconditioners". *Applied Numerical Mathematics* 30 (2–3) (1999), pp. 305–340.

[7] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., 1989.

[8] D. P. Bertsekas and J. N. Tsitsiklis. "Some aspects of parallel and distributed iterative algorithms - a survey, ." *Automatica* 27 (1) (1991), pp. 3–21.

[9] R. Bhatia. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton, NJ, USA: Princeton University Press, 2008, p. 264.

[10] M. D. Bingham. "A new method for obtaining the inverse matrix". *Journal of the American Statistical Association* 36 (216) (1941), pp. 530–534.

[11] R. F. Boisvert, R. Pozo, K. Remington, R. F. Barrett, and J. J. Dongarra. "Matrix Market : a web resource for test matrix collections". In: *The Quality of Numerical Software: Assessment and Enhancement*. Ed. by R. Boisvert. London: Chapman & Hall, 1997, pp. 125–137.

[12] J. M. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization*. Springer-Verlag New York, 2006.

[13] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. "Randomized gossip algorithms". *IEEE Transactions on Information Theory* 52 (6) (2006), pp. 2508 –2530.

[14] J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. "Parallel coordinate descent for $L1$-regularized loss minimization". In: *28th International Conference on Machine Learning*. 2011.

[15] C. G. Broyden. “A class of methods for solving nonlinear simultaneous equations”. *Mathematics of Computation* 19 (92) (1965), pp. 577–593.

[16] Y. Censor. “Row-action methods for huge and sparse systems and their applications”. *SIAM Review* 23 (4) (1981), pp. 444–455.

[17] C. C. Chang and C. J. Lin. “LIBSVM : a library for support vector machines”. *ACM Transactions on Intelligent Systems and Technology* 2 (3) (Apr. 2011), pp. 1–27.

[18] M. Charikar, K. Chen, and M. Farach-Colton. “Finding frequent items in data streams”. In: *Proceedings of the 29$^{th}$ International Colloquium on Automata, Languages and Programming (ICALP)*. Springer-Verlag London, 2002, pp. 693–703.

[19] E. Chow and Y. Saad. “Approximate inverse preconditioners via sparse-sparse iterations”. *SIAM Journal of Scientific Computing* 19 (3) (1998), pp. 995–1023.

[20] A. Chronopoulos and C. Gear. “S-step iterative methods for symmetric linear systems”. *Journal of Computational and Applied Mathematics* 25 (2) (1989), pp. 153 –168.

[21] G. Cormode and S. Muthukrishnan. “An improved data stream summary: the count-min sketch and its applications”. *Journal of Algorithms* (55) (2005), pp. 29–38.

[22] D. Csiba, Z. Qu, and P. Richtárik. “Stochastic dual coordinate ascent with adaptive probabilities”. *International Convferences on Machine Learning* (2015).

[23] L. Dai, M. Soltanalian, and K. Pelckmans. “On the randomized Kaczmarz algorithm”. *IEEE Signal Processing Letters* 21 (3) (2014), pp. 330–333.

[24] W. C Davidon. *Variable metric method for minimization*. Tech. rep. A.E.C. Research and Development Report, ANL-5990, 1959.

[25] W. C. Davidon. “Variance algorithms for minimization”. *Computer Journal* 10 (1968), pp. 406–410.

[26] T. A. Davis and Y. Hu. “The University of Florida sparse matrix collection”. *ACM Transactions on Mathematical Software* 38 (1) (2011), 1:1–1:25.

[27] A. Defazio, F. Bach, and S. Lacoste-Julien. “SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives”. *arXiv:1407.0202* (2014).

[28] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. “Inexact Newton methods”. *SIAM Journal on Numerical Analysis* 19 (2) (1982), pp. 400–408.

[29] J. W. Demmel. “The probability that a numerical analysis problem is difficult”. *Mathematics of Computation* 50 (182) (1988), pp. 449–449.

[30] C. A. Desoer and B. H. Whalen. “A note on pseudoinverses”. *Journal of the Society of Industrial and Applied Mathematics* 11 (2) (1963), pp. 442–447.

[31] J. Dongarra, M. A. Heroux, and P. Luszczek. “High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems.” *IJHPCA* 30 (1) (2016), pp. 3–10.

[32] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. “Faster least squares approximation”. *Numerische Mathematik* 117 (2) (2011), pp. 219–249.

[33] A. Edelman. "On the distribution of a scaled condition number". *Mathematics of Computation* 58 (197) (1992), pp. 185–185.

