Randomised Kaczmarz Method

Yutong Ma

May 2019

A thesis submitted for partial fulfilment of requirements of the degree of Bachelor of Science with Honours in Computational Mathematics of the Australian National University

Australian National University

For my beloved parents, Yaqun, Xiaowei, and grandma, Shuyin

Declaration

The work in this thesis is my own, and, to the best of my knowledge and belief, contains no material which has been published or accepted for the award of any other degree or diploma in any University, except where due reference is always made. All main sources of help are acknowledged in the thesis.

Yutong Ma

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr Qinian Jin, for his valuable advice, support, and patience throughout the year. Qinian introduced me to the topic of the thesis, while his guidance and encouragements helped me study much deeper. My sincere thanks to ANU MSI honours convenor, Dr Joan Licata, for her suggestions and supports.
I would like to thank my parents for their continuous support and encouragement. Also, I want to thank my friends, personally, Mingze Ni, Di Zhao, and Ping Zhang for their accompanying.

Abstract

Solving systems of linear equations, iterative methods are widely used for computing efficiency, though direct methods are robust and accurate. Kaczmarz method was first introduced in 1973 for square matrices in Euclidean space and was utilised in Computational Tomography, while it has been noticed after a few decades. Although Kaczmarz proved its convergence, the rate of convergence is hard to obtain. Until the recent decade, Strohmer and Vershynin obtained the convergence rate of the randomised version of the Kaczmarz method, which primarily increased the efficiency. In this thesis, we will construct the randomised Kaczmarz method based on original Kaczmarz method, and prove through the properties of convergence rate and pre-asymptotic convergence for both exact input and noised input. Then we introduced an accelerated version of randomised Kaczmarz methods and analysing its converging property. Additionally, we extend the randomised Kaczmarz method into solving systems of linear inequalities, with proving of convergence. After that, we will focus on solving a system of linear equations with sparse solutions. Randomised Kaczmarz method has its advantage of sparse solution problems due to its using less memory in computing. We thus construct a randomised sparse Kaczmarz method by applying soft-threshold function and based on this, we further build an exact-step randomised sparse Kaczmarz method based upon introducing concepts on Bregman distance and projection. Also, we carefully prove the convergence of these methods by adopting concepts and theorems in convex analysis. Moreover, by conducting numerical experiments to the methodologies we investigated on, we find that randomised Kaczmarz method has surpassed even Conjugate Gradient and Least Square (CGLS) method -a widely known iterative method- when solving overdetermined systems. Also, we find that the randomised sparse Kaczmarz method has much better performance than randomised Kaczmarz methods. In this way, we proposed a series of randomised versions of the Kaczmarz method, which may be implemented under many real-world problems.

Contents

Notation and terminology

In the following, x is a vector, A is a matrix, f is a function.

Notation

$\|\| \cdot \|\|$ or $\|\| \cdot \|\|_2$	2-norm
$\|\| \cdot \|\|_F$	Fröbenius norm or F-norm
∂f	subgradient of f
$\text{span}\{x_1, \cdots, x_n\}$	span of vectors

Terminology

convex function	A function $f : \mathbb{R}^n \to (-\infty, \infty]$ is called convex if for any points $x_0, x_1 \in \mathbb{R}^n$ and $0 < t < 1$ there holds $f(tx_1 + (1-t)x_0) < tf(x_1) + (1-t)f(x_0)$.

Chapter 1

Introduction

1.1 Background and Motivation

Kaczmarz's introduction of Kaczmarz methods into solving linear systems of Ax=b in 1937 [20] remained bare responses until Tompkins [41] in 1949 and Forsythe [11] in 1953 researched on Kaczmarz method again, and the procedure was utilised in a bunch of real-world problems in different discipline including computer tomography [14], image processing and contemporary harmonic analysis. This thesis aims to theoretically analyse convergence properties through existing randomised Kaczmarz method (RKM) to those further developed methods based on RKM, such as accelerated RKM, random sparse Kaczmarz method and RKM for solving inequalities, and apply these algorithms to compare their performances with that of the traditional Kaczmarz Method for solving systems of linear equations. In this way, we hope to assess how efficiency these algorithms compare with a well-known iterative method, conjugate descent method. In order to further examine the performance of RKM when implemented in solving linear systems, we chose matrices generated randomly by Gaussian distribution and solution vectors generated randomly by Gaussian distribution or sparsely, with adding noise level to construct an observed vector for our application respectively.

Methodologies in the thesis originate from the problem of solving the system of linear equations, specifically, given

$$Ax = b,$$

where A is a real matrix, b is a real vector, Gaussian elimination, a direct method, arises naturally as a basic solution to the problem, which is doing row operations of corresponding matrix into a row echelon form matrix to get the exact solution

[38]. It requires storage of all $n \times n$ entries, with about $2n^3/3$ operations for the square matrix $A \in \mathbb{R}^{n \times n}$ [13]. The Lower-Upper (LU) method can be viewed as an analogue of the Gaussian elimination method. The LU method is decomposing the corresponding matrix into a product of lower triangular matrix L and an upper triangular matrix U and it can be utilised to solve linear systems by calculating z from $Lz = b$, and thus to calculate x from $Ux = z$. There is also the Cholesky factorization method with about $m^3/3$ times of operations [15]. These direct methods are predictable and robust [32] [8]. However, it is widely aware that these methods are not suitable to be used while the system is huge, since the use of the methods when facing large corresponding matrix, required computing through the whole matrix to obtaining an associated matrix of crucial step for the final solution of linear equations. A similar problem carries through to solving linear equations when we need to do a minor change to information of the corresponding matrix, making it unsuitable to be used as computing resources consuming. Kaczmarz introduced the idea of the Kaczmarz method for solving systems of linear equations in 1937 [20], after which applied mathematician noticed that Kaczmarz method might ameliorate the efficiency of solving systems of linear equations.

As stated before, the Gaussian elimination method is not suitable to be used for very large linear systems. Iterative methods may be introduced into solving systems of linear equations. Generally, iterative methods are required well-understanding of the problem. Due to realising sparse solution problems arising from engineering, iterative methods may lessen memory requirements a lot [13]. Thus, iterative methods would be developed upon problem-related situations and given specific applications. Iterative methods have been regarded as the preferred method in many circumstances because it is conceptually simple and interpretable [32], whereas compared with inevitably expensive direct methods. On the other hand, iterative methods need careful analysis of convergence. Specifically, it can be slow in convergence or even stagnate. Rigorous proofs of convergence and convergence rate of methods remain crucial.

The iterative method this thesis based on, Kaczmarz method, is widely noticed after a long time since it was first proposed in 1937. Meanwhile, Cimmino proposed an iterative method similar but a little bit slower in 1938 [1]. However, due to lack of English translation [40], till few years passed, mathematicians began analysing the method, and Herman [14] extended the method into the non-square matrices, that is overdetermined systems of linear equations, and applying it on computed tomography (CT) and signal processing. It was implemented firstly in

1973 on medical equipment [5]. Recent decades, KM has been evaluated into randomised version, and Strohmer and Vershiynm [39] first derived its convergence rates, and then Needell [27] derived error estimates for noisy linear systems and modified it in several versions of RKM with massive improvement on the rate of convergence [28]. Inspired by such an idea, more and more specific versions of randomised Kaczmarz method has been developed in attempts for applications on certain types of problems, such as applications on improvement and accelerating[10, 24], solving system of inequalities [23] and linear system of equations with sparse solution [34]. Apart from these applications, it has also been extended on to Hilbert space [26] and even infinite-dimensions [21], which generalised in [7], showing that the method has been further applied and analysed theoretically.

1.2 Kaczmarz Method

We will firstly set up our ground method. Kaczmarz method is derived from the projection from one point to the hyperplane $\langle a, x\rangle = b$.

Consider a given system of linear equations $Ax = b$, which can be written as:

$$\begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix},$$

where row vectors of matrix A are denoted by a_i^T.

For any x, we can find its projection $P(x)$ on to the hyperplane $\langle a_i, x\rangle = b_i$ by the following lemma.

Lemma 1.1. *Projection of any $z \in \mathbb{R}^n$ onto the hyperplane $\langle a, x\rangle = b$, where $a, x \in \mathbb{R}^n, \quad b \in \mathbb{R}$ is given by*

$$P(z) = z + \frac{b - \langle a, z\rangle}{||a||_2^2} a.$$

Proof. Since $z - P(z)$ is a vector from the projection point $P(z)$ to the point z, $z - P(z)$ is perpendicular to the hyperplane $\langle a, x\rangle = b$. Thus, $z - P(z)$ is parallel to the normal vector of the hyperplane a.

Let t be a scalar such that $z - P(z) = ta$. Since $P(z)$ is on the hyperplane, by plugging it into $\langle a, x\rangle = b$, we obtain that

$$b = \langle a, P(z)\rangle = \langle a, z - ta\rangle = \langle a, z\rangle - \langle a, ta\rangle = \langle a, z\rangle - t||a||_2^2,$$

from which we can derive that

$$t = \frac{\langle a, z\rangle - b}{||a||_2^2}.$$

Therefore

$$P(z) = z - ta = z + \frac{b - \langle a, z\rangle}{||a||_2^2}a.$$

□

This introduces the basic geometry idea of Kaczmarz method: by finding projections cyclically, we can approach the solution of the given problem as stated before, systems of linear equations. For any $x \in \mathbb{R}^n$, we can find its projection onto a hyperplane defined by a row of the matrix, and then find projection point of the projection onto the hyperplane defined by next row, and by applying this continuously to sweep through each row of the matrix A in order, again and again, we have the Kaczmarz method, which is firstly proposed by Kaczmarz who originally proposed the method for square invertible matrices and proved its convergence [20], though KM was not noticed by mathematicians that time.

Algorithm 1 (KM). *For a given initial guess x_0, define for $k = 0, 1, \cdots$*

$$x_{k+1} = x_k + \frac{b_i - \langle a_i, x_k\rangle}{||a_i||_2^2}a_i,$$

where $i = (k \mod m) + 1$,and $||\cdot|2$ is Euclidean norm in $\mathbb{C}^n$.

We can see the index i by taking a few example steps: suppose $m = 3$, then

$$\begin{aligned}
x_1 &= x_0 + \frac{b_1 - \langle a_1, x_0\rangle}{||a_1||_2^2}a_1,\\
x_2 &= x_1 + \frac{b_2 - \langle a_2, x_1\rangle}{||a_2||_2^2}a_2,\\
x_3 &= x_2 + \frac{b_3 - \langle a_3, x_2\rangle}{||a_3||_2^2}a_3,\\
x_4 &= x_3 + \frac{b_1 - \langle a_1, x_3\rangle}{||a_1||_2^2}a_1,\\
&\cdots\cdots\cdots\cdots
\end{aligned}$$

The demonstration of the idea can be shown in Figure 1.1.

Recall that the Gaussian elimination method requires the whole matrix A when doing the algorithm, while the Kaczmarz method only needs one row of A at each iteration. It can be easily modified if a row of data is changed. Thus,

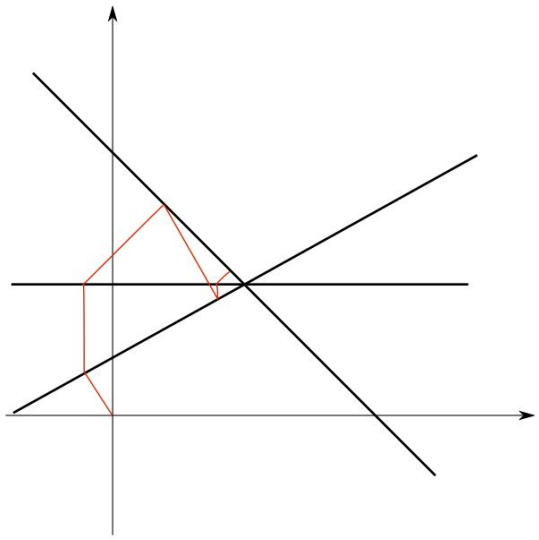


Figure 1.1: Illustration of iteration steps when applying KM

Kaczmarz method has been used in signal and image processing, such as computed tomography [14]. Although Kaczmarz proved the method is convergent [20], the convergence rate of this method is difficult to obtain and hard to compare with other methods.

1.3 Thesis Structure

In the previous sections, we introduced the traditional Kaczmarz method. In this thesis, we are going to develop the traditional Kaczmarz method into randomised Kaczmarz method (RKM), accelerated RKM, RKM for solving systems of linear inequalities, randomised sparse Kaczmarz method (RSKM), and exact-step randomised sparse Kaczmarz method (ERSKM).

We will start by explaining the RKM algorithm for exact data and its convergence properties, and application of it towards the noisy data in Chapter 2. In addition, our methodology involving the accelerated randomised Kaczmarz method will also be explicitly explained in that chapter. We will present an extension of RKM into solving systems of linear inequalities.

Problems on systems of linear equations with sparse solutions are so vital, and it has an advantage when solving by utilising RKM. Thus, we use Chapter 3 to explain the problem with sparse solution and construct the randomised sparse Kaczmarz method, while we develop the exact-step randomised sparse Kaczmarz

method and also analyse convergence properties of both exact and noisy data cases.

In Chapter 4, applying the methodologies described and explained in Chapter 2 and Chapter 3, we will present and discuss the results of our numerical experiments.

Finally, Chapter 5 gives out limitations and future research directions.

The key contribution of the thesis is integrating constructions and related proofs on contents mentioned above with my own approach building from a basic level of mathematical knowledge and with code programming independently with reference mainly based upon following publications: [39, 27, 24, 23, 18, 34].

Chapter 2

Randomised Kaczmarz Method

2.1 Motivating Example

Recall in Chapter 1, we derived the Kaczmarz Method. Firstly let us have a look on an example on solving linear equations.

Example 2.1. Consider the linear systems given by

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \\ 4 \end{pmatrix}.$$

For this problem, we conduct the KM with initial guess $(x_1, x_2)^T = (0, 0)^T$ so we have those steps shown by red lines in following Figure 2.1.

However, if we change the order of the rows of the matrix to reformulate the problem as

$$\begin{pmatrix} 0 & 1 \\ 1 & -2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \\ 4 \end{pmatrix}.$$

This is equivalent to the original problem. By performing KM on this system with the same initial guess, the iteration steps go through as shown by the green lines in Figure 2.1.

We can directly see from the Figure 2.1 that the second system has a faster convergence rate, though they are equivalent. If the first step in KM is performed by projecting the initial guess $(0, 0)^T$ onto the hyperplane $\langle (1, 1)^T, (x_1, x_2)^T \rangle = 4$, we can even directly get the solution. This shows that the order of rows of matrix have a significant influence to the speed of convergence.

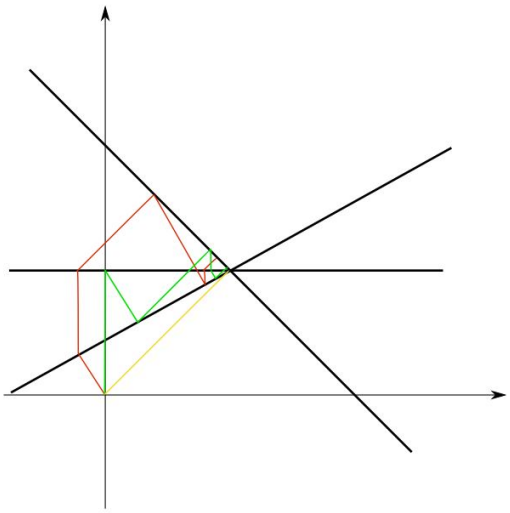


Figure 2.1: Iteration steps when applying KM in different order of matrix rows

The Kaczmarz Method iterates through the rows of A by a certain order leading to the dramatically high dependence of its convergence rate on the order. Intuitively, we can have the row chosen randomly in each iteration to avoid the impact of the order. The index chosen in the iteration can be viewed as a random variable.

2.2 Definitions and Construction

How shall we allocate probabilities to each hyperplane formulated by rows of a matrix? We can simply assign a probability to each row uniformly, that is, with equal probability $1/m$. Also, another method is that we can assign a probability to each row related to the "distance" between the row vector and zero, that is, the probability is proportional to the 2-norm square of each row. Hence, for better allocation of probabilities, we introduce the concept of Fröbenius norm (F-norm) of a matrix, the square root of the sum of all entries' square of the matrix, which is equal to the sum of each row's 2-norm square.

Definition 2.2. The Frobenius norm (F-norm) of a matrix $A \in \mathbb{R}^{m\times n}$ is denoted by $||A||_F$, and is defined by

$$||A||_F := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}|a_{ij}|^2},$$

where a_{ij} indicates the element of the matrix A in its ith row and jth column.

By using F-norm $||A||_F$, we can thus define $P(i_k = i) = \frac{||a_i||_2^2}{||A||_F^2}$. We can check that sum of each row's probability $\sum_{i=1}^{m} P(i_k = i) = 1$. From here, we can construct the randomised Kaczmarz Method based on Algorithm 1 as follows.

Algorithm 2 (RKM). *Given an initial guess x_0, we have the iteration method*

$$x_{k+1} = x_k + \frac{b_{r(k)} - \langle a_{r(k)}, x_k \rangle}{||a_{r(k)}||_2^2} a_{r(k)},$$

where the index $r(k)$ is drawn independent and identically distributed (i.i.d.) from the index set $\{1, 2, ..., m\}$ with the probability for the ith row given by

$$\mathbb{P}(r(k) = i) = \frac{||a_i||_2^2}{||A||_F^2}.$$

Instead of a certain preassigned order in the Kaczmarz's method, the randomised Kaczmarz method sweeps the rows selected randomly with the probability proportional to the 2-norm square of the row vectors of the matrix.

Here, some definitions are given for better discussion for the convergence rate in the following section.

Definition 2.3. The spectral norm of matrix A is $||A||_2$, the largest singular value of A, i.e.

$$||A||_2 = (\text{the maximum eigenvalue of } (A^H A)^{1/2},$$

where A^H is the conjugate transpose of A. We will also use $A^\dagger$ to denote the Moore-Penrose pseudoinverse of A, see [35].

Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ be all the nonzero singular values of A. It is known that

$$\|A\|_2 = \sigma_1, \quad \|A^\dagger\|_2 = \frac{1}{\sigma_r} \quad \text{and} \quad \|A\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2}. \tag{2.1}$$

When solving a linear system, we need to measure how sensitive the answer is to perturbations in the input data and to roundoff errors made during the solution process. For this purpose, we need the concept of a condition number [32].

Definition 2.4. The usual condition number is

$$k(A) := ||A||_2 ||A^\dagger||_2.$$

Definition 2.5. The scaled condition number is

$$\kappa(A) := ||A||_F||A^\dagger||_2.$$

Using (2.1) we can easily derive that

$$\sqrt{n} \leq \kappa(A) \leq \sqrt{n}k(A).$$

2.3 Convergence Properties of RKM

Because the index $r(k)$ is a random variable in the randomised Kaczmarz Method, we will analyse its convergence and derive its convergence rate in terms of the expectation $\mathbb{E}[||x_{k+1} - x^*||^2]$, where x^* denotes a solution of $Ax = b$ and $\mathbb{E}[\cdot]$ denotes expectation with respect to the random row index selection.

2.3.1 Exact Data

In this section, we derive the convergence rate for the randomised Kaczmarz method when the data is given exactly. The following lemma is the key step.

Lemma 2.6. *Let A have full column rank and assume the linear system $Ax = b$ is consistent. Let x^* be the solution of $Ax = b$. For any x let*

$$x^\dagger = x + \frac{b_i - \langle a_i, x\rangle}{\|a_i\|_2^2} a_i$$

with i drawn from the index set $\{1, 2, \cdots, m\}$ with the probability for the ith row given by $\|a_i\|_2^2/\|A\|_F^2$. Then

$$\mathbb{E}[\|x^+ - x^*\|^2] \leq \left(1 - \kappa(A)^{-2}\right) \|x - x^*\|^2.$$

Proof. Since A is of full column rank, there holds $A^\dagger A = I$. Thus, for any $z \in \mathbb{C}$, we have

$$\begin{aligned} \|z\|_2^2 = ||A^\dagger A z||_2^2 &\leq \|A^\dagger\|_2^2 \|Az\|_2^2 \\ &= \|A^\dagger\|_2^2 \sum_{i=1}^m |\langle a_i, z\rangle|^2 \end{aligned}$$

Thus

$$\sum_{i=1}^m |\langle z, a_i\rangle|^2 \geq \frac{\|z\|_2^2}{\|A^\dagger\|_2^2}.$$

By invoking the definition of the scaled condition number $\kappa(A)$, we have

$$\sum_{i=1}^{m} |\langle z, a_i \rangle|^2 \geq \frac{||z||_2^2 ||A||_F^2}{\kappa(A)^2}.$$

Rearranging the terms gives

$$\sum_{i=1}^{m} \frac{\|a_i\|_2^2}{\|A\|_F^2} \left| \langle z, \frac{a_i}{||a_i||} \rangle \right|^2 \geq \|z\|_2^2 \kappa(A)^{-2} \tag{2.2}$$

Let Z be the random vector defined by

$$Z = \frac{a_i}{\|a_i\|_2} \text{ with probability } \frac{\|a_i\|_2^2}{\|A\|_F^2}.$$

Then it follows from (2.2) that

$$\mathbb{E}[|\langle z, Z \rangle|^2] \geq ||z||_2^2 \kappa(A)^{-2}. \tag{2.3}$$

By the definition of x^+ we can see that x^+ is the projection of x onto the hyperplane $\langle a_i, x \rangle = b_i$. Thus $x^+ - x$ is orthogonal to this hyperplane. Because x^* is on this hyperplane, $x^+ - x$ is orthogonal to $x^+ - x^*$. Therefore, by the Pythagorean theorem, we have

$$||x^+ - x^*||_2^2 + ||x^+ - x||_2^2 = ||x - x^*||_2^2.$$

This implies that

$$\mathbb{E}[\|x^+ - x^*\|^2] = \|x - x^*\|^2 - \mathbb{E}[\|x^+ - x\|^2]. \tag{2.4}$$

Note that

$$\begin{aligned} x^+ - x &= \frac{b_i - \langle a_i, x \rangle}{\|a_i\|_2^2} a_i = \frac{\langle a_i, x^* \rangle - \langle a_i, x \rangle}{\|a_i\|_2^2} a_i \\ &= \left\langle \frac{a_i}{\|a_i\|_2}, x^* - x \right\rangle \frac{a_i}{\|a_i\|_2}. \end{aligned}$$

We have

$$\|x^+ - x\| = \left| \left\langle \frac{a_i}{\|a_i\|_2}, x^* - x \right\rangle \right|.$$

Therefore, it follows from the definition of Z and (2.3) that

$$\mathbb{E}[\|x^+ - x\|^2] = \mathbb{E}[|\langle Z, x^* - x \rangle|^2] \geq \kappa(A)^{-2} \|x - x^*\|^2.$$

Combining this with (2.4) gives the desired estimate. □

Using Lemma 2.6, we can easily derive the following convergence rate result for RKM with exact data.

