Row-Action Methods for Massive Inverse Problems

J. Tanner Slagel

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

Julianne Chung, Co-chair
Matthias Chung, Co-chair
Serkan Gugercin
Youssef M. Marzouk
Luis Tenorio

May 15, 2019
Blacksburg, Virginia

Keywords: inverse problems, Tikhonov regularization, row-action methods, Kaczmarz methods

Row-Action Methods for Massive Inverse Problems

J. Tanner Slagel

(ABSTRACT)

Numerous scientific applications have seen the rise of massive inverse problems, where there are too much data to implement an all-at-once strategy to compute a solution. Additionally, tools for regularizing ill-posed inverse problems are infeasible when the problem is too large. This thesis focuses on the development of row-action methods, which can be used to iteratively solve inverse problems when it is not possible to access the entire data-set or forward model simultaneously. We investigate these techniques for linear inverse problems and for separable, nonlinear inverse problems where the objective function is nonlinear in one set of parameters and linear in another set of parameters. For the linear problem, we perform a convergence analysis of these methods, which shows favorable asymptotic and initial convergence properties, as well as a trade-off between convergence rate and precision of iterates that is based on the step-size. These row-action methods can be interpreted as stochastic Newton and stochastic quasi-Newton approaches on a reformulation of the least squares problem, and they can be analyzed as limited memory variants of the recursive least squares algorithm. For ill-posed problems, we introduce sampled regularization parameter selection techniques, which include sampled variants of the discrepancy principle, the unbiased predictive risk estimator, and the generalized cross-validation. We demonstrate the effectiveness of these methods using examples from super-resolution imaging, tomography reconstruction, and image classification.

Row-Action Methods for Massive Inverse Problems

J. Tanner Slagel

(GENERAL AUDIENCE ABSTRACT)

Numerous scientific problems have seen the rise of massive data sets. An example of this is super-resolution, where many low-resolution images are used to construct a high-resolution image, or 3-D medical imaging where a 3-D image of an object of interest with hundreds of millions voxels is reconstructed from x-rays moving through that object. This work focuses on row-action methods that numerically solve these problems by repeatedly using smaller samples of the data to avoid the computational burden of using the entire data set at once. When data sets contain measurement errors, this can cause the solution to get contaminated with noise. While there are methods to handle this issue, when the data set becomes massive, these methods are no longer feasible. This dissertation develops techniques to avoid getting the solution contaminated with noise, even when the data set is immense. The methods developed in this work are applied to numerous scientific applications including super-resolution imaging, tomography, and image classification.

Dedication

To Lou Ann, Zach, Ashley, Koehler, Bear, & Wu Zhao

(in no particular order)

Acknowledgments

I am grateful for the support I have received from faculty, friends, and family during the composition of this work. First, I would like to thank Julianne Chung and Tia Chung, who have endlessly supported me throughout my graduate research experience. You all took extra time to help me develop as a mathematician, and for that I am a better researcher, presenter, writer, and person. There is no way I can repay you two for all you have done for me. I am truly thankful for the times we have had together.

I would also like to thank the other members of my committee. To Serkan Gugercin, who taught my first two semesters of Numerical Analysis. You are a great teacher, and I have always valued your questions and comments regarding my work. To Luis Tenorio and Youssef Marzouk, I thank you both for serving on my committee externally.

Thank you to the faculty, staff, graduate students, and undergraduate students in the Department of Mathematics at Virginia Tech for the experiences that helped shape my time as a Ph.D candidate. I have been so fortunate to be a part of this community of passionate researchers and teachers.

My professors at Berea College were the first people to show me that mathematics was something that was fun to do. To James Blackburn-Lynch, thank you for your contagious energy, and the hours you spent with me talking about mathematics and life. You went above and beyond the responsibilities of a professor, and I look back at the times we hung out as some of the best. To Larry Gratton thank you for your kindness, honesty, and humor in and out of the classroom. You gave me the blueprint for the type of mathematician I want to be.

I would also like to acknowledge the National Science Foundation for supporting my graduate research through the Division of Mathematical Science grants 1723005 and 1654175.

Last but not least I would like to thank my family, who have always supported me in following my interests and aspirations. To my parents Lou Ann and Zach, thank you for your continuous love and support. You have both worked tirelessly to provide me with the tools to build a meaningful life, and I'm grateful for the ways that you shaped the person I am. To my brother, sister, aunts, uncles, and grandparents, thank you all for the constant support as I journeyed through graduate school. I could not have asked for better community to grow up in, and be a part of.

Contents

List of Figures **xi**

List of Tables **xvi**

1 Introduction **1**

1.1 Mathematical Models 2

1.1.1 The Linear Inverse Problem 2

1.1.2 The Separable, Non-linear Inverse Problem 4

1.2 Outline 5

1.3 Overview of Contributions 6

2 Row-action Methods **8**

2.1 Definition and Notation 9

2.2 Sampling Methods 10

2.3 Step Sizes 11

2.4 The Kaczmarz Method 12

2.4.1 Consistent Linear System 12

2.4.2 Linear Least Squares 15

2.5 The Block Kaczmarz Method 18

2.6 The Damped Block Kaczmarz Method 19

2.7 Generalized Block Structure for Consistent Systems 20

2.8 Row-action Methods for the Tikhonov LS Problem 21

2.9 Other Row-action Methods 23

3 Stochastic Newton and Quasi-Newton methods 24

3.1 Stochastic Reformulation of the LS Problem 25

3.2 Stochastic Approximation Algorithms 26

3.2.1 Generalized Block Structure 26

3.2.2 Connection to Row-action Methods 29

3.2.3 Choice of Step Size 31

3.3 Almost Sure Convergence of the SN Method 32

3.4 Analysis of quasi-Newton Methods 36

3.4.1 Almost Sure Convergence 36

3.4.2 Convergence Rates 38

3.5 Numerical Experiments 41

3.5.1 Experiment 1: Asymptotic Behavior of Block Kaczmarz and `rrls` . . 41

3.5.2 Experiment 2: Convergence Behavior of `slimLS` 44

3.5.3 Experiment 3: Image Classification 46

3.6 Remarks and Future Directions 51

4 Sampled Tikhonov Regularization **54**

4.1 Problem Formulation 55

4.2 Full Memory Row-action Methods for Tikhonov Regularization 56

4.2.1 Random Sampling 59

4.2.2 Random Cyclic Sampling 62

4.2.3 An Illustration 63

4.3 Sampled Regularization Parameter Selection Methods 66

4.3.1 Sampled Discrepancy Principle 67

4.3.2 Sampled Unbiased Predictive Risk Estimator 68

4.3.3 Sampled Generalized Cross Validation 68

4.3.4 A Second Illustration 69

4.4 The `slimTik` Method 71

4.4.1 Limited Memory `sTik` 71

4.4.2 Connection to `slimLS` 74

4.4.3 A Third Illustration 76

4.5 Numerical Experiments 78

4.5.1 Experiment 4: `slimTik` Applied to X-ray Tomography 78

4.5.2 Experiment 5: Convergence of Sampled Regularization Methods . . . 80

4.5.3 Experiment 6: Super Resolution 83

4.6 Remarks and Future Directions 87

5 Extension to Separable, Non-linear Inverse Problems 91

5.1 Row-action Methods for Separable, Non-linear Inverse Problems 92

5.2 Numerical Results 94

5.2.1 Experiment 7: Comparing `sn-slimTik` to `variable projection` . . 96

5.2.2 Experiment 8: `sn-slimTik` for a Streaming Problem 100

5.3 Remarks and Future Directions 102

6 Conclusion 103

Bibliography 105

Appendices 123

Appendix A Proofs for Chapter 3 124

A.1 The Quasi-martingale Convergence Theorem 124

A.2 Almost Sure Convergence 125

A.3 Convergence Rates 131

Appendix B Derivation of the Sampled UPRE and GCV 141

B.0.1 Derivation of the Sampled UPRE 141

B.0.2 Derivation of the Sampled GCV 142

List of Figures

3.1 Error $\omega(\mu,\nu)$ of the stochastic Newton solution $\widetilde{\mathbf{x}}$ compared to $\widehat{\mathbf{x}}$. In the plot on the left, $\nu = 10$ and we vary μ. Notice that a pole exists at $\mu = 0$, where the relative error becomes arbitrarily large. The plot on the right illustrates the impact of varying ν for fixed $\mu = 1$. 35

3.2 Experiment 1: The top left panel contains relative errors for `rrls` iterates, computed as $\|\mathbf{x}_k - \mathbf{x}_{\text{LS}}\|/\|\mathbf{x}_{\text{LS}}\|$, where $\mathbf{x}_{\text{LS}}$ is the LS solution. The top right panel contains relative errors for block Kaczmarz iterates, computed as $\|\mathbf{x}_k - \widetilde{\mathbf{x}}\|/\|\widetilde{\mathbf{x}}\|$, where $\widetilde{\mathbf{x}}$ is defined in Theorem 3.1. Notice that we display 20,000 iterations for block Kaczmarz iterates and only 200 iterations for `rrls`. The bottom panel contains relative errors, $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$, for both `rrls` and the block Kaczmarz method. 43

3.3 Experiment 2: The top two plots contains relative errors for `rrls` and `slimLS` iterates for the first 200 and 100,000 iterations, computed as $\|\mathbf{x}_k - \mathbf{x}_{\text{LS}}\|/\|\mathbf{x}_{\text{LS}}\|$, where $\mathbf{x}_{\text{LS}}$ is the LS solution. The bottom plot relative errors for `rrls` and `slimLS` iterates for the first 100,000 iterations, computed as $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$, where $\mathbf{x}_{\text{true}}$ is the true solution. 45

3.4 Experiment 2: Relative errors for the `slimLS` iterates for the first 300 iterations for different values of constant step size α. The relative error is computed as $\|\mathbf{x}_k - \mathbf{x}_{\text{LS}}\|/\|\mathbf{x}_{\text{LS}}\|$, where $\mathbf{x}_{\text{LS}}$ is the LS solution. 46

3.5 Experiment 3: CPU times (median and 5^{th}–95^{th} percentiles) for solving LS problem (3.21) using `rrls`, global LSQR (Gl-LSQR), and the Moore-Penrose pseudoinverse for various numbers of training images m. 50

3.6 Expiroment 3: Classification (with corresponding accuracy) for the MNIST test images after training on 60,000 images, using different LS solvers. 52

4.1 Illustration of convergence behaviors of `rrls` and `sTik` iterates. Shown in the left panel are the true solution $\mathbf{x}_{\text{true}}$, the unregularized solution $\mathbf{x}(0)$, the Tikhonov solution $\mathbf{x}(\lambda)$, and `rrls` iterates after multiple epochs. Both `rrls` with random sampling iterates $\{\mathbf{y}_k^{\mathrm{r}}\}$ and `rrls` with random cyclic sampling iterates $\{\mathbf{y}_k^{\mathrm{c}}\}$ converge asymptotically to the unregularized solution. In the right panel, we provide `sTik` with random sampling iterates $\{\mathbf{x}_k^{\mathrm{r}}\}$ and confidence bounds. These iterates stay close to the Tikhonov solution. The axis for the right figure corresponds to the rectangular box in the left figure. The concentric gray circles represent the 95% confidence interval for these iterates after subsequent epochs. 64

4.2 "Asymptotic" behavior of the sampled regularization parameter selection methods for the `prolate` example. Corresponding regularization parameters computed using the full data are provided as horizontal lines for comparison. 70

4.3 Relative reconstruction errors of the sampled and full regularization methods for four test problems `prolate`, `baart`, `shaw`, and `gravity`. All solutions lie on the solid line, which corresponds to relative errors for Tikhonov solutions. Note that the UPRE and GCV estimation in the `prolate` and `baart` test problem underperform significantly and are therefore omitted. The relative errors for λ_{sUPRE} and λ_{DP} coincide in the `shaw` example. 72

4.4 Convergence of the `slimTik` method. The plot in the left panel contains the relative errors between the iterates after one epoch and the Tikhonov solution, for different memory levels. The plot in the right panel illustrates asymptotic convergence of the `slimTik` method for memory levels $r = 0, 2, 4, 6,$ and 8. For comparison we include relative errors for a sample gradient method. . . . 77

4.5 Experiment 4: Illustration of a 2-d parallel beam tomography set-up. 79

4.6 Experiment 4: The noisy sinogram from the parallel beam tomography problem. 80

4.7 Experiment 4: Error images the reconstructed images for the 2-D tomography example. Reconstructions correspond to `sg`, `sbK`, `sLBFGS` and `slimTik`. For comparisons we provide the error image after 100, 500, 1000, and 5,373 iterations. 81

4.8 Experiment 4: Comparing the relative error to the true images at each iteration of the `sg`, `sbK`, `sLBFGS` and `slimTik` algorithms. 82

4.9 Experiment 5: Comparison of relative reconstruction errors for `sg`, `sbK`, `slimTik`, and `sTik` iterates for `gravity` using various sampled regularization parameter selection methods for the first 10 iterations, i.e., one epoch. We compare sDP, sUPRE, and sGCV. The horizontal black line is the relative error corresponding to the optimal regularization parameter for the full problem, which is not feasible to obtain in practice. 84

4.10 Experiment 6: Super-resolution imaging example. On the left is the true high-resolution image, and on the right are three sample low-resolution images. The red-box corresponds to sub-images shown in Figure 4.12. 86

4.11 Experiment 6: Relative reconstruction errors for the super-resolution imaging example for one epoch. We note that sUPRE and sGCV produce good reconstructions. Additionally, `slimTik` produces a smaller relative reconstruction error, since it is using more curvature information. 88

4.12 Experiment 6: Sub-images of the reconstructed images for the super-resolution imaging example. Reconstructions correspond to `sg`, `sbK`, and `slimTik` with regularization parameter updates computed using sDP, sUPRE, and sGCV. For comparison, we provide reconstructions corresponding to no regularization, i.e., $\lambda = 0$. 89

5.1 Experiment 7: Super-resolution imaging example. The high-resolution (HR) image and a subimage corresponding to the yellow box are provided in (a) and (b) respectively. The subimage of one of the low-resolution (LR) images is provided in (c). 97

5.2 Experiment 7: Relative reconstruction error norms for the image $\mathbf{x}_k$ (left) and the nonlinear parameters $\mathbf{y}_k$ (right) for variable projection and `sn-slimTik` for various memory levels. Note that variable projection errors are only provided after every 100 iterations of `sn-slimTik`. 98

5.3 Experiment 7: Sub-images of `sn-slimTik` reconstructions for memory levels $r = 0, 1$, and 5 for iterates within the first two epochs of data access. 99

5.4 Experiment 8: Streaming super-resolution imaging example. The high-resolution (1,024 × 1,024) image is provided in (a), along with three of the low-resolution (64 × 64) images in (b)–(d). 100

5.5 Experiment 8: Relative reconstruction errors for both the linear (left) and nonlinear (right) parameters for the streaming data super-resolution problem. 101

5.6 Experiment 8: Sub-image of absolute error images for `sn-slimTik` reconstructions with different memory levels. 101

List of Tables

3.1 For various numbers of training images, we provide the mean and standard deviation for the relative reconstruction errors and the iteration counts for `rrls`. 50

Chapter 1

Introduction

The efficient computation of solutions to inverse problems is critical in many scientific applications. For example, x-ray tomography uses measurements of the intensity from x-rays passing through the human body to obtain an image of the internal bone structure. Another example is automated image classification, where a training set of images is used to create a model that classifies images outside of the training set. In addition to medical imaging and machine learning, inverse problems appear in applications such as geophysics, atmospheric science, astrophysics, and signal processing (see [4, 8, 78, 81, 121, 154, 160], and the sources therein).

An emerging challenge is finding a numerical solution to *massive* inverse problems, where the entire forward model or the observation data are not available all-at-once. This unavailability could be due to size, for example, in automated image classification where there are millions of training images [20] or 3-dimensional medical and scientific imaging where the number of voxels can be in the hundreds of millions [109, 113, 127]. Alternatively, the data may be unavailable all-at-once because it is being streamed. An example of this is super-resolution, where hundreds of low-resolution images are being collected in time and the goal is to reconstruct a high-resolution image with millions of unknown pixel values [41, 91].

When massive inverse problems are ill-posed, regularization must be introduced. An inverse problem is ill-posed if the solution is not unique, does not exist, or does not depend continuously on the observation data [75]. When the solution does not depend continuously on the observation data, small errors in the observations can result in substantial changes in the solution approximation. A regularization term is often included to remedy ill-posedness, but this often requires parameter tuning to balance the data-fit with the regularization [57, 77, 153]. When facing a massive inverse problem, the traditional tools to choose the correct regularization parameter are no longer feasible because they often require full access to the forward model.

This work focuses on two main thrusts of solving massive inverse problems–First, the analysis and implementation of *row-action* methods to numerically solve the enormous optimization problems, and second, the development of sampled regularization methods for finding an appropriate regularization parameter while performing row-action methods.

1.1 Mathematical Models

The implementation of row-action methods depends heavily on the underlying model of the inverse problem being solved. This section describes two models for massive inverse problems that are considered in this thesis. In both models, Tikhonov regularization is introduced as a tool to combat ill-posedness. We remark that there are other choices such as ℓ_1 and total variation regularization, but Tikhonov regularization is considered the most popular choice, and is thus the main regularization choice of this work [72].