[34] S. C. Eisenstat and H. F. Walker. "Choosing the forcing terms in an inexact Newton method". *SIAM Journal on Scientific Computing* 17 (1994), pp. 16–32.

[35] V. Faber and T. Manteuffel. "Necessary and sufficient conditions for the existence of a conjugate gradient method". *SIAM Journal on Numerical Analysis* 21 (1984), pp. 315–339.

[36] O. Fercoq. "Parallel coordinate descent for the Adaboost problem". In: *Proc. of the International Conference on Machine Learning and Applications*. 2013.

[37] O. Fercoq and P. Richtárik. "Accelerated, parallel and proximal coordinate descent". *SIAM Journal on Optimization* 25 (4) (2015), pp. 1997–2023.

[38] O. Fercoq and P. Richtárik. "Smooth minimization of nonsmooth functions by parallel coordinate descent". *arXiv:1309.5885* (2013).

[39] O. Fercoq, Z. Qu, P. Richtárik, and M. Takáč. "Fast distributed coordinate descent for minimizing non-strongly convex losses". *IEEE International Workshop on Machine Learning for Signal Processing* (2014).

[40] B. R. Fletcher and M. J. D. Powell. "A rapidly convergent descent method for minimization". *The Computer Journal* 6 (2) (1963), pp. 163–168.

[41] A. Gaul and N. Schlömer. "Preconditioned recycling Krylov subspace methods for self-adjoint problems". *arXiv:1208.0264* (2014).

[42] D. Goldfarb. "Modification methods for inverting matrices and solving systems of linear algebraic equations". *Mathematics of Computation* 26 (120) (1972), pp. 829–829.

[43] D. Goldfarb. "A family of variable-metric methods derived by variational means". *Mathematics of Computation* 24 (109) (1970), pp. 23–26.

[44] G. H. Golub and C. F. Van Loan. *Matrix Computations (4th Edition)*. 2013, p. 780.

[45] J. Gondzio. "Convergence analysis of an inexact feasible interior point method for convex quadratic programming". *SIAM Journal on Optimization* 23 (3) (2013), pp. 1510–1527.

[46] N. I. M. Gould and J. A. Scott. "Sparse approximate-inverse preconditioners using norm-minimization techniques". *SIAM Journal on Scientific Computing* 19 (2) (1998), pp. 605–625.

[47] R. M. Gower and A. L. Gower. "Higher-order reverse automatic differentiation with emphasis on the third-order". *Mathematical Programming* 155 (1) (2014), pp. 81–103.

[48] R. M. Gower and J. Gondzio. "Action constrained quasi-Newton methods". *arXiv:1412.8045v1* (2014).

[49] R. M. Gower and P. Richtárik. "Stochastic dual ascent for solving linear systems". *arXiv:1512.06890* (2015).

[50] R. M. Gower and M. P. Mello. "Computing the sparsity pattern of hessians using automatic differentiation". *ACM Transactions on Mathematical Software* 40 (2) (2014), 10:1–10:15.

[51] R. M. Gower and P. Richtárik. "Randomized iterative methods for linear systems". *SIAM Journal on Matrix Analysis and Applications* 36 (4) (2015), pp. 1660–1690.

[52] R. M. Gower and P. Richtárik. "Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms". *arXiv:1602.01768* (2016).

[53] M. Grant and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. 2014.

[54] S. Gratton, A. Sartenaer, and J. T. Ilunga. "On a class of limited memory preconditioners for large-scale nonlinear least-squares problems". *SIAM Journal on Optimization* 21 (3) (2011), pp. 912–935.

[55] B. J. Greenstadt. "Variations on variable-metric methods". *Mathematics of Computation* 24 (109) (1969), pp. 1–22.

[56] A. Griewank. "Broyden updating, the good and the bad!" *Optimization Stories,* Documenta Mathematica. Extra Volume: Optimization Stories (2012), pp. 301–315.

[57] P. Hennig. "Probabilistic interpretation of linear solvers". *SIAM Journal on Optimization* 25 (1) (2015), pp. 234–260.

[58] M. R. Hestenes and E. Stiefel. "Methods of conjugate gradients for solving linear systems". *Journal of research of the National Bureau of Standards* 49 (6) (1952).

[59] 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 25$^{th}$ International Conference on Machine Learning*. 2008, pp. 408–415.

[60] T. Huckle and A. Kallischko. "Frobenius norm minimization and probing for preconditioning". *International Journal of Computer Mathematics* 84 (8) (2007), pp. 1225–1248.

[61] R. Johnson and T. Zhang. "Accelerating stochastic gradient descent using predictive variance reduction". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 315–323.