Theorem 2.7. *Let A have full column rank and assume the linear system $Ax = b$ is consistent. Let x^* be the solution of $Ax = b$. Then the sequence $\{x_k\}$ generated by Algorithm 2 converges to x^* in expectation, with the average error*

$$\mathbb{E}[\|x_k - x\|_2^2] \leq (1 - \kappa(A)^{-2})^k ||x_0 - x^*||_2^2.$$

Proof. According to the definition of x_k, we may apply Lemma 2.6 to conclude that

$$\mathbb{E}[\|x_k - x^*\|_2^2 \mid x_{k-1}] \leq (1 - \kappa(A)^{-2})\|x_{k-1} - x^*\|_2^2.$$

Now we take the full expectation on the both sides of the above equation to obtain

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq (1 - \kappa(A)^{-2})\mathbb{E}[\|x_{k-1} - x^*\|_2^2].$$

By recursively using this inequality, we have

$$\begin{aligned}
\mathbb{E}[\|x_k - x^*\|_2^2] &\leq (1 - \kappa(A)^{-2})\mathbb{E}[||x_{k-1} - x^*||_2^2] \\
&\leq (1 - \kappa(A)^{-2})^2\mathbb{E}[||x_{k-2} - x^*||_2^2] \\
&\leq \cdots \\
&\leq (1 - \kappa(A)^{-2})^k\|x_0 - x^*\|_2^2.
\end{aligned}$$

□

2.3.2 Noisy Data

In practical applications, the data b is usually obtained by measurement, and it could be corrupted by noise. Thus, instead of b, we have a noisy data $b + \eta$ with an error vector η added to b. Consequently, instead of the consistent linear system $Ax = b$, we may need to consider the linear system

$$Ax \approx b + \eta.$$

For this noisy system, there may be no solution, so we do not expect it to be consistent. However, the randomised Kaczmarz method is still applicable to generated an iterative sequence which is formulated as follows.

Algorithm 3 (RKM with noisy data)**.** *Given an initial guess $\tilde{x}_0 = x_0$, we have the iteration method*

$$\tilde{x}_{k+1} = \tilde{x}_k + \frac{b_{r(k)} + \eta_{r(k)} - \langle a_{r(k)}, \tilde{x}_k \rangle}{||a_{r(k)}||_2^2} a_{r(k)},$$

where the index $r(k)$ is drawn independent and identically distributed (i.i.d.) from the index set $\{1, 2, ..., m\}$ with the probability for the ith row given by

$$\mathbb{P}(r(k) = i) = \frac{||a_i||_2^2}{||A||_F^2}.$$

We have the following error estimate for the randomised Kaczmarz method with noisy data.

Theorem 2.8. *Let A be of full column rank and assume that the system $Ax = b$ is consistent. Let x^* be the solution of $Ax = b$. Let $\{\tilde{x}_k\}$ be generated by Algorithm 3 using a noisy data $b + \eta$. Then there holds*

$$\mathbb{E}[\|\tilde{x}_k - x^*\|_2^2] \leq (1 - \kappa(A))^k \|x_0 - x^*\|_2^2 + \frac{\|\eta\|_2^2}{\sigma_{\min}(A)^2},$$

where the expectation is taken over the choice of rows in the algorithm, and $\sigma_{\min}(A)$ denotes the smallest nonzero singular value of A.

Proof. Let x_k denote the projection of $\tilde{x}_{k-1}$ onto the hyperplane $H_i = \{x : \langle a_i, x\rangle = b_i\}$. It then follows from Lemma 2.6 that

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq (1 - \kappa(A)^{-2})\|\tilde{x}_{k-1} - x^*\|_2^2. \tag{2.5}$$

According to the definition of $\tilde{x}_k$, we can see that $\tilde{x}_k$ is the projection of $\tilde{x}_{k-1}$ to the hyperplane $\tilde{H}_i = \{a : \langle a_i, x\rangle = b_i + r_i\}$. Then $x_k - \tilde{x}_{k-1}$ is perpendicular to H_i and $\tilde{x}_k - \tilde{x}_{k-1}$ is perpendicular to $\tilde{H}_i$. Both H_i and $\tilde{H}_i$ are parallel with the same normal vector a_i. Thus both $x_k - \tilde{x}_{k-1}$ and $\tilde{x}_k - \tilde{x}_{k-1}$ are parallel to a_i. Consequently $\tilde{x}_k - x_k$ is parallel to a_i. We can write $\tilde{x}_k - x_k = ta_i$ for some scalar t. Then

$$t\|a_i\|_2^2 = \langle a_i, ta_i\rangle = \langle a_i, \tilde{x}_k - x_k\rangle = \langle a_i, \tilde{x}_k\rangle - \langle a_i, x_k\rangle = b_i + \eta_i - b_i = \eta_i.$$

This shows that $t = \eta_i/\|a_i\|_2^2$ and hence

$$\tilde{x}_k - x_k = \frac{\eta_i}{\|a_i\|_2^2} a_i.$$

Note that $x_k - x^*$ is in H_i which has the normal vector a_i. Thus $x_k - x^*$ and $\tilde{x}^k - x_k$ are perpendicular to each other. By the Pythagorean theorem, it follows that

$$\|\tilde{x}_k - x^*\|_2^2 = \|x_k - x^*\|_2^2 + \|\tilde{x}_k - x_k\|_2^2.$$

Taking the conditional expectation on knowing $\tilde{x}_{k-1}$ gives

$$\mathbb{E}[\|\tilde{x}_k - x^*\|_2^2|\tilde{x}_{k-1}] = \mathbb{E}[\|x_k - x^*\|_2^2|\tilde{x}_{k-1}] + \mathbb{E}[\|\tilde{x}_k - x_k\|_2^2|\tilde{x}_{k-1}].$$

Note that $\|\tilde{x}_k - x_k\|_2 = |\eta_i|/\|a_i\|_2$, we have

$$\mathbb{E}[\|\tilde{x}_k - x_k\|_2^2 | \tilde{x}_{k-1}] = \sum_{i=1}^{m} \mathbb{P}(r(k) = i) \frac{|\eta_i|^2}{\|a_i\|_2^2} = \sum_{i=1}^{m} \frac{\|a_i\|_2^2}{\|A\|_F^2} \frac{|\eta_i|^2}{\|a_i\|_2^2} = \frac{\|\eta\|_2^2}{\|A\|_F^2}.$$

Therefore

$$\mathbb{E}[\|\tilde{x}_k - x^*\|_2^2 | \tilde{x}_{k-1}] = \mathbb{E}[\|x_k - x^*\|_2^2 | \tilde{x}_{k-1}] + \frac{\|\eta\|_2^2}{\|A\|_F^2}.$$

In view of (2.5) we then have

$$\mathbb{E}[\|\tilde{x}_k - x^*\|_2^2 | \tilde{x}_{k-1}] \leq (1 - \kappa(A)^{-2}) \|\tilde{x}_{k-1} - x^*\|_2^2 + \frac{\|\eta\|_2^2}{\|A\|_F^2}.$$

By taking the full expectation on the both sides it yields

$$\mathbb{E}[\|\tilde{x}_k - x^*\|_2^2] \leq (1 - \kappa(A)^{-2}) \mathbb{E}[\|\tilde{x}_{k-1} - x^*\|_2^2] + \frac{\|\eta\|_2^2}{\|A\|_F^2}.$$

Now we can apply this inequality recursively to obtain

$$\begin{aligned}
\mathbb{E}[\|\tilde{x}_k - x^*\|_2^2] &\leq (1 - \kappa(A)^{-2})^k \|x_0 - x^*\|_2^2 \\
&\quad + \left(1 + (1 - \kappa(A)^{-2}) + \cdots + (1 - \kappa(A)^{-2})^{k-1}\right) \frac{\|\eta\|_2^2}{\|A\|_F^2} \\
&\leq (1 - \kappa(A)^{-2})^k \|x_0 - x^*\|_2^2 + \frac{1}{1 - (1 - \kappa(A)^{-2})} \frac{\|\eta\|_2^2}{\|A\|_F^2} \\
&= (1 - \kappa(A)^{-2})^k \|x_0 - x^*\|_2^2 + \kappa(A)^2 \frac{\|\eta\|_2^2}{\|A\|_F^2} \\
&= (1 - \kappa(A)^{-2})^k \|x_0 - x^*\|_2^2 + \|A^\dagger\|_2^2 \|\eta\|_2^2.
\end{aligned}$$

Note that $\|A^\dagger\|_2 = 1/\sigma_{\min}(A)$, we therefore complete the proof. □

2.3.3 Pre-asymptotic Convergence Properties

Theorem 2.7 gives error estimates in expectation for any iterate x_k and the convergence rate is determined by $\kappa(A)$. For ill-conditioned problems, the condition number $\kappa(A)$ can be very huge, and thus Theorem 2.7 predicts a very slow convergence. However, in practice, RKM converges rapidly during the initial stage of iterations. The estimate in Theorem 2.8 is also deceptive for noisy data due to the presence of the term $\|\eta\|_2^2/\sigma_{\min}(A)^2$, which implies blowup at the very beginning of the iterations, which is however not the case in practice. Hence, Theorem 2.7 and Theorem 2.8 do not fully explain the excellent empirical convergence of RKM for ill-conditioned problems.

In this section, we will provide an explanation on the fast empirical convergence behaviour of the randomised Kaczmarz method by analysing its pre-asymptotic convergence behaviour and showing that the low-frequency error (with respect to the right singular vectors) decays faster during the first iterations than the high-frequency error.

To start, let the singular value decomposition (SVD) of $A \in \mathbb{R}^{m\times n}$ given by

$$A = U\Sigma V^T, \qquad U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}.$$

$$U = \begin{pmatrix} u_1^T \\ \vdots \\ u_n^T \end{pmatrix} \qquad V^T = \begin{pmatrix} v_1^T \\ \vdots \\ v_m^T \end{pmatrix}$$

where U and V are orthogonal matrices whose columns are called the left and right singular vectors of A, and $\Sigma \in \mathbb{R}^{m\times b}$ is diagonal with the nonzero elements being the singular values of A, ordered non-increasingly by $\sigma_1 \geq \cdots \geq \sigma_r > 0$ with $r \leq \min\{m, n\}$ [38].

Given a frequency cut-off number L with $1 \leq L \leq n$, we can define two subspaces of $\mathbb{R}^n$ by

$$\mathcal{L} = \text{span}\{v_1, \cdots, v_L\}, \qquad \mathcal{H} = \text{span}\{v_{L+1}, \cdots, v_m\}$$

denoting by low frequency and high frequency solution spaces. $\mathcal{L}$ and $\mathcal{H}$ are orthogonal.

For any vector $z \in \mathbb{R}^m$, there exists a unique decomposition

$$z = P_L z + P_H z,$$

where P_L and P_H are the orthogonal projection operators onto $\mathcal{L}$ and $\mathcal{H}$ respectively, defined by:

$$P_L z = \sum_{i=1}^{L} \langle v_i, z\rangle v_i, \qquad P_H z = \sum_{i=L+1}^{m} \langle v_i, z\rangle v_i.$$

We have the following lemmas based on SVD properties.

Lemma 2.9. *For any $e_L \in \mathcal{L}$ and $e_H \in \mathcal{H}$, there hold*

$$\sigma_L||e_L|| \leq ||Ae_L|| \leq \sigma_1||e_L||, \qquad ||Ae_H|| \leq \sigma_{L+1}||e_H||, \qquad \langle Ae_L, Ae_H\rangle = 0.$$

Proof. Since $e_L \in \mathcal{L}$, we have

$$e_L = \sum_{i=1}^{L} \langle e_L, v_i\rangle v_i, \quad Ae_L = \sum_{i=1}^{L} \sigma_i \langle e_L, v_i\rangle u_i.$$

Therefore

$$\|e_L\|_2^2 = \sum_{i=1}^{L} |\langle e_l, v_i\rangle|^2, \quad \|Ae_L\|_2^2 = \sum_{i=1}^{L} \sigma_i^2 |\langle e_L, v_i\rangle|^2$$

from which we immediately have $\sigma_L\|e_L\|_2 \leq \|Ae_l\|_2 \leq \sigma_1\|e_L\|_2$. By similar argument we can show that $\|Ae_H\|_2 \leq \sigma_{L+1}\|e_H\|_2$ for $e_H \in \mathcal{H}$. Note that $\mathcal{L}$ and $\mathcal{H}$ are invariant under A^A, i.e.

$$A^*A(\mathcal{L}) \subset \mathcal{L}, \quad A^*A(\mathcal{H}) \subset \mathcal{H}.$$

Thus

$$\langle Ae_L, Ae_H\rangle = \langle A^*Ae_L, e_H\rangle = 0$$

because $A^*Ae_L \in \mathcal{L}$, $e_H \in \mathcal{H}$, and $\mathcal{L}$ and $\mathcal{H}$ are orthogonal. □

Lemma 2.10. *For $i = 1, \cdots, n$, there hold*

$$||P_H a_i||^2 \leq \sigma_{L+1}^2, \qquad \sum_{i=1}^{n} ||P_H a_i||^2 \leq \sum_{j=L+1}^{r} \sigma_j^2.$$

Proof. Since

$$A = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} u_1^T \\ \vdots \\ u_n^T \end{pmatrix} \Sigma V^T$$

we have $a_i = u_i^T \Sigma V^T$. Note also that

$$V^T v_j = \begin{pmatrix} v_1^T v_j \\ \vdots \\ v_m^T v_j \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = e_j.$$

Therefore

$$\begin{aligned} P_H a_i &= \sum_{j=L+1}^{m} \langle v_j, a_i\rangle v_j = \sum_{j=L+1}^{m} (a_i^T v_j) v_j \\ &= \sum_{j=L+1}^{m} (u_i^T \Sigma V^T v_j) v_j = \sum_{j=L+1}^{m} (u_i^T \Sigma e_j) v_j \\ &= \sum_{j=L+1}^{m} (u_i^T \sigma_j e_j) v_j = \sum_{j=L+1}^{m} \sigma_j \langle u_i^T, e_j\rangle v_j. \end{aligned}$$

Consequently

$$||P_H a_i||_2^2 = \sum_{j=L+1}^{m} \sigma_j^2 \langle u_i, e_j \rangle^2 \leq \sigma_{L+1}^2 \sum_{j=L+1}^{m} \langle u_i, e_j \rangle^2 \leq \sigma_{L+1}^2 \|u_i\|^2 = \sigma_{L+1}^2.$$

The second statement can be proved similarly. □

Let the error of kth iteration be $e_k := x_k - x^*$, where x_k is the kth iterate, and x^* is the exact solution. We then have the following theorem on the pre-asymptotic convergence for Algorithm 2 concerning the situation without noise.

Theorem 2.11. *Let* $c_1 = \frac{\sigma_L}{||A||_F^2}$, *and* $c_2 = \frac{1}{||A||_F^2} \sum_{i=L+1}^{r} \sigma_i^2$. *Then there hold*

$$\mathbb{E}[||P_L e_{k+1}||^2 | e_k] \leq (1 - c_1)||P_L e_k||^2 + c_2||P_H e_k||^2,$$
$$\mathbb{E}[||P_H e_{k+1}||^2 | e_k] \leq c_2||P_L e_k||^2 + (1 + c_2)||P_H e_k||^2.$$

Proof. Adjust the algorithm formula for the error e_k and e_{k+1}. Without losing generality, let $i := r(k)$. Then

$$x_{k+1} - x^* = x_k - x^* + \frac{b_i - \langle a_i, x_k \rangle}{||a_i||_2^2} a_i.$$

Therefore

$$\begin{aligned} e_{k+1} &= e_k + \frac{b_i - \langle a_i, x_k \rangle}{||a_i||_2^2} a_i = e_k + \frac{\langle a_i, x^* \rangle - \langle a_i, x_k \rangle}{||a_i||_2^2} a_i \\ &= e_k + \frac{\langle a_i, x^* - x_k \rangle}{||a_i||_2^2} a_i = e_k - \frac{\langle a_i, e_k \rangle}{||a_i||_2^2} a_i \\ &= \left(I - \frac{a_i a_i^T}{||a_i||_2^2} \right) e_k. \end{aligned}$$

Error e_k can be decomposed into the projections onto $\mathcal{L}$ and $\mathcal{H}$, i.e. $e_k = e_L + e_H$, where $e_L := P_L e_k$ and $e_H := P_H e_k$. Then we have

$$P_L e_{k+1} = P_L e_k - \frac{\langle a_i, e_k \rangle}{||a_i||_2^2} P_L a_i = e_L - \frac{\langle a_i, e_k \rangle}{||a_i||_2^2} P_L a_i.$$

Based on property $\langle P_L a_i, e_L \rangle = \langle a_i, e_L \rangle$, we can obtain

$$\begin{aligned} ||P_L e_{k+1}||_2^2 &= ||e_L - \frac{\langle a_i, e_k \rangle}{||a_i||_2^2} P_L a_i||_2^2 \\ &= ||e_L||_2^2 - \frac{2\langle a_i, e_k \rangle}{||a_i||_2^2} \langle P_L a_i, e_L \rangle + ||\frac{\langle a_i, e_k \rangle}{||a_i||_2^2} P_L a_i||_2^2 \\ &= ||e_L||_2^2 - \frac{2\langle a_i, e_k \rangle}{||a_i||_2^2} \langle a_i, e_L \rangle + \frac{\langle a_i, e_k \rangle^2}{||a_i||_2^4} ||P_L a_i||_2^2 \end{aligned}$$

$$
\begin{aligned}
&\leq ||e_L||_2^2 - \frac{2\langle a_i, e_k\rangle}{||a_i||_2^2}\langle a_i, e_L\rangle + \frac{\langle a_i, e_k\rangle^2}{||a_i||_2^2} \\
&= ||e_L||_2^2 - \frac{2\langle a_i, e_k\rangle}{||a_i||_2^2}\langle a_i, e_L\rangle + \frac{\langle a_i, e_L + e_H\rangle^2}{||a_i||_2^2} \\
&= ||e_L||_2^2 - \frac{2\langle a_i, e_L + e_H\rangle}{||a_i||_2^2}\langle a_i, e_L\rangle \\
&\quad + \frac{\langle a_i, e_L\rangle^2 + \langle a_i, e_H\rangle^2 + 2\langle a_i, e_L\rangle\langle a_i, e_H\rangle}{||a_i||_2^2} \\
&= ||e_L||_2^2 + \frac{\langle a_i, e_H\rangle^2 - \langle a_i, e_L\rangle^2}{||a_i||_2^2}.
\end{aligned}
$$

By taking the expectation conditional on knowing e_k and using Lemma 2.9, it follows that

$$
\begin{aligned}
\mathbb{E}[||P_L e_{k+1}||_2^2 | e_k] &\leq ||e_L||_2^2 + \sum_{i=1}^{n} \frac{||a_i||_2^2}{||A||_F^2} \frac{-\langle a_i, e_L\rangle^2 + \langle a_i, e_H\rangle^2}{||a_i||_2^2} \\
&= ||e_L||_2^2 + \frac{1}{||A||_F^2}(-\sum_{i=1}^{n}\langle a_i, e_L\rangle^2 + \sum_{i=1}^{n}\langle a_i, e_H\rangle^2) \\
&= ||e_L||_2^2 + \frac{1}{||A||_F^2}(-||Ae_L||_2^2 + ||Ae_H||_2^2) \\
&\leq (1 - c_1)||P_L e_k||^2 + c_2||P_H e_k||^2
\end{aligned}
$$

Similarly we have

$$
P_H e_{k+1} = e_H - \frac{P_H a_i}{||a_i||_2^2} P_H a_i.
$$

Then

$$
\begin{aligned}
||P_H e_{k+1}||_2^2 &= ||e_H - \frac{\langle a_i, e_k\rangle}{||a_i||_2^2} P_H a_i||_2^2 \\
&= ||e_H||_2^2 - \frac{2\langle a_i, e_k\rangle}{||a_i||_2^2}\langle P_H a_i, e_H\rangle + ||\frac{\langle a_i, e_k\rangle}{||a_i||_2^2} P_H a_i||_2^2 \\
&\leq ||e_H||_2^2 - \frac{2\langle a_i, e_k\rangle}{||a_i||_2^2}\langle a_i, e_H\rangle + \frac{||P_H a_i||_2^2}{||a_i||_2^4}||a_i||_2^2||e_k||_2^2 \\
&= ||e_H||_2^2 - \frac{2\langle a_i, e_k\rangle}{||a_i||_2^2}\langle a_i, e_H\rangle + \frac{||P_H a_i||_2^2}{||a_i||_2^2}(||e_L||_2^2 + ||e_H||_2^2)
\end{aligned}
$$

By taking the conditional expectation on both sides and using Lemma 2.9 and Lemma 2.10, we can obtain

$$
\begin{aligned}
&\mathbb{E}[||P_H e_{k+1}||_2^2 | e_k] \\
&\leq ||e_H||_2^2 - \sum_{i=1}^{n} \frac{||a_i||_2^2}{||A||_F^2} \left(\frac{2\langle a_i, e_k\rangle}{||a_i||_2^2}\langle a_i, e_H\rangle + \frac{||P_H a_i||_2^2}{||a_i||_2^2}(||e_L||_2^2 + ||e_H||_2^2) \right)
\end{aligned}
$$

$$= ||e_H||_2^2 - \frac{2||Ae_H||_2^2}{||A||_F^2} + \sum_{i=1}^{n} \frac{||P_H a_i||_2^2}{||A||_F^2}(||e_L||_2^2 + ||e_H||_2^2)$$
$$\leq c_2||e_L||_2^2 + (1+c_2)||e_H||_2^2.$$

□

By Theorem 2.11, the decay of the error $\mathbb{E}[[\|P_L e_{k+1}\|^2|e_k]$ is largely determined by the factor $1-c_1$ and only mildly affected by $\|P_H e_k\|^2$ by a factor c_2. By taking expectation of both sides of the estimates in Theorem 2.11, we can obtain

$$\mathbb{E}[\|P_L e_{k+1}\|^2] \leq (1-c_1)\mathbb{E}[\|P_L e_k\|^2] + c_2\mathbb{E}[\|P_H e_k\|^2],$$
$$\mathbb{E}[\|P_H e_{k+1}\|^2] \leq c_2\mathbb{E}[\|P_L e_k\|^2] + (1+c_2)\mathbb{E}[\|P_H e_k\|^2]$$

By using this estimate recursively, it follows that

$$\begin{pmatrix} \mathbb{E}[\|P_L e_k\|^2] \\ \mathbb{E}[\|P_H e_k\|^2] \end{pmatrix} \leq D^k \begin{pmatrix} \|P_L e_0\|^2 \\ \|P_H e_0\|^2 \end{pmatrix}, \qquad D = \begin{pmatrix} 1-c_1 & c_2 \\ c_2 & 1+c_2 \end{pmatrix}.$$

By studying the spectral property of the matrix D, one can obtain for $\alpha := c_2/c_1 \ll 1$ the following approximate error propagation for $k = O(1)$:

$$\mathbb{E}[\|P_L e_k\|^2] \approx (1-c_1)^k\|P_L e_0\|^2 + \alpha(1-(1-c_1)^k)\|P_H e_0\|^2,$$
$$\mathbb{E}[\|P_H e_k\|^2] \approx \alpha(1-(1-c_1)^k)\|P_L e_0\|^2 + (1+k\alpha c_1)\|P_H e_0\|^2$$

which explains the fast decay property of the randomised Kaczmarz method during the first iterations, see [18] for details.

2.4 Accelerated Randomised Kaczmarz method

By applying the Nesterov's acceleration strategy [29] to the RKM, in this section we consider an accelerated version of RKM for solving the consistent linear system

$$Ax = b$$

with the assumptions on A as before. Moreover, we assume that the rows of A are normalised in the sense that $\|a_j\|_2 = 1$ for $j = 1, \cdots, m$.

2.4.1 Construction

For the minimisation problem

$$\min_{x\in\mathbb{R}^n} f(x)$$

with a differentiable objective function $f(x)$, the gradient method takes the form

$$x_{k+1} = x_k - \theta_k \nabla f(x_k), \quad k = 0, 1, \cdots,$$

where θ_k is the step size. Nesterov proposed several schemes to accelerate the gradient method [29], one of the acceleration scheme can be formulated as follows

$$\begin{aligned} y_k &= \alpha_k v_k + (1 - \alpha_k) x_k, \\ x_{k+1} &= y_k - \theta_k \nabla f(y_k), \\ v_{k+1} &= \beta_k v_k + (1 - \beta_k) y_k - \gamma_k \nabla f(y_k) \end{aligned} \tag{2.6}$$

with suitable choices of α_k, β_k, γ_k and θ_k. We can use this strategy to propose an accelerated randomised Kaczmarz method (ARKM) by modifying the projection step of RKM in Algorithm 2, that is, instead of defining x_{k+1} by

$$x_{k+1} = x_k + \frac{b_i - \langle a_i, x_k \rangle}{\|a_i\|_2^2} a_i$$

directly, we will use an analog of the three step strategy in (2.6) to define x_{k+1}. This leads to the following accelerated randomised Kaczmarz method.