1.1.1 The Linear Inverse Problem

Consider the linear inverse problem

$$\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon},$$

where $\mathbf{x}_{\text{true}} \in \mathbb{R}^n$ contains the unknown and desired input parameters, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a matrix describing the forward model, $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is random additive noise, and $\mathbf{b} \in \mathbb{R}^m$ contains the noisy output measurements. Here the linear inverse problem is assumed to be massive which means that $m, n \in \mathbb{R}$ are so large that a $\min\{m, n\} \times \min\{m, n\}$ matrix cannot fit in computer memory, or that the entries of the observation vector $\mathbf{b}$ are being streamed and thus not available all-at-once.

When the noise $\boldsymbol{\epsilon}$ is assumed to be have independent and identically distributed entries with mean zero, it is appropriate to find the minimizer of the linear least squares (LS) problem [27]

$$\min_{\mathbf{x}} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_2^2 ,$$

where $\left\| \cdot \right\|_2$ is the vector 2-norm defined as $\left\| \mathbf{y} \right\|_2 = \sqrt{\sum_{i=1}^m y_i}$, for $\mathbf{y} \in \mathbb{R}^m$ with entries $\{y_i\}_{i=1}^m$. The LS problem above can have infinitely many minimizers. If $\mathbf{A}$ is full column rank then $\mathbf{x}_{\text{LS}} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{b}$ is the unique minimizer, which is an unbiased estimator of $\mathbf{x}_{\text{true}}$ [107].

When the linear inverse problem is ill-posed, regularization must be introduced. The Tikhonov-regularized linear LS problem is given by

$$\min_{\mathbf{x}} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_2^2 + \lambda \left\| \mathbf{L}\mathbf{x} \right\|_2^2 , \tag{1.1}$$

where $\lambda > 0$ is the regularization parameter and $\mathbf{L} \in \mathbb{R}^{n \times n}$ the regularization matrix. $\mathbf{L}$ is often chosen to be the identity matrix, but it can be chosen to reflect prior knowledge about the solution $\mathbf{x}_{\text{true}}$ [8]. When $\begin{bmatrix} \mathbf{A}^\top & \mathbf{L}^\top \end{bmatrix}^\top$ is full column rank the unique solution is given by

$$\mathbf{x}_{\text{Tik}} = \left(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{L}^\top \mathbf{L}\right)^{-1} \mathbf{A}^\top \mathbf{b}.$$

Massive linear inverse problems of this form arise in applications such as 3-D tomography, image classification, support vector machines, compressed sensing, rigid body dynamics, and computer vision [47, 52, 92, 99, 127, 158, 165].

1.1.2 The Separable, Non-linear Inverse Problem

The separable, nonlinear inverse problem is given by

$$\mathbf{b} = \mathbf{A}(\mathbf{y}_{\text{true}})\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}\,, \tag{1.2}$$

where $\mathbf{x}_{\text{true}} \in \mathbb{R}^n$ contains the desired linear input parameters, $\mathbf{y}_{\text{true}} \in \mathbb{R}^p$ contains the desired nonlinear input parameters, $\mathbf{A}(\cdot) : \mathbb{R}^p \to \mathbb{R}^{m \times n}$ is a nonlinear operator describing the forward model, $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is random additive noise, and $\mathbf{b} \in \mathbb{R}^m$ contains the noisy output measurements. In this scenario, $\mathbf{x}_{\text{true}}$ and $\mathbf{y}_{\text{true}}$ contain the unknown parameters. As with the linear inverse problem, the separable nonlinear problem is assumed to be massive which means that $m, n \in \mathbb{R}$ are so large that is it not feasible to fit a $\min\{m, n\} \times \min\{m, n\}$ matrix in computer memory, or the entries of the observation vector $\mathbf{b}$ are being streamed and thus not available all-at-once.

Similar to the linear inverse problem, when the problem is ill-posed, regularization is needed to compute a reasonable solution. The Tikhonov-regularized optimization problem is of the form

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}, \mathbf{y}) = \left\|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\right\|_2^2 + \lambda \left\|\mathbf{L}\mathbf{x}\right\|_2^2\,, \tag{1.3}$$

where $\lambda > 0$ is the regularization parameter and $\mathbf{L} \in \mathbb{R}^{n \times n}$ the regularization matrix. Problem (1.3) is called separable since the objective function f is a linear function in $\mathbf{x}$ and a nonlinear function in $\mathbf{y}$. Separable nonlinear inverse problems are sometimes called partially separable [129].

Massive separable nonlinear inverse problems of this form arise in applications such as super-resolution image reconstruction, neural networks, biomedical system dynamics, and molecular imaging [38, 39, 64, 109, 133, 141, 162].

1.2 Outline

When an inverse problem is massive, the challenge is two-fold. First, numerical algorithms that can approximate the solution without requiring the full forward model at once must be introduced, analyzed, and effectively implemented. Second, numerical methods to find a regularization parameter that balance the data-fit with regularization must be developed and implemented despite the inverse problem's massive size. This dissertation addresses both of these challenges.

In Chapter 2, a thorough background on row-action methods is provided. In Chapter 3, novel row-action methods are introduced to solve the linear LS problem. Convergence theory is presented to show the benefits of these methods analytically. This analysis is extended to the Tikhonov LS problem in Chapter 4, and sampling techniques to update the regularization parameter are investigated and implemented. The row-action methods introduced in Chapters 3 and 4 are extended to the separable nonlinear inverse problem in Chapter 5. Numerical results are offered at the end of Chapters 3, 4, and 5 to illustrate the computational effectiveness of the methods in each chapter. Proofs of theorems in each of these chapters are provided in the Appendices. Concluding remarks are made in Chapter 6.

1.3 Overview of Contributions

The contributions of this work include developing numerical algorithms to solve and effectively regularize massive inverse problems, providing convergence analysis for these methods, and implementing the methods effectively on applications related to medical imaging, astrological imaging, and data science. Below is a detailed list of contributions made in this dissertation. The corresponding chapters are listed at the end of each description, and the citation to the relevant paper or manuscript is provided.

- **Linear Inverse Problems**
  - The linear LS problem is recast as a stochastic optimization problem. The connection between stochastic approximation methods and row-action methods allow for development of asymptotic convergence theory for some well-known row-action methods, including the Kaczmarz, block Kaczmarz, and damped block Kaczmarz algorithms. We show that these algorithms are stochastic Newton methods applied to the stochastic reformulation of the linear LS problem. Chapter 3, appears in [42, 144].
  - The row-action method `slimLS` is introduced, with appropriate convergence theory. This algorithm is shown to be more favorable than other methods due to the lack of bias in the asymptotic convergence, as well as the quicker convergence due to the use of more information from previous iterates. Chapter 3, appears in [144]
  - The algorithm `slimLS` is shown to be an extension of the damped block Kaczmarz method. Non-asymptotic convergence analysis for the damped block Kaczmarz shows an expected linear convergence rate. Bounds on the expected mean square error show the trade-off between convergence rate and precision of iterates, that

depends on step size. Chapter 3, appears in [144].

- The Tikhonov LS problem is recast as a stochastic optimization problem. Connections between `slimLS` and the recursive LS algorithm lead to the development of the `slimTik` method, which uses sampled regularization parameter selection methods to choose an appropriate regularization parameter. Chapter 4, appears in [143, 144].
- The recursive LS algorithm is shown to be a full memory variant of the `slimTik` algorithm. Convergence theory by epoch compares cyclic, random, and random without replacement sampling. Chapter 4, appears in [143].
- Sampled variants of regularization parameter selection methods including the discrepancy principle, the unbiased predictive risk estimator method, and the generalized cross validation are analyzed for massive inverse problems. Chapter 4, appears in [143].
- Efficient implementations of `slimLS` and `slimTik` are applied to massive inverse problems, including tomography, image classification, and super-resolution. Chapters 3 and 4, appears in [42, 143, 144].

- **Non-linear Separable inverse problems**
  - The separable, nonlinear Tikhonov LS problem is reformulated as a stochastic optimization problem, and `SlimTik` is extended to `nl-slimTik`, a row-action version of the variable projection method for separable, nonlinear Tikhonov problems. Chapter 5, appears in [43].
  - An efficient implementation of the `nl-slimTik` method is applied to a massive super-resolution problem. Chapter 5, appears in [43].

Chapter 2

Row-action Methods

The focus of this chapter is on row-action methods for solving a system of linear equations,

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.1}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^{m}$, for $m, n \in \mathbb{N}$. If there exists an $\mathbf{x} \in \mathbb{R}^{n}$ that satisfies (2.1), then the system of linear equations is *consistent*. When (2.1) is consistent, we denote the minimum norm solution to be $\mathbf{x}_{\text{true}}$, meaning $\mathbf{x}_{\text{true}}$ is the unique vector that satisfies (2.1) and has the property that if a vector $\mathbf{z} \in \mathbb{R}^{n}$ satisfies (2.1), then $\|\mathbf{x}_{\text{true}}\|_2 \leq \|\mathbf{z}\|_2$. Otherwise, we consider the linear LS problem,

$$\min_{\mathbf{x}} f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2\,, \tag{2.2}$$

for which there is always a minimum norm solution, namely $\mathbf{x}_{\text{LS}}$.

Row-action methods are known for their cheap computational cost and their ability to approximate solutions rapidly. Row-action methods have been used in a number of applications from signal processing [103], compressed sensing [124], artificial intelligence [97], medical imaging [123], geophysics [145], and game theory [128]. The methods discussed here will provide a foundation for the methods developed in Chapters 3 and 4 for solving massive linear inverse problems and eventually in Chapter 5 for solving massive separable nonlinear inverse problems. We begin by defining row-action methods in 2.1, with various choices of sampling methods and step sizes discussed in Sections 2.2 and 2.3 respectively. Section 2.4 introduces the Kaczmarz method, which is the most common and widely used row-action method. Extension of this method including the block Kaczmarz and damped block Kaczmarz methods are described in Sections 2.5 and 2.6. An extension of the block Kaczmarz method that allows for more general sampling for the consistent linear system is described in Section 2.7. We will review work done on row-action methods for solving the Tikhonov LS problem in 2.8 and end with a brief discussion of row-action methods in a more general optimization context 2.9. To the best of our knowledge, no row-action methods have been developed to solve the separable nonlinear inverse problem.

2.1 Definition and Notation

In the context of (2.1), a *row-action* method is defined as an iterative method that at each iteration

1. makes no change to the original matrix $\mathbf{A}$,
2. does not require any operations involving the entire matrix $\mathbf{A}$, and
3. uses only a selection of rows of $\mathbf{A}$ and $\mathbf{b}$.

The above definition is the same as the one in [30], except for the third criterion which allows each iteration to use a selection of rows of $\mathbf{A}$ and $\mathbf{b}$ instead of only one row of $\mathbf{A}$ and $\mathbf{b}$ at each iteration. In the following discussion, we denote the ith row of $\mathbf{A}$ and $\mathbf{b}$ as $\mathbf{a}_i$ and $\mathbf{b}_i$ respectively, and we denote a block of rows of $\mathbf{A}$ and the corresponding entries in $\mathbf{b}$ as $\mathbf{A}_i$ and $\mathbf{b}_i$ respectively. Unless otherwise mentioned, the blocks are assumed to correspond to the following partition of $\mathbf{A}$ and $\mathbf{b}$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_M \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{bmatrix}, \tag{2.3}$$

where $M \in \mathbb{N}$ is chosen such that, for each i such that $1 \leq i \leq M$, $\mathbf{A}_i \in \mathbb{R}^{\ell \times n}$ where $\ell = m/M \in \mathbb{N}$.

Given an initial vector $\mathbf{x}_0 \in \mathbb{R}^n$, row-action methods are iterative methods that take the form

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{s}_k \left(\mathbf{x}_{k-1}\right), \tag{2.4}$$

where $\alpha_k > 0$ is the step size and $\mathbf{s}_k$ can depend on the current and previously sampled rows. The next two sections describe common choices of sampling strategies and different methods for determining the step size.

2.2 Sampling Methods

At the kth iteration, a row-action method described in (2.4) uses a new selection of rows $\mathbf{A}_{\tau(k)}$ and $\mathbf{b}_{\tau(k)}$. Here the function $\tau\left(k\right)$ represents the selection strategy at the kth iteration. We describe four sampling methods.

- *Cyclic* sampling sweeps through the blocks in order, by setting $\tau(k) = (k-1 \mod M)+1$. After every M iterations, every block has been visited the same number of times and in the same order.

- *Random uniform* sampling chooses a random block at each iteration, by setting $\tau(k)$ to the random uniform variable on the set $\{1, \dots M\}$. In Chapter 4 this is referred to as *sampling without replacement.*

- *Random non-uniform* sampling chooses a random block based on its relative size to other blocks. Here $\tau(k)$ is the random variable such that $p(\tau(k) = i) = \frac{\|\mathbf{A}_i\|_{\mathrm{F}}}{\|\mathbf{A}\|_{\mathrm{F}}}$ for each $i \in \{1, \dots M\}$, where $\| \cdot \|_{\mathrm{F}}$ denotes the Frobenius norm.

- *Random cyclic* sampling is an extension of sampling without replacement that allows access to blocks more than once. This is done by, for each $j \in \mathbb{N}$, setting $\{\tau(k)\}_{k=jM+1}^{(j+1)M}$ to a random permutation on the set $\{1, \dots, M\}$. This way after every M iterations, every block has been visited the same number of times in random order.

Most of the work on row-action methods use cyclic or random uniform sampling, with a few results utilizing random non-uniform sampling. Random cyclic sampling, or sampling without replacement, has been noted as an area that needs more development [149], and we give some attention to this in Chapter 4.

2.3 Step Sizes

The sequence of step sizes $\{\alpha_k\}$ can take various forms. In this work, we consider two options.

- *Constant* step size where $\alpha_k = \alpha$ for some $\alpha > 0$. Here, α is called the relaxation parameter. When $\alpha > 1$ this is called over-relaxation, and when $\alpha < 1$ this is called under-relaxation.

- *Decaying* step size chooses a sequence $\{\alpha_k\}$ such that $\alpha_k \overset{k\to\infty}{\longrightarrow} 0$. A decaying step size often helps to guarantee asymptotic convergence of a row-action method, but it will slow convergence.

2.4 The Kaczmarz Method

The first row-action method is accredited to Stephan Kaczmarz in 1937 [111, 150]. Given an initial guess $\mathbf{x}_0$, the Kaczmarz method is defined as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \frac{\mathbf{a}_{\tau(k)}\mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)}}{\left\|\mathbf{a}_{\tau(k)}\right\|_2^2}\mathbf{a}_{\tau(k)}^\top. \tag{2.5}$$

The Kaczmarz method has an intuitive geometric explanation–for $\alpha_k = 1$, the Kaczmarz method orthogonally projects $\mathbf{x}_{k-1}$ onto the hyperplane $\mathbf{a}_{\tau(k)}\mathbf{x} = \mathbf{b}_{\tau(k)}$. Due to the geometric nature of the Kaczmarz method, it has a natural connection to the method of projection onto convex sets (POCS) [30] and alternating projection methods [16, 58].

Much of the early work on the Kaczmarz method was for consistent linear systems of equations. First, we present the theory for this case. Then we look at the corresponding theory for LS problems.

2.4.1 Consistent Linear System

Kaczmarz proved that for a consistent linear system, with the additional assumption that $\mathbf{x}_{\text{true}}$ is unique, the iterates defined in (2.5) converge to $\mathbf{x}_{\text{true}}$ under cyclic sampling with $\alpha_k = 1$ [100, 101]. The Kaczmarz method initially received little attention (except in [17, 61, 156]), but in the 1970s the method began to receive recognition within the medical imaging community under the name algebraic reconstruction technique (ART) [67]. ART was recognized for its cheap computational cost and its ability to produce approximates of

$\mathbf{x}_{\text{true}}$ rather quickly [2, 18, 29, 63, 84, 90, 114, 123, 151, 155, 159]. Due to its simplicity, the Kaczmarz method was used in the first medical commercial CT scanner in 1972 [10, 89].

Convergence theory was developed to incorporate different step sizes. For constant step size, $\alpha_k = \alpha \in (0, 2)$ it was shown that the iterates in (2.5) converge at a linear rate to the minimum norm solution, $\mathbf{x}_{\text{true}}$, of the linear consistent system under cyclic sampling. The convergence rates at this time relied heavily on the geometric nature of the problem and included values that were hard to compute, and thus hard to compare to other existing methods [84, 123, 159]. In 2014, a more accessible bound for convergence of the Kaczmarz method with cyclic sampling was derived. Assuming that $\mathbf{A} \in \mathbb{R}^{m\times m}$ is invertible in (2.1), the iterates in (2.5) with cyclic sampling satisfy, for each $j \in \mathbb{N}$,

$$\|\mathbf{x}_{jm} - \mathbf{x}_{\text{true}}\|_2^2 \leq \left[\left(1 - \det\left(\mathbf{D}^{-1}\mathbf{A}\right)^{\frac{2}{m}}\right)\right]^{jm} \|\mathbf{x}_0 - \mathbf{x}_{\text{true}}\|_2^2 , \tag{2.6}$$

where $\mathbf{D} \in \mathbb{R}^{m\times m}$ is the diagonal matrix whose entries are the 2-norms of the rows of $\mathbf{A}$, i.e.

$$\mathbf{D} = \begin{bmatrix} \|\mathbf{a}_1\|_2 & & \\ & \ddots & \\ & & \|\mathbf{a}_m\|_2 \end{bmatrix} . \tag{2.7}$$

It is easily verified that $0 < \det\left(\mathbf{D}^{-1}\mathbf{A}\right) \leq 1$ [161].