[62] M. S. Kaczmarz. "Angenäherte auflösung von systemen linearer gleichungen". *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques* 35 (1937), pp. 355–357.

[63] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995.

[64] L. Y. Kolotilina and A. Y. Yeremin. "Factorized sparse approximate inverse preconditionings I. Theory". *SIAM Journal on Matrix Analysis and Applications* 14 (1) (1993), pp. 45–58.

[65] J. Konečný and P. Richtárik. "S2GD: semi-stochastic gradient descent methods". *arXiv:1312.1666* (2014).

[66] J. Konečný, J. Liu, P. Richtárik, and M. Takáč. "Mini-batch semi-stochastic gradient descent in the proximal setting". *IEEE Journal of Selected Topics in Signal Processing* 10 (2) (2016), pp. 242–255.

[67] Y. T. Lee and A. Sidford. “Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems”. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS* (2013), pp. 147–156.

[68] D. Leventhal and A. S. Lewis. “Randomized methods for linear constraints: convergence rates and conditioning”. *Mathematics of Operations Research* 35 (3) (2010), pp. 641–654.

[69] D. Leventhal and A. Lewis. “Randomized Hessian estimation and directional search”. *Optimization* 60 (3) (2011), pp. 329–345.

[70] W. Li and Z. Li. “A family of iterative methods for computing the approximate inverse of a square matrix and inner inverse of a non-square matrix”. *Applied Mathematics and Computation* 215 (9) (2010), pp. 3433–3442.

[71] J. Liesen and Z. Strakos. *Krylov Subspace Methods : Principles and Analysis*. Oxford: Oxford University Press, 2014.

[72] Q. Lin, Z. Lu, and L. Xiao. “An accelerated proximal coordinate gradient method”. In: *Advances in Neural Information Processing Systems 27*. 2014.

[73] J. Liu and S. J. Wright. “An accelerated randomized Kaczmarz algorithm”. *Mathematics of Computation* 85 (297) (2016), pp. 153–178.

[74] J. Liu, S. J. Wright, and S. Srikrishna. “An asynchronous parallel randomized Kaczmarz algorithm”. *arXiv:1401.4780* (2014).

[75] Y. Lu, P. Dhillon, D. P. Foster, and L. Ungar. “Faster ridge regression via the subsampled randomized Hadamard transform”. In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 369–377.

[76] A. Ma, D. Needell, A. Ramdas, and N. A. Mar. “Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods”. *arXiv:1503.08235* (2015), pp. 1–16.

[77] M. W. Mahoney. “Randomized algorithms for matrices and data”. *Foundations and Trends® in Machine Learning* 3 (2) (2011), pp. 123–224.

[78] E. H. Moore. “Abstract for “On the reciprocal of the general algebraic matrix””. *Bulletin of the American Mathematical Society* 26 (1920), pp. 394–395.

[79] B. A. Murtagh and R. W. H. Sargent. “A constrained minimization method with quadratic convergence”. In: *Optimization*. Ed. by R. Fletcher. London: Academic Press, 1969.

[80] I. Necoara and D. Clipici. *Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC*. Tech. rep. Politehnica University of Bucharest, 2012.

[81] I. Necoara and A. Patrascu. “A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints”. *Computational Optimization and Applications* 57 (2) (2014), pp. 307–337.

[82] D. Needell. “Randomized Kaczmarz solver for noisy linear systems”. *BIT* 50 (2) (2010), pp. 395–403.

[83] D. Needell, N. Srebro, and R. Ward. “Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm”. *Mathematical Programming* 155 (1) (2015), pp. 549–573.

[84] D. Needell and J. A. Tropp. “Paved with good intentions: analysis of a randomized block Kaczmarz method”. *Linear Algebra and Its Applications* 441 (August) (2012), pp. 199–221.

[85] D. Needell, R. Zhao, and A. Zouzias. “Randomized block Kaczmarz method with projection for solving least squares”. *Linear Algebra and its Applications* 484 (2015), pp. 322–343.

[86] Y. Nesterov. “Efficiency of coordinate descent methods on huge-scale optimization problems.” *SIAM Journal on Optimization* 22 (2) (2012), pp. 341–362.

[87] Y. Nesterov. *Random gradient-free minimization of convex functions*. Tech. rep. Louvain: ECORE Universite catholique de Louvain, Feb. 2011, pp. 1–32.

[88] F. Niu, B. Recht, C. Ré, and S. Wright. “Hogwild!: a lock-free approach to parallelizing stochastic gradient descent”. In: *Advances in Neural Information Processing Systems 24*. 2011.