Algorithm 4 (ARKM)**.** *Given an initial guess x_0, let $v_0 = x_0$, $\gamma_{-1} = 0$, and $\lambda \in [0, \lambda_{\min}]$, where $\lambda_{\min}$ is the smallest non-zero singular value of A, that is, the minimum eigenvalue of $A^T A$. For $k = 0, 1, \cdots$ we choose γ_k to be the larger root of*

$$\gamma_k^2 - \frac{\gamma_k}{m} = \left(1 - \frac{\gamma_k \lambda}{m}\right) \gamma_{k-1}^2$$

and define

$$\begin{aligned} \alpha_k &= \frac{m - \gamma_k \lambda}{\gamma_k (m^2 - \lambda)} \\ \beta_k &= 1 - \frac{\gamma_k \lambda}{m}; \\ y_k &= \alpha_k v_k + (1 - \alpha_k) x_k; \\ x_{k+1} &= y_k + \frac{b_i - a_i^T y_k}{\|a_i\|^2} a_i; \\ v_{k+1} &= \beta_k v_k + (1 - \beta_k) y_k + \gamma_k \frac{b_i - a_i^T y_k}{\|a_i\|_2^2} a_i, \end{aligned}$$

where i is drawn from the step $\{1, 2, \cdots, m\}$ with uniform probability distribution.

From this we can develop a more efficient algorithm by calculating the parameters. Let

$$g_k := \frac{a_i^T y_k - b_i}{||a_i||_2^2} a_i.$$

Then we can represent parameters of Algorithm 4 in a new way for developing a equivalent algorithm with less cost. Recall the last two iterate steps of Algorithm 4, we have

$$v_{k+1} = \beta_k v_k + (1-\beta_k) y_k - \gamma_k g_k, \quad x_{k+1} = y_k - g_k.$$

Recall also that $y_k = \alpha_k v_k + (1-\alpha_k) x_k$, we have $v_k = \frac{1}{\alpha_k}(y_k - (1-\alpha_k)x_k)$. Therefore

$$\begin{aligned} v_{k+1} &= \frac{\beta_k}{\alpha_k} y_k - (1-\alpha_k)x_k) + (1-\beta_k) y_k - \gamma_k g_k \\ &= \left(\frac{\beta_k}{\alpha_k} + 1 - \beta_k \right) y_k - \frac{\beta_k(1-\alpha_k)}{\alpha_k} x_k - \gamma_k g_k. \end{aligned}$$

Similarly, substitute this v_{k+1} and $x_{k+1} = y_k - g_k$ into the formula

$$y_{k+1} = \alpha_{k+1} v_{k+1} + (1-\alpha_{k+1}) x_{k+1}$$

for y_{k+1}, we can obtain

$$\begin{aligned} y_{k+1} = \alpha_{k+1} &\left(\left(\frac{\beta_k}{\alpha_k} + 1 - \beta_k \right) y_k - \frac{\beta_k(1-\alpha_k)}{\alpha_k} x_k - \gamma_k g_k \right) \\ &+ (1-\alpha_{k+1})(y_k - g_k) \\ = &\left(1 + \frac{1-\alpha_k}{\alpha_k} \alpha_{k+1} \beta_k \right) y_k - \frac{1-\alpha_k}{\alpha_k} \alpha_{k+1} \beta_k x_k \\ &- (1 - \alpha_{k+1} + \alpha_{k+1}\gamma_k) g_k. \end{aligned}$$

According to the formulae of α_k and β_k in Algorithm 4, we have

$$\begin{aligned} \frac{1-\alpha_k}{\alpha_k} \beta_k &= \left(\frac{1}{\alpha_k} - 1 \right) \beta_k \\ &= \left(\frac{\gamma_k(m^2-\lambda)}{m - \gamma_k \lambda} - 1 \right) \left(1 - \frac{\gamma_k \lambda}{m} \right) \\ &= \left(\frac{\gamma_k(m^2-\lambda)}{m - \gamma_k \lambda} - 1 \right) \frac{m - \gamma_k \lambda}{m} \\ &= \frac{\gamma_k(m^2-\lambda)}{m} - \frac{m - \gamma_k \lambda}{m} \\ &= \frac{\gamma_k m^2 - m}{m} = \gamma_k m - 1. \end{aligned}$$

Thus

$$y_{k+1} = \alpha_{k+1}(1 - m\gamma_k)x_k + (1 - \alpha_{k+1} + m\alpha_{k+1}\gamma_k)y_k - (1 - \alpha_{k+1} + \alpha_{k+1}\gamma_k)g_k. \tag{2.7}$$

According to (2.7) we can see that there is no need to calculate β_k and v_k. This leads to the following equivalent but more efficient algorithm.

Algorithm 5 (EARKM). *Given an initial guess x_0, let $v_0 = x_0$, $\gamma_{-1} = 0$, and $\lambda \in [0, \lambda_{\min}]$, where $\lambda_{\min}$ is the smallest non-zero singular value of A, that is, the minimum eigenvalue of $A^T A$. For $k = 0, 1, \cdots$, generate γ_k and α_k as in Algorithm 4, and define*

$$\begin{aligned} s_k &= \frac{a_i^T y_k - b_i}{||a_i||_2^2}; \\ g_k &= s_k a_i; \\ y_{k+1} &= \alpha_{k+1}(1 - m\gamma_k)x_k + (1 - \alpha_{k+1} + m\alpha_{k+1}\gamma_k)y_k \\ &\quad - (1 - \alpha_{k+1} + \alpha_{k+1}\gamma_k)g_k; \\ x_{k+1} &= y_k - g_k, \end{aligned}$$

where i is drawn from $\{1, 2, \cdots, m\}$ with uniform probability distribution.

2.4.2 Convergence Properties

Now we will give the convergence analysis on the accelerated randomised Kaczmarz method. For a positive semi-definite metric B, we set

$$||x||_B := \sqrt{\text{trace}(X^T B X)}.$$

Theorem 2.12. *Consider Algorithm 4. Let*

$$\sigma_1 = 1 + \frac{\sqrt{\lambda}}{2m} \quad \text{and} \quad \sigma_2 = 1 - \frac{\sqrt{\lambda}}{2m}.$$

Then we have

$$\mathbb{E}[\|v_k - x^*\|^2_{(A^T A)^+}] \leq \frac{4\|x_0 - x^*\|^2_{(A^T A)^+}}{(\sigma_1^k + \sigma_2^k)^2}, \quad \mathbb{E}[\|x_k - x^*\|^2] \leq \frac{4m^2\|x_0 - x^*\|^2_{(A^T A)^+}}{(\sigma_1^k + \sigma_2^k)^2}$$

for $k = 1, 2, \cdots$.

The proof of Theorem 2.12 is based on a series lemmas.

Lemma 2.13. *For a row normalised matrix $A \in \mathbb{R}^{m \times n}$ there holds $\lambda_{\min} \leq m$.*

Proof. By the definition of $\lambda_{\min}$, there is an $x \neq 0$ such that $A^TAx = \lambda_{\min}x$. Note that

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \implies A^TA = a_1a_1^T + \cdots + a_ma_m^T.$$

We have

$$\lambda_{\min}x = (a_1a_1^T + \cdots + a_ma_m^T)x.$$

Therefore

$$\begin{aligned} \lambda_{\min}\|x\|_2 &= \langle (a_1a_1^T + \cdots + a_ma_m^T)x, x\rangle = (a_1^Tx)^2 + \cdots + (a_m^Tx)^2 \\ &\leq \|a_1\|_2^2\|x\|_2^2 + \cdots + \|a_m\|_2^2\|x\|_2^2 \\ &= m\|x\|_2^2. \end{aligned}$$

Since $\|x\|_2 \neq 0$, we must have $\lambda_{\min} \leq m$. □

Lemma 2.14. *For the sequence $\{\gamma_k\}$, $\{\alpha_k\}$ and $\{\beta_k\}$ defined in Algorithm 4, we have*

$$\gamma_{k-1} \leq \gamma_k \leq \frac{1}{\sqrt{\lambda}}, \quad \gamma_k \geq \frac{1}{m} \quad \text{and} \quad 0 \leq \alpha_k, \beta_k \leq 1$$

for all $k \geq 0$.

Proof. We first show the assertion of $\{\gamma_k\}$ by an induction argument. Since $\gamma_{-1} = 0$, γ_0 is the larger root of the equation $\gamma^2 - \gamma/m = 0$ and hence $\gamma_0 = 1/m$. By Lemma 2.13 we have $\lambda \leq \lambda_{\min} \leq m$. Thus the assertion for γ_0 follows. Now we assume the assertion for γ_k for some $k \geq 0$, and show that the assertion is also true for γ_{k+1}. Consider the function

$$t(\gamma) := \gamma^2 - \frac{1 - \lambda\gamma_k^2}{m}\gamma - \gamma_k^2.$$

Clearly $t(\gamma) \to +\infty$ as $|\gamma| \to \infty$. Note that

$$t(\frac{1}{m}) = \frac{1}{m^2} - \frac{1 - \lambda\gamma_k^2}{m}\frac{1}{m} - \gamma_k^2 = \left(\frac{\lambda}{m} - 1\right)\gamma_k^2 \leq 0$$

since $\lambda/m^2 - 1 \leq m/m^2 - 1 = 1/m - 1 \leq 0$. This shows that $t(\gamma)$ must have a root on $[1/m, \infty)$. Thus $\gamma_{k+1} \geq 1/m$ by the definition of γ_{k+1}. Note also that

$$t(\gamma_k) = \gamma_k^2 - \frac{1 - \lambda\gamma_k^2}{m}\gamma_k - \gamma_k^2 = -\frac{\gamma_k(1 - \lambda\gamma_k^2)}{m} \leq 0$$

by the induction hypothesis on γ_k. We also have $\gamma_{k+1} \geq \gamma_k$ by the definition of γ_{k+1}. Moreover,

$$f(\frac{1}{\sqrt{\lambda}}) = \frac{1}{\lambda} - \frac{1}{m\sqrt{\lambda}}(1 - \lambda\gamma_k^2) - \gamma_k^2 = \frac{1}{\lambda} - \frac{1}{m\sqrt{\lambda}} + \gamma_k^2\left(\frac{\sqrt{\lambda}}{m} - 1\right)$$
$$\geq \frac{1}{\lambda} - \frac{1}{m\sqrt{\lambda}} + \frac{1}{\lambda}\left(\frac{\sqrt{\lambda}}{m} - 1\right) = 0$$

since $\gamma_k^2 \leq 1/\sqrt{\lambda}$ and $\sqrt{\lambda}/m - 1 \leq 0$. Note that $t(\gamma)$ is a strong convex quadratic function, we must have $t(\gamma) > 0$ for all $\gamma > 1/\sqrt{\lambda}$. Therefore $\gamma_{k+1} \leq 1/\sqrt{\lambda}$.

Using the assertion for $\{\gamma_k\}$ and the definition of α_k and β_k, we can see immediately that $0 \leq \alpha_k, \beta_k \leq 1$ for all $k \geq 0$. □

Lemma 2.15. *Let $U \in \mathbb{R}^{m\times r}$ with $U^TU = I$. Let $U^T = [u_1, u_2, \cdots, u_m]$. Then*

$$\|u_i\|_2 \leq 1 \quad \text{for } i = 1, \cdots, m.$$

Proof. Let $U = [\tilde{u}_1, \tilde{u}_2, \cdots, \tilde{u}_r]$. Then $U^TU = I$ implies that $\tilde{u}_i^T\tilde{u}_j = 1$ if $i = j$ and $= 0$ if $i \neq j$. Let W be the space spanned by $\{\tilde{u}_1, \cdots, \tilde{u}_r\}$. Then $\{\tilde{u}_1, \cdots, \tilde{u}_r\}$ is an orthonormal basis of W. Thus for any $y \in \mathbb{R}^m$, the projection of x onto W is

$$P_W(y) = \sum_{i=1}^{r}(\tilde{u}_i^Ty)\tilde{u}_i = \sum_{i=1}^{r}\tilde{u}_i\tilde{u}_i^Ty = UU^Ty.$$

Thus

$$\|U^Ty\|_2^2 = \langle UU^Ty, y\rangle = \langle P_W(y), y\rangle = \langle P_W(y), P_W(y)\rangle = \|P_W(y)\|_2^2 \leq \|y\|_2^2.$$

By taking $y = e_j$ which is the vector whose entries are zeros except the jth spot where the entry is 1, we then obtain

$$\|u_j\|_2^2 = \|U^Te_j\|_2^2 \leq \|e_j\|_2^2 = 1.$$

□

Lemma 2.16. *For every $w \in \mathbb{R}^n$,*

$$\mathbb{E}[||a_i(a_i^Tw - b_i)||_{(A^TA)^+}^2] \leq \frac{1}{m}||Aw - b||^2,$$

where expectation are taken with respect to i which is uniformly distributed from index set $\{1, \cdots, m\}$.

Proof. By using the singular value decomposition, we can write

$$A = U\Sigma V^T,$$

where $U \in \mathbb{R}^{m\times r}$ satisfies $U^TU = I$, $V \in \mathbb{R}^{n\times r}$ satisfies $V^TV = I$ and $\Sigma \in \mathbb{R}^{r\times r}$ is a positive diagonal matrix with r being the rank of A. Thus $A^T = (U\Sigma V^T)^T = V\Sigma U^T$ and hence

$$A^TA = V\Sigma U^TU\Sigma V^T = V\Sigma^2V^T.$$

Therefore the Moore-Penrose pseudoinverse of (A^TA) is $(A^TA)^+ = V\Sigma^{-2}V^T$. By taking the expectation with respect to i, and using the commutativity and linearity of trace, i.e.

$$\text{trace}(BC) = \text{trace}(CB), \qquad \text{trace}(B + C) = \text{trace}(B) + trace(C)$$

we have

$$\begin{aligned}
&\mathbb{E}[||a_i(a_i^Tw - b_i)||^2_{(A^TA)^+}] \\
&= \frac{1}{m}\sum_{i=1}^{m} \text{trace}\left((a_i(a_i^Tw - b_i))^T(A^TA)^+(a_i(a_i^Tw - b_i))\right) \\
&= \frac{1}{m}\sum_{i=1}^{m} \text{trace}\left((A^TA)^+(a_i(a_i^Tw - b_i))(a_i(a_i^Tw - b_i))^T\right) \\
&= \frac{1}{m}\,\text{trace}\left((A^TA)^+\sum_{i=1}^{m} a_i(a_i^Tw - b_i)^2a_i^T\right) \\
&= \frac{1}{m}\,\text{trace}\left((A^TA)^+A^T\,\text{diag}(Aw - b)A\right) \\
&= \frac{1}{m}\,\text{trace}(V\Sigma^{-2}V^TV\Sigma U^T\,\text{diag}(Aw - b)^2U\Sigma V^T) \\
&= \frac{1}{m}\,\text{trace}(V\Sigma^{-1}U^T\,\text{diag}(Aw - b)^2U\Sigma V^T) \\
&= \frac{1}{m}\,\text{trace}(\Sigma V^TV\Sigma^{-1}U^T\,\text{diag}(Aw - b)^2U) \\
&= \frac{1}{m}\,\text{trace}(U^T\,\text{diag}(Aw - b)^2U) \\
&= \frac{1}{m}\sum_{i=1}^{m}(a_i^Tw - b)^2||u_i||^2 \\
&\leq \frac{1}{m}||Aw - b||^2,
\end{aligned}$$

where, for the last inequality, we used Lemma 2.15. □

Lemma 2.17. *Let x^* be any solution of $Ax = b$, where A is assumed to be normalised. For any $w \in \mathbb{R}^n$ let $p_i(w)$ be defined as the orthogonal projection of w onto $a_i^T w = b_i$, i.e.*

$$p_i(w) := w - \frac{a_i}{||a_i||^2}(a_i^T w - b_i) = w - (a_i^T w - b_i),$$

Then

$$\mathbb{E}[||p_i(w) - x^*||^2] = ||w - x^*||^2 - \frac{1}{m}||Aw - b||^2,$$

where the expectation is taken with respect to i which is uniformly distributed from the index set $\{1, \cdots, m\}$.

Proof. Because $Ax^* = b$ and $\|a_i\| = 1$ for $i = 1, \cdots, m$, we have

$$\begin{aligned}
\mathbb{E}[||p_i(w) - x^*||^2] &= \mathbb{E}[||w - a_i(a_i^T w - b_i) - x^*||^2] \\
&= ||w - x^*||^2 + \mathbb{E}[(a_i^T w - b_i)^2] - 2\mathbb{E}[\langle w - x^*, a_i(a_i^T w - b_i)\rangle] \\
&= ||w - x^*||^2 + \mathbb{E}[(a_i^T w - b_i)^2] - 2\mathbb{E}[\langle a_i^T(w - x^*), a_i^T w - b_i\rangle \\
&= ||w - x^*||^2 + \mathbb{E}[(a_i^T w - b_i)^2] - 2\mathbb{E}[(a_i^T w - b_i)^2] \\
&= ||w - x^*||^2 - \mathbb{E}[(a_i^T w - b_i)^2] \\
&= ||w - x^*||^2 - \frac{1}{m}\sum_{i=1}^{m}(a_i^T w - b_i)^2 \\
&= ||w - x^*||^2 - \frac{1}{m}||Aw - b||^2.
\end{aligned}$$

□

Lemma 2.18. *Consider Algorithm 4. Let*

$$r_k = \|v_k - x^*\|_{(A^TA)^+} \quad \text{and} \quad s_k = \|x_k - x^*\|.$$

Then there holds

$$\mathbb{E}[r_{k+1}^2|k] + \gamma_k^2\mathbb{E}[s_{k+1}^2|k] \leq \beta_k r_k^2 + \beta_k\gamma_{k-1}^2 s_k^2, \tag{2.8}$$

where the expectation is conditioned on knowing all the previous k iterations.

Proof. Recall that

$$v_{k+1} = \beta_k v_k + (1 - \beta_k)y_k - \gamma_k(a_i^T y_k - b_i)a_i,$$

here we used the normalisation $\|a_i\|_2 = 1$. Then we have

$$\begin{aligned}
r_{k+1}^2 &= ||v_{k+1} - x^*||_{(A^TA)^+}^2 = ||\beta_k v_k + (1 - \beta_k)y_k - \gamma_k(a_i^T y_k - b_i)a_i - x^*||_{(A^TA)^+}^2 \\
&= I_1 + I_2 + I_3,
\end{aligned} \tag{2.9}$$

where

$$I_1 = ||\beta_k v_k + (1-\beta_k)y_k - x^*||^2_{(A^TA)^+},$$
$$I_2 = ||\gamma_k a_i(a_i^T y_k - b_i)||^2_{(A^TA)^+},$$
$$I_3 = -2\langle \beta_k v_k + (1-\beta_k)y_k - x^*, (A^TA)^+\gamma_k a_i(a_i^T y_k - b_i)\rangle.$$

Now we consider these three terms separately. From Lemma 2.14 we know that $0 \leq \beta_k \leq 1$. Since the semi-norm $||\cdot||^2_{(A^TA)^+}$ is convex, by using convexity we have

$$\begin{aligned} I_1 &= ||\beta_k(v_k - x^*) + (1-\beta_k)(y_k - x^*)||^2_{(A^TA)^+} \\ &\leq \beta_k||v_k - x^*||^2_{(A^TA)^+} + (1-\beta_k)||y_k - x^*||^2_{(A^TA)^+}. \end{aligned}$$

Recall from the definition of β_k that $1-\beta_k = \gamma_k\lambda/m$. Thus

$$I_1 \leq \beta_k r_k^2 + \frac{\gamma_k\lambda}{m}||y_k - x^*||^2_{(A^TA)^+}.$$

Since in Algorithm 4 we take $x_0 = 0$, we can show bu induction that x_k, y_k, v_k and x^* are all in $\mathcal{R}(A^T)$. Thus, by the definition of $\lambda_{\min}$ we have

$$\|y_k - x^*\|^2_{(A^TA)^+} \leq \frac{1}{\lambda_{\min}}\|y - x\|_2^2.$$

This together with $\lambda \leq \lambda_{\min}$ gives

$$I_1 \leq \beta_k r_k^2 + \frac{\gamma_k}{m}\|y_k - x^*\|_2^2. \tag{2.10}$$

Recall that x_{k+1} is the projection of y_k onto the plane $\{x : a_i^T x = b\}$, we may use Lemma 2.16 and Lemma 2.17 to conclude that

$$\begin{aligned} \mathbb{E}[I_2|k] = \mathbb{E}[\gamma_k^2||a_i(a_i^T y_k - b_i)||^2_{(A^TA)^+}|k] &\leq \frac{\gamma_k^2}{m}||Ay_k - b||_2^2 \\ &= \gamma_k^2\left(||y_k - x^*||^2 - \mathbb{E}[||x_{k+1} - x^*||^2|k]\right) \\ &= \gamma_k^2\left(\|y_k - x^*\|^2 - \mathbb{E}[s_{k+1}^2|k]\right). \end{aligned} \tag{2.11}$$

For I_3 we have

$$\begin{aligned} \mathbb{E}[I_3|k] &= -2\gamma_k\langle \beta_k v_k + (1-\beta_k)y_k - x^*, (A^TA)^+\mathbb{E}[a_i(a_i^T y_k - b_i)]\rangle \\ &= -\frac{2\gamma_k}{m}\langle \beta_k v_k + (1-\beta_k)y_k - x^*, (A^TA)^+\sum_{i=1}^m a_i(a_i^T y_k - b_i)\rangle \\ &= -\frac{2\gamma_k}{m}\langle \beta_k v_k + (1-\beta_k)y_k - x^*, (A^TA)^+A^T(Ay_k - b)\rangle \\ &= -\frac{2\gamma_k}{m}\langle \beta_k v_k + (1-\beta_k)y_k - x^*, (A^TA)^+A^TA(y_k - x^*)\rangle \\ &= -\frac{2\gamma_k}{m}\langle \beta_k v_k + (1-\beta_k)y_k - x^*, y_k - x^*\rangle. \end{aligned}$$

Recall that $y_k = \alpha_k v_k + (1-\alpha_k)x_k$, we have

$$v_k = \frac{1}{\alpha_k} y_k - \frac{1-\alpha_k}{\alpha_k} x_k.$$

Thus

$$\begin{aligned}
\mathbb{E}[I_3|k] &= -\frac{2\gamma_k}{m}\left\langle \beta_k\left(\frac{1}{\alpha_k}y_k - \frac{1-\alpha_k}{\alpha_k}x_k\right) + (1-\beta_k)y_k - x^*, y_k - x^*\right\rangle \\
&= \frac{2\gamma_k}{m}\left\langle x^* - y_k + \frac{1-\alpha_k}{\alpha_k}\beta_k(x_k - y_k), y_k - x^*\right\rangle \\
&= -\frac{2\gamma_k}{m}\|y_k - x^*\|^2 + \frac{2\gamma_k}{m}\frac{1-\alpha_k}{\alpha_k}\beta_k\langle x_k - y_k, y_k - x^*\rangle.
\end{aligned}$$

Recall that

$$\alpha_k = \frac{m - \gamma_k\lambda}{\gamma_k(m^2-\lambda)} \quad \text{and} \quad \gamma_k^2 - \frac{\gamma_k}{m} = \left(1 - \frac{\gamma_k\lambda}{m}\right)\gamma_{k-1}^2.$$

We have

$$\frac{1-\alpha_k}{\alpha_k} = \frac{m^2\gamma_k - m}{m - \gamma_k\lambda} = \frac{m\gamma_{k-1}^2}{\gamma_k}.$$

Therefore

$$\begin{aligned}
\mathbb{E}[I_3|k] &= -\frac{2\gamma_k}{m}\|y_k - x^*\|^2 + 2\beta_k\gamma_{k-1}^2\langle x_k - y_k, y_k - x^*\rangle \\
&= -\frac{2\gamma_k}{m}||y_k - x^*||^2 + \beta_k\gamma_{k-1}^2\left(\|x_k - x^*\|^2 - \|y_k - x^*\|^2 - \|x_k - y_k\|^2\right) \\
&= -\left(\frac{2\gamma_k}{m} + \beta_k\gamma_{k-1}^2\right)\|y_k - x^*||^2 + \beta_k\gamma_{k-1}^2\left(\|x_k - x^*||^2 - \|x_k - y_k\|^2\right) \\
&\le -\left(\frac{2\gamma_k}{m} + \beta_k\gamma_{k-1}^2\right)\|y_k - x^*||^2 + \beta_k\gamma_{k-1}^2\|x_k - x^*\|^2. \qquad (2.12)
\end{aligned}$$

Combining (2.10), (2.11) and (2.12) with (2.9) we obtain

$$\mathbb{E}[r_{k+1}^2|k] \le \beta_k r_k^2 - \gamma_k^2\mathbb{E}[s_{k+1}^2|k] + \beta_k\gamma_{k-1}^2 s_k^2 + \left(\gamma_k^2 - \frac{\gamma_k}{m} - \beta_k\gamma_{k-1}^2\right)\|y_k - x^*\|^2.$$

According to the definition of γ_k and β_k we have $\gamma_k^2 - \frac{\gamma_k}{m} - \beta_k\gamma_{k-1}^2 = 0$. Therefore

$$\mathbb{E}[r_{k+1}^2|k] \le \beta_k r_k^2 - \gamma_k^2\mathbb{E}[s_{k+1}^2|k] + \beta_k\gamma_{k-1}^2 s_k^2.$$

and the proof is complete. □

Lemma 2.19. *Let $\{a_k\}$ and $\{b_k\}$ be two sequence of non-negative numbers with $a_0 = 0$, $b_0 \neq 0$ and*

$$a_{k+1}^2 = \gamma_k^2 b_{k+1}^2, \qquad b_{k+1}^2 = \frac{b_k^2}{\beta_k}, \quad k = 0, 1, \cdots,$$

where $\{\beta_k\}$ *and* $\{\gamma_k\}$ *are defined by Algorithm 4. Then* $a_k \geq a_{k-1}$, $b_k \geq b_{k-1}$ *and*

$$a_k \geq \frac{b_0}{2}(\sigma_i^k - \sigma_2^k), \qquad b_k \geq \frac{b_0}{2}(\sigma_1^k + \sigma_2^k)$$

for all $k \geq 1$, *where*

$$\sigma_1 = 1 + \frac{\sqrt{\lambda}}{2m}, \qquad \sigma_2 = 1 - \frac{\sqrt{\lambda}}{2m}.$$

Proof. Since $0 < \beta_k \leq 1$ and $\gamma_k^2 - \gamma_k/m = \beta_k \gamma_{k-1}^2$, we have

$$b_{k+1}^2 = \frac{b_k^2}{\beta_k} \geq b_k^2$$

and

$$a_{k+1}^2 = \gamma_k^2 b_{k+1}^2 = \frac{\gamma_k^2 b_k^2}{\beta_k} = \frac{\gamma_k^2 a_k^2}{\beta_k \gamma_k^2} = \frac{\gamma_k^2 a_k^2}{\gamma_k^2 - \gamma_k/m} \geq a_k^2.$$

Therefore $a_{k+1} \geq a_k$ and $b_{k+1} \geq b_k$ for $k \geq 0$.