The converge rate in (2.6) depends on the number of rows in $\mathbf{A}$. For the Kaczmarz method under cyclic control, convergence rates often depend on m, which is not reflective of the favorable quick convergence of the algorithm. Furthermore, convergence theory of the Kaczmarz method often uses cyclic sampling, but it was observed that when the Kaczmarz method was performed with random uniform sampling, the algorithm convergence was faster

[59, 85, 123, 151]. In 2009, a convergence rate for the Kaczmarz method with non-uniform random sampling was shown to have a linear convergence rate based on the scaled condition number of $\mathbf{A}$, $\kappa\left(\mathbf{A}\right) = \left\|\mathbf{A}\right\|_{\mathrm{F}} \left\|\mathbf{A}^{\dagger}\right\|_{2}$, where $\mathbf{A}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$ and $\left\|\cdot\right\|_{\mathrm{F}}$ is the matrix Frobenius norm defined as $\left\|\mathbf{A}\right\|_{\mathrm{F}} = \left\|\mathrm{vec}\left(\mathbf{A}\right)\right\|_{2}$. This rate is given by

$$\mathbb{E}\left\|\mathbf{x}_{k} - \mathbf{x}_{\mathrm{true}}\right\|_{2}^{2} \leq \left(1 - \kappa\left(\mathbf{A}\right)^{-2}\right)^{k} \left\|\mathbf{x}_{0} - \mathbf{x}_{\mathrm{true}}\right\|_{2}^{2},$$

assuming that $\mathbf{A}$ has full column rank [148, 149]. Notice that this convergence rate does not depend on the number of rows of $\mathbf{A}$.

Later it was noted that non-uniform random sampling has overall little influence on the convergence rate [33]. For uniform random sampling, the bound on the convergence rate can easily be modified

$$\mathbb{E}\left\|\mathbf{x}_{k} - \mathbf{x}_{\mathrm{true}}\right\|_{2}^{2} \leq \left(1 - \kappa\left(\mathbf{D}^{-1}\mathbf{A}\right)^{-2}\right)^{k} \left\|\mathbf{x}_{0} - \mathbf{x}_{\mathrm{true}}\right\|_{2}^{2},$$

where $\mathbf{D} \in \mathbb{R}^{m\times m}$ is defined as in (2.7) [161]. Additionally when $\mathbf{A} \in \mathbb{R}^{n\times n}$ is invertible,

$$\kappa(\mathbf{D}^{-1}\mathbf{A}) = \left\|\mathbf{D}^{-1}\mathbf{A}\right\|_{\mathrm{F}} \left\|\mathbf{A}^{-1}\mathbf{D}\right\|_{2} \leq \frac{\max_{i}\left\|\mathbf{a}_{i}\right\|_{2}}{\sqrt{\sum_{i}^{m}\left\|\mathbf{a}_{i}\right\|_{2}^{2}}} \left\|\mathbf{A}\right\|_{\mathrm{F}} \left\|\mathbf{A}^{-1}\right\|_{2} \leq \kappa(\mathbf{A}), \tag{2.8}$$

implying that the bound for the rate of convergence from sampling uniformly is better than the rate of convergence from sampling non-uniformly, i.e.,

$$\left(1 - \kappa\left(\mathbf{D}^{-1}\mathbf{A}\right)^{-2}\right) \leq \left(1 - \kappa\left(\mathbf{A}\right)^{-2}\right). \tag{2.9}$$

To complete the convergence theory for the Kaczmarz method under random uniform sampling for the consistent linear system of equations, the asymptotic behavior of the randomized Kaczmarz method was studied in [35] showing that the iterates $\mathbf{x}_k$ converged almost surely to the true solution, i.e., $\mathbf{x}_k \xrightarrow{\text{a.s.}} \mathbf{x}_{\text{true}}$.

2.4.2 Linear Least Squares

For inconsistent systems, the Kaczmarz method does not necessarily converge to a minimizer of LS problem in (2.2). Under cyclic sampling, for constant step size $\alpha_k = \alpha \in (0, 2)$ and $\mathbf{x}_0 \in \text{Range}\left(\mathbf{A}^\top\right) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{A}^\top \mathbf{y} \text{ for some } y \in \mathbb{R}^m\}$, the cyclic subsequences $\{\mathbf{x}_{jm+i}\}_{j=1}^{\infty}$ converge for each $i \in \mathbb{N}$ such that $0 \leq i \leq m-1$. More specifically, for matrices

$$\mathbf{B}_{(i)} = \begin{bmatrix} \mathbf{a}_{((i-1) \bmod m)+1} \\ \mathbf{a}_{(i \bmod m)+1} \\ \vdots \\ \mathbf{a}_{((i+m-2) \bmod m)+1} \end{bmatrix}, \qquad \mathbf{y}_{(i)} = \begin{bmatrix} \mathbf{b}_{((i-1) \bmod m)+1} \\ \mathbf{b}_{(i \bmod m)+1} \\ \vdots \\ \mathbf{b}_{((i+m-1) \bmod m)+1} \end{bmatrix}, \tag{2.10}$$

$\mathbf{D}_{(i)} = \text{diag}\left(\mathbf{B}_i \mathbf{B}_i^\top\right)$, and

$$\mathbf{L}_{(i)} = \begin{bmatrix} 0 & & & \\ \mathbf{a}_{(i \bmod M)+1}\mathbf{a}_{((i-1) \bmod m)+1}^\top & 0 & & \\ \vdots & & \ddots & \\ \mathbf{a}_{((i+m-2) \bmod m)+1}\mathbf{a}_{((i-1) \bmod m)+1}^\top & \cdots & \mathbf{a}_{((i+m-2) \bmod m)+1}\mathbf{a}_{((i+m-3) \bmod m)+1}^\top & 0 \end{bmatrix}, \tag{2.11}$$

the cyclic subsequence $\{\mathbf{x}_{jm+i}\}_{j=1}^{\infty}$ converges

$$\lim_{j\to\infty} \mathbf{x}_{jm+i} = \widetilde{\mathbf{x}}_i,$$

where $\widetilde{\mathbf{x}}_i$ is the unique solution in Range $\left(\mathbf{A}^\top\right)$ to

$$\mathbf{B}_{(i)}^\top \left(\mathbf{D}_{(i)} + \alpha \mathbf{L}_{(i)}\right)^{-1} \mathbf{B}_{(i)}\mathbf{x} = \mathbf{B}_{(i)} \left(\mathbf{R}_{(r)} + \alpha \mathbf{L}_{(r)}\right)^{-1} \mathbf{y}_{(i)}.$$

The vector $\widetilde{\mathbf{x}}_i$ is not necessarily the LS solution. Although, as $\alpha \to 0$, $\widetilde{\mathbf{x}}_i \to \widetilde{\mathbf{x}}$, where $\widetilde{\mathbf{x}}$ is the minimum norm solution to

$$\min \left\|\mathbf{D}^{-1}\left(\mathbf{A}\mathbf{x} - \mathbf{b}\right)\right\|_2^2, \tag{2.12}$$

with $\mathbf{D} \in \mathbb{R}^{m\times m}$ is defined as in (2.7) [98, 114, 123, 151, 159].

This result has two significant contributions. First, for constant step size, the iterates do not converge in the traditional sense but vary from one “solution” to the next. A diminishing step size remedies this variation in the iterates. Second, as the step size gets smaller, the iterates do not converge to a minimizer of the LS problem but rather to a minimizer of the weighted LS problem in (2.12). These two quantities can be arbitrarily far apart, see Chapter 3, Section 3.3 for an illustration.

For random uniform samples, similar results have been derived. For constant step size $\alpha < 1$ and $\mathbf{x}_0 \in$ Range $\left(\mathbf{A}^\top\right)$,

$$\mathbb{E}\left\|\mathbf{x}_k - \widetilde{\mathbf{x}}\right\|_2^2 \leq \left[\left(1 - \frac{2\alpha(1-\alpha)}{\kappa\left(\mathbf{D}^{-1}\mathbf{A}\right)}\right)\right]^k \left\|\mathbf{x}_0 - \widetilde{\mathbf{x}}\right\|_2^2 + \frac{\alpha}{1-\alpha}\kappa\left(\mathbf{D}^{-1}\mathbf{A}\right)\widetilde{\mathbf{r}}, \tag{2.13}$$

where $\widetilde{\mathbf{r}} = \left\|\mathbf{D}^{-1}\left(\mathbf{A}\widetilde{\mathbf{x}} - \mathbf{b}\right)\right\|_2^2/m$ and $\widetilde{\mathbf{x}}$ is the minimum norm solution to (2.12) [125]. This bound shows that the mean square error between iterates of the randomized Kaczmarz algorithm converge to a weighted LS solution linearly up to what is known as a "convergence horizon." As α gets closer to zero, the rate of convergence goes to 1, while the convergence horizon gets smaller. This demonstrates a trade-off between speed of convergence and precision of iterates that is based on the step size.

By modifying the step sizes, the Kaczmarz method can be made to converge to the LS solution under cyclic and random sampling [125]. For cyclic sampling, setting $\alpha_k = \alpha\left\|\mathbf{a}_{\tau(k)}\right\|_2^2$ for $\alpha \in \left(0, 2\text{min}_i \left\|\mathbf{a}_i\right\|_2^{-2}\right)$ ensures that $\mathbf{x}_k \to \mathbf{x}_\alpha$, where $\mathbf{x}_\alpha \in \mathbb{R}^n$ depends on step size α, such that $\mathbf{x}_\alpha \overset{\alpha\to 0}{\longrightarrow} \mathbf{x}_{\text{LS}}$ [32]. For random uniform sampling, setting $\alpha_k = \alpha\left\|\mathbf{a}_{\tau(k)}\right\|_2^2$ for $\alpha \in (0, \left\|\mathbf{A}\right\|_{\text{F}}^{-2})$ gives the following bound

$$\mathbb{E}\left\|\mathbf{x}_k - \mathbf{x}_{\text{LS}}\right\|_2^2 \leq \left[\left(1 - \frac{2\alpha(1-2\alpha)}{\kappa\left(\mathbf{A}\right)}\right)\right]^k \left\|\mathbf{x}_0 - \mathbf{x}_{\text{LS}}\right\|_2^2 + \frac{\alpha}{1-2\alpha}\kappa\left(\mathbf{A}\right)\frac{2\hat{\mathbf{r}}}{n\left\|\mathbf{A}\right\|_{\text{F}}^2}, \tag{2.14}$$

where $\hat{\mathbf{r}} = \mathbb{E}\left\|\mathbf{a}_{\tau(k)}^\top\left(\mathbf{a}_{\tau(k)}\widetilde{\mathbf{x}} - \mathbf{b}_{\tau(k)}\right)\right\|_2^2$ [125].

As we will see in Chapter 3, these choices of step sizes are effectively transforming the Kaczmarz method from a stochastic Newton method to a stochastic gradient method, to fix the bias in the convergence of the iterates. For further discussion see Section 3.2.2.

2.5 The Block Kaczmarz Method

The Kaczmarz method can be naturally extended to the block Kaczmarz method. For an initial guess $\mathbf{x}_0 \in \mathbb{R}^n$, the block Kaczmarz iterates are defined by

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{A}_{\tau(k)}^{\dagger} \left(\mathbf{A}_{\tau(k)} \mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)} \right). \tag{2.15}$$

For $\alpha_k = 1$ and a consistent linear system, this method projects the iterate $\mathbf{x}_{k-1}$ onto the hyperplane $\mathbf{A}_{\tau(k)}\mathbf{x} = \mathbf{b}_{\tau(k)}$. In the case of block size $\ell = 1$, this method is equivalent to the Kaczmarz method. The block Kaczmarz method was first studied under cyclic control [54], and was studied extensively in the medical imaging context. For a consistent system of linear equations, $\mathbf{x}_0 \in \text{Range}\left(\mathbf{A}^\top\right)$, and $\alpha_k = \alpha \in (0, 2)$, the iterates in (2.15) converge to the minimum norm solution $\mathbf{x}_{\text{true}}$ [152]. If the matrix $\mathbf{A}$ has full column rank and each block $\mathbf{A}_i$ has full row rank, then for each $j \in \mathbb{N}$,

$$\left\| \mathbf{x}_{jM} - \mathbf{x}_{\text{true}} \right\|_2 \leq \left(\left(1 - \frac{(\det(\mathbf{A}))^2}{\prod_{i=1}^{M} \det(\mathbf{A}_j^\top \mathbf{A}_j)} \right)^{\frac{1}{M}} \right)^{\frac{jM}{2}} \left\| \mathbf{x}_0 - \mathbf{x}_{\text{true}} \right\|_2, \tag{2.16}$$

where $\mathbf{x}_{\text{true}}$ is the unique solution to the system [7].

For the LS problem, the iterates are shown to have convergent cyclic subsequences, which is similar to the Kaczmarz method [53, 54]. In a randomized context, convergence bounds on the mean square error have been developed, similar to those for the Kaczmarz method. For $\mathbf{A}$ full column rank, step size $\alpha = 1$, and $\mathbf{x}_0 \in \mathbb{R}^n$,

$$\mathbb{E} \left\| \mathbf{x}_k - \mathbf{x}_{\text{LS}} \right\|_2^2 \leq \left[1 - \frac{\sigma_{\min}^2(\mathbf{A})}{m A_{\max}} \right]^k \left\| \mathbf{x}_0 - \mathbf{x}_{\text{LS}} \right\|_2 + \frac{A_{\max}}{A_{\min}} \frac{\left\| \mathbf{A}\mathbf{x}_{\text{LS}} - \mathbf{b} \right\|_2^2}{\sigma^2(\mathbf{A})}$$

where $A_{\max}$ and $A_{\min}$ are positive scalars such that $A_{\min} \leq \lambda_{\min}(\mathbf{A}_i)$ and $\lambda_{\max}(\mathbf{A}_i) \leq A_{\max}$

for each i such that $1 \leq i \leq M$. For all k, this bound again shows a linear convergence rate up to a convergence horizon. For more general step sizes, no similar bound has been derived.

For a decaying step size, no asymptotic convergence has been shown for the block Kaczmarz method with random sampling. However, the randomized extended block Kaczmarz method utilizes blocks of columns of $\mathbf{A}$ in addition to blocks of rows of $\mathbf{A}$ to guarantee convergence of iterates to the LS solution. In Section 3.3 we show that convergence to the LS problem is not guaranteed when only row samples are available.

2.6 The Damped Block Kaczmarz Method

When the matrix blocks $\mathbf{A}_i$ are ill-conditioned, the inversion in (2.15) can cause iterates to become contaminated with noise. To remedy this, the damped block Kaczmarz method introduces a damping term in the inversion,

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \left(\alpha_k^{-1}\mathbf{I} + \mathbf{A}_{\tau(k)}^\top \mathbf{A}_{\tau(k)}\right)^{-1} \mathbf{A}_{\tau(k)}^\top \left(\mathbf{A}_{\tau(k)}\mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)}\right). \tag{2.17}$$

Notice that the sequence of step sizes has been moved and is now represented as the sequence of damping terms.