[89] J. Nocedal. “Updating quasi-Newton matrices with limited storage”. *Mathematics of Computation* 35 (151) (1980), p. 773.

[90] A. Olshevsky and J. N. Tsitsiklis. “Convergence speed in distributed consensus and averaging”. *SIAM Journal on Control and Optimization* 48 (1) (2009), pp. 33–55.

[91] P. Oswald and W. Zhou. “Convergence analysis for Kaczmarz-type methods in a Hilbert space framework”. *Linear Algebra and its Applications* 478 (2015), pp. 131–161.

[92] C. C. Paige and M. A. Saunders. “Solution of sparse indefinite systems of linear equations”. *SIAM Journal on Numerical Analysis* 12 (4) (1975), pp. 617–629.

[93] V. Pan and R. Schreiber. “An improved Newton iteration for the generalized inverse of a matrix, with applications”. *SIAM J. Scientific Computing* 12 (5) (1991), pp. 1109–1130.

[94] R. Penrose. “A generalized inverse for matrices”. *Mathematical Proceedings of the Cambridge Philosophical Society* 51 (3) (1955), pp. 406–413.

[95] M. Pilanci and M. Wainwright. “Randomized sketches of convex programs with sharp guarantees”. *Information Theory, IEEE Transactions on* 61 (9) (2015), pp. 5096–5115.

[96] M. Pilanci and M. J. Wainwright. “Iterative Hessian sketch : Fast and accurate solution approximation for constrained least-squares”. *Journal of Machine Learning Research* 17 (2016), pp. 1–33.

[97] M. Pilanci and M. J. Wainwright. “Newton sketch : A linear-time optimization algorithm with linear-quadratic convergence”. *arXiv:1505.02250* (2015).

[98] Z. Qu and P. Richtárik. “Coordinate descent with arbitrary sampling I: algorithms and complexity”. *arXiv:1412.8060* (2014).

[99] Z. Qu and P. Richtárik. “Coordinate descent with arbitrary sampling II: expected separable overapproximation”. *arXiv:1412.8063* (2014).

[100] Z. Qu, P. Richtárik, and T. Zhang. “Quartz: randomized dual coordinate ascent with arbitrary sampling”. In: *Advances in Neural Information Processing Systems 28*. 2015, pp. 1–34.

[101] Z. Qu, P. Richtárik, M. Takáč, and O. Fercoq. "SDNA: Stochastic dual Newton ascent for empirical risk minimization". In: *Proceedings of the 33rd International Conference on Machine Learning*. 2016.

[102] A. Ramdas. "Rows vs columns for linear systems of equations - randomized Kaczmarz or coordinate descent ?" *arXiv:1406.5295* (2014).

[103] P. Richtárik and M. Takáč. "Distributed coordinate descent method for learning with big data". *Journal of Machine Learning Research* (2013).

[104] P. Richtárik and M. Takáč. "Efficient serial and parallel coordinate descent method for huge-scale truss topology design". In: *Operations Research Proceedings*. Springer, 2012, pp. 27–32.

[105] P. Richtárik and M. Takáč. "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function". *Mathematical Programming* 144 (1) (2014), pp. 1–38.

[106] P. Richtárik and M. Takáč. "On optimal probabilities in stochastic coordinate descent methods". *Optimization Letters* (2015), pp. 1–5.

[107] P. Richtárik and M. Takáč. "Parallel coordinate descent methods for big data optimization problems". *Mathematical Programming* (2015), pp. 1–52.

[108] H. Robbins and S. Monro. "A stochastic approximation method". *Annals of Mathematical Statistics* 22 (1951), pp. 400–407.

[109] D. G. Robert M. Gower and P. Richtárik. "Stochastic block BFGS: squeezing more curvature out of data". *Proceedings of the 33rd International Conference on Machine Learning* (2016).

[110] D. P. Robinson and R. E. H. Tappenden. "A flexible admm algorithm for big data applications". *arXiv:1502.04391* (2015).

[111] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. Vol. 104. Monographs on Statistics and Applied Probability. London: Chapman & Hall, 2005.

[112] Y. Saad. *Iterative Methods for Sparse Linear Systems*. 2nd. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.

[113] Y. Saad and M. H. Schultz. "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM Journal on Scientific and Statistical Computing* 7 (3) (1986), pp. 856–869.

[114] M. Schmidt, N. Le Roux, and F. Bach. "Minimizing finite sums with the stochastic average gradient". *arXiv:1309.2388* (2013).