Next we derive the growth rate on a_k and b_k. We have

$$b_k^2 = \beta_k b_{k+1}^2 = \left(1 - \frac{\lambda}{m}\gamma_k\right) b_{k+1}^2 = \left(1 - \frac{\lambda a_{k+1}}{m b_{k+1}}\right) b_{k+1}^2 = b_{k+1}^2 - \frac{\lambda}{m} a_{k+1} b_{k+1}.$$

Thus

$$\frac{\lambda}{m} a_{k+1} b_{k+1} = b_{k+1}^2 - b_k^2 = (b_{k+1} + b_k)(b_{k+1} - b_k) \leq 2b_{k+1}(b_{k+1} - b_k).$$

Consequently

$$b_{k+1} \geq b_k + \frac{\lambda}{m} a_{k+1} \geq b_k + \frac{\lambda}{m} a_k. \tag{2.13}$$

Similarly, we have that

$$\frac{A a_{k+1}^2}{b_{k+1}^2} - \frac{a_{k+1}}{m b_{k+1}} = \gamma_k^2 - \frac{\gamma_k}{m} = \beta_k \gamma_{k-1}^2 = \beta_k \frac{a_k^2}{b_k^2} = \frac{a_k^2}{b_{k+1}^2}.$$

Thus

$$a_{k+1}^2 - \frac{1}{m} a_{k+1} b_{k+1} = a_k^2$$

which implies that

$$\frac{1}{m} a_{k+1} b_{k+1} = a_{k+1}^2 - a_k^2 = (a_{k+1} + a_k)(a_{k+1} - a_k) \leq 2a_{k+1})(a_{k+1} - a_k).$$

Therefore

$$a_{k+1} \geq a_k + \frac{1}{2m} b_{k+1} \geq a_k + \frac{1}{2m} b_k. \tag{2.14}$$

Combining (2.13) and (2.14) we have

$$\begin{pmatrix} a_{k+1} \\ b_{k+1} \end{pmatrix} \geq D \begin{pmatrix} a_k \\ b_k \end{pmatrix}, \tag{2.15}$$

where

$$D = \begin{pmatrix} 1 & \frac{1}{2m} \\ \frac{\lambda}{2m} & 1 \end{pmatrix}.$$

Since all entries of D are non-negative, we may use (2.15) recursively to obtain

$$\begin{pmatrix} a_k \\ b_k \end{pmatrix} \geq D^k \begin{pmatrix} a_0 \\ b_0 \end{pmatrix}, \tag{2.16}$$

From the characteristic equation

$$0 = \det(\sigma I - D) = (\sigma - 1)^2 - \frac{\lambda}{4m^2},$$

of D, we can see that D has two distinct eigenvalues

$$\sigma_1 = 1 + \frac{\sqrt{\lambda}}{2m}, \qquad \sigma_2 = 1 - \frac{\sqrt{\lambda}}{2m}.$$

Direct calculation shows that

$$\begin{pmatrix} 1 \\ \sqrt{\lambda} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 \\ -\sqrt{\lambda} \end{pmatrix}$$

are eigenvectors of D corresponding to σ_1 and σ_2 respectively. Therefore

$$D = \begin{pmatrix} 1 & 1 \\ \sqrt{\lambda} & -\sqrt{\lambda} \end{pmatrix}^{-1} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ \sqrt{\lambda} & -\sqrt{\lambda} \end{pmatrix},$$

Consequently

$$\begin{aligned} D^k &= \begin{pmatrix} 1 & 1 \\ \sqrt{\lambda} & -\sqrt{\lambda} \end{pmatrix}^{-1} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}^k \begin{pmatrix} 1 & 1 \\ \sqrt{\lambda} & -\sqrt{\lambda} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 1 & \frac{1}{\sqrt{\lambda}} \\ 1 & -\frac{1}{\sqrt{\lambda}} \end{pmatrix} \begin{pmatrix} \sigma_1^k & 0 \\ 0 & \sigma_2^k \end{pmatrix} \begin{pmatrix} 1 & 1 \\ \sqrt{\lambda} & -\sqrt{\lambda} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} \sigma_1^k + \sigma_2^k & \sigma_1^k - \sigma_2^k \\ \sigma_1^k - \sigma_2^k & \sigma_1^k + \sigma_2^k \end{pmatrix}, \end{aligned}$$

Combining this with (2.16) yields

$$\begin{aligned} a_k &\geq \frac{a_0}{2}(\sigma_1^k + \sigma_2^k) + \frac{b_0}{2}(\sigma_1^k - \sigma_2^k), \\ b_k &\geq \frac{a_0}{2}(\sigma_1^k - \sigma_2^k) + \frac{b_0}{2}(\sigma_1^k + \sigma_2^k). \end{aligned}$$

The proof is thus complete. □

Now we are ready to complete the proof of Theorem 2.12.

Proof of Theorem 2.12. By taking the full expectation on the equation (2.8) in Lemma 2.18, we have

$$\mathbb{E}[r_{k+1}^2] + \gamma_k^2 \mathbb{E}[s_{k+1}^2] \leq \beta_k \left(\mathbb{E}[r_k^2] + \gamma_{k-1}^2 \mathbb{E}[s_k^2] \right). \tag{2.17}$$

Recursively using this equation we can obtain

$$\mathbb{E}[r_k^2] + \gamma_{k-1}^2 \mathbb{E}[s_k^2] \leq \beta_0 \cdots \beta_{k-1} \left(\mathbb{E}[r_0^2] + \gamma_{-1}^2 \mathbb{E}[s_0^2] \right) = \beta_0 \cdots \beta_{k-1} r_0^2.$$

Let $\{a_k\}$ and $\{b_k\}$ be the sequences defined in Lemma 2.19. Then we have

$$\beta_0 \cdots \beta_{k-1} = \frac{b_0^2}{b_k^2}.$$

Recall that $\gamma_k \geq 1/m$ for $k \geq 0$, we therefore obtain

$$\mathbb{E}[r_k^2] + \frac{1}{m^2} \mathbb{E}[s_k^2] = \frac{b_0^2}{b_k^2} r_0^2.$$

By making use of the estimate of $\{b_k\}$ given in Lemma 2.19 we thus complete the proof. □

2.5 RKM Solving Inequalities

In this section, we consider to generalise the randomised Kaczmarz method for linear equations to solve system of linear inequalities of the form

$$\begin{aligned} a_i^T x &\leq b_i, \quad i \in \mathcal{I}, \\ a_i^T x &= b_1, \quad i \in \mathcal{E}, \end{aligned} \tag{2.18}$$

where $\mathcal{I}$ is the set of indices of inequalities and $\mathcal{E}$ is the set of indices of equalities, we assume that $\mathcal{I}$ and $\mathcal{E}$ are disjoint and form a partition of the set $\{1, 2, \cdots, m\}$. We will modify Algorithm 2 so that it can be used to solve (2.18). We will let A to be the matrix whose rows are a_i^T with $i \in \mathcal{I} \cup \mathcal{E}$. We will also use the notation

$$x^+ := \max\{x, 0\}$$

for a vector x, where max acts on x componentwisely.

Algorithm 6. *Consider the systems of linear inequalities (2.18). Let x_0 be an arbitrary initial point. For $k = 0, 1, \cdots$, compute*

$$\beta_k = \begin{cases} (a_i x_k - b_i)^+, & i \in \mathcal{I}, \\ a_i^T x_k - b_i, & i \in \mathcal{E}, \end{cases}$$

$$x_{k+1} = x_k - \frac{\beta_k}{||a_i||_2^2} a_i,$$

where, at each iteration k, the index i is drawn independently from the set $\{1, 2, \cdots, m\}$ with probability

$$\mathbb{P}(i = j) = \frac{||a_j||_2^2}{||A||_F^2}.$$

In order to analyse Algorithm 6, we need the following theorem due to Hoffman [16] which tells about the distance between any x and a non-empty set of a polyhedron, which may be a solution set of a linear inequality system, is bounded by a linear function.

Theorem 2.20 (Hoffman)**.** *There is a constant L such that, for any $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ such that the solution set*

$$S_b := \{x \in \mathbb{R}^n : a_i^T x \leq b_i \text{ for } i \in \mathcal{I} \text{ and } a_i^T x = b_i \text{ for } i \in \mathcal{E}\}$$

of the system of linear inequalities (2.18) is nonempty, there holds

$$d(x, S_b) \leq L||e(Ax - b)||,$$

where $d(x, S_b)$ denotes the distance from x to S_b, i.e.

$$d(x, S_b) = \inf_{z \in S_b} \|x - z\|_2,$$

and $e : \mathbb{R}^m \to \mathbb{R}^m$ is a function defined by

$$e(y)_i = \begin{cases} \max\{y_i, 0\}, & i \in \mathcal{I}, \\ y_i, & i \in \mathcal{E}. \end{cases}$$

The smallest constant L is called the Hoffman constant.

From Theorem 2.20 we can obtain the convergence properties about Algorithm 6 by observing that $\beta_k = e(Ax_k - b)_i$.

Theorem 2.21. *Assume that the system of linear inequalities (2.18) has a nonempty solution set S. Then Algorithm 6 converges linearly in expectation, that is,*

$$\mathbb{E}[d(x_k, S)^2] \leq \left(1 - \frac{1}{L^2||A||_F^2}\right)^k d(x_0, S)^2,$$

where L is the Hoffman constant.

Proof. Let P_S denote the projection operator onto S. Then

$$d(x, S) = \|x - P_S(x)), \quad \forall x \in |mathbbR^n.$$

By the definition og distance function and the fact $P_S(x_k) \in S$ we have

$$d(x_{k+1}, S)^2 \leq \|x_{k+1} - P_S(x_k)\|^2.$$

Since $x_{k+1} = x_k - \frac{\beta_k}{\|a_i\|^2} a_i$, we have

$$\begin{aligned} d(x_{k+1}, S) &\leq \left\| x_k - P_S(x_k) - \frac{\beta_k}{\|a_i\|^2} a_i \right\|^2 \\ &= \|x_k - P_S(x_k)\|^2 + \frac{\beta_k^2}{\|a_i\|^2} - 2\frac{\beta_k}{\|a_i\|^2} a_i^T (x_k - P_S(x_k)) \\ &= d(s_k, S)^2 + \frac{\beta_k^2}{\|a_i\|^2} - 2\frac{\beta_k}{\|a_i\|^2} a_i^T (x_k - P_S(x_k)). \end{aligned} \tag{2.19}$$

Note that $P_S(x_k) \in S$. For $i \in \mathcal{I}$ we have $a_i^T P_S(x_k) \leq b_i$. Since $\beta_k \geq 0$, we must have

$$\frac{\beta_k}{\|a_i\|^2} a_i^T (x_k - P_S(x_k)) \geq \frac{\beta_k}{\|a_i\|^2} (a_i^T x_k - b_i) = \frac{\beta_k^2}{\|a_i\|^2}.$$

For $i \in \mathcal{E}$ we have $a_i^T P_S(x_k) = b_i$ and thus

$$\frac{\beta_k}{||a_i||^2} a_i^T (x_k - P_S(x_k)) = \frac{\beta_k}{||a_i||_2^2} (a_i^T x_k - b_i) = \frac{\beta_k^2}{||a_i||_2^2}.$$

Combining the above two cases with (2.19), it follows that

$$\begin{aligned} d(x_{k+1}, S)^2 &\leq d(x_k, S)^2 - \frac{\beta_k^2}{\|a_i\|^2} \\ &= d(x, S)^2 - \frac{[e(Ax_k - b)_i]^2}{\|a_i\|^2}. \end{aligned}$$

Therefore, by taking the conditional expectation on knowing $x)k$ we have

$$\begin{aligned} \mathbb{E}[d(x_{k+1}, S)^2 | x_k] &\leq d(x_k, S)^2 - \mathbb{E}\left[\frac{[e(Ax_k - b)_i]^2}{||a_i||^2} \right] \\ &= d(x_k, S)^2 - \sum_{i=1}^{m} \frac{\|a_i\|^2}{\|A\|_F^2} \frac{[e(Ax_k - b)_i]^2}{\|a_i\|^2} \\ &= d(x_k, S)^2 - \frac{1}{\|A\|_F^2} \sum_{i=1}^{m} [e(Ax_k - b)_i]^2 \\ &= d(x_k, S) - \frac{1}{\|A\|_F^2} \|e(Ax_k - b)\|^2. \end{aligned}$$

With the help of Theorem 2.20, it yields

$$\begin{aligned}\mathbb{E}[d(x_{k+1},S)^2|x_k] &\leq d(x_k,S)^2 - \frac{1}{L^2\|A\|_F^2}d(x_k,S)^2 \\ &= \left(1 - \frac{1}{L^2\|A\|_F^2}\right) d(x_k,S)^2.\end{aligned}$$

Taking the full expectation gives

$$\mathbb{E}[d(x_{k+1},S)^2] = \left(1 - \frac{1}{L^2\|A\|_F^2}\right) \mathbb{E}[d(x_k,S)^2].$$

Recursively using this inequality we then obtain

$$\begin{aligned}\mathbb{E}[d(x_k,S)^2] &\leq \left(1 - \frac{1}{L^2\|A\|_F^2}\right)^k \mathbb{E}[d(x_0,S)^2] \\ &= \left(1 - \frac{1}{L^2\|A\|_F^2}\right)^k d(x_0,S)^2\end{aligned}$$

which shows the desired linear convergence rate. □

Chapter 3

Randomised Sparse Kaczmarz Method

In this chapter, we focus on problems of recovering sparse solution of systems of linear equations. Sparsity indicates that most entries are zero. This characteristic runs prevalent in real-world situations. For instance, in computational biology, the alignment of DNA sequences from similar ancestors has its entries largely zeros [9]. Moreover, it is widely-appeared not only on statistical learning [36], but on signal processing, [17] especially image signal processing, as well.

Now, let us consider the linear system with sparse solutions

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

where $A \in \mathbb{R}^{m\times n}$. The interest is to find a solution as sparse as possible. Here a vector is said to be sparse if most of its entries are zeros. Given a vector $x \in \mathbb{R}^n$, its support is defined as

$$\text{supp}(x) := \{i \in \{1, \cdots, n\} : x_j \neq 0\}.$$

The number of elements in $\text{supp}(x)$ is called the sparsity level of x and is denoted by $\|x\|_0$. A natural procedure for finding a sparse solution of (3.1) is of course to solve the constrained minimisation problem

$$\min \{\|x\|_0 : Ax = b\}. \tag{3.2}$$

Unfortunately, due to the non-smoothness and non-convexity of $\|x\|_0$, this minimisation problem becomes extremely difficult to solve. Instead of solving (3.2) directly, one may consider its convex relaxation

$$\min\{\|x\|_1 : Ax = b\}, \tag{3.3}$$

where $\|\cdot\|_1$ denotes the ℓ^1-norm, i.e.

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|.$$

It turns out that the model (3.3) can recover sparse solution very well, see [6], and many algorithms have been developed for solving (3.3. Because the function $\|x\|_1$ is not strong convex, it poses numerical challenges for solving (3.3). In order to develop fast numerical algorithms, one may consider its augmented version

$$\min\left\{f(x) := \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2 : Ax = b\right\}, \tag{3.4}$$

where $\lambda > 0$ is a parameter. According to an exact penalty theory [12, 42], if λ is suitably large, the solution of (3.4) must be a solution of (3.3). However, the objective function

$$f(x) := \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2 \tag{3.5}$$

is strong convex which present a lot of advantage to solve (3.4) by numerical algorithms. Among many algorithms developed for solving (3.4), the sparse Kaczmarz method developed in [19, 25] can be used to deal with the problem where A has huge size. Assuming

$$A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix},$$

where a_i^T denotes the ith row of A, the method in [19, 25] can be formulated as

$$\begin{aligned} \xi_{k+1} &= \xi_k - \frac{a_i^T x_k - b_i}{\|a_i\|_2^2} a_i, \\ x_{k+1} &= \arg\min_{x\in\mathbb{R}^n} \left\{f(x) - \langle \xi_{k+1}, x\rangle\right\}, \end{aligned} \tag{3.6}$$

where $i = \mathrm{mod}(k-1, m) + 1$ and $f(x)$ is given by (3.5). Among other things, it has been shown in [19, 25] that the sequence $\{x_k\}$ converges to the solution of (3.4). However, no convergence rate is available for (3.6). In this chapter we will consider the randomised sparse Kaczmarz method (RSKM) and exact-step randomised sparse Kaczmarz method (ERSKM) developed in [34] for which a linear convergence rate in expectation is available.

3.1 Definitions and Construction

Let us first introduce the soft threshold function. We start with the one-dimensional case.

Definition 3.1. The function $s_\lambda : \mathbb{R} \to \mathbb{R}$ defined by

$$s_\lambda(\tau) := \arg\min_{t\in\mathbb{R}} \left\{ \lambda|t| + \frac{1}{2}(t-\tau)^2 \right\}, \qquad \tau \in \mathbb{R}$$

is called the soft-threshold function with threshold $\lambda > 0$.

The soft threshold function $s_\lambda(\tau)$ has the explicit formula

$$s_\lambda(\tau) := \begin{cases} \tau - \lambda & \text{if } \tau > \lambda, \\ 0 & \text{if } |\tau| \le \lambda, \\ \tau + \lambda & \text{if } \tau < -\lambda \end{cases} \tag{3.7}$$

which can be obtained by solving the one-dimensional minimisation problem. To see this, let $\phi(t) = \lambda|t| + \frac{1}{2}(t-\tau)^2$ which is a convex function. Then $\hat{\tau} := s_\lambda(\tau)$ is the minimiser of $\phi(t)$ over $\mathbb{R}$. We note that

$$\begin{aligned} \hat{\tau} > 0 \text{ is a minimiser of } \phi &\iff \phi'(\hat{\tau}) = 0 \text{ and } \hat{\tau} > 0 \\ &\iff 1 + \frac{1}{\lambda}(\hat{\tau} - \tau) = 0 \text{ and } \hat{\tau} > 0 \\ &\iff \hat{\tau} = \tau - \lambda \text{ and } \tau > \lambda \end{aligned}$$

and

$$\begin{aligned} \hat{\tau} < 0 \text{ is a minimiser of } \phi &\iff \phi'(\hat{\tau}) = 0 \text{ and } \hat{\tau} < 0 \\ &\iff -1 + \frac{1}{\lambda}(\hat{\tau} - \tau) = 0 \text{ and } \hat{\tau} < 0 \\ &\iff \hat{\tau} = \tau + \lambda \text{ and } \tau < -\lambda. \end{aligned}$$

If $|\tau| \le \lambda$, the above two observations show that neither $\hat{\tau} > 0$ nor $\hat{\tau} < 0$ holds. Thus $\hat{\tau} = 0$. Combining the above analysis we obtain (3.7). From (3.7) it is easy to see that

$$s_\lambda(\tau) = \operatorname{sign}(\tau)\max\{|\tau| - \lambda, 0\} = \tau - P_{[-\lambda,\lambda]}(\tau), \quad \tau \in \mathbb{R}, \tag{3.8}$$

where $P_{[-\lambda,\lambda]}$ denotes the projection of $\mathbb{R}$ onto $[-\lambda, \lambda]$.

Similarly, we can also introduce the soft threshold function for vectors as follows.

Definition 3.2. The function $S_\lambda : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$S_\lambda(v) := \arg\min_{x\in\mathbb{R}^n} \left\{ \lambda\|x\|_1 + \frac{1}{2}\|x - v\|_2^2 \right\}, \quad v \in \mathbb{R}^n$$

is called the soft threshold function with threshold $\lambda > 0$.

Note that

$$\lambda\|x\|_1 + \frac{1}{2}\|x - v\|_2^2 = \sum_{i=1}^{n} \left(\lambda|x_i| + \frac{1}{2}(x_i - v_i)^2 \right).$$

Therefore $S_\lambda(v) := (\hat{x}_1, \cdots, \hat{x}_n)$ is a minimiser of $\lambda\|x\|_1 + \frac{1}{2}\|x - v\|_2^2$ over $\mathbb{R}^n$ if and only if $\hat{x}_i$ is a minimiser of $\lambda|x_i| + \frac{1}{2}(x_i - v_i)^2$ over $\mathbb{R}$ for each $i = 1, \cdots, n$. Consequently $\hat{x}_i = s_\lambda(v_i)$ for $i = 1, \cdots, n$. Therefore

$$S_\lambda(v) = (s_\lambda(v_1), \cdots, s_\lambda(v_n)).$$

By using (3.8) we also have

$$S_\lambda(v) = \text{sign}(v) \cdot \max\{|v| - \lambda, 0\} = v - P_{[-\lambda,\lambda]^n}(v), \quad v \in \mathbb{R}^n,$$

where all the operations sign, $\cdot$, max and $|\cdot|$ are understood in componentwise, and $P_{[-\lambda,\lambda]^n}$ denotes the Euclidean projection of $\mathbb{R}^n$ onto the cubic $[-\lambda, \lambda]^n$.