A convergence analysis of this method has only been investigated in a cyclic context. Assuming that the step sizes α_k satisfy $\sum \alpha_k = \infty$ and $\alpha_k \to 0$, and that there is a $c \in \mathbb{R}$ such that $\mathbf{A}_{\tau(k)}^\top \left(\mathbf{A}_{\tau(k)}\mathbf{x}_k - \mathbf{b}_k\right) \leq c$ for all k, the iterates in (2.17) satisfies the condition

$$\lim_{k\to\infty} \|\mathbf{A}\mathbf{x}_k - \mathbf{b}\|_2^2 = \|\mathbf{A}\mathbf{x}_{\mathrm{LS}} - \mathbf{b}\|_2^2. \tag{2.18}$$

This shows, in the case of $\mathbf{A}$ being full column rank, that $\mathbf{x}_k \to \mathbf{x}_{\mathrm{LS}}$. For constant step size

$\alpha \in (0, 2)$,

$$\liminf_{k\to\infty} \|\mathbf{A}\mathbf{x}_k - \mathbf{b}\|_2 = \|\mathbf{A}\mathbf{x}_{\text{LS}} - \mathbf{b}\|_2 + \frac{\alpha M^2 c^2}{2}\left(4 + \frac{1}{M}\right). \tag{2.19}$$

This shows a subsequence of $\mathbf{x}_k$ produces values $\mathbf{A}\mathbf{x}_k - \mathbf{b}$ to within a threshold of the optimal residual $\mathbf{A}\mathbf{x}_{\text{LS}} - \mathbf{b}$. Note that as the step size gets smaller, the threshold goes to zero [3]

2.7 Generalized Block Structure for Consistent Systems

Row-action methods for consistent linear systems have been developed that allow for a more general selection of blocks $\mathbf{A}_k$ and $\mathbf{b}_k$. For a random matrix $\mathbf{W} \in \mathbb{R}^{m\times\ell}$, blocks may be define as $\mathbf{A}_k = \mathbf{W}_k^\top \mathbf{A}$ and $\mathbf{b}_k = \mathbf{W}_k^\top \mathbf{b}$ where the matrices $\{\mathbf{W}_k\}_{k=1}^\infty$ are independent and each identically distributed to $\mathbf{W}$. This allows the blocks $\mathbf{A}_k$ and $\mathbf{b}_k$ to contain linear combination of rows of $\mathbf{A}$ and corresponding entries of $\mathbf{b}$. For $\mathbf{x}_0 \in \mathbb{R}^n$ the iterates

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{A}_k^\top \left(\mathbf{A}_k\mathbf{A}_k^\top\right)^{-1} (\mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k)$$

satisfy the following convergence properties

$$\|\mathbb{E}\left[\mathbf{x}_k - \mathbf{x}_{\text{true}}\right]\|_2^2 \le \rho^k \|\mathbf{x}_0 - \mathbf{x}_{\text{true}}\|_2^2,$$

and

$$\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2^2 \le \rho^{2k} \|\mathbf{x}_0 - \mathbf{x}_{\text{true}}\|_2^2,$$

where $\mathbf{x}_{\text{true}}$ is the unique solution to the consistent linear system with $\mathbf{A}$ full column rank, and $\mathbf{W}$ is chosen in such a way to guarantee $\rho = 1 - \lambda_{\min}\left(\mathbb{E}\left[\mathbf{A}\mathbf{W}\left(\mathbf{W}^\top\mathbf{A}\mathbf{A}\mathbf{W}\right)^\dagger\mathbf{W}^\top\mathbf{A}\right]\right)$ is between 0 and 1. These bounds show a linear convergence rate to the true solution of a consistant system for the sequence of first moments $\mathbb{E}\left[\mathbf{x}_k\right]$ and the mean square error $\mathbb{E}\left\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\right\|_2^2$. Specific choices of $\mathbf{W}$ produce the randomized Kaczmarz and block Kaczmarz method [69], which we will also see in Section 3.2.

2.8 Row-action Methods for the Tikhonov LS Problem

There have been fewer works on extensions of row-action methods to solve the Tikhonov problem, (1.1). In the case of $\mathbf{L} = \mathbf{I}_n$, the minimum norm solution to (1.1), $\mathbf{x}_{\text{Tik}}$, can be found by applying the Kaczmarz method to the consistent system of linear equations

$$\begin{bmatrix} \mathbf{I}_m & \frac{\mathbf{A}}{\sqrt{\lambda}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{x} \end{bmatrix} = \frac{\mathbf{b}}{\sqrt{\lambda}}. \tag{2.20}$$

This is because the minimum norm solution in (2.20) is given by

$$\begin{bmatrix} \mathbf{u}^* \\ \mathbf{x}^* \end{bmatrix} = \begin{bmatrix} \mathbf{I}_m \\ \frac{\mathbf{A}^\top}{\sqrt{\lambda}} \end{bmatrix} \left(\mathbf{I}_n + \frac{1}{\lambda}\mathbf{A}^\top\mathbf{A}\right)^{-1} \frac{\mathbf{b}}{\sqrt{\lambda}}, \tag{2.21}$$

which implies

$$\mathbf{x}^* = \mathbf{A}^\top\left(\mathbf{I}_m + \mathbf{A}\mathbf{A}^\top\right)^{-1}\mathbf{b} = \left(\mathbf{I}_n + \mathbf{A}^\top\mathbf{A}\right)\mathbf{A}^\top\mathbf{b} = \mathbf{x}_{\text{Tik}}$$

[86, 87].

A row-action method that uses past samples to obtain the Tikhonov least square problem has been introduced. Under cyclic control, the Sherman Morrison iteration defines iterates at the kth iteration for $1 \leq k \leq m$ as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \frac{\mathbf{a}_{\tau(k)}^\top \mathbf{x}_{k-1}}{1 + \mathbf{a}_{\tau(k)}^\top \mathbf{z}_{k-1,k}} \mathbf{z}_{k-1,k}, \tag{2.22}$$

and for $k+1 \leq j \leq m$ as

$$\mathbf{z}_{k,j} = \mathbf{z}_{k-1,j} - \frac{\mathbf{a}_{\tau(k)}^\top \mathbf{z}_{k-1,j}}{1 + \mathbf{a}_{\tau(k)}^\top \mathbf{z}_{k-1,k}} \mathbf{z}_{k-1,k}, \tag{2.23}$$

where $\mathbf{x}_0 = \frac{1}{\lambda^2} \left(\mathbf{L}^\top \mathbf{L}\right)^{-1}$ and $\mathbf{z}_{0,j} = \frac{1}{\lambda^2} \left(\mathbf{L}^\top \mathbf{L}\right)^{-1} \mathbf{a}_{\tau(j)}^\top$ for $1 \leq j \leq m$.

At the Mth iteration $\mathbf{x}_M = \mathbf{x}_{\text{Tik}}$ [112, 142]. We will see that this algorithm can be generalized to include larger blocks of $\mathbf{A}$ and has connections to the recursive LS algorithm discussed in Chapter 4.

To solve the Tikhonov problem, any row-action method can be applied to

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2, \tag{2.24}$$

however when the Kaczmarz and block Kaczmarz methods are applied to the underlying linear system, convergence is not guaranteed. In Chapter 4, we consider the approach of applying row-action methods to (2.24), developing row-action methods that guarantee asymptotic convergence to $\mathbf{x}_{\text{Tik}}$ while utilizing regularization parameter updates.

2.9 Other Row-action Methods

The row-action methods described in this chapter were primarily focused on the linear system of equations; however, row-action methods can be applied to more general problems. Row-action can be seen as a much more broad class of algorithms for unconstrained and constrained optimization [30]. The Kaczmarz method has been applied to optimization problems involving linear operators in a Hilbert space [62, 118], finding the inverse of a matrix [68] quadratic equations [37], convex optimization [147] and has been applied to nonlinear functions of different forms [24, 48, 117, 118].

Row-action methods have been connected to other classes of algorithms. The Kaczmarz and block Kaczmarz method can be seen as a particular case of the projection onto convex sets algorithm and alternating projections methods [28, 31, 58]. There are also connections between the Kaczmarz method the Gauss-Seidel method for the LS problem [15]. Row-action methods have recently been connected with proximal gradient methods, where variants of the Kaczmarz method have been applied to the LS problem with total variation and ℓ_1 regularization [3, 13, 14].

Chapter 3

Stochastic Newton and Quasi-Newton methods

Randomized row-action methods can be interpreted as stochastic Newton and quasi-Newton methods applied to a stochastic reformulation of the LS problem. This observation allows new insights into the convergence properties of known algorithms including the block and damped block Kaczmarz algorithms. Furthermore, from this stochastic optimization framework, we will develop a new row-action method, called `slimLS`, that utilizes memory of past samples and has favorable convergence properties.

This chapter is concerned will solving the massive LS problem,

$$\min_{\mathbf{x}} f(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \,, \tag{3.1}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, with $m, n \in \mathbb{N}$ being *massive* in the sense that a $\min\{m,n\} \times \{m,n\}$ is too large to store in computer memory and matrix vector multiplications of $\mathbf{A}$ with a dense vector are not feasible.

We begin by reformulating the LS problem as a stochastic optimization problem in Section 3.1. The equivalence between stochastic approximation methods applied to this reformulation and row-action methods for the LS problem is presented in Section 3.2, and the stochastic quasi-Newton methods `rrls` and `slimLS` are defined. We present novel asymptotic convergence theory for the block Kaczmarz method in Section 3.3, showing that under random sampling the iterates converge to a weighted LS solution. In Section 3.4, we show that the `rrls` and `slimLS` methods converge to the LS solution. We study `slimLS` and find a trade-off between precision in iterates and convergence rate that is based on step size. Numerical examples are presented in 3.5, Proofs for theorems and lemmas in this chapter are shown in Appendix A.2. In Chapter 4 we extend the described methods to the Tikhonov LS problem.

3.1 Stochastic Reformulation of the LS Problem

Let $\mathbf{W} \in \mathbb{R}^{m \times \ell}$ be a random variable such that $\mathbb{E}\left[\mathbf{W}\mathbf{W}^\top\right] = \beta \mathbf{I}_m$, where $\ell \ll m$ and $\beta > 0$. Define $f_{\mathbf{W}}(\mathbf{x}) := \left\|\mathbf{W}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})\right\|_2^2$. For each $\mathbf{x} \in \mathbb{R}^n$,

$$\begin{aligned}\mathbb{E}\left\|\mathbf{W}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})\right\|_2^2 &= \mathbb{E}\left[(\mathbf{A}\mathbf{x} - \mathbf{b})^\top \mathbf{W}\mathbf{W}^\top (\mathbf{A}\mathbf{x} - \mathbf{b})\right] && (3.2)\\ &= \beta(\mathbf{A}\mathbf{x} - \mathbf{b})^\top(\mathbf{A}\mathbf{x} - \mathbf{b}) && (3.3)\\ &= \beta\left\|\mathbf{A}\mathbf{x} - \mathbf{b}\right\|_2^2. && (3.4)\end{aligned}$$

Therefore the stochastic optimization problem,

$$\min_{\mathbf{x}} \mathbb{E}\left\|\mathbf{W}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})\right\|_2^2, \tag{3.5}$$

has the same solution set as the LS problem in (3.1). In the context of randomized linear algebra, $\mathbf{W}$ would be called a sketching matrix [9, 36, 50, 51, 135, 163].

3.2 Stochastic Approximation Algorithms

In this section, we describe stochastic approximation (SA) methods for computing a solution to (3.5), and connect them to row-action methods for the LS problem (3.1). Stochastic approximation algorithms are iterative optimization methods for objective functions that contain an expected value. For a general introduction to SA methods, see [11, 105, 106, 140]. Given an initial vector $\mathbf{x}_0 \in \mathbb{R}^n$, SA methods applied to (3.1) define a sequence of iterates

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{s}_k, \tag{3.6}$$

where $\{\mathbf{s}_k\}$ is a sequence of search directions such that $\mathbf{s}_k$ depends on the iterate $\mathbf{x}_{k-1}$ and the random variables $\mathbf{W}_1, \dots, \mathbf{W}_k$, where each $\mathbf{W}_k$ independent and has an identical distributed to $\mathbf{W}$.

At the kth iteration $\mathbf{A}_k^\top = \mathbf{W}_k^\top \mathbf{A} \in \mathbb{R}^{\ell \times n}$, $\mathbf{b}_k = \mathbf{W}_k^\top \mathbf{b} \in \mathbb{R}^\ell$ will denote the block of rows of $\mathbf{A}$ and $\mathbf{b}$ that are sampled. The size of these blocks allows them to be used in numerical computations unlike the full matrix and vector $\mathbf{A}$ and $\mathbf{b}$.

3.2.1 Generalized Block Structure

The choice of $\mathbf{W}$ described below determines the linear combination of rows of $\mathbf{A}$ and $\mathbf{b}$ that are sampled from the LS problem. There are many choices of $\mathbf{W}$ that are available:

1. *Random sparse matrices.* Let $\mathbf{W} \in \mathbb{R}^{m \times \ell}$ be a random matrix with i.i.d. random elements w_{ij} where, for a fixed $0 < \psi \leq 1$, w_{ij} takes the values $\pm\sqrt{\beta/\ell\psi}$ for a $\beta > 0$ each with probability $\psi/2$ and the value zero with probability $1 - \psi$. It is straightforward to verify that $\mathbb{E}(\,\mathbf{W}\mathbf{W}^\top\,) = \beta \mathbf{I}_m$. Notice that as ψ gets closer to zero, more sparsity is introduced in $\mathbf{W}$. It is worth mentioning that this choice of $\mathbf{W}$ is a generalization of Achlioptas

random matrix ($\psi = 1/3$ and $\beta = \ell$) and the Rademacher distribution ($\psi = 1$ and $\beta = \ell$), see [1, 74].

2. *Generalized Kaczmarz matrices.* For $i = 1, \ldots, p$, let $\mathbf{Q}_i \in \mathbb{R}^{m \times \ell_i}$ be such that $\mathbf{Q} = [\mathbf{Q}_1, \ldots, \mathbf{Q}_p] \in \mathbb{R}^{m \times m}$ is an orthogonal matrix. Define the distribution of $\mathbf{W}$ to be uniform on $\{\mathbf{Q}_1, \ldots, \mathbf{Q}_p\}$. Then

$$\mathbb{E}\left(\mathbf{W}\mathbf{W}^\top\right) = \tfrac{1}{p}\sum_{i=1}^{p} \mathbf{Q}_i\mathbf{Q}_i^\top = \tfrac{1}{p}\mathbf{I}_m.$$

Notice that selecting $\mathbf{Q} = \mathbf{I}_m$, and leads to sampling blocks of rows of $\mathbf{A}$, i.e. $\mathbf{W}^\top\mathbf{A} = \mathbf{A}_{\tau(k)}$, where $\tau(k)$ is the random uniform variable on the set $\{1, \ldots, M/\ell\}$, this is the typical block choice for row-action methods, described in Chapter 2, Section 2.1. We will refer to this particular choice of blocks as *Kaczmarz blocks*. Choosing the elements of $\mathbf{Q}$ to be ±1 (or in $\{0, \pm 1\}$) leads to (sparse) randomized Hadamard matrices [23, 83]. Notice, that sparsity may be introduced by the particular choice of $\mathbf{Q}$ and that the number of columns in the $\mathbf{Q}_i$'s can differ.

3. *Sparse Rademacher matrices.* Fix $p \leq m$. The columns $\mathbf{w}_i$ of $\mathbf{W} \in \mathbb{R}^{m \times \ell}$ are i.i.d. and each column can be any $m \times 1$ vector with p-nonzero entries in $\{\pm 1\}$ with equal probability. Hence, conditional on a vector configuration, C, of p ones and $m - p$ zeros, each column $\mathbf{w}_i$ has conditional expectation

$$\mathbb{E}(\, \mathbf{w}_i\mathbf{w}_i^\top \mid C\,) = \mathbf{I}_{m,C},$$

where $\mathbf{I}_{m,C}$ is the diagonal matrix with the configuration C in the diagonal. It follows that

$$\mathbb{E}(\,\mathbf{w}_i\mathbf{w}_i^\top\,) = \mathbb{E}\,\mathbb{E}(\,\mathbf{w}_i\mathbf{w}_i^\top \mid C\,) = \frac{\begin{pmatrix} m-1 \\ p-1 \end{pmatrix}}{\begin{pmatrix} m \\ p \end{pmatrix}}\,\mathbf{I}_m = \tfrac{p}{m}\,\mathbf{I}_m,$$

and therefore $\mathbb{E}(\,\mathbf{W}\mathbf{W}^\top\,) = (\ell\,p/m)\,\mathbf{I}_m$. Note that the case $p = m$ generates full Rademacher matrices. The distinction between the other choices of $\mathbf{W}$ is that entries of the sparse Rademacher matrices are not i.i.d., as in the random sparse matrices, and do not necessarily come from partitions of orthogonal matrices, as in the generalized Kaczmarz matrices.

As discussed in Section 3.1, the solutions to problems (3.1) and (3.5) are equivalent if $\mathbb{E}(\mathbf{W}\mathbf{W}^\top) = \beta\mathbf{I}_m$. Adjusting the sampling matrix $\mathbf{W}$ such that $\mathbb{E}(\mathbf{W}\mathbf{W}^\top) = \mathbf{\Gamma}^{-1}$ for a positive definite matrix $\mathbf{\Gamma} \in \mathbb{R}^{m\times m}$ would make the stochastic optimization method in (3.5) have the same solution set as the weighted LS problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2_{\mathbf{\Gamma}^{-1}}\,.$$

This could be useful when, for example, solving the inverse problem

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon},$$

in the case where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ [27]. However, for simplicity of presentation, we consider $\mathbf{\Gamma} = \beta\mathbf{I}_m$.

3.2.2 Connection to Row-action Methods

Different choices of $\mathbf{s}_k$ in (3.6) produce different row-action methods.

- *Stochastic gradient methods.* The most common SA approach is the stochastic gradient method, where $\mathbf{s}_k = \alpha_k \nabla f_{\mathbf{W}_k}(\mathbf{x}_{k-1})$ and iterates are defined as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right), \tag{3.7}$$

  for a given $\mathbf{x}_0 \in \mathbb{R}^n$, where $\{\alpha_k\}$ is the sequence of step sizes. A discussion on the choice of step size is made in the section below. Iterates in (3.7) have been studied as variants of the Kaczmarz and block Kaczmarz algorithms, see Chapter 2 Sections 2.4 and 2.5. In general, the popularity of the stochastic gradient method stems from its proven consistency properties and its easy implementation [19]. However, the stochastic gradient method is known to converge slowly [164] and is sensitive to the choice of step size α_k [139, 166]. Thus, higher order methods are desired.