[115] G. Schulz. "Iterative berechung der reziproken matrix". *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik* 13 (1) (1933), pp. 57–59.

[116] S. Shalev-Shwartz and A. Tewari. "Stochastic methods for $\ell_1$-regularized loss minimization". *Journal of Machine Learning Research* 12 (2011), pp. 1865–1892.

[117] S. Shalev-Shwartz and T. Zhang. "Accelerated mini-batch stochastic dual coordinate ascent". In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 378–385.

[118] S. Shalev-Shwartz and T. Zhang. "Stochastic dual coordinate ascent methods for regularized loss". *Journal of Machine Learning Research* 14 (1) (Feb. 2013), pp. 567–599.

[119] D. F. Shanno. "Conditioning of quasi-Newton methods for function minimization". *Mathematics of Computation* 24 (111) (1971), pp. 647–656.

[120] V. Simoncini. "Computational methods for linear matrix equations". *(Survey article) to appear in SIAM Review* (2014), pp. 1–58.

[121] S. U. Stich, C. L. Müller, and B. Gärtner. "Optimization of convex functions with random pursuit". *SIAM Journal on Optimization* 23 (2) (2014), pp. 1284–1309.

[122] S. U. Stich, C. L. Müller, and B. Gärtner. "Variable metric random pursuit". *Mathematical Programming* 156 (1) (2015), pp. 549–579.

[123] S. U. Stich and C. L. Müller. "On spectral invariance of randomized Hessian and covariance matrix adaptation schemes". In: *Parallel Problem Solving from Nature - PPSN XII*. Springer Berlin Heidelberg, 2012, pp. 448–457.

[124] S. U. Stich. "Convex Optimization with Random Pursuit". PhD thesis. ETH Zurich, 2014.

[125] T. Strohmer and R. Vershynin. "A randomized Kaczmarz algorithm with exponential convergence". *Journal of Fourier Analysis and Applications* 15 (2) (2009), pp. 262–278.

[126] M. Takáč, A. Bijral, P. Richtárik, and N. Srebro. "Mini-batch primal and dual methods for SVMs". *In 30th International Conference on Machine Learning* 28 (2013), pp. 537–552.

[127] Q. Tao, K. Kong, D. Chu, and G. Wu. "Stochastic coordinate descent methods for regularized smooth and nonsmooth losses". *Machine Learning and Knowledge Discovery in Databases* (2012), pp. 537–552.

[128] R. Tappenden, P. Richtárik, and J. Gondzio. "Inexact block boordinate descent method: complexity and preconditioning". *arXiv:1304.5530* (2013).

[129] C. Wang and A. Xu. "An inexact accelerated proximal gradient method and a dual Newton-CG method for the maximal entropy problem". *Journal of Optimization Theory and Applications* 157 (2) (2013), pp. 436–450.

[130] D. Williams. *Probability with Martingales*. Cambridge mathematical textbooks. Cambridge University Press, 1991.

[131] M. A. Woodbury. *Inverting modified matrices*. Tech. rep. Rep. no. 42, Statistical Research Group, Princeton University, 1950.

[132] S. J. Wright. "Accelerated block-coordinate relaxation for regularized optimization". *SIAM Journal on Optimization* (1) (2012), pp. 159–186.

[133] S. J. Wright. "Coordinate descent methods". *Mathematical Programming* (1) (2015), pp. 3–34.

[134] L. Xiao and T. Zhang. "A proximal stochastic gradient method with progressive variance reduction". *arXiv:1403.4699* (2014).

[135] T. Yang. "Trading computation for communication: distributed stochastic dual coordinate ascent". In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 629–637.

[136] Y. Zhang and L. Xiao. "Stochastic primal-dual coordinate method for regularized empirical risk minimization". In: *Proceedings of the 32$^{nd}$ International Conference on Machine Learning*. 2015, pp. 353–361.

[137] P. Zhao and T. Zhang. "Stochastic optimization with importance sampling for regularized loss minimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. 2015, pp. 1–9.

[138] X. Y. Zhao, D. Sun, and K. C. Toh. "A Newton-CG augmented lagrangian method for semidefinite programming". *SIAM Journal on Optimization* 20 (4) (2010), pp. 1737–1765.

[139] A. Zouzias and N. M. Freris. "Randomized extended Kaczmarz for solving least-squares". *SIAM Journal on Matrix Analysis and Applications* 34 (2) (2013), pp. 773–793.

[140] A. Zouzias and N. M. Freris. "Randomized gossip algorithms for solving Laplacian systems". In: *IEEE European Control Conference (ECC)*. 2015, pp. 1920–1925.