The following result gives further properties of the soft threshold function.

Lemma 3.3. *(i)* $\lambda\|S_\lambda(v)\|_1 = \langle v - S_\lambda(v), S_\lambda(v)\rangle$ *for any* $v \in \mathbb{R}^n$.

(ii) S_λ *is Lipschitz continuous with*

$$\|S_\lambda(v_1) - S_\lambda(v_2)\|_2 \le \|v_1 - v_2\|_2, \quad \forall v_1, v_2 \in \mathbb{R}^n.$$

(iii) The function $v \to \|S_\lambda(v)\|_2^2$ *is continuous differentiable and its gradient is*

$$\nabla(\|S_\lambda(v)\|_2^2) = 2S_\lambda(v), \quad \forall v \in \mathbb{R}^n.$$

Proof. (i) According to the formula of S_λ, we have for any $v \in \mathbb{R}^n$ that

$$\begin{aligned}\langle S_\lambda(v), v - S_\lambda(v)\rangle &= \sum_{i=1}^{n} s_\lambda(v_i)(v_i - s_\lambda(v_i)) \\ &= \left\{ \sum_{i:v_i>\lambda} + \sum_{i:v_i<-\lambda} + \sum_{i:|v_i|\le\lambda} \right\} s_\lambda(v_i)(v_i - s_\lambda(v_i))\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i:v_i>\lambda} (v_i-\lambda)(v_i-(v_i-\lambda)) \\
&\quad + \sum_{i:v_i<-\lambda} (v_i+\lambda)(v_i-(v_i+\lambda)) + \sum_{i:|v_i|\le\lambda} 0 \\
&= \lambda \sum_{i:v_i>\lambda} (v_i-\lambda) + \lambda \sum_{i:v_i<-\lambda} (-v_i-\lambda) + \lambda \sum_{i:|v_i|\le\lambda} 0 \\
&= \lambda \sum_{i:v_i>\lambda} |s_\lambda(v_i)| + \lambda \sum_{i:v_i<-\lambda} |s_\lambda(v_i)| + \lambda \sum_{i:|v_i|\le\lambda} |s_\lambda(v_i)| \\
&= \lambda \sum_{i=1}^{n} |s_\lambda(v_i)| = \lambda \|S_\lambda(v)\|_1.
\end{aligned}
$$

(ii) It suffices to show that

$$|s_\lambda(\tau) - s_\lambda(t)| \le |\tau - t|, \qquad \forall \tau, t \in \mathbb{R} \tag{3.9}$$

for the single variable soft threshold function s_λ. This can be verified case by case. For the cases $(\tau, t > \lambda)$, $(-\lambda \le \tau, t \le \lambda)$ and $(\tau, t < -\lambda)$, the verification is straightforward. For the case $\tau > \lambda$ and $t < -\lambda$, we have

$$0 \le s_\lambda(\tau) - s_\lambda(t) = (\tau - \lambda) - (t + \lambda) = \tau - t - 2\lambda \le \tau - t.$$

For the case $\tau > \lambda$ and $-\lambda \le t \le \lambda$, we have

$$0 \le s_\lambda(\tau) - s_\lambda(t) = \tau - \lambda \le \tau - t.$$

For the case $-\lambda \le \tau \le \lambda$ and $t < -\lambda$, we have

$$0 \le s_\lambda(\tau) - s_\lambda(t) = -t - \lambda \le \tau - t.$$

Therefore (3.9) is verified.

(iii) Note that for the single variable soft threshold function s_λ we have

$$[s_\lambda(\tau)]^2 = \begin{cases} (\tau-\lambda)^2 & \text{if } \tau > \lambda, \\ 0 & \text{if } |\tau| \le \lambda, \\ (\tau+\lambda)^2 & \text{if } \tau < -\lambda \end{cases}$$

Therefore $[s_\lambda(\tau)]^2$ is continuous differentiable with

$$
\begin{aligned}
\frac{d}{d\tau}[s_\lambda(\tau)]^2 &= 2 \begin{cases} \tau-\lambda & \text{if } \tau > \lambda, \\ 0 & \text{if } |\tau| \le \lambda, \\ \tau+\lambda & \text{if } \tau < -\lambda \end{cases} \\
&= 2 s_\lambda(\tau).
\end{aligned}
$$

Since

$$\|S_\lambda(v)\|_2^2 = \sum_{i=1}^{n} [s_\lambda(v_i)]^2,$$

we can see that $\|S_\lambda(v)\|_2^2$ is continuous differentiable with

$$\nabla [S_\lambda(v)]^2 = 2(s_\lambda(v_1), \cdots, s_\lambda(v_n))^T = 2S_\lambda(v).$$

The proof is complete. □

By using the soft threshold function, we can formulate the sparse Kaczmarz method as the following form

$$\begin{aligned} \xi_{k+1} &= \xi_k - \frac{a_i^T x_k - b_i}{\|a_i\|_2^2} a_i, \\ x_{k+1} &= S_\lambda(\xi_{k+1}), \end{aligned} \tag{3.10}$$

where $i = \mathrm{mod}(k-1, m) + 1$. By invoking the randomisation idea in Algorithm 2, it leads to the following randomised sparse Kaczmarz method.

Algorithm 7 (RSKM)**.** *Given the initial guess $x_0 = \xi_0 = 0$. For $k = 1, 2, \cdots$ we define*

$$\begin{aligned} \xi_{k+1} &= \xi_k - \frac{\langle a_i, x_k \rangle - b_i}{\|a_i\|_2^2} a_i, \\ x_{k+1} &= S_\lambda(\xi_{k+1}), \end{aligned}$$

where the index i is drawn randomly from the set $\{1, 2, ..., m\}$ with the probability

$$\mathbb{P}(i = j) = \frac{||a_j||_2^2}{||A||_F^2}.$$

Next we will use the Bregman projection induced by $f(x) = \lambda\|x| + \|x\|_2^2/2$ to derive an exact-step (randomised) sparse Kaczmarz method of the form

$$\begin{aligned} \xi_{k+1} &= \xi_k - t_k a_i, \\ x_{k+1} &= S_\lambda(\xi_{k+1}), \end{aligned}$$

where the step size t_k is obtained by an exact line search procedure, and the choice $t_k = \frac{\langle a_i, x_k \rangle - b_i}{\|a_i\|_2^2}$ corresponds to an inexact line search. We need to introduce the concepts on Bregman distance, Bregman projection and their properties.

We start with the basic elements of convex analysis. A function $f : \mathbb{R}^n \to (-\infty, \infty]$ is called proper if $f(x) < \infty$ for some point $x \in \mathbb{R}^n$. We use $\mathrm{dom}(f)$

to denote the set of all points where f takes finite values and call it the domain of f. A proper function $f : \mathbb{R}^n \to (-\infty, \infty]$ is called lower semi-continuous at a point $x \in \mathbb{R}^n$ if

$$f(x) \leq \liminf_{y \to x} f(x).$$

f is called lower semi-continuous if f is lower semi-continuous at every point. A proper function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex if

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

for all $x_1, x_2 \in \mathbb{R}^n$ and $0 \leq t \leq 1$. A function $f : \mathbb{R}^n \to (-\infty, \infty]$ is called α-strongly convex for some $\alpha > 0$ if

$$f(tx_1 + (1-t)x_2) + \alpha t(1-t)\|x_1 - x_2\|_2^2 \leq tf(x_1) + (1-t)f(x_2)$$

for all $x_1, x_2 \in \mathbb{R}^n$ and $0 \leq t \leq 1$.

Since the functions we will deal with may not differentiable everywhere, so the gradient may have no definition at some points. We need to find a substitute for gradient. For convex functions, subgradient was introduced exactly for this purpose.

Definition 3.4 (Subgradient)**.** Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function and let $x \in \mathbb{R}^n$.

(i) $\xi \in \mathbb{R}^n$ is called a subgradient of f at x if

$$f(y) \geq f(x) + \langle \xi, y - x \rangle$$

for all $y \in \mathbb{R}^n$.

(ii) The set $\partial f(x)$ consists of all subgradients of f at x and is called the subdifferential of f at x;

(iii) If $\partial f(x) \neq \emptyset$, then f is said to be subdifferentiable at x.

The following simple but useful result gives a characterisation of minimisers for proper convex functions.

Theorem 3.5 (Fermat's rule)**.** *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function. Then*

$$x^* \in \arg\min_{x \in \mathbb{R}^n} f(x) \iff \mathbf{0} \in \partial f(x^*).$$

Proof.

$$\begin{aligned} x^* \in \arg\min_{x\in\mathbb{R}^n} f(x) &\iff f(x) \geq f(x^*),\ \forall x \in \mathbb{R}^n \\ &\iff f(x) \geq f(x^*) + \langle \mathbf{0}, x - x^* \rangle,\ \forall x \in \mathbb{R}^n \\ &\iff \mathbf{0} \in \partial f(x^*). \end{aligned}$$

□

The following sum rule on subdifferential is important whose proof can be found in [43]

Theorem 3.6 (Moreau-Rockafellar)**.** *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ and $g : \mathbb{R}^n \to (-\infty, \infty]$ be two convex functions. Then for any $x \in \mathbb{R}^n$ there holds*

$$\partial f(x) + \partial g(x) \subset \partial(f + g)(x).$$

If there is a point $x_0 \in dom(f) \cap dom(g)$ such that f is continuous at x_0, then for any $x \in \mathbb{R}^n$ there holds

$$\partial(f + g)(x) = \partial f(x) + \partial g(x).$$

In order to introduce further results on subdifferential calculus, we need to use the Legendre-Fenchel conjugate of convex functions.

Definition 3.7. Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function. For any $\xi \in \mathbb{R}^n$ define

$$f^*(\xi) := \sup_{x\in\mathbb{R}^n} \{\langle \xi, x \rangle - f(x)\}$$

which is called the Legendre-Fenchel conjugate of f.

The following theorem summarises some relevant results related to the Legendre-Fenchel conjugate.

Theorem 3.8. *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper, lower semi-continuous, convex function. Then f^* is also a proper, lower semi-continuous, convex function from $\mathbb{R}^n$ to $(-\infty, \infty]$. Moreover*

$$\xi \in \partial f(x) \iff f(x) + f^*(\xi) = \langle \xi, x \rangle \iff x \in \partial f^*(\xi).$$

If, in addition, f is α-strongly convex with $\alpha > 0$, then $dom(f^*) = \mathbb{R}^n$, f^* is differentiable and its gradient is Lipschitz continuous with

$$\|\nabla f^*(\xi) - \nabla f^*(\eta)\|_2 \leq \frac{1}{2\alpha}\|\xi - \eta\|_2$$

for all $\xi, \eta \in \mathbb{R}^n$.

Proof. See [30, 31, 43]. □

Example 3.9. Consider the function $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ with $\lambda > 0$. It is straightforward to show that f is α-strongly convex with $\alpha = 1/2$. Its Legendre-Fenchel conjugate, by definition, is given by

$$\begin{aligned} f^*(\xi) &= \sup_{x\in\mathbb{R}^n}\{\langle\xi, x\rangle - f(x)\} = -\inf_{x\in\mathbb{R}^n}\left\{\lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2 - \langle\xi, x\rangle\right\} \\ &= -\inf_{x\in\mathbb{R}^n}\left\{\lambda\|x\|_1 + \frac{1}{2}\|x-\xi\|_2^2\right\} + \frac{1}{2}\|\xi\|_2^2. \end{aligned}$$

By the definition of the soft threshold function, we can see that the infimum is achieved as $x = S_\lambda(\xi)$. Thus

$$\begin{aligned} f^*(\xi) &= -\lambda\|S_\lambda(\xi)\|_1 - \frac{1}{2}\|S_\lambda(\xi) - \xi\|_2^2 + \frac{1}{2}\|\xi\|_2^2 \\ &= -\lambda\|S_\lambda(\xi)\|_1 - \frac{1}{2}\|S_\lambda(\xi)\|_2^2 + \langle S_\lambda(\xi), \xi\rangle. \end{aligned}$$

By virtue of Lemma 3.3 (i) we have

$$f^*(\xi) = \langle S_\lambda(\xi) - \xi, S_\lambda(\xi)\rangle - \frac{1}{2}\|S_\lambda(\xi)\|_2^2 + \langle S_\lambda(\xi), \xi\rangle = \frac{1}{2}\|S_\lambda(\xi)\|_2^2.$$

From Lemma 3.3 (iii) it then follows that f^* is continuous differentiable with

$$\nabla f^*(\xi) = S_\lambda(\xi), \quad \xi \in \mathbb{R}^n.$$

An application of Lemma 3.3 (ii) shows that

$$\|\nabla f^*(\xi) - \nabla f^*(\eta)\|_2 = \|S_\lambda(\xi) - S_\lambda(\eta)\|_2 \leq \|\xi - \eta\|_2$$

which coincides with the result in Theorem 3.8.

Next we introduce the Bregman distance induced by a convex function.

Definition 3.10 (Bregman distance)**.** Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function. Let $x \in \mathbb{R}^n$ be such that $\partial f(x) \neq \emptyset$ and let $\xi \in \partial f(x)$. The Bregman distance induced by f at x in the direction ξ is defined by

$$D_f^\xi(x, y) := f(y) - f(x) - \langle\xi, y - x\rangle, \quad y \in \mathbb{R}^n.$$

By definition it is easy to see that

$$D_f^\xi(x, y) \geq 0.$$

For any $x, y \in \mathbb{R}^n$ with $\partial f(x) \neq \emptyset$ and $\partial f(y) \neq \emptyset$, we have

$$\begin{aligned} D_f^\xi(x,y) &\leq D_f^\xi(x,y) + D_f^\eta(y,x) \\ &= [f(y) - f(x) - \langle \xi, y - x\rangle] + [f(x) - f(y) - \langle \eta, x - y\rangle] \\ &= \langle \xi - \eta, x - y\rangle \end{aligned}$$

for any $\xi \in \partial f(x)$ and $\eta \in \partial f(y)$.

When f is strongly convex, we may use $D_f^\xi(x,y)$ to bound $\|x-y\|_2^2$.

Lemma 3.11. *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be α-strongly convex. Then for any $x \in \mathbb{R}^n$ with $\partial f(x) \neq \emptyset$ and any $\xi \in \partial f(x)$ there holds*

$$D_f^\xi(x,y) \geq \alpha \|x-y\|_2^2, \quad \forall y \in \mathbb{R}^n.$$

Consequently $D_f^\xi(x,y) = 0 \iff y = x$.

Proof. By the strong convexity of f we have

$$f(ty + (1-t)x) + \alpha t(1-t)\|x-y\|_2^2 \leq tf(y) + (1-t)f(x)$$

for all $0 < t < 1$. Since $\xi \in \partial f(x)$, we have

$$\begin{aligned} f(ty + (1-tx) &\geq f(x) + \langle \xi, ty + (1-t)x - x\rangle \\ &= f(x) + t\langle \xi, y - x\rangle. \end{aligned}$$

Consequently

$$tf(y) + (1-t)f(x) \geq f(x) + t\langle \xi, y - x\rangle + \alpha t(1-t)\|x-y\|_2^2,$$

or

$$t[f(y) - f(x) - \langle \xi, y - x\rangle] \geq \alpha t(1-t)\|x-y\|_2^2$$

for all $0 < t \leq 1$. Dividing both sides by t gives

$$D_f^\xi(x,y) = f(y) - f(x) - \langle \xi, y - x\rangle \geq (1-t)\alpha\|x-y\|_2^2$$

for all $0 < t \leq 1$. Letting $t \to 0$ we can obtain $D_f^\xi(x,y) \geq \alpha\|x-y\|_2^2$. □

Definition 3.12 (Bregman projection)**.** Let $C \subset \mathbb{R}^n$ be a nonempty closed convex set and let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper lower semi-continuous convex function. For $x \in \mathbb{R}^n$ with $\partial f(x) \neq \emptyset$ and $\xi \in \partial f(x)$, an element $\hat{x} \in C$ such that

$$D_f^\xi(x, \hat{x}) = \min_{y \in C} D_f^\xi(x,y),$$

if exists, is called a Bregman projection of x onto C with respect to f and ξ.

For a general convex function, the Bregman projection may not exist. However, if f is strongly convex, then any $x \in \mathbb{R}^n$ with $\partial f(x) \neq \emptyset$ has a unique Bregman projection onto C with respect to f and any $\xi \in \partial f(x)$ because the function

$$y \to D_f^{\xi}(x, y)$$

is strong convex and hence coercive.

Lemma 3.13. *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a continuous convex function. The following statements about Bregman projection are equivalent:*

(i) $\hat{x} \in C$ is a Bregman projection of x onto C with respect to f and $\xi \in \partial f(x)$.

(ii) there exists $\hat{\xi} \in \partial f(\hat{x})$ such that

$$\langle \hat{\xi} - \xi, \hat{x} - y \rangle \leq 0, \quad \forall y \in C.$$

(iii) there exists $\hat{\xi} \in \partial f(\hat{x})$ such that

$$D_f^{\hat{\xi}}(\hat{x}, y) \leq D_f^{\xi}(x, y) - D_f^{\xi}(x, \hat{x}), \quad \forall y \in C.$$

The vector $\hat{\xi}$ in (ii) and (iii) is called an admissible subgradient for $\hat{x}$.

Proof. (i) $\Longleftrightarrow$ (ii). Let δ_C denote the indicator function of C, i.e.

$$\delta_C(y) = \begin{cases} 0 & \text{if } y \in C, \\ +\infty & \text{if } y \notin C \end{cases}$$

which is a proper, lower semi-continuous, convex function because C is a nonempty closed convex set. Then, by Theorem 3.5 and Theorem 3.6, we have

$$\begin{aligned}
& D_f^{\xi}(x, \hat{x}) = \min_{y \in C} D_f^{\xi}(x, y) \\
& \Longleftrightarrow \ \hat{x} \in \arg\min_{y \in \mathbb{R}^n} \left\{ D_f^{\xi}(x, y) + \delta_C(y) \right\} \\
& \Longleftrightarrow \ 0 \in \partial \left(D_f^{\xi}(x, \cdot) + \delta_C \right)(\hat{x}) = \partial f(\hat{x}) - \xi + \partial \delta_C(\hat{x}) \\
& \Longleftrightarrow \ \exists \hat{\xi} \in \partial f(\hat{x}) \text{ such that } \xi - \hat{\xi} \in \partial \delta_C(\hat{x}) \\
& \Longleftrightarrow \ \exists \hat{\xi} \in \partial f(\hat{x}) \text{ such that} \\
& \qquad \delta_C(y) \geq \delta_C(\hat{x}) + \langle \xi - \hat{\xi}, y - \hat{x} \rangle, \quad \forall y \in \mathbb{R}^n \\
& \Longleftrightarrow \ \exists \hat{\xi} \in \partial f(\hat{x}) \text{ such that} \\
& \qquad 0 \geq \langle \xi - \hat{\xi}, y - \hat{x} \rangle, \quad \forall y \in C.
\end{aligned}$$

(ii) $\Longleftrightarrow$ (iii). By the definition of Bregman distance we have

$$\begin{aligned}
&D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) + D_f^{\xi}(x, \hat{x}) \\
&= [f(y) - f(\hat{x}) - \langle \hat{\xi}, y - \hat{x} \rangle] - [f(y) - f(x) - \langle \xi, y - x \rangle)] \\
&\quad + [f(\hat{x}) - f(x) - \langle \xi, \hat{x} - x \rangle] \\
&= -\langle \hat{\xi}, y - \hat{x} \rangle + \langle \xi, y - x \rangle - \langle \xi, \hat{x} - x \rangle \\
&= \langle \xi - \hat{\xi}, y - \hat{x} \rangle.
\end{aligned}$$

Therefore (ii) and (iii) are equivalent. □

With the preparation of Bregman projection, the exact-step randomised Kaczmarz method for solving $Ax = b$ defines its next iterate as a Bregman projection of the current iterate onto a hyperplane $\{a_i^T x = b_i\}$ with i randomly chosen from $\{1, 2, \cdots, m\}$ with suitable probability distribution. In order to formulate the method more explicitly, we need to determine how to find a Bregman projection onto a hyperplane. We need the following special version of the Karush-Kuhn-Tucker theory for constrained minimisation problems.

Proposition 3.14. *Consider the minimisation problem*

$$\min\{f(x) : Ax = b\}, \tag{3.11}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Then $x^* \in \mathbb{R}^n$ is an optimal solution of (3.11) if and only if there exists $\mu \in \mathbb{R}^m$ such that

$$0 \in \partial f(x^*) + A^T \mu \quad \text{and} \quad Ax^* = b.$$

Proof. Please refer to [43]. □

Lemma 3.15. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be α-strongly convex with $\alpha > 0$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$. Let f^* denote the Legendre-Fenchel conjugate of f. Then the Bregman projection of $x \in \mathbb{R}^n$ onto the affine subspace*

$$L(A, b) := \{x \in \mathbb{R}^n | Ax = b\} \neq \emptyset$$

with respect to f and $\xi \in \partial f(x)$ is given by

$$\hat{x} = \nabla f^*(\xi - A^T \hat{\mu}),$$

where $\hat{\mu} \in \mathbb{R}^m$ is a solution of

$$\min_{\mu \in \mathbb{R}^m} \left\{ f^*(\xi - A^T \mu) + \langle b, \mu \rangle \right\}. \tag{3.12}$$

Moreover, $\hat{\xi} := \xi - A^T\hat{\mu}$ *is an admissible subgradient for* $\hat{x}$*. If* A *has full row rank then*

$$D_f^{\hat{\xi}}(\hat{x}, y) \leq D_f^{\xi}(x, y) - \alpha\|(AA^T)^{-1/2}(Ax - b)\|_2^2 \tag{3.13}$$

for all $y \in L(A, b)$*.*

Proof. Since $\hat{x}$ is the Bregman projection of x onto $L(A, b)$ with respect to f and $\xi \in \partial f(x)$, we have

$$\hat{x} = \arg\min \left\{ D_f^{\xi}(x, y) : Ay = b \right\}.$$

Recall the definition of Bregman distance,

$$D_f^{\xi}(x, y) = f(y) - f(x) - \langle \xi, y - x \rangle.$$

By the Karush-Kuhn-Tucker theory for constrained minimisation problems, i.e. Proposition 3.14, there is $\hat{\mu} \in \mathbb{R}^m$ such that

$$0 \in \partial(D_f^{\hat{\xi}}(x, \cdot))(\hat{x}) + A^T\hat{\mu} = \partial f(\hat{x}) - \xi + A^T\hat{\mu} \quad \text{and} \quad A\hat{x} = b.$$

Therefore

$$\hat{\xi} := \xi - A^T\hat{\mu} \in \partial f(\hat{x}).$$

Consequently, it follows from Theorem 3.8 that

$$\hat{x} \in \partial f^*(\xi - A^T\hat{\mu}).$$

Since f^* is differentiable, we thus obtain $\hat{x} = \nabla f^*(\xi - A^T\hat{\mu})$.

Next we use the equation $A\hat{x} = b$ to obtain

$$b = A\hat{x} = A\nabla f^*(\xi - A^T\hat{\mu}).$$

Therefore

$$\nabla_\mu(f^*(\xi - A^T\hat{\mu}) + \langle b, \hat{\mu} \rangle) = 0.$$

This means that $\hat{\mu}$ is a solution of (3.12).

The vector $\hat{\xi} := \xi - A^T\hat{\mu}$ is an admissible subgradient of $\hat{x}$ because

$$\langle \hat{\xi} - \xi, \hat{x} - y \rangle = -\langle A^T\hat{\mu}, \hat{x} - y \rangle = \langle \hat{\mu}, A\hat{x} - b \rangle = 0$$

for all $y \in L(A, b)$.