- *Stochastic Newton methods.* For the stochastic Newton (SN) method, the search direction is typically defined as

$$\mathbf{s}_k = \alpha_k \left(\nabla^2 f_{\mathbf{W}_k}\right)^\dagger \nabla f_{\mathbf{W}_k}(\mathbf{x}_{k-1}), \tag{3.8}$$

  where the sample Hessian is given by $\nabla^2 f_{\mathbf{W}} = \mathbf{A}_k^\top \mathbf{A}_k$, and † denotes the Moore-Penrose pseudoinverse. Using properties of the pseudoinverse, the iterates become identical to the block Kaczmarz method,

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{A}_k^\dagger(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k). \tag{3.9}$$

When the sample Hessian is ill-conditioned, a damping term is added to the sample Hessian in the search direction,

$$\mathbf{s}_k = \left(\alpha_k^{-1}\mathbf{I} + \nabla^2 f_{\mathbf{W}_k}\right)^{\dagger} \nabla f_{\mathbf{W}_k}(\mathbf{x}_{k-1}),$$

which yields the damped block Kaczmarz method,

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \left(\alpha_k^{-1}\mathbf{I} + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right).$$

- *Stochastic quasi-Newton methods.* For the stochastic quasi-Newton method, the search direction is given by

$$\mathbf{s}_k = -\mathbf{B}_k \nabla f_{\mathbf{W}_k}(\mathbf{x}_{k-1}), \tag{3.10}$$

where the sequence of positive definite matrices $\{\mathbf{B}_k\}$ approximate the inverse Hessian $\left(\mathbf{A}^\top\mathbf{A}\right)^{-1}$, choices discussed below.

The randomized recursive least squares algorithm, `rrls`, uses all previous samples to approximate the Hessian, and the iterates are given as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \left(\alpha_0 \mathbf{C} + \sum_{i=1}^{k} \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right),$$

where $\mathbf{C}$ is a positive definite matrix and $\alpha_0 \in \mathbb{R}^+$. We will see in Chapter 4 the connection between this algorithm and the recursive least squares algorithm, which gives it powerful convergence properties. The main disadvantage is that this algorithm requires storing an $n \times n$ matrix in memory or solving a progressively larger linear LS system at each iteration, which makes it impractical for massive problems. The sampled limited memory method for the LS problem, `slimLS`, uses only a few previous samples to approximate the Hessian $\left(\mathbf{A}^\top\mathbf{A}\right)^{-1}$ to avoid this computational bottleneck. Given a memory parameter $r \in \mathbb{N}$, the kth `slimLS` iterate is defined as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k\mathbf{A}_k^\top\left(\mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k\right) \tag{3.11}$$

with

$$\mathbf{B}_k = \left(\alpha_k^{-1}/(r+1)\mathbf{C}_k + \mathbf{M}_k^\top\mathbf{M}_k\right)^{-1} \quad \text{and} \quad \mathbf{M}_k = \begin{bmatrix}\mathbf{A}_{k-r}, & \ldots, & \mathbf{A}_k\end{bmatrix}^\top.$$

Here $\{\mathbf{C}_k\}$ is a sequence of positive semi-definite matrices. On the one hand `slimLS` is a limited memory variant of the `rrls` algorithm that only uses r previous samples instead of all previous samples. The `slimLS` method could also be interpreted as a generalization of the damped block Kaczmarz method, where the block Kaczmarz method is recovered when $r = 0$ and $\mathbf{C}_k = \mathbf{I}_n$.

There are many benefits of the `slimLS` method. `slimLS` exhibits favorable initial convergence, similar to stochastic Newton-type methods, but with the added benefit of asymptotic convergence to the LS solution, which we will discuss in Section 3.4.

3.2.3 Choice of Step Size

Selecting a good step size α_k (or learning rate, as it is referred to in machine learning) is critical. A variety of methods have been proposed to improve convergence rates, see for instance [19, 46, 146]. To ensure asymptotic convergence, a decaying step size is often necessary. In this case we will assume the step size meets the following conditions

$$\sum \alpha_k = \infty \quad \text{and} \quad \sum \alpha_k^2 < \infty, \tag{3.12}$$

when proving asymptotic convergence in Sections 3.3 and 3.4. This is satisfied, for example, by setting the step sizes to the harmonic sequence $\alpha_k = 1/k$, see [137]. We will also consider the case when $\alpha_k = \alpha$ is constant, for $\alpha \in (0,2)$. By using a constant step size, we sacrifice the asymptotic convergence properties of the algorithm, but we see the favorable initial convergence properties that these algorithms are known for.

3.3 Almost Sure Convergence of the SN Method

In this section, we study the consistency of the SN method, which provides new convergence theory for the randomized block Kaczmarz method. The following result shows that the SN method does not necessarily converge to an LS solution, but instead to a weighted LS solution. See Section A.2 in the appendix for the proof.

Theorem 3.1 (a.s. convergence of the SN method)**.** *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *have rank* n *and* $\mathbf{b} \in \mathbb{R}^m$. *Let* $\mathbf{W} \in \mathbb{R}^{m\times \ell}$ *be a random variable with* M *realizations and with the property that* $\mathbb{E}\left[\mathbf{W}\mathbf{W}^\top\right] = \beta \mathbf{I}_n$ *for some* $\beta > 0$. *Let* $\{\alpha_k\}$ *be a positive sequence of scalars such that*

$$\sum_{k=1}^{\infty} \alpha_k = \infty \quad \textit{and} \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

Set $\mathbf{P} = \mathbb{E}\left[\mathbf{A}_k^\dagger \mathbf{W}_k^\top\right]$ *and* $\widetilde{\mathbf{x}} = (\mathbf{P}\mathbf{A})^{-1}\mathbf{P}\mathbf{b}$, *and let* $\mathbf{x}_0 \in \mathbb{R}^n$ *be an arbitrary initial vector. Define*

$$\mathbf{x}_k \;=\; \mathbf{x}_{k-1} + \alpha_k \mathbf{A}_k^\dagger \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)$$

Then $\mathbf{x}_k \xrightarrow{\text{a.s.}} \widetilde{\mathbf{x}}$.

When $\mathbf{W}$ is chosen so that $\mathbf{A}_k$ are Kaczmarz blocks as defined in Section 3.2.1. Theorem 3.1 shows asymptotic convergence to a weighted LS solution for the Kaczmarz and block Kaczmarz method. For the Kaczmarz method, iterates converge to

$$\widetilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\|\mathbf{D}^{\dagger}\left(\mathbf{A}\mathbf{x} - \mathbf{b}\right)\right\|_2^2,$$

where $\mathbf{D} \in \mathbb{R}^{m\times m}$ is defined as in (2.7) For the block Kaczmarz method, iterates converge to

$$\widetilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\|\mathbf{F}^{\dagger}\left(\mathbf{A}\mathbf{x} - \mathbf{b}\right)\right\|_2^2$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_M \end{bmatrix}.$$

These are the first almost sure convergence results for the Kaczmarz and block Kaczmarz methods for a non-consistent linear system [42]. In general, the solution to this weighted LS problem can be arbitrarily far away from the solution to the standard LS problem.

We now provide some insight regarding the potential discrepancy between the desired LS solution $\widehat{\mathbf{x}}$ and the solution to which the SN method converges, namely,

$$\widetilde{\mathbf{x}} = (\mathbf{P}\mathbf{A})^{-1}\mathbf{P}\mathbf{b} = \left(\mathbb{E}\left[(\mathbf{W}^{\top}\mathbf{A})^{\dagger}\mathbf{W}^{\top}\mathbf{A}\,\right]\right)^{-1}\mathbb{E}\left[\,(\mathbf{W}^{\top}\mathbf{A})^{\dagger}\mathbf{W}^{\top}\,\right]\mathbf{b}. \tag{3.13}$$

The difference between $\widehat{\mathbf{x}}$ and $\widetilde{\mathbf{x}}$ depends on $\mathbf{P}$, and we can say the following. Assuming that

$$\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}, \tag{3.14}$$

where the random noise $\boldsymbol{\epsilon}$ has zero mean and covariance matrix $\mathrm{Var}(\boldsymbol{\epsilon}) = \sigma^2 \mathbf{I}_m$, we have

$$\begin{aligned} \mathbb{E}\,\mathbf{x}_{\mathrm{LS}} &= \mathbf{x}_{\mathrm{true}}, \qquad \mathrm{Var}(\widehat{\mathbf{x}}) = \sigma^2\,(\mathbf{A}^\top\mathbf{A})^{-1}, \\ \mathbb{E}\,\widetilde{\mathbf{x}} &= \mathbf{x}_{\mathrm{true}}, \qquad \mathrm{Var}(\widetilde{\mathbf{x}}) = \sigma^2\,(\mathbf{P}\mathbf{A})^{-1}\mathbf{P}\mathbf{P}^\top(\mathbf{A}^\top\mathbf{P}^\top)^{-1}. \end{aligned}$$

This shows that $\widehat{\mathbf{x}}$ and $\widetilde{\mathbf{x}}$ are both unbiased estimators of $\mathbf{x}_{\mathrm{true}}$, but by the Gauss-Markov theorem, $\widehat{\mathbf{x}}$ is expected to have smaller variance. Consider the following simple example:

Example. Consider the LS problem, where

$$\mathbf{A} = \begin{bmatrix} \mu & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ \nu \end{bmatrix},$$

for some fixed $\mu, \nu \in \mathbb{R}$. We compare the LS solution and the solution obtained via stochastic Newton with random Kaczmarz vectors $\mathbf{w} \in \mathbb{R}^{m\times 1}$ (see page 27). It is easy to see that in this case we have

$$\mathbf{P} = \mathbf{A}^\top\mathbf{D}^{-1} \quad \text{with} \quad \mathbf{D}^{-1} = \mathrm{diag}\{1/\|\mathbf{a}_1\|^2, 1/\|\mathbf{a}_2\|^2, 1/\|\mathbf{a}_3\|^2\},$$

where $\mathbf{a}_i$ are the rows of $\mathbf{A}$. It follows that $\widetilde{\mathbf{x}}$ minimizes the weighted LS functional $(\mathbf{b} - \mathbf{A}\mathbf{x})^\top\mathbf{D}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x})$. We obtain the following solutions:

$$\widehat{\mathbf{x}} = \frac{1}{2\mu^2+1}\begin{bmatrix} 2\mu+\nu+1 \\ \mu - \mu^2\nu + \mu^2 + 1 \end{bmatrix} \qquad \text{and} \qquad \tilde{\mathbf{x}} = \frac{1}{4}\begin{bmatrix} 1+\nu+3/\mu \\ 3-\nu+1/\mu \end{bmatrix},$$

respectively. The covariance matrices of $\widehat{\mathbf{x}}$ and $\widetilde{\mathbf{x}}$ are

$$\mathrm{Var}(\widehat{\mathbf{x}}) = \frac{\sigma^2}{2\mu^2+1}\begin{bmatrix} 2 & 1 \\ 1 & \mu^2+1 \end{bmatrix}, \qquad \mathrm{Var}(\widetilde{\mathbf{x}}) = \frac{\sigma^2}{4\mu^2}\begin{bmatrix} 2\mu^2+9 & 8\mu^2+3 \\ 8\mu^2+3 & 10\mu^2+1 \end{bmatrix}.$$

It is clear that the variances of the components $\widetilde{\mathbf{x}}$ can be much larger than those of $\widehat{\mathbf{x}}$. The solution $\widetilde{\mathbf{x}}$ would have smaller variance if the covariance matrix of the noise was proportional to $\mathbf{P}^{-1}$ instead of $\mathbf{I}_m$. Figure 3.1 shows the error $\omega(\mu, \nu) = \|\widehat{\mathbf{x}} - \widetilde{\mathbf{x}}\|$ for various choices of μ and ν. The left panel shows that $\omega \to \infty$ as $\mu \to 0$, which makes sense as the first row of $\mathbf{A}$ becomes all zeros. The right panel shows that even for $\mu \neq 0$, a significant error can be incurred by varying ν – and therefore the "observation vector" $\mathbf{b}$.

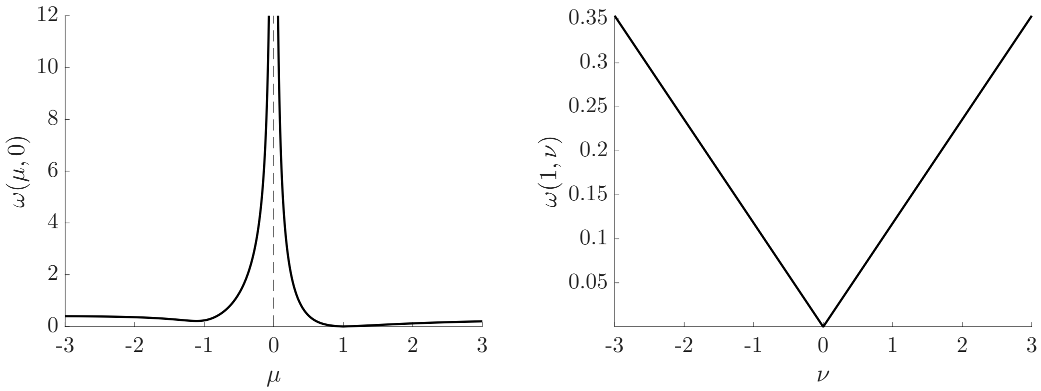

Figure 3.1: Error $\omega(\mu, \nu)$ of the stochastic Newton solution $\widetilde{\mathbf{x}}$ compared to $\widehat{\mathbf{x}}$. In the plot on the left, $\nu = 10$ and we vary μ. Notice that a pole exists at $\mu = 0$, where the relative error becomes arbitrarily large. The plot on the right illustrates the impact of varying ν for fixed $\mu = 1$.

Although the difference between $\widetilde{\mathbf{x}}$ and $\widehat{\mathbf{x}}$ can be significant, there are cases where they are identical. Some previous works have studied the problem of how close $\widetilde{\mathbf{x}}$ is to $\widehat{\mathbf{x}}$, e.g., [50, 163]. However, their assumptions do not apply to our matrix $\mathbf{P}$. For our problem, $\widetilde{\mathbf{x}} = \widehat{\mathbf{x}}$ when the linear system is consistent, since in this case, $\mathbf{P}\mathbf{A}\widehat{\mathbf{x}} = \mathbf{P}\mathbf{b}$, or when $(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top = (\mathbf{P}\mathbf{A})^{-1}\mathbf{P}$, which is equivalent to $\mathrm{null}(\mathbf{A}^\top) \subseteq \mathrm{null}(\mathbf{P})$, since $\mathbf{A}\widehat{\mathbf{x}} - \mathbf{b} \in \mathrm{null}(\mathbf{A}^\top)$ implies that

$\mathbf{P}(\mathbf{A}\widehat{\mathbf{x}}-\mathbf{b}) = \mathbf{0}$. In the example above, this occurs when $\nu = 1/\mu - 1$ (e.g., $\mu = 1$ and $\nu = 0$).

3.4 Analysis of quasi-Newton Methods

In this section, we present the convergence properties of `rrls` and `slimLS`. First, we analyze the asymptotic behavior, showing almost sure convergence to the LS solution. Then, a non-asymptotic analysis is done to show expected linear convergence of the first moment and mean square error of the iterates.

3.4.1 Almost Sure Convergence

Iterates defined by `rrls` and `slimLS` converge to the LS solution. For the linear inverse problem discussed in Section 1.1.1, this convergence is more favorable than the convergence to a weighted LS solution of the stochastic Newton methods.

Theorem 3.2. *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be full column rank and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{W} \in \mathbb{R}^{m \times \ell}$ be a random variable with M realizations and with the property that $\mathbb{E}\left[\mathbf{W}\mathbf{W}^\top\right] = \beta \mathbf{I}_n$ for some $\beta > 0$. For an initial $\mathbf{x}_0 \in \mathbb{R}^n$, define*

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)$$

with

$$\mathbf{B}_k = \left(\alpha_0 \mathbf{C} + \sum_{i=1}^{k} \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1}$$

where

- $\alpha_0 \geq 0$ *and*

- $\mathbf{C}$ *is a positive definite matrix*

Then $\mathbf{x}_k \xrightarrow{\text{a.s.}} \mathbf{x}_{\text{LS}}$.

Theorem 3.3. *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *be full column rank and* $\mathbf{b} \in \mathbb{R}^m$. *Let* $\mathbf{W} \in \mathbb{R}^{m\times \ell}$ *be a random variable with* M *realizations and with the property that* $\mathbb{E}\left[\mathbf{W}\mathbf{W}^\top\right] = \beta \mathbf{I}_n$ *for some* $\beta > 0$. *For memory parameter* $r \in \mathbb{N}$ *and initial* $\mathbf{x}_0 \in \mathbb{R}^n$, *define*

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)$$

with

$$\mathbf{B}_k = \left(\alpha_k^{-1}/(r+1)\mathbf{C}_k + \mathbf{M}_k^\top \mathbf{M}_k\right)^{-1} \quad \textit{and} \quad \mathbf{M}_k = \begin{bmatrix} \mathbf{A}_{k-r}, & \ldots, & \mathbf{A}_k \end{bmatrix}^\top,$$

where

- $\sum \alpha_k = \infty$ *and* $\sum \alpha_k^2$ *converges,*
- $\{\mathbf{C}_k\}$ *is a sequence of symmetric positive definite matrices with* $\mathbf{C}_k$ *being* $\mathbf{W}_1, \ldots, \mathbf{W}_{k-1}$ *measurable, with eigenvalues bounded below and above by* $\eta_{\min}, \eta_{\max} \in \mathbb{R}^+$ *respectively, and*
- $\left\|\mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)\right\|_2 \le g$ *for* $g \ge 0$ *and all* $k \in \mathbb{N}$

Then $\mathbf{x}_k \xrightarrow{\text{a.s.}} \mathbf{x}_{\text{LS}}$.