Finally we show (3.13). We have

$$\begin{aligned} D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) &= [f(y) - f(\hat{x}) - \langle \hat{\xi}, y - \hat{x} \rangle] - [f(y) - f(x) - \langle \xi, y - x \rangle] \\ &= f(x) - f(\hat{x}) - \langle \hat{\xi}, y - \hat{x} \rangle + \langle \xi, y - x \rangle \\ &= -[f(\hat{x}) - \langle \hat{\xi}, \hat{x} \rangle] + [f(x) - \langle \xi, x \rangle] - \langle \hat{\xi} - \xi, y \rangle. \end{aligned}$$

From Theorem 3.8 it follows that

$$D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) = f^*(\hat{\xi}) - f^*(\xi) - \langle \hat{\xi} - \xi, y \rangle.$$

Recall that $\hat{\xi} = \xi - A^T \hat{\mu}$. We have

$$\begin{aligned} D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) &= f^*(\xi - A^T \hat{\mu}) + \langle A^T \hat{\mu}, y \rangle - f^*(\xi) \\ &= f^*(\xi - A^T \hat{\mu}) + \langle \hat{\mu}, b \rangle - f^*(\xi). \end{aligned}$$

Since $\hat{\mu}$ is a solution of (3.12), we have for any $\mu \in \mathbb{R}^m$ that

$$\begin{aligned} D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) &\leq f^*(\xi - A^T \mu) + \langle b, \mu \rangle - f^*(\xi), \\ &= f^*(\xi - A^T \mu) - f^*(\xi) - \langle \nabla f^*(\xi), -A^T \mu \rangle \\ &\quad + \langle \nabla f^*(\xi), -A^T \mu \rangle + \langle b, \mu \rangle. \end{aligned}$$

By using the Lipschitz continuity of ∇f^*, see Theorem 3.8, we can obtain

$$D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) \leq \frac{1}{4\alpha} \| - A^T \mu \|_2^2 + \langle \mu, b - A \nabla f^*(\xi) \rangle.$$

Since $\xi \in \partial(x)$, we have from Theorem 3.8 that $x = \nabla f^*(\xi)$. Therefore

$$D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) \leq \frac{1}{4\alpha} \| A^T \mu \|_2^2 + \langle \mu, b - Ax \rangle \tag{3.14}$$

for any $\mu \in \mathbb{R}^m$.

Now we choose μ to minimise the right hand side

$$h(\mu) = \frac{1}{4\alpha} \| A^T \mu \|_2^2 + \langle \mu, b - Ax \rangle.$$

It gives

$$\frac{1}{2\alpha} A A^T \mu + b - Ax = 0.$$

Since A has full row rank, AA^T is positive definite matrix. Therefore we can solve the above equation to obtain

$$\mu = 2\alpha (AA^T)^{-1} (Ax - b).$$

Substituting this μ into the equation (3.14) and noting $\|A^T\mu\|_2 = \|(AA^T)^{1/2}\mu\|_2$, we have

$$\begin{aligned} D_f^{\hat{\xi}}(\hat{x}, y) - D_f^{\xi}(x, y) &\leq \frac{1}{4\alpha}||2\alpha(AA^T)^{1/2}(AA^T)^{-1}(Ax-b)||_2^2 \\ &\quad + 2\alpha\langle b - Ax, (AA^T)^{-1}(Ax-b)\rangle \\ &= -\alpha\|(AA^T)^{-1/2}(Ax-b)\|_2^2. \end{aligned}$$

The proof is complete. □

Corollary 3.16. Let $f : \mathbb{R}^n \to \mathbb{R}$ be α-strongly convex with $\alpha > 0$, $0 \neq u \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$. The Bregman projection of $x \in \mathbb{R}^n$ onto the hyperplane

$$H(u, \beta) := \{x \in \mathbb{R}^n : \langle u, x\rangle = \beta\}$$

with respect to f and $\xi \in \partial f(x)$ is given by

$$\hat{x} = \nabla f^*(\xi - \hat{t}u),$$

where $\hat{t} \in \mathbb{R}$ is a solution of

$$\min_{t\in\mathbb{R}} \left\{f^*(\xi - tu) + t\beta\right\}.$$

Moreover, $\hat{\xi} := \xi - \hat{t}u$ is an admissible subgradient for $\hat{x}$ and

$$D_f^{\hat{\xi}}(\hat{x}, y) \leq D_f^{\xi}(x, y) - \alpha\frac{(\langle u, x\rangle - \beta)^2}{||u||_2^2}$$

for all $y \in H(u, \beta)$.

With the help of this corollary, we are ready to formulate the exact-step randomised sparse Kaczmarz method (ERSKM).

Algorithm 8 (ERSKM)**.** *Let f be given by (3.5) and let $x_0 = \xi_0 = 0$. For $k = 1, 2, \cdots$ define*

$$\begin{aligned} t_k &= \arg\min_{t\in\mathbb{R}} \left\{f^*(\xi_k - ta_i) + tb_i\right\}, \\ \xi_{k+1} &= \xi_k - t_k a_i, \\ x_{k+1} &= \nabla f^*(\xi_{k+1}) = S_\lambda(\xi_{k+1}), \end{aligned}$$

where the index i is randomly chosen from the set $\{1, 2, ..., m\}$ with the probability

$$\mathbb{P}(i = j) = \frac{\|a_j\|_2^2}{||A||_F^2}, \quad j = 1, \cdots, m.$$

3.2 Convergence

In this section we will show that the sequences defined in Algorithm 7 and Algorithm 8 converge in expectation to the unique solution of (3.4) with linear rates. The proof is based on the restricted strong convexity of the dual objective function of (3.4). So let us first determine the dual problem of (3.4). Note that the associated Lagrangian function is

$$\mathscr{L}(x,\mu) := \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2 + \langle \mu, b - Ax\rangle.$$

By definition, the dual function is

$$\begin{aligned} d(\mu) &= \inf_{x\in\mathbb{R}^n}\left\{\mathscr{L}(x,\mu) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2 + \langle \mu, b - Ax\rangle\right\} \\ &= \inf_{x\in\mathbb{R}^n}\left\{\lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2 - \langle A^T\mu, x\rangle + \langle \mu, b\rangle\right\} \\ &= \inf_{x\in\mathbb{R}^n}\left\{\lambda\|x\|_1 + \frac{1}{2}\|x - A^T\mu\|_2^2 - \frac{1}{2}\|A^T\mu\|_2^2 + \langle \mu, b\rangle\right\} \end{aligned}$$

According to the definition of the soft threshold function, one can see that the above infimum is achieved at $x = S_\lambda(A^T\mu)$. Therefore

$$\begin{aligned} d(\mu) &= \lambda\|S_\lambda(A^T\mu)\|_1 + \frac{1}{2}\|S_\lambda(A^T\mu)\|_2^2 - \langle A^T\mu, S_\lambda(A^T\mu)\rangle + \langle \mu, b\rangle \\ &= \lambda\|S_\lambda(A^T\mu)\|_1 + \langle S_\lambda(A^T\mu) - A^T\mu, S_\lambda(A^T\mu)\rangle - \frac{1}{2}\|S_\lambda(A^T\mu)\|_2^2 + \langle \mu, b\rangle. \end{aligned}$$

From Lemma 3.3 (i) it follows that

$$\lambda\|S_\lambda(A^T\mu)\|_1 + \langle S_\lambda(A^T\mu) - A^T\mu, S_\lambda(A^T\mu)\rangle = 0. \tag{3.15}$$

Therefore

$$d(\mu) = -\frac{1}{2}\|S_\lambda(A^T\mu)\|_2^2 + \langle \mu, b\rangle.$$

Let

$$g(\mu) := -d(\mu) = \frac{1}{2}\|S_\lambda(A^T\mu)\|_2^2 - \langle \mu, b\rangle.$$

Then $g(\mu)$ is a convex function and the dual problem has the form

$$\max_{\mu\in\mathbb{R}^m} d(\mu) = -\min_{\mu\in\mathbb{R}^m} g(\mu).$$

For the minimisation problem (3.4), the strong duality holds because it only involves a linear equality constraint with a continuous objective function. Thus

$$\min_{x\in\mathbb{R}^n}\left\{f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2 : Ax = b\right\} = -\min_{\mu\in\mathbb{R}^m} g(\mu).$$

Lemma 3.17. *Let x^* be the unique solution of (3.4). Then for any solution $\hat{\mu}$ of the minimisation problem*

$$\min_{\mu\in\mathbb{R}^m}\left\{g(\mu)=\frac{1}{2}\|S_\lambda(A^T\mu)\|_2^2-\langle b,\mu\rangle\right\} \tag{3.16}$$

there holds $A^T\hat{\mu}\in\partial f(x^*)$.

Proof. By the strong duality we have $f(x^*)=-g(\hat{\mu})$, i.e.

$$\lambda\|x^*\|_1+\frac{1}{2}\|x^*\|_2^2=-\frac{1}{2}\|S_\lambda(A^T\hat{\mu})\|_2^2+\langle b,\hat{\mu}\rangle.$$

Since $Ax^*=b$, we have

$$\lambda\|x^*\|_1+\frac{1}{2}\|x^*\|_2^2=-\frac{1}{2}\|S_\lambda(A^T\hat{\mu})\|_2^2+\langle x^*,A^T\hat{\mu}\rangle.$$

By using the identity (3.15) it is easy to see that

$$\lambda\|S_\lambda(A^T\hat{\mu})\|_1+\frac{1}{2}\|S_\lambda(A^T\hat{\mu})\|_2^2=-\frac{1}{2}\|S_\lambda(A^T\hat{\mu})\|_2^2+\langle S_\lambda(A^T\hat{\mu}),A^T\hat{\mu}\rangle.$$

By the uniqueness of x^*, we must have

$$x^*=S_\lambda(A^T\hat{\mu}).$$

Recall that for the Legendre-Fenchel conjugate f^* of f we have $\nabla f^*(\xi)=S_\lambda(\xi)$. Thus

$$x^*=S_\lambda(A^T\hat{\mu})=\nabla f^*(A^T\hat{\mu}).$$

According to Theorem 3.8, we therefore have $A^T\hat{\mu}\in\partial f(x^*)$. □

In order to introduce the next result, we need some notations. For any nonzero vector x, we define

$$|x|_{\min}:=\min\{|x_j|:j\in\mathrm{supp}(x)\}>0.$$

For any subset J of the index set $\{1,2,\cdots,n\}$, we use A_J to denote the matrix that is formed by the columns of A indexed by J. We then define

$$\tilde{\sigma}_{\min}(A):=\min\{\sigma_{\min}(A_J):J\subset\{1,\cdots,n\}\text{ and }A_J\neq 0\},$$

where $\sigma(A_J)$ denotes the smallest singular value of A_J.

Lemma 3.18 (Restricted strong convexity)**.** *Let U be the solution set of the dual problem (3.16) and let P_U be the Euclidean projection of $\mathbb{R}^m$ onto U. Then*

$$\langle \mu - P_U(\mu), \nabla g(\mu) \rangle \geq \nu \|\mu - P_U(\mu)\|_2^2 \tag{3.17}$$

for all $\mu \in \mathbb{R}^m$, where

$$\nu := \tilde{\sigma}_{\min}^2(A) \frac{|x^*|_{\min}}{|x^*|_{\min} + 2\lambda} > 0.$$

The proof of this result is very lengthy. So we will not include its proof here and one may refer to [22] for the details. For a differentiable function g, it is easy to see that g is ν-strongly convex if and only if

$$\langle \mu - \bar{\mu}, \nabla g(\mu) - \nabla g(\bar{\mu}) \rangle \geq \nu \|\mu - \bar{\mu}\|_2^2 \tag{3.18}$$

for all $\mu, \bar{\mu} \in \mathbb{R}^m$. Note that $P_U(\mu)$ is a minimiser of g, we have $\nabla g(P_U(\mu)) = 0$. Consequently (3.17) can be written as

$$\langle \mu - P_U(\mu), \nabla g(\mu) - \nabla g(P_U(\mu)) \rangle \geq \nu \|\mu - P_U(\mu)\|_2^2.$$

Compared with (3.18), we can see that (3.17) is the restriction of (3.18) to the specially chosen $\bar{\mu} = P_U(\mu)$. Because of this reason, the result in Lemma 3.18 is called the restricted strong convexity of g.

Lemma 3.19. *Let x^* be the unique solution of (3.4). For any $x \in \mathbb{R}^n$ with $\partial f(x) \cap \mathcal{R}(A^T) \neq \emptyset$ and any $\xi \in \partial f(x) \cap \mathcal{R}(A^T)$ there holds*

$$D_f^{\xi}(x, x^*) \leq \frac{1}{\tilde{\sigma}_{\min}^2(A)} \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \|Ax - b\|_2^2.$$

Proof. For $\xi \in \partial f(x) \cap \mathcal{R}(A^T)$ we have $x = \nabla f^*(\xi) = S_\lambda(\xi)$ and $\xi = A^T \hat{\mu}$ for some $\hat{\mu} \in \mathbb{R}^m$. By using Lemma 3.18 we have

$$\begin{aligned} \|\hat{\mu} - P_U(\hat{\mu})\|_2^2 &\leq \frac{1}{\tilde{\sigma}_{\min}^2(A)} \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \langle \hat{\mu} - P_U(\hat{\mu}), \nabla g(\hat{\mu}) \rangle \\ &\leq \frac{1}{\tilde{\sigma}_{\min}^2(A)} \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \|\hat{\mu} - P_U(\hat{\mu})\|_2 \|\nabla g(\hat{\mu})\|_2. \end{aligned}$$

Therefore

$$\|\hat{\mu} - P_U(\hat{\mu})\|_2 \leq \frac{1}{\tilde{\sigma}_{\min}^2(A)} \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \|\nabla g(\hat{\mu})\|_2.$$

According to the definition of $g(\mu)$ and Lemma 3.3 (iii) we have

$$\nabla g(\hat{\mu}) = AS_\lambda(A^T\hat{\mu}) - b = AS_\lambda(\xi) - b = Ax - b.$$

Consequently

$$\|\hat{\mu} - P_U(\hat{\mu})\|_2 \leq \frac{1}{\tilde{\sigma}^2_{\min}(A)} \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \|Ax - b\|_2.$$

Since $P_U(\hat{\mu})$ is a solution of (3.16), it follows from Lemma 3.17 that $\xi^* := A^T P_U(\hat{\mu}) \in \partial f(x^*)$. Therefore

$$\begin{aligned} D_f^\xi(x, x^*) &\leq \langle \xi - \xi^*, x - x^* \rangle = \langle A^T\hat{\mu} - A^T P_U(\hat{\mu}), x - x^* \rangle \\ &= \langle \hat{\mu} - P_U(\hat{\mu}), Ax - b \rangle \\ &\leq \|\hat{\mu} - P_U(\hat{\mu})\|_2 \|Ax - b\|_2 \\ &\leq \frac{1}{\tilde{\sigma}^2_{\min}(A)} \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \|Ax - b\|_2^2 \end{aligned}$$

which completes the proof. □

Now we are ready to provide the main convergence results on Algorithm 7 and Algorithm 8.

Theorem 3.20. *Let x^* be the unique solution of (3.4). Then the iterates x_k generated by both Algorithm 7 and Algorithm 8 converge to x^* in expectation with a linear rate, i.e.*

$$\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1}, x^*)] \leq q\mathbb{E}[D_f^{\xi_k}(x_k, x^*)]$$

and

$$\mathbb{E}[\|x_k - x^*\|_2^2] \leq q^k(2\lambda\|x^*\|_1 + \|x^*\|_2^2),$$

where

$$q = 1 - \frac{1}{2\tilde{\kappa}^2} \frac{|x^*|_{\min}}{|x^*|_{\min} + 2\lambda}, \qquad \tilde{\kappa} = \frac{\|A\|_F}{\tilde{\sigma}_{\min}(A)}$$

and the expectations is taken with respect to the choice of rows in the algorithms.

Proof. We first claim that

$$D_f^{\xi_{k+1}}(x_{k+1}, x^*) \leq D_f^{\xi_k}(x_k, x^*) - \frac{1}{2} \frac{(\langle a_i, x_k \rangle - b_i)^2}{\|a_i\|_2^2}. \tag{3.19}$$

For Algorithm 8, this claim follows immediately from Corollary 3.16 and the α-strong convexity of f with $\alpha = 1/2$. For Algorithm 7, this claim can be confirmed

as follows: By using the argument in the proof of Lemma 3.15 we have

$$\begin{aligned} D_f^{\xi_{k+1}}(x_{k+1}, x^*) - D_f^{\xi_k}(x_k, x^*) &= f^*(\xi_{k+1}) - f^*(\xi_k) - \langle \xi_{k+1} - \xi_k, x^* \rangle \\ &= f^*(\xi_{k+1}) - f^*(\xi_k) - \langle \xi_{k+1} - \xi_k, \nabla f^*(\xi_k) \rangle \\ &\quad + \langle \xi_{k+1} - \xi_k, \nabla f^*(\xi_k) - x^* \rangle. \end{aligned}$$

With the help of Theorem 3.8 and the fact $x_k = S_\lambda(\xi_k) = \nabla f^*(\xi_k)$, we can obtain

$$D_f^{\xi_{k+1}}(x_{k+1}, x^*) - D_f^{\xi_k}(x_k, x^*) \leq \frac{1}{2}\|\xi_{k+1} - \xi_k\|_2^2 + \langle \xi_{k+1} - \xi_k, x_k - x^* \rangle,$$

where we used the α-strong convexity of f with $\alpha = 1/2$. Since

$$\xi_{k+1} - \xi_k = -\frac{\langle a_i, x_k \rangle - b_i}{\|a_i\|_2^2} a_i$$

and $\langle a_i, x^* \rangle = b_i$, we have

$$\begin{aligned} D_f^{\xi_{k+1}}(x_{k+1}, x^*) - D_f^{\xi_k}(x_k, x^*) &\leq \frac{1}{2}\frac{(\langle a_i, x_k \rangle - b_i)^2}{\|a_i\|_2^2} - \frac{\langle a_i, x_k \rangle - b_i}{\|a_i\|_2^2}\langle a_i, x_k - x^* \rangle \\ &= \frac{1}{2}\frac{(\langle a_i, x_k \rangle - b_i)^2}{\|a_i\|_2^2} - \frac{(\langle a_i, x_k \rangle - b_i)^2}{\|a_i\|_2^2} \\ &= -\frac{1}{2}\frac{(\langle a_i, x_k \rangle - b_i)^2}{\|a_i\|_2^2} \end{aligned}$$

which shows the claim (3.19).

By taking the expectation of (3.19) conditioned on knowing all the previous k iterations, it follows

$$\begin{aligned} \mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1}, x^*)|k] &\leq D_f^{\xi_k}(x_k, x^*) - \sum_{i=1}^m \frac{\|a_i\|_2^2}{\|A\|_F^2} \frac{(\langle a_i, x_k \rangle - b_i)^2}{2\|a_i\|_2^2} \\ &= D_f^{\xi_k}(x_k, x^*) - \frac{1}{2||A||_F^2} \sum_{i=1}^m (\langle a_i, x_k \rangle - b_i)^2 \\ &= D_f^{\xi_k}(x_k, x^*) - \frac{||Ax_k - b||_2^2}{2||A||_F^2}. \end{aligned}$$

With the help of Lemma 3.19, we have

$$\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1}, x^*)|k] \leq q D_f^{\xi_k}(x_k, \hat{x}).$$

By taking the full expectation, we obtain

$$\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1}, x^*)] \leq q \mathbb{E}[D_f^{\xi_k}(x_k, x^*)].$$

By using this inequality recursively, we have

$$\mathbb{E}[D_f^{\xi_k}(x_k, x^*)] \leq q^k D_f^{\xi_0}(x_0, x^*) = q^k f(x^*) = \left(\lambda\|x^*\|_1 + \frac{1}{2}\|x^*\|_2^2\right) q^k.$$

Note that

$$D_f^{\xi_k}(x_k, x^*) \geq \frac{1}{2}\|x_k - x^*\|,$$

see Lemma 3.11, we immediately have

$$\mathbb{E}[\|x_k - x^*\|^2] \leq \left(2\lambda\|x^*\|_1 + \|x^*\|_2^2\right) q^k.$$

The proof is therefore complete. □

We can also consider convergence rate of Algorithm 7 and Algorithm 8 for noisy data case.

Theorem 3.21. *Let x^* be the unique solution of (3.4). Let $\tilde{b} \in \mathbb{R}^m$ be a noisy data satisfying $\|\tilde{b} - b\|_2 \leq \delta$. Let $\{x_k\}$ and $\{\xi_k\}$ be the computed results from Algorithm 7 and Algorithm 8 with b replaced by $\tilde{b}$. Then we have the following error estimates:*

(i) For Algorithm 7 there holds

$$\mathbb{E}[D_f^{\xi_k}(x_k, x^*)] \leq q^k f(x^*) + \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \cdot \frac{\delta^2}{\tilde{\sigma}^2(A)}.$$

(ii) For Algorithm 8 there holds

$$\mathbb{E}[D_f^{\xi_k}(x_k, x^*)] \leq q^k f(x^*) + \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \cdot \frac{\delta^2 + 4\|A\|_{1,2}\delta}{\tilde{\sigma}^2(A)},$$

where $q < 1$ is defined as in Theorem 3.20 and $\|A\|_{1,2} = \sqrt{\sum_{i=1}^m \|a_i\|_1^2}$.

Proof. Let

$$\tilde{x} := x^* + \frac{\tilde{b}_i - b_i}{\|a_i\|_2^2} a_i.$$

Then

$$a_i^T \tilde{x} = a_i^T x^* + \frac{\tilde{b}_i - b_i}{\|a_i\|_2^2} a_i^T a_i = a_i^T x^* + (\tilde{b}_i - b_i) = b_i + (\tilde{b}_i - b_i) = \tilde{b}_i.$$

This shows that $\tilde{x} \in H(a_i, \tilde{b}_i)$. For Algorithm 8, by definition x_{k+1} is a Bregman projection of x_k onto the hyperplane $H(a_i, \tilde{b}_i)$ and ξ_{k+1} is an admissible subgradient for x_{k+1}. Therefore, we may use Corollary 3.16 to conclude that

$$D_f^{\xi_{k+1}}(x_{k+1}, \tilde{x}) \leq D_f^{\xi_k}(x_k, \tilde{x}) - \frac{1}{2}\frac{(\langle a_i, x_k\rangle - \tilde{b}_i)^2}{\|a_i\|_2^2}. \tag{3.20}$$

For Algorithm 7, (3.20) can be derived by the exactly same argument in the proof of Theorem 3.20.

By the definition of Bregman distance,we can write (3.20) explicitly as

$$f(\tilde{x}) - f(x_{k+1}) - \langle \xi_{k+1}, \tilde{x} - x_{k+1} \rangle \leq f(\tilde{x}) - f(x_k) - \langle \xi_k, \tilde{x} - x_k \rangle - \frac{1}{2} \frac{(\langle a_i, x_k \rangle - \tilde{b}_i)^2}{\|a_i\|_2^2}.$$

By regrouping the terms, we have

$$\begin{aligned} & f(x^*) - f(x_{k+1}) - \langle \xi_{k+1}, x^* - x_{k+1} \rangle + \langle \xi_{k+1}, x^* - \tilde{x} \rangle \\ & \leq f(x^*) - f(x_k) - \langle \xi_k, x^* - x_k \rangle + \langle \xi_k, x^* - \tilde{x} \rangle - \frac{1}{2} \frac{(\langle a_i, x_k \rangle - \tilde{b}_i)^2}{\|a_i\|_2^2}. \end{aligned}$$

Consequently

$$D_f^{\xi_{k+1}}(x_{k+1}, x^*) \leq D_f^{\xi_k}(x_k, x^*) + \langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle - \frac{1}{2} \frac{(\langle a_i, x_k \rangle - \tilde{b}_i)^2}{\|a_i\|_2^2}. \quad (3.21)$$

We need to treat the term $\langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle$ by considering Algorithm 7 and Algorithm 8 separately.

(i) For Algorithm 7, we have

$$\xi_{k+1} - \xi_k = -\frac{\langle a_i, x_k \rangle - \tilde{b}_i}{\|a_i\|_2^2} a_i.$$

Therefore

$$\langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle = -\frac{\langle a_i, x_k \rangle - \tilde{b}_i}{\|a_i\|_2^2} \langle a_i, \tilde{x} - x^* \rangle$$

According to the definition of $\tilde{x}$, we have

$$\langle a_i, \tilde{x} - x^* \rangle = \tilde{b}_i - b_i.$$

Therefore

$$\begin{aligned} \langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle &= -\frac{\langle a_i, x_k \rangle - \tilde{b}_i}{\|a_i\|_2^2} (\tilde{b}_i - b_i) \\ &= \frac{(\tilde{b}_i - b_i)^2}{\|a_i\|_2^2} - \frac{(\tilde{b}_i - b_i)(\langle a_i, x_k \rangle - b_i)}{\|a_i\|_2^2}. \end{aligned}$$

By expanding the terms, we also have

$$\begin{aligned} \frac{(\langle a_i, x_k \rangle - \tilde{b}_i)^2}{\|a_i\|_2^2} &= \frac{((\langle a_i, x_k \rangle - b_i) - (\tilde{b}_i - b_i))^2}{\|a_i\|_2^2} \\ &= -\frac{(\langle a_i, x_k \rangle - b_i)^2 + (\tilde{b}_i - b_i)^2 - 2(\tilde{b}_i - b_i)(\langle a_i, x_k \rangle - b_i)}{\|a_i\|_2^2}. \end{aligned}$$

Therefore

$$\langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle - \frac{1}{2}\frac{(\langle a_i, x_k\rangle - \tilde{b}_i)^2}{\|a_i\|_2^2} = \frac{(\tilde{b}_i - b_i)^2 - (\langle a_i, x_k\rangle - b_i)^2}{2\|a_i\|_2^2}.$$

Combining this with (3.21) we have

$$D_f^{\xi_{k+1}}(x_{k+1}, x^*) \leq D_f^{\xi_k}(x_k, x^*) + \frac{(\tilde{b}_i - b_i)^2 - (\langle a_i, x_k\rangle - b_i)^2}{2\|a_i\|_2^2}.$$

Now we take the expectation of both sides conditional on knowing the previous iterates to obtain

$$\begin{aligned}
\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1}, x^*)|k] &\leq D_f^{\xi_k}(x_k, x^*) + \sum_{i=1}^{m} \frac{\|a_i\|_2^2}{\|A\|_F^2} \frac{(\tilde{b}_i - b_i)^2 - (\langle a_i, x_k\rangle - b_i)^2}{2\|a_i\|_2^2} \\
&= D_f^{\xi_k}(x_k, x^*) - \frac{||Ax_k - b||_2^2 - ||b^\delta - b||_2^2}{2||A||_F^2} \\
&\leq D_f^{\xi_k}(x_k, x^*) - \frac{\|Ax_k - b\|_2^2}{2\|A\|_F^2} + \frac{\delta^2}{2\|A\|_F^2}.
\end{aligned}$$

With the help of Lemma 3.19 we then obtain

$$\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1}, x^*)|k] \leq q D_f^{\xi_k}(x_k, x^*) + \frac{\delta^2}{2||A||_F^2}$$

By taking the full expectation on both sides, it follows

$$\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1}, x^*)] \leq q\mathbb{E}[D_f^{\xi_k}(x_k, x^*)] + \frac{\delta^2}{2||A||_F^2}.$$

Recursively using this equation we can obtain

$$\begin{aligned}
\mathbb{E}[D_f^{\xi_k}(x_k, x^*)] &\leq q^k \mathbb{E}[D_f^{\xi_0}(x_0, x^*)] + (1 + q + \cdots + q^{k-1}) \frac{\delta^2}{2||A||_F^2} \\
&\leq q^k D_f^{\xi_0}(x_0, x^*) + \frac{1}{1-q} \frac{\delta^2}{2\|A\|_F^2} \\
&= q^k f(x^*) + \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \cdot \frac{\delta^2}{\tilde{\sigma}^2(A)}.
\end{aligned}$$

(ii) Next we consider Algorithm 8. Recall that $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$, we have

$$\begin{aligned}
\partial f(x) &= x + \lambda \partial(\|\cdot\|_1)(x) \\
&= x + \lambda \left\{ \xi \in \mathbb{R}^n : \|\xi\|_\infty \leq 1 \text{ and } \langle \xi, x\rangle = \|x\|_1 \right\}.
\end{aligned}$$

Since $\xi_k \in \partial f(x_k)$ and $\xi_{k+1} \in \partial f(x_{k+1})$, we must have

$$\xi_k = x_k + \lambda s_k \quad \text{and} \quad \xi_{l+1} = x_{k+1} + \lambda s_{k+1}$$

with $\|s_k\|_\infty, \|s_{k+1}\|_\infty \leq 1$. Therefore, using the definition of $\tilde{x}$, we have

$$\langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle = \frac{\tilde{b}_i - b_i}{\|a_i\|_2^2} \left(\langle a_i, x_{k+1} - x_k \rangle + \langle a_i, s_{k+1} - s_k \rangle \right).$$

Recall that $x_{k+1} = \nabla f^*(\xi_{k+1}) = \nabla f^*(\xi_k - t_k a_i)$, where t_k is a minimiser of $f^*(\xi_k - t a_i) + t\tilde{b}_i$. We have

$$0 = \frac{d}{dt}\Big|_{t=t_k} (f^*(\xi_k - t a_i) + t\tilde{b}_i) = -\langle a_i, \nabla f^*(\xi_k - t_k a_i) \rangle + \tilde{b}_i = -\langle a_i, x_{k+1} \rangle + \tilde{b}_i.$$

This shows that $\langle a_i, x_{k+1} \rangle = \tilde{b}_i$. Therefore

$$\begin{aligned} \langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle &= \frac{\tilde{b}_i - b_i}{\|a_i\|_2^2} \left(\tilde{b}_i - \langle a_i, x_k \rangle + \langle a_i, s_{k+1} - s_k \rangle \right) \\ &\leq \frac{(\tilde{b}_i - b_i)^2 - (\tilde{b}_i - b_i)(\langle a_i, x_k \rangle - b_i)}{\|a_i\|_2^2} + \frac{2\|a_i\|_1 |\tilde{b}_i - b_i|}{\|a_i\|_2^2}. \end{aligned}$$

Thus

$$\begin{aligned} \langle \xi_{k+1} - \xi_k, \tilde{x} - x^* \rangle - \frac{1}{2} \frac{(\langle a_i, x_k \rangle - \tilde{b}_i)^2}{\|a_i\|_2^2} &= \frac{(\tilde{b}_i - b_i)^2 - (\langle a_i, x_k \rangle - b_i)^2}{2\|a_i\|_2^2} \\ &\quad + \frac{2\|a_i\|_1 |\tilde{b}_i - b_i|}{\|a_i\|_2^2}. \end{aligned}$$

Combining this with (3.21) we have

$$D_f^{\xi_{k+1}}(x_{k+1}, x^*) \leq D_f^{\xi_k}(x_k, x^*) + \frac{(\tilde{b}_i - b_i)^2 - (\langle a_i, x_k \rangle - b_i)^2}{2\|a_i\|_2^2} + \frac{2\|a_i\|_1 |\tilde{b}_i - b_i|}{\|a_i\|_2^2}.$$

Now we take the expectation of both sides conditional on knowing the previous

iterates and use Lemma 3.19 to obtain

$$\begin{aligned}
\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1},x^*)|k] &\leq D_f^{\xi_k}(x_k,x^*) + \sum_{i=1}^{m} \frac{\|a_i\|_2^2}{\|A\|_F^2} \frac{(\tilde{b}_i - b_i)^2 - (\langle a_i, x_k\rangle - b_i)^2}{2\|a_i\|_2^2} \\
&\quad + \sum_{i=1}^{m} \frac{\|a_i\|_2^2}{\|A\|_F^2} \frac{2\|a_i\|_1|\tilde{b}_i - b_i|}{\|a_i\|_2^2} \\
&= D_f^{\xi_k}(x_k,x^*) - \frac{||Ax_k - b||_2^2 - ||b^\delta - b||_2^2}{2||A||_F^2} \\
&\quad + \frac{2}{\|A\|_F^2} \sum_{i=1}^{m} \|a_i\|_1|\tilde{b}_i - b_i| \\
&\leq qD_f^{\xi_k}(x_k,x^*) + \frac{\delta^2}{2\|A\|_F^2} \\
&\quad + \frac{2}{\|A\|_F^2} \left(\sum_{i=1}^{m} \|a_i\|_1^2\right)^{1/2} \left(\sum_{i=1}^{m} |\tilde{b}_i - b_i|^2\right)^{1/2} \\
&\leq qD_f^{\xi_k}(x_k,x^*) + \frac{\delta^2}{2\|A\|_F^2} + \frac{2}{\|A\|_F^2}\|A\|_{1,2}\delta.
\end{aligned}$$

By taking the full expectation on both sides, it follows

$$\mathbb{E}[D_f^{\xi_{k+1}}(x_{k+1},x^*)] \leq q\mathbb{E}[D_f^{\xi_k}(x_k,x^*)] + \frac{\delta^2 + 4\|A\|_{1,2}\delta}{2||A||_F^2}.$$

Recursively using this equation we can obtain

$$\begin{aligned}
\mathbb{E}[D_f^{\xi_k}(x_k,x^*)] &\leq q^k D_f^{\xi_0}(x_0,x^*) + \frac{1}{1-q}\frac{\delta^2 + 4\|A\|_{1,2}\delta}{2\|A\|_F^2} \\
&= q^k f(x^*) + \frac{|x^*|_{\min} + 2\lambda}{|x^*|_{\min}} \cdot \frac{\delta^2 + 4\|A\|_{1,2}\delta}{\tilde{\sigma}^2(A)}.
\end{aligned}$$

□

Chapter 4

Numerical Experiments

Recall that this research aims to theoretically analyse convergence properties through existing randomised Kaczmarz method (RKM) to those further developed methods based on RKM, such as random sparse Kaczmarz method, and apply these algorithms, and compare their performances with that of the traditional Kaczmarz Method for solving systems of linear equations. In this chapter, we do numerical experiments to present and discuss the performances in solving linear equation systems using the methodologies described in Chapter 2 and Chapter 3. Here we do not include the method for solving linear inequalities, because of lacking popular comparison benchmark. Furthermore, we conduct a comparison of RKM with CGLS, which is a well-known iterative method, performing well for solving linear equation systems. Here we adopt and modify the CGLS algorithm Matlab code from Saunders [33].

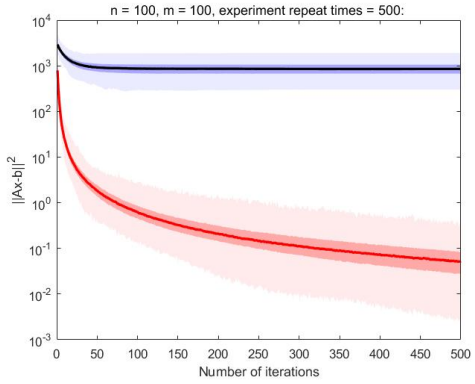
4.1 Comparison between KM and RKM

This section presents and discusses the results of the methods discussed in Chapter 2 for solving linear system $Ax = b$ for x, where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ is the exact data. We generate matrix A and x respectively by normal distributed random numbers.

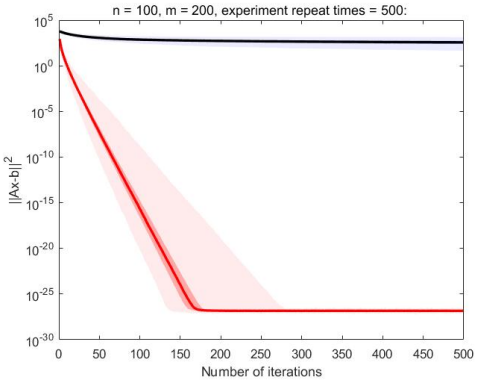
4.1.1 Exact Data

We build our exact data b by calculating Ax. We then input A and b into Algorithm 1 and Algorithm 2, comparing their residuals $||Ax_k - b||_2^2$ at each iteration. The results of numerical experiments are shown in the following Figure

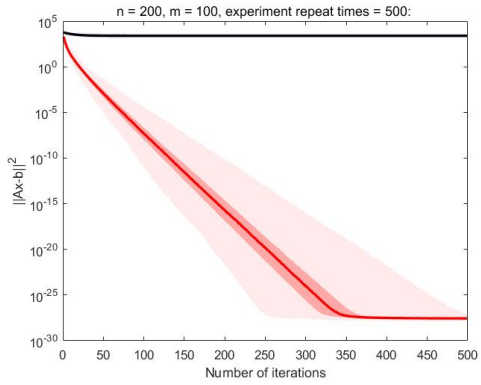
4.1.



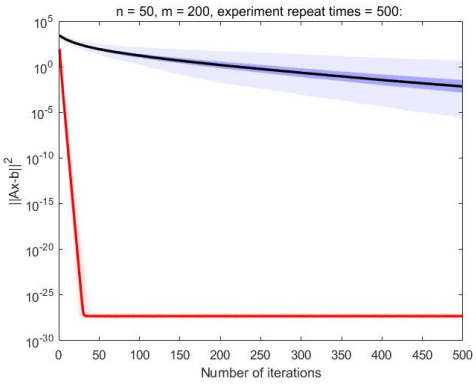
(a) n = 100, m = 100



(b) n = 100, m = 200



(c) n = 200, m = 100



(d) n = 50, m = 200

Figure 4.1: Comparison for KM and RKM with exact data for different n,m

In Figure 4.1, blue lines are the residuals of iterations of KM, while red lines are the residuals of iterations of RKM. The lighter areas of each colour represent all values of residual ranges, while the darker areas around the lines are ranges in the quantiles.

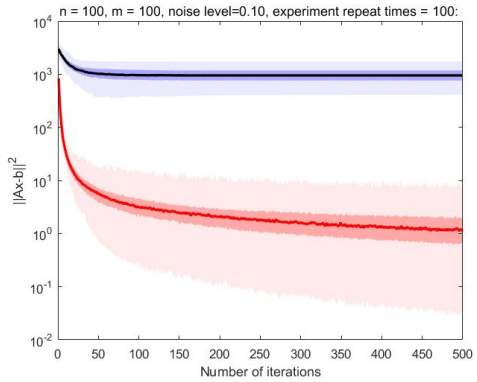
Here we choose four different scenarios for demonstration, that is, the sizes of matrices are: square random matrix of 100 rows and 100 columns in Figure 4.1(a), underdetermined system with matrix of 100 rows and 200 columns shown in Figure 4.1(c), and two overdetermined systems with matrix of 100 columns and 200 rows shown in Figure 4.1(b), another one with 50 columns shown in Figure 4.1(d).

From Figure 4.1, we can see that without noise, RKM always has better performance compared with KM. For $n = m$ case, RKM has converged much

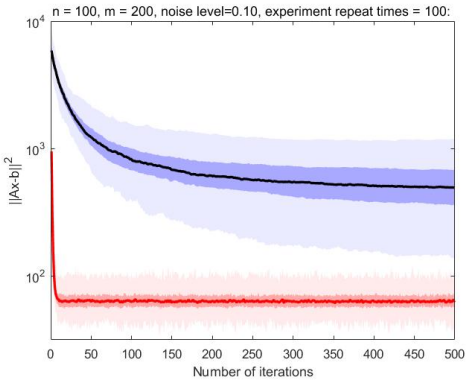
slower than underdetermined case $n > m$, while RKM is specifically working well for overdetermined systems, that is $m >> n$. As m goes up, it converges faster, and the residuals are much smaller, as shown in Figure 4.1(b).

4.1.2 Noisy Data

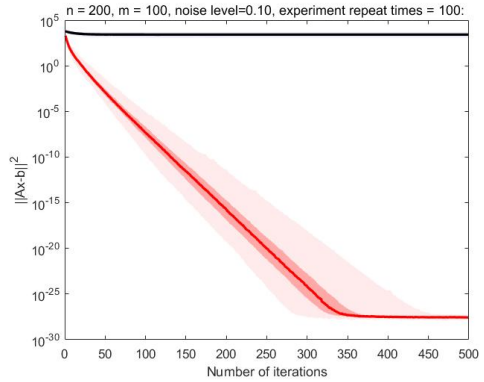
For noisy data case, we add noise constructed as a normalised normal distributed random vector proportion to a given noise level. We denote noise as ϵ, and input A and $\tilde{b} := b + \epsilon$ into Algorithm 1 and Algorithm 2, comparing their residuals $||Ax_k - \tilde{b}||_2^2$ for each iteration. The results of the noised numerical experiments are shown in the following Figure 4.2.



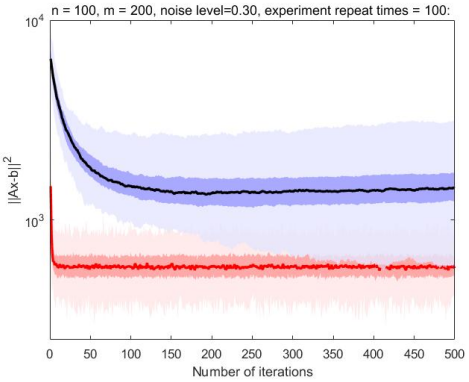
(a) n = 100, m = 100, noise level = 0.1



(b) n=100, m=200, noise level = 0.1



(c) n = 200, m = 100, noise level = 0.1



(d) n = 100, m = 200, noise level = 0.3

Figure 4.2: Comparison for KM and RKM with noise for different n,m, and different noise level

In Figure 4.2, the red lines represents RKM, and the blue lines represent KM. The lighter areas of each colour represent all values of residual ranges, while the

darker areas around the lines are ranges in the quantiles. Here we choose noisy level as 10% shown in Figure 4.2(a),(b) and (c), while we choose a bit larger noisy level, 30%, shown in Figure 4.2(d).

It can be directly seen that RKM has much better performance compared with KM in the perturbed systems. To be specific, RKM still converges fast in overdetermined systems, but the residuals remain very large as shown in Figure 4.2(b), relatively to the underdetermined system in Figure 4.2(c). Also, as the relative noise level goes up, the residuals of RKM becomes very large and tend to be similar to KM in the overdetermined system.

4.2 Comparison between RKM and RSKM

When facing a sparse problem of linear equations, we will check performances of both RKM and RSKM. We will generate a random sparse vector x and a random Gaussian matrix A to calculate the vector b with add-on noise level for our input at each time of the experiment. The results of experiments inputting exact b and noised b^{δ} are shown in the following Figure 4.3.

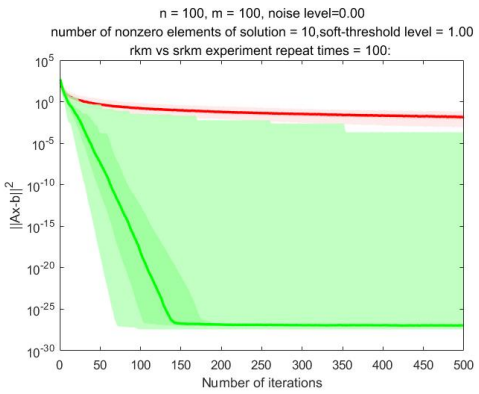
In both Figure 4.3 and 4.4, the green lines represents residuals of RSKM and red lines represent residuals of RKM when recovering sparse solutions. The lighter areas of each colour represent all residual value ranges respectively, while the darker areas around the lines are ranges in the quantiles.

In Figure 4.3, we choose to generate random matrices with 100 rows and 100 columns in Figure 4.3 (a) (b), and 100 rows and 200 columns in Figure 4.3 (c) (d). It is obvious to see that RSKM does not perform well extremely when solving the underdetermined sparse solution problem, while RKM performs with no such drastic differences. Although RSKM converge slower and residuals of it are larger, we can see from Table 4.1, the iterations RSKM given have most of it entries zero, which make the results much more sparse than RKM.

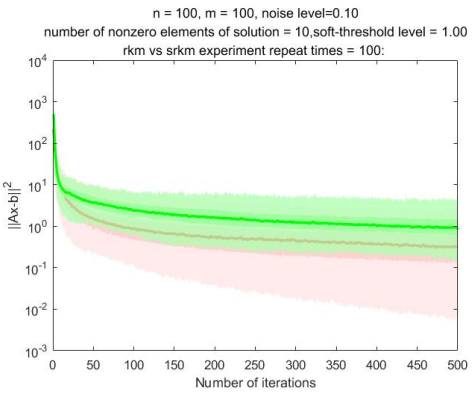
As for overdetermined sparse solution systems in Figure 4.4, we choose the random matrices with 200 rows and 100 columns in Figure 4.4 (a) (b), and 200 rows and 50 columns in Figure 4.4 (c) (d).

We can see it converges faster than RKM, but the residuals of RSKM converging to is a little bit larger than RKM. At the same time, we can observe that RSKM has huge variances as shown by the green areas representing quantiles. This may be due to the utilisation of the soft-threshold function.