The assumption that $\left\|\mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_k - \mathbf{b}_k\right)\right\|_2 \le g$ is essentially a bound on the second moment of the search direction $\mathbf{B}_k \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)$, since the eigenvalues of $\mathbf{B}_k$ are assumed to be bounded. Bounds of this type are assumed frequently when proving asymptotic convergence of such algorithms, e.g., in stochastic optimization [11, 19, 20, 22, 71, 106, 120] and for cyclic control [3, 13, 14].

As seen in Section 3.3, there is bias in the asymptotic behavior of the stochastic Newton methods. This bias is from using only the sample Hessian information $\mathbf{A}_k^\top \mathbf{A}_k$. In `slimLS` as the step size α_k gets small, the sample Hessian $\mathbf{A}_k^\top \mathbf{A}_k$ is given less weight in the curvature matrix $\mathbf{B}_k$. This allows asymptotic convergence to the LS solution but allows early iterations to use the sample Hessian to get a fast initial convergence, which is why row-action methods are so popular. In the next section, we show that for a constant step size there is a trade-off between convergence rate and accuracy of iterations that is based on the step size or how much weight is put on the sample Hessian.

3.4.2 Convergence Rates

One benefit of row-action methods is their favorable initial convergence properties. With a decaying step size, the convergence of stochastic approximation algorithms can be quite slow (sub-linear) in general [21, 126]. In many cases, it is much more practical to use a constant step size to obtain fast initial convergence, at the cost of sacrificing accuracy of the approximation, see, e.g., [11, Ch. 3].

For constant step size $\alpha \in \mathbb{R}^+$, memory level $r = 0$ and $\mathbf{C}_k = \mathbf{I}_n$, we present converge rate properties of `slimLS`. More specifically, we show linear convergence of the expectation of the iterates and a linear convergence up to a convergence horizon of the mean squared error. Such analyses have been done for the Kaczmarz and block Kaczmarz methods, but not for the damped block Kaczmarz method; see Chapter 2 for details.

In the case of constant step size, there will be bias in solution. Define

$$\mathbf{B} = \mathbb{E}\left[\left(\alpha^{-1}\mathbf{I}_n + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \mathbf{A}_k^\top \mathbf{A}_k\right]. \tag{3.15}$$

When $\mathbf{A}$ is full column rank, $\mathbf{B}$ is symmetric positive definite, see A.2. Define

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\| \mathbf{B}\mathbf{x} - \mathbb{E}\left[\mathbf{B}_k \mathbf{A}_k^\top \mathbf{b}_k\right] \right\|_2^2 \tag{3.16}$$

$$= \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{B}_k \mathbf{A}_k^\top \mathbf{b}_k\right]. \tag{3.17}$$

We would like to quantify the difference between $\widehat{\mathbf{x}}$ and $\mathbf{x}_{\text{LS}}$. Notice that when the system is consistent, i.e., $\mathbf{A}\mathbf{x}_{\text{LS}} = \mathbf{b}$, then $\widehat{\mathbf{x}} = \mathbf{x}_{\text{LS}}$. This is because we may re-write

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\| \mathbb{E}\left[\mathbf{B}_k \mathbf{A}_k^\top \mathbf{W}_k^\top\right] (\mathbf{A}\mathbf{x} - \mathbf{b}) \right\|_2^2. \tag{3.18}$$

Clearly, $\left\| \mathbb{E}\left[\mathbf{B}_k \mathbf{A}_k^\top \mathbf{W}_k^\top\right] (\mathbf{A}\mathbf{x}_{\text{LS}} - \mathbf{b}) \right\|_2^2 = 0$, and so $\widehat{\mathbf{x}} = \mathbf{x}_{\text{LS}}$.

For a general LS problem (including inconsistent problems), we provide the following theorem that gives a loose bound on the difference between $\mathbf{x}_{\text{LS}}$ and $\widehat{\mathbf{x}}$.

Theorem 3.4. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be full column rank and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{W} \in \mathbb{R}^{m\times \ell}$ be a random variable with M realizations $\{\mathbf{W}^{(i)}\}_{i=1}^M$ with the property that $\mathbb{E}\left[\mathbf{W}\mathbf{W}^\top\right] = \beta \mathbf{I}_n$ for some $\beta > 0$. Define $\widehat{\mathbf{x}} = \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{B}_k \mathbf{A}_k^\top \mathbf{b}_k\right]$ where $\mathbf{B} = \mathbb{E}\left[\left(\alpha^{-1}\mathbf{I}_n + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \mathbf{A}_k^\top \mathbf{A}_k\right]$ and $\mathbf{x}_{\text{LS}} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b}$. Then*

$$\|\widehat{\mathbf{x}} - \mathbf{x}_{\text{LS}}\|_2 \le \alpha \frac{1 + \alpha A_{\min}}{1 + \alpha A_{\max}} \frac{A_{\max}}{p_{min} A_{\min}} \mathbb{E}\left[\left\| \mathbf{A}_k^\top \mathbf{A}_k \right\|_2 \left\| \left(\beta \mathbf{A}^\top \mathbf{A}\right)^{-1} \right\|_2 \left\| \mathbf{A}^\top \mathbf{b} \right\|_2 + \left\| \mathbf{A}_k^\top \mathbf{b}_k \right\|_2 \right]$$

where $A_{\max}$ and $A_{\min}$ are the largest and smallest non-zero eigenvalue of $\mathbf{A}_k^\top \mathbf{A}_k$ across all realizations of $\mathbf{W}$, and $p_{min} = \min_i p\left(\mathbf{W} = \mathbf{W}^{(i)}\right)$ where p is the probability density function of $\mathbf{W}$.

In Theorem 3.4, it is important to notice the relationship between the step size α and the upper bound. Notice that as α gets smaller, the difference between $\widehat{\mathbf{x}}$ and $\mathbf{x}_{\text{LS}}$ gets smaller.

In light of the asymptotic properties that $\mathbf{x}_k \to \mathbf{x}_{\text{LS}}$ for a decaying step size in Section 3.4.1, this result makes sense. In the following theorem, we show linear convergence of the first moment of $\mathbf{x}_k$ to $\widehat{\mathbf{x}}$, and linear convergence of the mean squared error between $\mathbf{x}_k$ and $\widehat{\mathbf{x}}$ to what is known as a convergence horizon. We see that there is a trade-off between step size and the bias in the iterates. Additionally, there will be a trade-off between step size and convergence rate.

Theorem 3.5. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be full column rank and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{W} \in \mathbb{R}^{m\times \ell}$ be a random variable with M realizations and with the property that $\mathbb{E}\left[\mathbf{W}\mathbf{W}^\top\right] = \beta \mathbf{I}_n$ for some $\beta > 0$. For memory parameter $r \in \mathbb{N}$ and step length $\alpha \in \mathbb{R}^+$, define*

$$\begin{aligned}\mathbf{x}_k &= \mathbf{x}_{k-1} - \mathbf{B}_k \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)\\ \mathbf{B}_k &= \left(\alpha^{-1}\mathbf{I}_n + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1}\end{aligned}$$

Define $\mathbf{B} = \mathbb{E}\left[\mathbf{B}_k \mathbf{A}_k^\top \mathbf{A}_k\right]$ and $\widehat{\mathbf{x}} = \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{B}_k \mathbf{A}_k^\top \mathbf{b}_k\right]$, then

1. $\mathbb{E}\left[\mathbf{x}_k\right] \to \widehat{\mathbf{x}}$, *more specifically*

$$\left\|\mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}}\right]\right\|_2 \leq \rho^k \left\|\mathbb{E}\left[\mathbf{x}_0 - \widehat{\mathbf{x}}\right]\right\|_2$$

where $\rho = \left\|\mathbb{E}\left(\mathbf{I}_n + \alpha \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1}\right\|_2 < 1$,

2.

$$\mathbb{E}\left[\left\|\mathbf{x}_k - \widehat{\mathbf{x}}\right\|_2^2\right] \leq (1-2c)^k \left\|\mathbf{x}_0 - \widehat{\mathbf{x}}\right\|_2^2 + \alpha^2 c^{-1}\sigma^2$$

where

(a) $0 < (1-2c) < 1$, *with* $c = \frac{\alpha\lambda_{\min}(\mathbf{B})}{(1+\alpha A_{\max})}$, $\lambda_{\min}\left(\mathbf{B}\right)$ *is the minimum eigenvalue of* $\mathbf{B}$,

$A_{\max}$ *maximum possible eigenvalue of* $\mathbf{A}_k^\top \mathbf{A}_k$*, and*

(b) $\sigma = \mathbb{E}\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2$.

The first part of this theorem shows an expected linear convergence to the weighted LS solution $\widehat{\mathbf{x}}$. The second bound shows linear convergence of the mean square error of iterates up to a convergence horizon. Notice that as $\alpha \to 0$ the convergence rate approaches one, but the convergence horizon gets smaller. Additionally as $\alpha \to 0$, $\widehat{\mathbf{x}} \to \mathbf{x}_{\text{LS}}$. This shows a trade-off between convergence rate and precision of the iterates.

3.5 Numerical Experiments

In this section, we present three experiments. The first experiment is on synthetic data, numerically comparing the asymptotic behavior of the randomized block Kaczmarz and the `rrls` algorithms. The second experiment considers the convergence behavior of `slimLS`. The last experiment implements `rrls` on a large scale image classification problem. Numerical experiments that implement `slimLS` on massive inverse problems are deferred to Chapter 4, when `slimLS` is applied to the Tikhonov LS problem.

3.5.1 Experiment 1: Asymptotic Behavior of Block Kaczmarz and `rrls`

First, we compare the asymptotic behavior of `rrls` and the block Kaczmarz method. We consider a linear regression problem, where $\mathbf{A} \in \mathbb{R}^{50{,}000\times 1{,}000}$ is a random matrix with elements drawn from a standard normal distribution. We let $\mathbf{x}_{\text{true}} = \mathbf{1} \in \mathbb{R}^{1{,}000}$ and $\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}$, where the additive noise $\boldsymbol{\epsilon}$ is also assumed to be standard normal. We choose $\mathbf{W} \in \mathbb{R}^{50{,}000\times 625}$ to be a block Kaczmarz matrix with block size $\ell = 625$. For the block Kaczmarz method we use step size $\alpha_k = 1/k$. For `rrls`, we chose the initial $\alpha_0 = 10^{-5}$ and $\mathbf{C} = \mathbf{I}_n$. To avoid a $n \times n$ inversion at each step, we update the $\mathbf{B}_k$ matrix with the Woodbury formula [65] at each iteration

$$\mathbf{B}_k = \mathbf{B}_{k-1} - \mathbf{B}_{k-1}\mathbf{A}_k^\top \left(\mathbf{I}_\ell + \mathbf{A}_k\mathbf{A}_k^\top\right)^{-1} \mathbf{A}_k\mathbf{B}_{k-1}. \tag{3.19}$$

For both algorithms, we start at a random initial guess $\mathbf{x}_0$. In Figure 3.2 we provide plots of relative errors. In the top left panel, the relative errors are computed as $\|\mathbf{x}_k - \mathbf{x}_{\text{LS}}\|/\|\mathbf{x}_{\text{LS}}\|$, where $\mathbf{x}_k$ are the `rrls` iterates, and in the top right panel, the relative errors are computed as $\|\mathbf{x}_k - \widetilde{\mathbf{x}}\|/\|\widetilde{\mathbf{x}}\|$, where $\mathbf{x}_k$ are block Kaczmarz iterates. These plots illustrate convergence of `rrls` and block Kaczmarz iterates to $\mathbf{x}_{\text{LS}}$ and $\widetilde{\mathbf{x}}$ respectively, as shown in Sections 3.3 and 3.4.1. Notice that the block Kaczmarz method exhibits much slower convergence to $\tilde{\mathbf{x}}$ than the convergence of `rrls` (solid blue line) to $\mathbf{x}_{\text{LS}}$. The block Kaczmarz method requires 20,000 iterations to reach a relative error of $3.3 \cdot 10^{-3}$, while the `rrls` iterates reach a relative error of $3.3 \cdot 10^{-3}$ after 175 iterations. Moreover, it takes the stochastic quasi-Newton only 22 iterations to achieve a relative error of 10^{-2}.

In the bottom panel of Figure 3.2, we provide reconstruction errors relative to the true solution $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\| \,/\, \|\mathbf{x}_{\text{true}}\|$, for both methods, which demonstrates that `rrls` is faster than the block Kaczmarz method at providing a better approximation of the true solution. For this experiment the relative error between $\mathbf{x}_{\text{LS}}$ and $\widehat{\mathbf{x}}$ is $\|\tilde{\mathbf{x}} - \mathbf{x}_{\text{LS}}\| \,/\, \|\mathbf{x}_{\text{LS}}\| = 5.69 \cdot 10^{-3}$. It is worth noting that the moderate size of this problem still allows one to use a QR solver (e.g., Matlab's "backslash") to solve the LS problem, which takes about 6 seconds whereas `rrls` requires about 12 seconds to run $k = 200$ iterations.

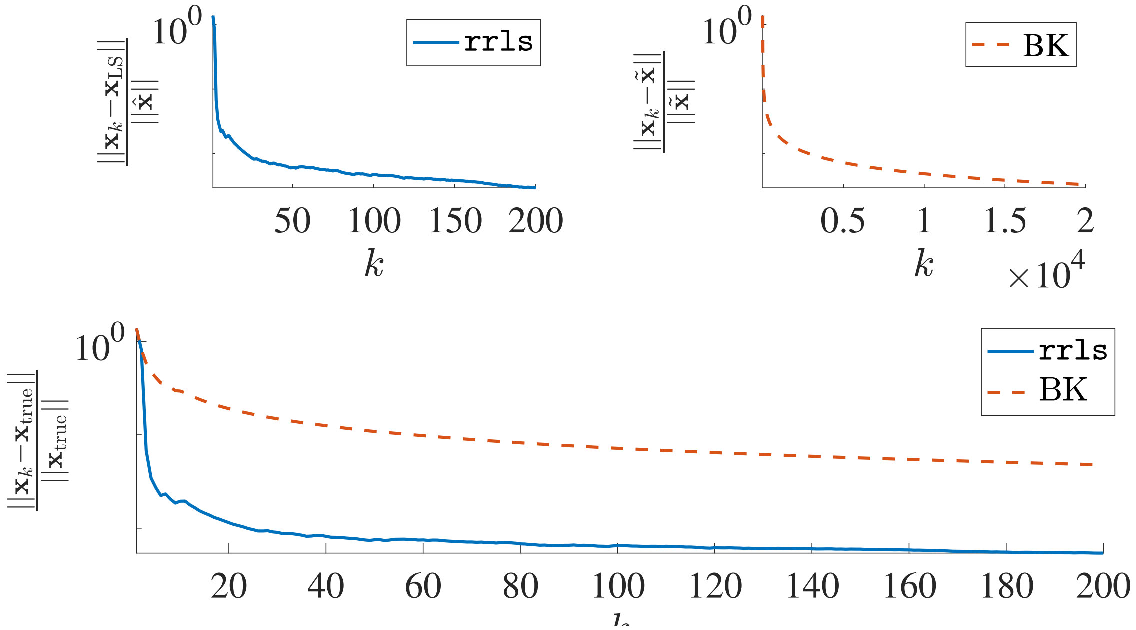

Figure 3.2: Experiment 1: The top left panel contains relative errors for `rrls` iterates, computed as $\|\mathbf{x}_k - \mathbf{x}_{\rm LS}\|/\|\mathbf{x}_{\rm LS}\|$, where $\mathbf{x}_{\rm LS}$ is the LS solution. The top right panel contains relative errors for block Kaczmarz iterates, computed as $\|\mathbf{x}_k - \widetilde{\mathbf{x}}\|/\|\widetilde{\mathbf{x}}\|$, where $\widetilde{\mathbf{x}}$ is defined in Theorem 3.1. Notice that we display 20,000 iterations for block Kaczmarz iterates and only 200 iterations for `rrls`. The bottom panel contains relative errors, $\|\mathbf{x}_k - \mathbf{x}_{\rm true}\|/\|\mathbf{x}_{\rm true}\|$, for both `rrls` and the block Kaczmarz method.

3.5.2 Experiment 2: Convergence Behavior of `slimLS`

Decaying Step Size

We now compare the asymptotic behavior of `rrls` and the `slimLS` algorithm for the same experimental set-up as Experiment 1 in Section 3.5.1. For `slimLS` we choose a step size $\alpha_k = 1/k$ with $\mathbf{C}_k = \mathbf{I}_n$, and vary the memory level r for $r = 0, 3, 5$. To efficiently compute the search direction in (3.11), we notice it can be computed by solving the regularized least squares problem

$$\mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) = \arg\min_{\mathbf{s}} \left\| \mathbf{M}_k \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k \end{bmatrix} \right\|_2^2 + \frac{r-1}{\alpha_k} \|\mathbf{s}\|_2^2 , \tag{3.20}$$

which can be solved using standard Krylov subspace methods such as LSQR [131, 132].

In Figure 3.3 we provide plots of relative errors. The top two plots contains relative errors for `rrls` and `slimLS` iterates for the first 200 and 100,000 iterations, computed as $\|\mathbf{x}_k - \mathbf{x}_{\mathrm{LS}}\| / \|\mathbf{x}_{\mathrm{LS}}\|$, where $\mathbf{x}_{\mathrm{LS}}$ is the LS solution. The bottom plot relative errors for `rrls` and `slimLS` iterates for the first 100,000 iterations, computed as $\|\mathbf{x}_k - \mathbf{x}_{\mathrm{true}}\| / \|\mathbf{x}_{\mathrm{true}}\|$, where $\mathbf{x}_{\mathrm{true}}$ is the true solution.

Notice that the convergence of `slimLS` speeds up the memory level increases. The `rrls` method uses all the information from past samples, so its convergence is the fastest. However `rrls` is very slow, and for problems where n is massive, not feasible because it requires storing a $n \times n$ matrix in memory. The `slimLS` method avoids this storage by only keeping r of the past samples, and replacing the $n \times n$ matrix inversion with the linear solve in (3.20).

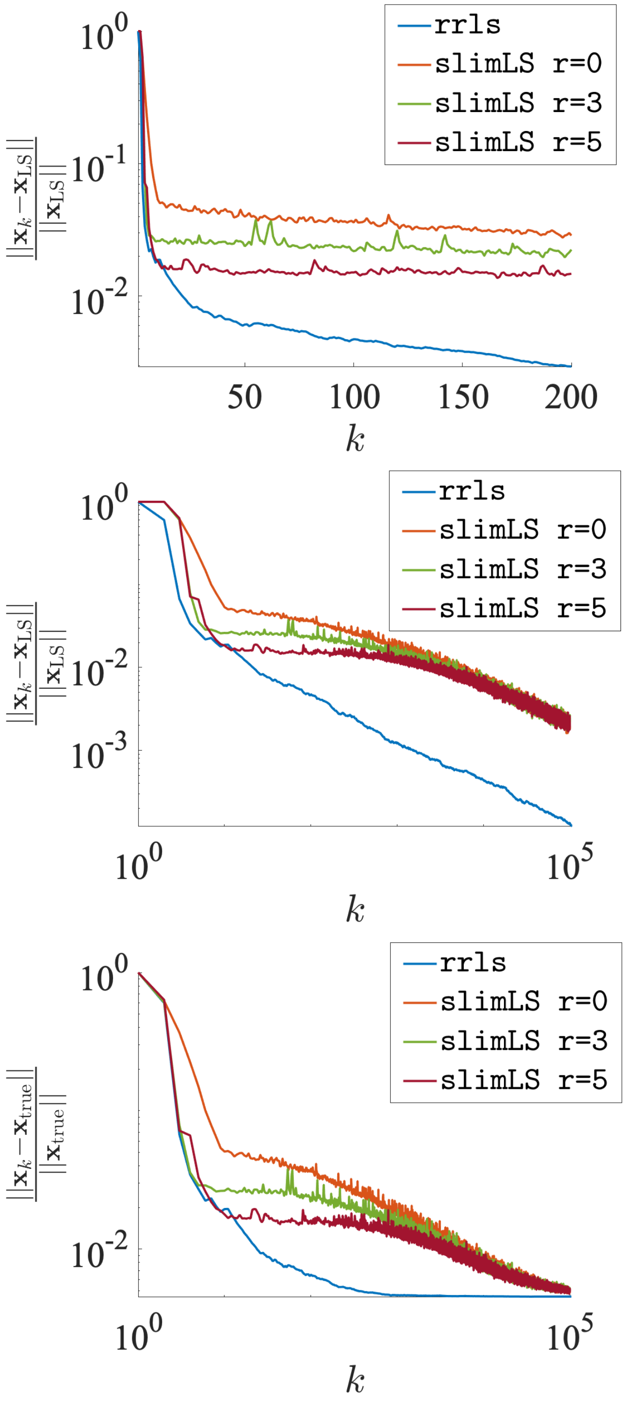

Figure 3.3: Experiment 2: The top two plots contains relative errors for `rrls` and `slimLS` iterates for the first 200 and 100,000 iterations, computed as $\|\mathbf{x}_k - \mathbf{x}_{\rm LS}\|/\|\mathbf{x}_{\rm LS}\|$, where $\mathbf{x}_{\rm LS}$ is the LS solution. The bottom plot relative errors for `rrls` and `slimLS` iterates for the first 100,000 iterations, computed as $\|\mathbf{x}_k - \mathbf{x}_{\rm true}\|/\|\mathbf{x}_{\rm true}\|$, where $\mathbf{x}_{\rm true}$ is the true solution.

Constant Step Size

We now numerically illustrate the convergence properties of `slimLS` for constant step size. We us the same problem set-up as Section (3.5.2). For memory level three we vary the step size $\alpha = \{10^{-i}\}_{i=1}^{4}$. In Figure 3.4 we see that for larger step size, the initial convergence is quick, but the error levels off very early in the iterative process. For smaller step sizes the initial convergence is slower, but the relative error gets smaller at later iterations before leveling off. This illustrates the trade-off between convergence rate and precision of iterates, as shown in Section 3.4.2.

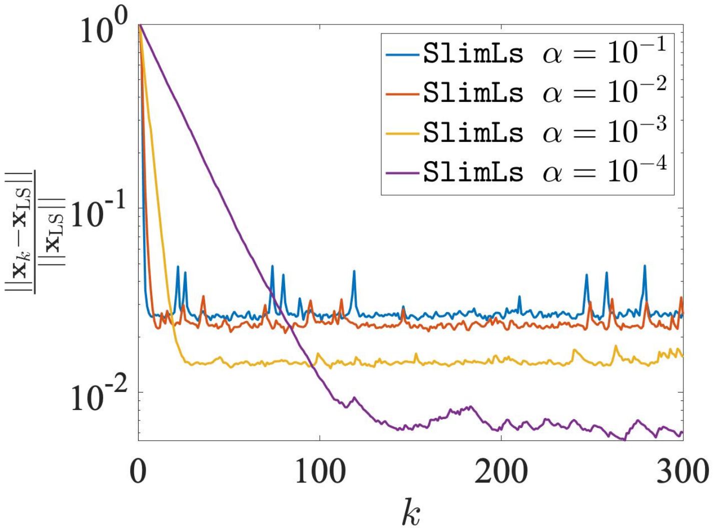

Figure 3.4: Experiment 2: Relative errors for the `slimLS` iterates for the first 300 iterations for different values of constant step size α. The relative error is computed as $\|\mathbf{x}_k - \mathbf{x}_{\mathrm{LS}}\|/\|\mathbf{x}_{\mathrm{LS}}\|$, where $\mathbf{x}_{\mathrm{LS}}$ is the LS solution.

3.5.3 Experiment 3: Image Classification

Next, we investigate the use `rrls` that arises in extreme learning machines (ELMs). ELM is a machine learning technique that uses random hidden nodes or neurons in a feedforward network to mimic biological learning techniques. The literature on ELM in the machine learning community is vast, with cited benefits that include higher scalability, less computational complexity, no requirement of tuning, and smaller training errors than generic machine learning techniques. ELM is commonly used for clustering, regression, and classification. Full details and comparisons are beyond the scope of this paper, and we refer the interested reader to papers such as [73, 92, 93, 94, 95, 96] and references therein.

At the core of ELM is a very large and potentially dynamically growing linear regression problem. In this experiment, we investigate the use of stochastic algorithms for efficiently solving these LS problems. In particular, we consider the problem of handwritten digit classification using the "MNIST" database [45], which contains 60,000 training images and 10,000 testing images of handwritten digits ranging from 0 to 9. Each image is 28×28 pixels and converted into a vector $\boldsymbol{\xi} \in \mathbb{R}^{784}$ (e.g., corresponding to 784 features).

We begin with a brief description of the classification problem for the MNIST dataset. Suppose we are given a set of m examples in the form of a training set

$$S = \{(\boldsymbol{\xi}_1, c_1), \cdots, (\boldsymbol{\xi}_m, c_m)\},$$

where $\boldsymbol{\xi}_i \in \mathbb{R}^{784}$ and c_i takes values from the set of classes $\mathcal{C} = \{0, 1, \cdots, 9\}$. Consider an ELM with a hidden layer of n nodes. Then the goal is to solve an LS problem of the form,

$$\min_{\mathbf{X}} \|\mathbf{H}\mathbf{X} - \mathbf{Y}\|_{\mathrm{F}}^2, \tag{3.21}$$

where the hidden-layer output matrix is defined as

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\boldsymbol{\xi}_1) \\ \vdots \\ \mathbf{h}(\boldsymbol{\xi}_m) \end{bmatrix} \in \mathbb{R}^{m \times n}$$

with $\mathbf{h}(\boldsymbol{\xi}) = \begin{bmatrix} h_1(\boldsymbol{\xi}) & \cdots & h_n(\boldsymbol{\xi}) \end{bmatrix}$ being the output (row) vector of the hidden layer with respect to the input $\boldsymbol{\xi}$, $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_{10} \end{bmatrix}$ where $\mathbf{x}_j \in \mathbb{R}^n$ contains the desired output weights for class j , and the training data target matrix $\mathbf{Y} \in \mathbb{R}^{m \times 10}$ takes entries

$$y_{ij} = \begin{cases} 1, & \text{if} \quad c_i = j - 1, \\ -1, & \text{else.} \end{cases}$$

For this example, $\mathbf{h}(\boldsymbol{\xi})$ can be interpreted as a map from the image pixel space to the n-dimensional hidden-layer feature space. Although various activation functions could be used, we employ a standard choice of sigmoid additive hidden nodes with

$$h_j(\boldsymbol{\xi}) = G(\mathbf{d}_j, \delta_j, \boldsymbol{\xi}) = 1/(1 + \exp(-\mathbf{d}_j^\top \boldsymbol{\xi} + \delta_j)),$$

where all of the hidden-node parameters $(\mathbf{d}_j, \delta_j)_{j=1}^n$ are randomly generated based on a uniform distribution [73]. For our experiments, we set the number of hidden neurons to be $n = 300$.

The main computational work of ELM is to solve (3.21). Regularized or constrained solutions have been investigated (e.g., [6, 73, 110]). However, *our focus will be on solving the unconstrained LS problem efficiently and for enormous sets of training data.* In order to generate larger datasets, we performed multiple random rotations of the original 60,000 training images. More specifically, each image was rotated by $20(\eta - 0.5)$ degrees, where η is a random number drawn from a beta distribution with shape parameters equal to 2. In our experiments, we consider up to 15 random rotations per image, resulting in up to 900,000 training images. Notice that as the number of training images increases, the number of rows of $\mathbf{H}$ increases accordingly, while the number of columns remains the same.

We consider three approaches to solve (3.21) and compare CPU timings. In the original implementation of ELM [93, 94], the LS solution was computed as $\widehat{\mathbf{X}} = \mathbf{H}^{\dagger}\mathbf{Y}$ where $\mathbf{H}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathbf{H}$. We denote this approach "PINV". Another approach to solving large (often sparse) LS problems is to use an iterative method such as LSQR [131, 132], but since the LS problem needs to be solved for multiple right-hand sides (here, 10 solves), we use a global LS method (Gl-LSQR) [157] with a maximum of 50 iterations and a residual tolerance of 10^{-6}. It was experimentally shown in [157] that Gl-LSQR is more effective and less expensive than LSQR applied to each right-hand side. We use the `rrls` method where $\mathbf{W}$ corresponds to the sparse Rademacher matrix with $\ell = 50$ and $\lambda_1 = 10^{-5}$. We use a maximum number of iterations of 1,000, a stopping tolerance of $\texttt{tol} = 10^{-4}$, and an initial guess of $\mathbf{0}$. Since the $\mathbf{B}_k$ matrices only depend on $\mathbf{H}$ and $\mathbf{W}$, `rrls` can be applied to multiple right-hand sides simultaneously.

Each LS solver is repeated 20 times in Matlab R2015b on a MacBook Pro with 2.9 GHz Intel Core i7 and 8G memory, and in Figure 3.5, we provide the median and 5^{th}–95^{th} percentiles of the CPU times vs. the number of training images (e.g., number of rows in $\mathbf{H}$). It is evident that for smaller training sets, all three methods perform similarly, but as the number of training images increases, `rrls` quickly surpasses PINV and Gl-LSQR in terms of faster CPU time. For various numbers of training data, we provide in Table 3.1 the mean and standard deviation of the relative reconstruction error for the `rrls` estimate, $\texttt{rel} = \|\mathbf{X}_{\texttt{rrls}} - \widehat{\mathbf{X}}\|_{\mathrm{F}}/\|\widehat{\mathbf{X}}\|_{\mathrm{F}}$, and of the number of `rrls` iterations, k. Our results demonstrate that `rrls`does not necessarily provide the most accurate solutions; however, it can be used to achieve sufficiently good solutions efficiently.

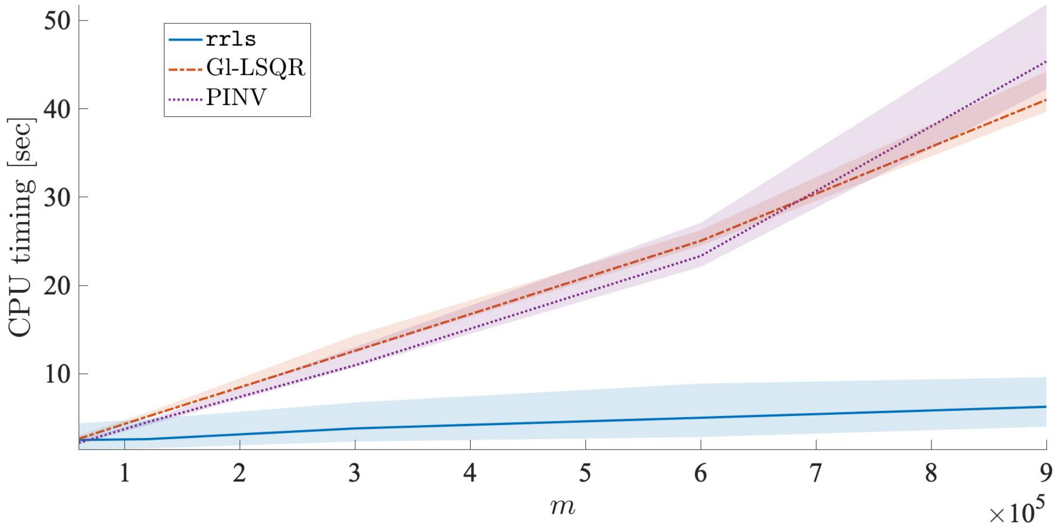

Figure 3.5: Experiment 3: CPU times (median and 5th–95th percentiles) for solving LS problem (3.21) using `rrls`, global LSQR (Gl-LSQR), and the Moore-Penrose pseudoinverse for various numbers of training images m.

Table 3.1: For various numbers of training images, we provide the mean and standard deviation for the relative reconstruction errors and the iteration counts for `rrls`.

m	60,000	120,000	300,000	600,000	900,000
`rel`	0.2705 ±0.046	0.2665 ±0.045	0.2456 ±0.035	0.2649 ±0.044	0.2568 ±0.036
k	632 ±219	675 ±227	706 ±194	631 ±196	663 ±180

Next, we test the performance of these estimates for classification of the MNIST testing dataset. That is, once computed, the output weights $\mathbf{X}$ can be used to classify images in the following way. For each test image, the predicted class is given by

$$\text{Class of } \boldsymbol{\xi} = \arg\max_{j} \mathbf{h}(\boldsymbol{\xi})\mathbf{x}_j \,.$$

In Figure 3.6 we provide a visualization of the computed classifications for the 10,000 testing images, where accuracy values in the titles are calculated as $1 - r/10000$ where r is the number of misclassified images. An accuracy value that is close to 1 corresponds to a good performance of the classifier. These results correspond to training on 60,000 images, and the testing set was sorted by class for easier visualization. Notice that in Figure 3.6 the misclassified images are almost identical for all three methods, and the classification accuracy for `rrls` is only slightly smaller than that of PINV and Gl-LSQR. Thus, we have shown that the `rrls` method can achieve comparable classification performance as PINV and GL-LSQR with much faster learning speed.

It is worth noting that the matrices considered here, though large, can still be loaded into memory. For problems where this is not the case (e.g., data too large or being dynamically generated [167]), PINV and Gl-LSQR would not be feasible, while `rrls` could still be used.

3.6 Remarks and Future Directions

In this chapter, we introduced row-action methods for the LS problems as stochastic approximation methods. Utilizing the connection between SA methods and row-action methods, the asymptotic behavior of the block Kaczmarz method was derived. We developed a new row-action method called `slimLS`, which uses information from past blocks to speed up the convergence of iterations. This method with memory level zero is identical to the damped block Kaczmarz method. We show that there is a trade-off between the convergence rate of this method and the precision of the iterates that is based on step size, and applied these methods to a large scale classification problem.