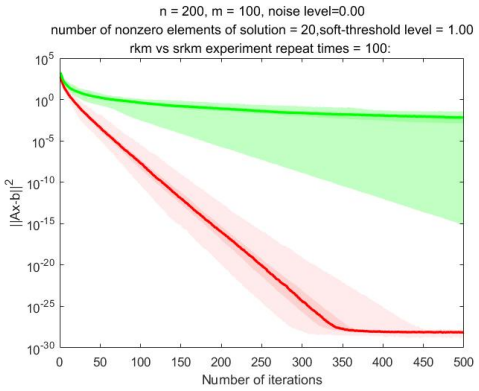
Additionally, Figure 4.4 (b) and 4.4 (d) shows RSKM does converge to bet-



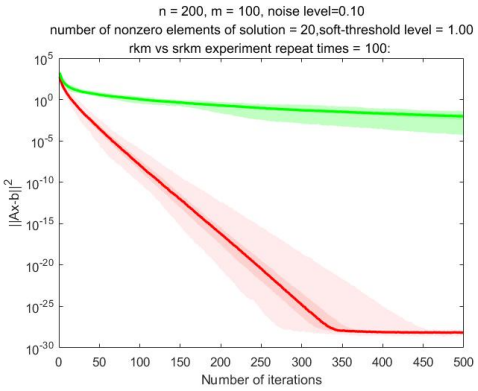
(a) n = 100, m = 100, without noise



(b) n = 100, m = 100, noise level = 0.1



(c) n = 200, m = 100, without noise

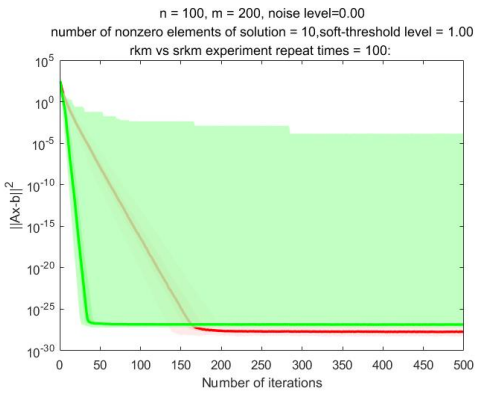


(d) n = 200, m = 100, noise level = 0.1

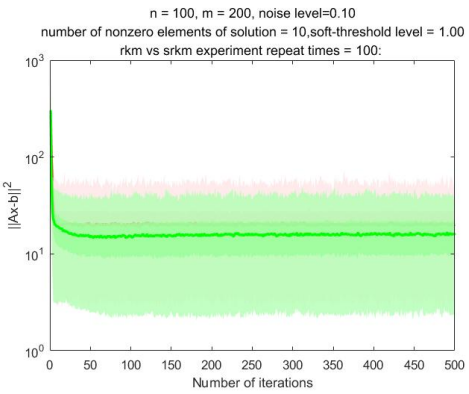
Figure 4.3: Comparison for RKM and RSKM for equal n,m, and overdetermined system with 10% of non-zero element in x, for both exact and noisy data

n=20,m=5, underdetermined system without noise with the sparse solution of 2 non-zero entries after 200 iterations soft-threshold level in RSKM = 1.0					
experiment no.1 result		experiment no. 2 result		experiment no.3 result	
RKM	RSKM	RKM	RSKM	RKM	RSKM
-0.03	-0.06	0.09	0.00	-0.00	0.00
-0.04	-0.00	-0.00	0.00	0.09	0.00
-0.02	-0.00	-0.24	-0.17	-0.02	-0.00
-0.01	-0.00	0.42	0.58	-0.12	-0.14
-0.03	-0.00	0.04	0.00	-0.11	-0.00
0.01	-0.00	0.08	0.00	0.06	0.00
-0.04	-0.00	-0.12	-0.00	-0.07	-0.00
-0.00	-0.00	0.16	0.00	0.01	-0.00
-0.07	-0.19	0.25	0.04	-0.02	-0.00
0.03	0.00	-0.31	-0.23	-0.15	-0.00
0.03	0.00	-0.15	-0.00	0.23	0.64
0.07	0.03	-0.08	-0.00	-0.03	-0.00
-0.01	-0.00	0.02	0.00	-0.04	-0.00
-0.06	-0.08	0.29	0.30	0.17	0.00
0.01	0.00	-0.42	-0.24	0.03	0.00
-0.01	-0.00	-0.52	-1.04	0.08	0.00
0.00	-0.00	-0.10	-0.00	0.03	0.00
-0.02	-0.00	0.02	-0.00	0.02	-0.00
0.04	0.08	-0.27	-0.09	0.06	0.00
-0.04	-0.00	-0.14	-0.00	0.00	-0.00

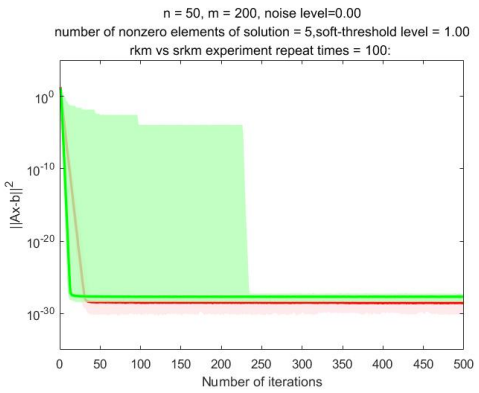
Table 4.1: Comparison of RKM and RSKM in underdetermined systems



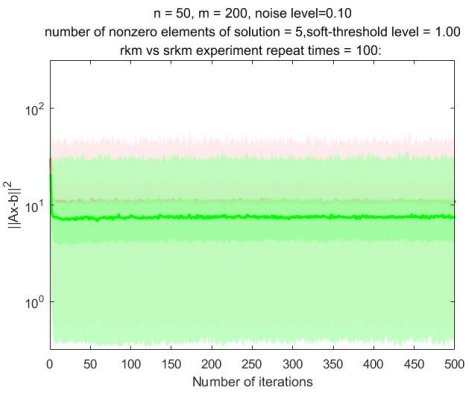
(a) n = 100, m = 200, without noise



(b) n = 100, m = 200, noise level = 0.1



(c) n = 50, m = 200, without noise



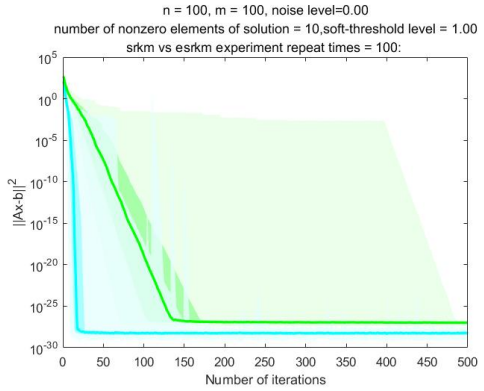
(d) n = 50, m = 200, noise level = 0.1

Figure 4.4: Comparison for RKM and RSKM for different n,m, in overdetermined systems, with 10% of non-zero element in x, for both exact and noisy data

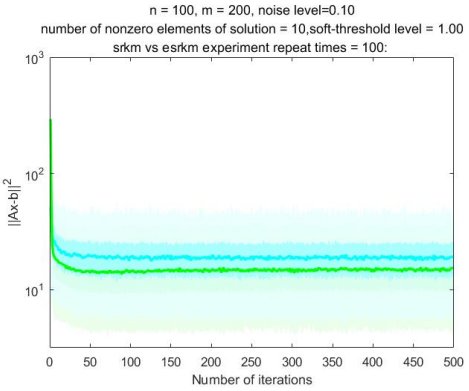
ter iteration comparing with RKM, though both of their residuals remain large. From here we can see that it is not very suitable for solving the perturbed systems. Meanwhile, we can observe from Figure 4.3(b) and 4.3(d) that RSKM did not do well with noise perturbations when associated with square matrices and underdetermined matrices.

4.3 Comparison between RSKM and ERSKM

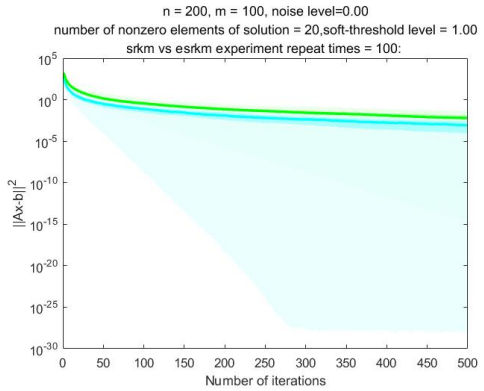
In previous Chapter 3, we develop ERSKM by using the properties of Bregman projection. In this section, we apply the comparison experiments similar to the previous section. The numerical experiments results are shown in Figure 4.5.



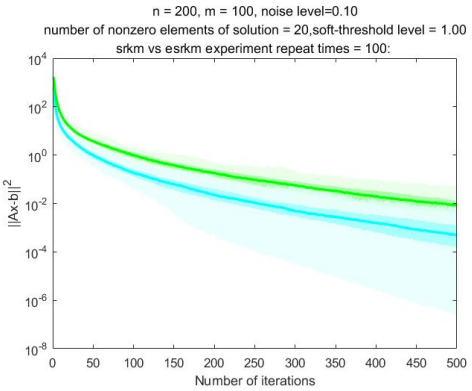
(a) n = 100, m = 100, without noise



(b) n = 100, m = 100, noise level = 0.1



(c) n = 200, m = 100, without noise

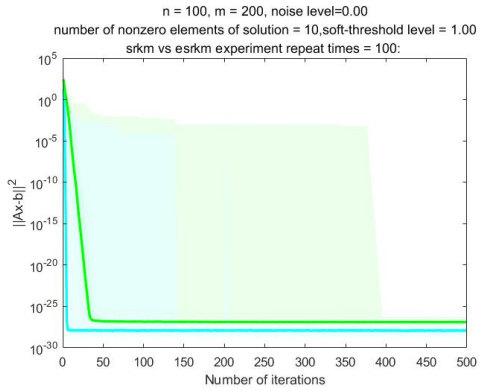


(d) n = 200, m = 100, noise level = 0.1

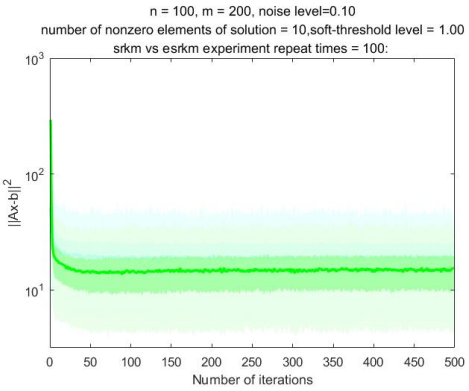
Figure 4.5: Comparison for ERSKM and RSKM for equal n,m, and overdetermined system with 10% of non-zero element in x, for both exact and noisy data

In both Figure 4.5 and 4.6, where we choose the four scenarios same as be-

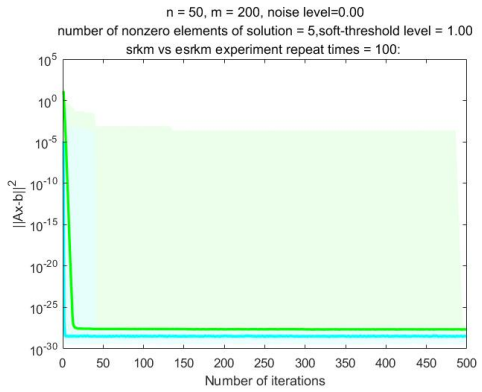
fore, the cyan lines represent the residuals of ERSKM iterates, while the green lines represent the residuals of RSKM iterates. The lighter areas of each colour represent all residual value ranges respectively, while the darker areas around the lines are ranges in the quantiles.



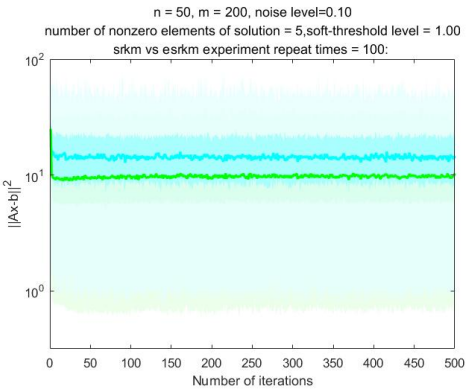
(a) n = 200, m = 100, without noise



(b) n = 200, m = 100, noise level = 0.1



(c) n = 50, m = 200, without noise



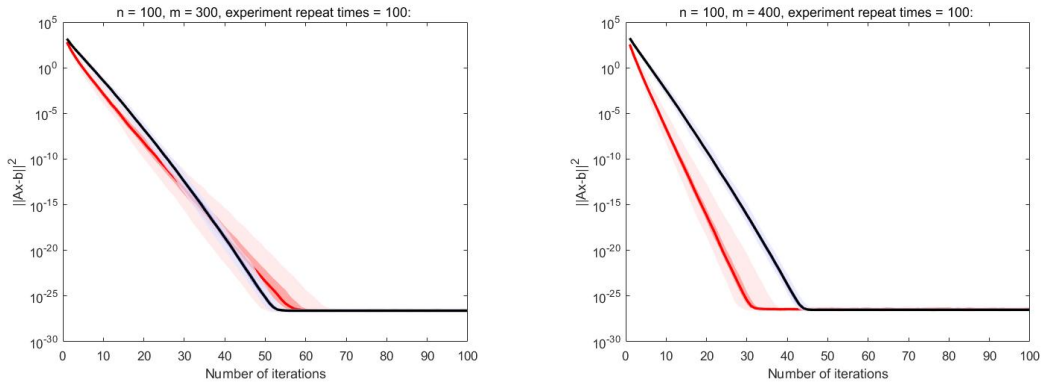
(d) n = 50, m = 200, noise level = 0.1

Figure 4.6: Comparison for RKM and RSKM for different n,m, in overdetermined systems, with 10% of non-zero element in x, for both exact and noisy data

We can see from Figure 4.5 and 4.6, ERSKM does a much better job compared with RSKM in all systems without perturbations. Comparing the figures, we observed that as the m/n goes up, the advantage of ERSKM is dropping, while from Figure 4.5(b)(d) and 4.6(b)(d), the performance shown in noisy cases of ERSKM are not consistently efficient.

4.4 Comparison between CGLS and RKM

Similarly, as previous sections generating methods, we compare RKM with CGLS method. CGLS is a widely used algorithm for solving linear systems. The Matlab code of CGLS is modified from Saunders [33]. The comparison experiments results are shown in the following Figure 4.7.



(a) n = 100, m = 300 (b) n = 100, m = 400

Figure 4.7: Comparison for RKM and CGLS for different n,m

In Figure 4.7, the dark blue lines represent residuals of CGLS and red lines represent residuals of RKM when solving linear systems of equations. The lighter areas of each colour represent all residual value ranges respectively, while the darker areas around the lines are ranges in the quantiles.

We can see that from [3] CGLS has its convergence rate as

$$||x_k - x||_{A^*A} \leq 2||x_0 - x||_{A^*A}(\frac{k(A)-1}{k(A)+1})^k,$$

from which [39] derive that the complexity of CGLS is about $\frac{4nm}{\log(n/m)} \log \frac{2}{\epsilon}$, while the complexity of RKM is approximately $\frac{2n^2m}{m+n-2\sqrt{mn}} \log \frac{1}{\epsilon}$.

Figure 4.7 clearly ascertains that RKM has better performance when $m/n > 3$, where it converges faster than CGLS.

Chapter 5

Conclusion

This thesis aimed to analyse the convergence related properties of randomised Kaczmarz method, accelerated version, randomised Kazcmarz method for solving linear inequalities, randomised sparse Kaczmarz method and exact-step randomised sparse Kaczmarz method by proofs and convex analysis, and to assess how RKM for solving linear systems associated with large scale matrices compares with the traditional Kaczmarz Method, and how RKM performs when the problem is sparse compared with RSKM. Also, the comparison between RKM and well-known iterative method, CGLS, is given out. Unlike the traditional iterative method, KM, we examined our randomised algorithm by analysing the expectation of error. For experiments, we implemented the methodologies by random matrices and vectors, sparse versions, and noised add-on. In conclusion, we view the RKM as a better iterative method to the overdetermined linear systems in general.

5.1 Summary

The KM method is our primary iterative method for solving linear systems of equations because it is commonly applied in the area of Computed Tomography (CT). The KM is a cyclic process of finding projections through each hyperplane defined by rows of the matrix. As an iterative method, it has an advantage in solving the systems that need modification, compared with direct methods. Most importantly, it has evolved into the randomised version, which is a tremendous enhancement. For this thesis, we construct the randomised Kaczmarz method by introducing a probability distribution assigned to the choosing of the indices. This is unlike the KM method, which only sweeps the rows in a certain order causing

drastically different from changing orders of rows. RKM surpasses KM because of its fast converging rate compared with KM, and this has been theoretically analysed in this thesis. Applying it on noised data, we have analysed that our RKM method also converges. After that, we extend RKM onto solving systems of linear inequalities based on Hoffman's theorem[16].

When the system is overdetermined (number of rows of the associated matrix is larger than the number of columns), this common situation appeared in many real-world problems such as Computed Tomography; we expect our RKM converges exponentially and the performance is much better when compared with solving underdetermined systems. Also, with noised data, it is more prominent.

When the solution is sparse (entries are mainly zeros), this also are encountered in many engineering problems. We construct a randomised sparse Kaczmarz method (RSKM) by using soft-threshold function and build exact-step randomised sparse method (ERSKM) by using the properties of Bregman projection. Through theoretical analysis and computation experiments, we show that RSKM converges better than RKM when solving sparse problems.

Overall, we can view RKM, RSKM and ERSKM as good iterative algorithms in solving systems of linear equations, especially those with sparse solutions. As the size of matrices going up and overdetermined, our methodologies remain quite efficient performance, even compared with CGLS when solving some specific linear systems.

5.2 Limitations and Future Research Directions

For this study, we planned to examine RKM and the methodologies developing from RKM, and RSKM that specialised on systems with sparse solution, about their convergence properties of both exact situations and noisy situations. Furthermore, we implemented the algorithms for numerical experiments. The results may seem pleasing, however, for noisy data, we find it has a relatively disappointing result of converging point suggested in Chapter 4 of applying both RKM and RSKM, as the residuals remain large. Future improvement can be made to further develop the algorithms in noisy cases.

We observed the noteworthy properties and numerical results of those based on RKM, whereas we could also adopt the idea behind applying it further not only on solving systems of linear equations but onto those non-linear optimisation problems as well. Recall our basic idea is an improvement from cyclically

computing iterates to randomised computing iterates.

Let's have a look on an optimisation problem, that is, minimising the function $f(x) := f(x_1, \cdots, x_n)$ on feasible set $\{x := (x_1, \cdots, x_n) | a_i \leq x_i \leq b_i, \quad i \in \{1, \cdots, n\}\}$. In order to find the minimiser, we can use Coordinate-wise descent method [44], which is minimising $f(x)$ along only one coordinate cyclically at each iterate. The iterates are as follows.

$$\begin{aligned} x_1^{k+1} &= x_1^k - \alpha_k \nabla_1 f(x_1^k, x_2^k, \cdots, x_n^k) \\ x_2^{k+1} &= x_2^k - \alpha_k \nabla_2 f(x_1^{k+1}, x_2^k, \cdots, x_n^k) \\ &\cdots \end{aligned}$$

This kind of methods can be generalised into alternating minimisation methods, where the iterates are generally in the following form.

$$\begin{aligned} x_1^{k+1} &= \arg\min f(x_1^k, x_2^k, \cdots, x_n^k) \\ x_2^{k+1} &= \arg\min f(x_1^{k+1}, x_2^k, \cdots, x_n^k) \\ &\cdots \end{aligned}$$

The alternating minimisation has been developed into a wide range of methods, as Expectation Maximisation (EM) method, an important algorithm applying in the statistical inference aimed at finding the maximum of log likelihood function. The iterates of EM method is based on finding expectation of each maximum on each coordinate [37]. Additionally, an arrays of methods develop through alternating minimisation, such as alternating proximal method and alternating direction method of multipliers (ADMM).

From this, by applying our basic idea from RKM, further research leaves to be done to construct randomised versions of those methods by examining convergence properties and numerical computing results.

Bibliography

[1] Ansorge, R. (1984). Connections between the Cimmino-method and the Kaczmarz-method for the solution of singular and regular systems of equations. *Computing*, 33(3-4), 367-375.

[2] Bertsekas, D. P. (2009). *Convex optimization theory* (pp. 157-226). Belmont: Athena Scientific.

[3] Bjorck, A. (1996). *Numerical methods for least squares problems* (Vol. 51). Siam.

[4] Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[5] Brooks, M. (2010). *A survey of algebraic algorithms in computerized tomography* (Doctoral dissertation, UOIT).

[6] Candes, E. J., Romberg, J. K., & Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 59*(8), 1207-1223.

[7] Czaja, W., & Tanis, J. H. (2013). Kaczmarz algorithm and frames. *International Journal of Wavelets, Multiresolution and Information Processing, 11*(05), 1350036.

[8] Duff, I. S., Erisman, A. M., & Reid, J. K. (2017). *Direct methods for sparse matrices*. Oxford University Press.

[9] Edelkamp, S., & Schroedl, S. (2011). *Heuristic search: theory and applications*. Elsevier.

[10] Eldar, Y. C., & Needell, D. (2011). Acceleration of randomized Kaczmarz method via the Johnson-Lindenstrauss lemma. *Numerical Algorithms, 58*(2), 163-177.

[11] Forsythe, G. E. (1953). Solving linear algebraic equations can be interesting. *Bulletin of the American Mathematical Society, 59*(4), 299-329.

[12] Friedlander, M. P., & Tseng, P. (2007). Exact regularization of convex programs. *SIAM Journal on Optimization, 18*(4), 1326-1350.

[13] Greenbaum, A. (1997). *Iterative methods for solving linear systems* (Vol. 17). Siam.

[14] Herman, G. T. Image reconstruction from projections: the fundamentals of computerized tomography. 1980. *New York, Academic*.

[15] Higham, N. J. (2008). *Functions of matrices: theory and computation* (Vol. 104). Siam.

[16] Hoffman, A. J. (2003). On approximate solutions of systems of linear inequalities. In *Selected Papers Of Alan J Hoffman: With Commentary* (pp. 174-176).

[17] Huang, J., & Li, Y. (2016). Advanced sparsity techniques in magnetic resonance imaging. In *Machine Learning and Medical Imaging* (pp. 183-236). Academic Press.

[18] Jiao, Y., Jin, B., & Lu, X. (2017). Preasymptotic convergence of randomized Kaczmarz method. *Inverse Problems, 33*(12), 125012.

[19] Jin, Q., & Wang, W. (2013). Landweber iteration of Kaczmarz type with general non-smooth convex penalty functionals. *Inverse Problems, 29*(8), 085011.

[20] Kaczmarz, S. (1993). Approximate solution of systems of linear equations. *International Journal of Control, 57*(6), 1269-1271.

[21] Kwapien, S., & Mycielski, J. (2001). On the Kaczmarz algorithm of approximation in infinite-dimensional spaces. *Studia Mathematica, 1*(148), 75-86.

[22] Lai, M. J., & Yin, W. (2013). Augmented ℓ_1 and nuclear-norm models with a globally linearly convergent algorithm. *SIAM Journal on Imaging Sciences, 6*(2), 1059-1091.

[23] Leventhal, D., & Lewis, A. S. (2010). Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research, 35*(3), 641-654.

[24] Liu, J., & Wright, S. (2016). An accelerated randomized Kaczmarz algorithm. *Mathematics of Computation, 85*(297), 153-178.

[25] Lorenz, D. A., Schöpfer, F., & Wenger, S. (2014). The linearized Bregman method via split feasibility problems: analysis and generalizations. *SIAM Journal on Imaging Sciences, 7*(2), 1237-1262.

[26] McCormick, S. F. (1977). The methods of Kaczmarz and row orthogonalization for solving linear equations and least squares problems in Hilbert space. *Indiana University Mathematics Journal, 26*(6), 1137-1150.

[27] Needell, D. (2010). Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics, 50*(2), 395-403.

[28] Needell, D., Ward, R., & Srebro, N. (2014). Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Advances in Neural Information Processing Systems* (pp. 1017-1025).

[29] Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course* (Vol. 87). Springer Science & Business Media.

[30] Rockafellar, R.T., Wets, R.J.B. (2009). *Variational analysis*. Berlin: Springer

[31] Rockafellar, R.T. (1970). *Convex analysis* (Vol. 28). Princeton university press.

[32] Saad, Y. (2003). *Iterative methods for sparse linear systems* (Vol. 82). siam.

[33] Saunders, M. (2015). CGLS: CG method for Ax = b and Least Squares Avaliable from: http://web.stanford.edu/group/SOL/software/cgls/

[34] Schöpfer, F., & Lorenz, D. A. (2018). Linear convergence of the randomized sparse Kaczmarz method. *Mathematical Programming*, 1-28.

[35] Stoer, J. (1993). *Introduction to numerical analysis*, 2nd Ed, New York : Springer-Verlag

[36] Sugiyama, M. (2015). *Introduction to statistical machine learning*. Morgan Kaufmann.

[37] Sundberg, R. (1974). Maximum likelihood theory for incomplete data from an exponential family. *Scandinavian Journal of Statistics*, 49-58.

[38] Strang, G. (2016). *Introduction to Linear Algebra*, 5th Ed, Wellesley - Cambridge Press, ISBN 978-0-9802327-7-6

[39] Strohmer, T., & Vershynin, R. (2009). A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications, 15*(2), 262.

[40] Sznajder, R. (2016). Kaczmarz algorithm revisited. *Czasopismo Techniczne*.

[41] Tompkins, C. (1949). Projection methods in calculation of some linear problems. In *BULLETIN OF THE AMERICAN MATHEMATICAL SOCIETY* (Vol. 55, No. 5, pp. 520-520). 201 CHARLES ST, PROVIDENCE, RI 02940-2213: AMER MATHEMATICAL SOC.

[42] Yin, W. (2010). Analysis and generalizations of the linearized Bregman method. *SIAM Journal on Imaging Sciences, 3*(4), 856-877.

[43] Zalinescu, C. (2002). *Convex analysis in general vector spaces*. World scientific.

[44] Zangwill, W. I. (1969). *Nonlinear programming: a unified approach* (Vol. 196, No. 9). Englewood Cliffs, NJ: Prentice-Hall.