There are many future directions that this research can go. For a constant step size, a thorough analysis of the convergence rates for `slimLS` for a general memory parameter r

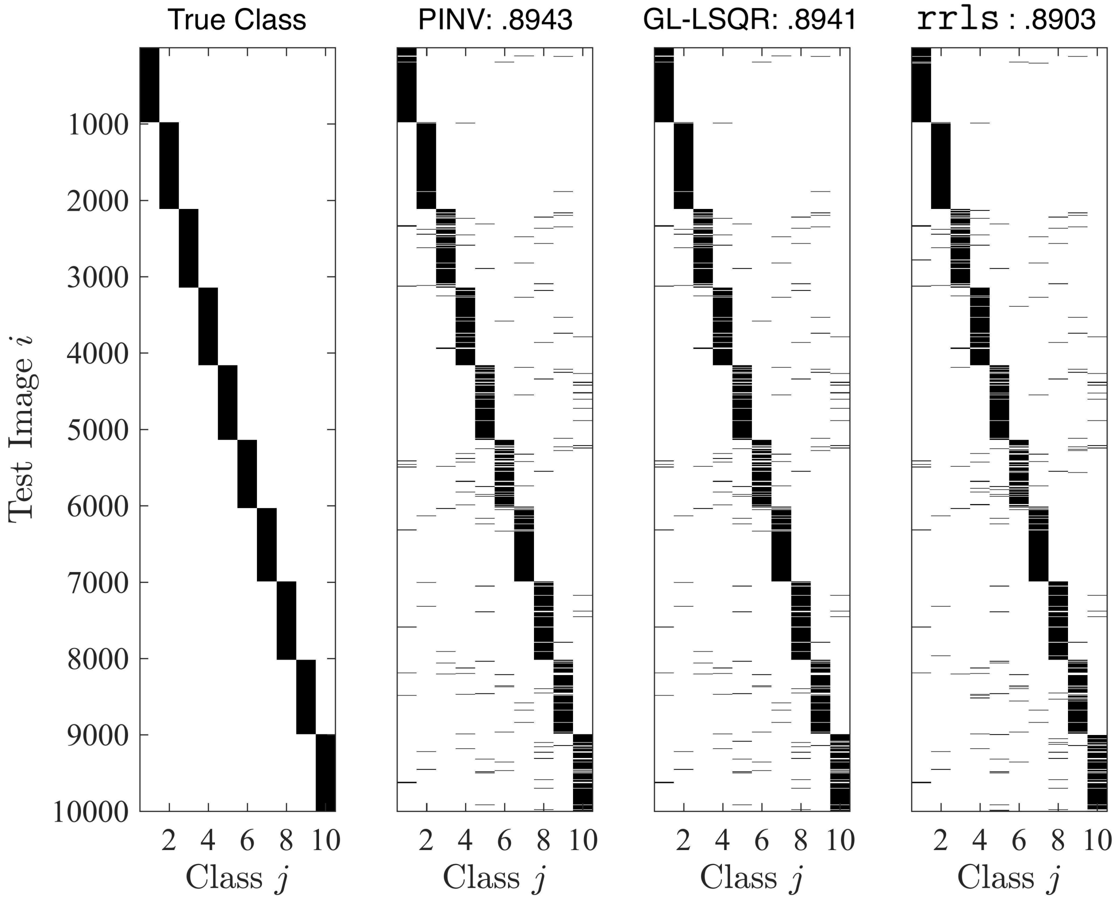

Figure 3.6: Expiroment 3: Classification (with corresponding accuracy) for the MNIST test images after training on 60,000 images, using different LS solvers.

would help understand the properties of `slimLS`. Additionally adapting the choice of the random variable $\mathbf{W}$ to sample the more essential parts of $\mathbf{A}$ could speed up convergence. An adaptive sampling strategy would be related to importance sampling.

Although `rrls` has the most favorable convergence properties, it is not feasible when n is massive, because it requires the storage of an $n \times n$ matrix in computer memory. The algorithm `slimLS` is a limited memory variant of `rrls` that can be applied to massive problems where n is massive. In the next chapter, we extend `rrls` and `slimLS` to the Tikhonov LS problem. The connection between these methods and the recursive least squares algorithm gives new insight into sampling methods and choice of the regularization parameter in the Tikhonov LS problem.

Chapter 4

Sampled Tikhonov Regularization

When the linear inverse problem is ill-posed, regularization must be introduced to recover a meaningful solution. This chapter focuses on row-action methods to solve the massive Tikhonov-regularized problem,

$$\min_{\mathbf{x}} f_\lambda(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{L}\mathbf{x}\|_2^2 \,, \tag{4.1}$$

where $\lambda > 0$ is the regularization parameter, and for simplicity we assume that $\mathbf{L}$ has full column rank. When all of $\mathbf{b}$ and $\mathbf{A}$ are available or can be accessed at once (e.g., via matrix-vector multiplication with $\mathbf{A}$), the Tikhonov solution,

$$\mathbf{x}(\lambda) = (\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{L}^\top \mathbf{L})^{-1} \mathbf{A}^\top \mathbf{b} \,, \tag{4.2}$$

can be computed using a plethora of existing iterative methods (e.g., Krylov or other optimization methods [78, 102]). There are also many techniques to chose an appropriate regularization parameter λ when all of $\mathbf{A}$ and $\mathbf{b}$ are available [25, 56, 72, 76, 78]. These techniques are not possible for the massive Tikhonov LS problem. Note that $\mathbf{x}(0)$ in (4.2) is the unregularized solution, which is defined if $\mathbf{A}$ has full column rank.

In this chapter we extend the row action methods introduced in Chapter 3 to the massive Tikhonov LS problem in (4.1), and develop sample based regularization techniques to find a useful regularization parameter during the iteration process. We introduce the block structure of the problem in Section 4.1. In Section 4.2 we describe two row-action methods for Tikhonov regularization, `rrls` and `sTik`, that utilize all previous blocks of data at each iteration. The `sTik` method converges asymptotically to a Tikhonov-regularized solution, while `rrls` converges to the LS solution, making `sTik` more favorable for the Tikhonov LS problem, as seen in Theorem 4.2. Asymptotic convergence results for random uniform sampling and random cyclic sampling are provided. In Section 4.3 we describe sampled regularization parameter selection methods that can be used to update the regularization parameter. In Section 4.4 we show that `slimLS` applied to the Tikhonov LS problem in (4.1) produces a limited memory version of `sTik`, which we call `slimTik`. Applying the convergence results of Section (3.4) shows asymptotic convergence of `slimTik` to the Tikhonov solution. Numerical illustrations are provided throughout the chapter. Numerical experiments in 4.5 apply `slimTik` to a tomography problem and a massive super resolution problem. Conclusions and future work are discussed in Section 4.6. Derivation of sampled regularization parameter selection methods are available in Appendix B

4.1 Problem Formulation

In the following, we describe a mathematical formulation of the problem that allows us to solve (4.1) in situations where samples of $\mathbf{A}$ and $\mathbf{b}$ become available over time. Since we would like to use random, cyclic, and random cyclic sampling (introduced in Section 2.2), we introduce the block structure in a slightly different way than in Chapter 3.

Formally, at the kth iteration, we assume that a set of rows of $\mathbf{A}$ and corresponding elements of $\mathbf{b}$ become available, which we denote by $\mathbf{W}_k^\top \mathbf{A}$ and $\mathbf{W}_k^\top \mathbf{b}$ respectively. Here the matrix $\mathbf{W}_k \in \mathbb{R}^{m \times \ell}$ is a *sampling* matrix, which selects rows of $\mathbf{A}$ and $\mathbf{b}$. For a fixed $M \in \mathbb{N}$ we assume that matrices $\{\mathbf{W}_i\}_{i=1}^M$ satisfy the following properties:

1. for each $i \in \{1, \ldots, M\}$, $\mathbf{W}_i \in \mathbb{R}^{m \times \ell}$, where $\ell = \frac{m}{M}$ and

2. the sum $\sum_{i=1}^{M} \mathbf{W}_i \mathbf{W}_i^\top = \mathbf{I}_m$.

The first assumption implies that the size of $\mathbf{W}_i^\top \mathbf{A}$ is smaller than the size of $\mathbf{A}$, and thus computationally manageable. The second assumption guarantees that all rows of $\mathbf{A}$ are given equal weight; however, importance sampling could be included and results in a weighted LS problem.

There are many suitable choices for $\{\mathbf{W}_i\}_{i=1}^{M}$. We will primarily consider the case where $\{\mathbf{W}_i\}$ are the realizations of block Kaczmarz matrices described in Section 3.2.1. Notice that the $\mathbf{W}_i$ defined in this chapter are matrices and not random variables as in Chapter 3. We will introduce the sampling method at the kth iteration through the variable $\tau(k)$ in Section 4.2, as previously seen in Chapter 2.

4.2 Full Memory Row-action Methods for Tikhonov Regularization

We investigate two row-action methods that use all previous blocks of data at each iteration. Let $\mathbf{y}_0, \mathbf{x}_0 \in \mathbb{R}^n$ be initial iterates and let $\mathbf{W}_i \in \mathbb{R}^{m \times \ell}$, $i = 1, \ldots, k$ be arbitrary matrices. For notational convenience, we denote $\mathbf{A}_i = \mathbf{W}_i^\top \mathbf{A}$ and $\mathbf{b}_i = \mathbf{W}_i^\top \mathbf{b}$. Assuming a fixed regularization parameter λ, the first method that we consider is *regularized recursive least squares* (`rrls`)[1], which is defined as

$$\mathbf{y}_k = \mathbf{y}_{k-1} - \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{y}_{k-1} - \mathbf{b}_k), \quad k \in \mathbb{N}, \tag{4.3}$$

[1] This should not be confused with the residual-reducing LS (RRLS) algorithm referenced in [132].

where $\mathbf{B}_k = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$. If $\mathbf{W}_i$ is the ith column of the identity matrix, `rrls` is an extension of the recursive LS algorithm [15] that includes a Tikhonov term. Since it may be difficult to know a good regularization parameter in advance, we propose a *sampled Tikhonov* (`sTik`) method, where the iterates are defined as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k \left(\mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) + \Lambda_k \mathbf{L}^\top \mathbf{L} \mathbf{x}_{k-1}\right), \quad k \in \mathbb{N}, \tag{4.4}$$

where $\mathbf{B}_k = \left(\sum_{i=1}^k \Lambda_i \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$. Compared to `rrls`, the main advantages of the `sTik` method are that the regularization parameter can be updated during the iterative process and that in a sampled framework, the `sTik` iterates converge asymptotically to a Tikhonov solution whereas the `rrls` iterates converge asymptotically to an unregularized solution. Of course, selecting a suitable regularization parameter can be difficult, especially for problems with a small range of good values. In any case, for inverse problems, it is desirable that the numerical method for solution computation converges to a regularized solution.

In this section, we begin by showing that for arbitrary matrices $\mathbf{W}_i$, both `rrls` and `sTik` iterates can be recast as solutions to regularized LS problems.

Theorem 4.1. *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{L} \in \mathbb{R}^{s \times n}$ have full column rank and $\mathbf{W}_i \in \mathbb{R}^{m \times \ell}$, $i = 1, \ldots, k$ be an arbitrary sequence of matrices.*

*(i) For $\lambda > 0$ and an arbitrary initial guess $\mathbf{y}_0 \in \mathbb{R}^n$, the **`rrls`** iterate (4.3) with $\mathbf{B}_k = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$ is the solution of the LS problem*

$$\min_{\mathbf{x}} \quad \left\|[\mathbf{W}_1, \ldots, \mathbf{W}_k]^\top (\mathbf{A}\mathbf{x} - \mathbf{b})\right\|_2^2 + \lambda \left\|\mathbf{L}(\mathbf{x} - \mathbf{y}_0)\right\|_2^2. \tag{4.5}$$

*(ii) For $\lambda_k = \sum_{i=1}^k \Lambda_i > 0$ for any k and an arbitrary initial guess $\mathbf{x}_0 \in \mathbb{R}^n$, the **`sTik`** iter-*

ate (4.4) with $\mathbf{B}_k = \left(\sum_{i=1}^k \Lambda_i \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$ *is the solution of the LS problem*

$$\min_{\mathbf{x}} \quad \left\| [\mathbf{W}_1, \ldots, \mathbf{W}_k]^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) \right\|_2^2 + \lambda_k \left\| \mathbf{L}\mathbf{x} \right\|_2^2 . \tag{4.6}$$

Proof of Theorem 4.1. For (ii), note that the solution of the LS problem (4.6) is given by

$$\mathbf{x}(\lambda_k) = \mathbf{B}_k \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{b}_i .$$

Noticing the relationship $\mathbf{B}_k^{-1} = \mathbf{B}_{k-1}^{-1} + \mathbf{A}_k^\top \mathbf{A}_k + \Lambda_k \mathbf{L}^\top \mathbf{L}$, we get the following equivalencies for the `sTik` iterates

$$\begin{aligned} \mathbf{x}_k &= \mathbf{x}_{k-1} - \mathbf{B}_k \left(\mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) + \Lambda_k \mathbf{L}^\top \mathbf{L} \mathbf{x}_{k-1} \right) \\ &= \mathbf{B}_k \left(\mathbf{B}_k^{-1} \mathbf{x}_{k-1} - \mathbf{A}_k^\top \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{A}_k^\top \mathbf{b}_k - \Lambda_k \mathbf{L}^\top \mathbf{L} \mathbf{x}_{k-1} \right) \\ &= \mathbf{B}_k \left(\mathbf{B}_{k-1}^{-1} \mathbf{x}_{k-1} + \mathbf{A}_k^\top \mathbf{b}_k \right) = \mathbf{B}_k \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{b}_i = \mathbf{x}(\lambda_k). \end{aligned}$$

A similar proof can be made for (i). □

The above results are true for any arbitrary sequence of matrices $\{\mathbf{W}_k\}$. Next, we consider a fixed set of matrices, as described in the introduction, and allow random sampling from this set. To be precise, define $\mathbf{W}_{\tau(k)}$ to be a random variable at the kth iteration, where $\tau(k)$ is a random variable that indicates a sampling strategy. For example, if we let $\tau(k)$ be a uniform random variable on the set $\{1, \ldots, M\}$, then we would be sampling with replacement. In Section 4.2.1 we prove asymptotic convergence of `rrls` and `sTik` iterates using this sampling strategy. We then focus on random cyclic sampling, where for each $j \in \mathbb{N}$, $\{\tau(k)\}_{jM+1}^{(j+1)M}$ is a random permutation on the set $\{1, \ldots, M\}$. Note, cyclic sampling, where $\tau(k) = k \mod M$, is a special case of random cyclic sampling. We note that, until all blocks have been sampled, random cyclic sampling is just sampling without replacement. For random cyclic sampling, we characterize iterates after each epoch and prove asymptotic convergence of `rrls` and `sTik` iterates in Section 4.2.2. An illustrative example comparing the behavior of the solutions is provided in Section 4.2.3. For notational simplicity we denote $\mathbf{A}_{\tau(k)} = \mathbf{W}_{\tau(k)}^\top \mathbf{A}$ and $\mathbf{b}_{\tau(k)} = \mathbf{W}_{\tau(k)}^\top \mathbf{b}$.

Notice that for both random sampling and random cyclic sampling, we have the following property,

$$\mathbb{E}\left[\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^\top\right] = \frac{1}{M}\mathbf{I}_m = \frac{\ell}{m}\mathbf{I}_m\,. \tag{4.7}$$

There are many choices for $\{\mathbf{W}_i\}$, see e.g., [42, 108, 116], but a simple choice is a block column partition of a permutation matrix. For the choice of $\{\mathbf{W}_i\}$ we will consider, $\mathbf{A}_{\tau(k)}$ is just a predefined block of rows of $\mathbf{A}$.

4.2.1 Random Sampling

Next we investigate the asymptotic convergence of `rrls` and `sTik` iterates for the case of uniform random sampling. This is also referred to as sampling with replacement.

Theorem 4.2. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{L} \in \mathbb{R}^{s\times n}$ have full column rank and define $\mathbf{x}(\lambda)$ as in (4.2). Let $\{\mathbf{W}_i\}_{i=1}^M$ be a set of real valued $m \times \ell$ matrices with the property that $\sum_{i=1}^M \mathbf{W}_i\mathbf{W}_i^\top = \mathbf{I}_m$, and let $\tau(k)$ be a uniform random variable on the set $\{1,\dots M\}$.*

(i) Let $\lambda > 0$, $\mathbf{y}_0 \in \mathbb{R}^n$ be arbitrary, and define the sequence $\{\mathbf{y}_k\}$ as

$$\mathbf{y}_k = \mathbf{y}_{k-1} - \mathbf{B}_k\mathbf{A}_{\tau(k)}^\top(\mathbf{A}_{\tau(k)}\mathbf{y}_{k-1} - \mathbf{b}_{\tau(k)}), \quad k \in \mathbb{N}, \tag{4.8}$$

where $\mathbf{B}_k = \left(\lambda\mathbf{L}^\top\mathbf{L} + \sum_{i=1}^k \mathbf{A}_{\tau(i)}^\top\mathbf{A}_{\tau(i)}\right)^{-1}$. If $\mathbf{A}$ has full column rank, then $\mathbf{y}_k \xrightarrow{\text{a.s.}} \mathbf{x}(0)$.

(ii) *Let* $\sum_{i=1}^{k}\Lambda_i > 0$ *for all* k*, and* $\lambda = \lim_{k\to\infty}\frac{M}{k}\sum_{i=1}^{k}\Lambda_i > 0$ *be finite. Let* $\mathbf{x}_0 \in \mathbb{R}^n$ *be arbitrary, and define the sequence* $\{\mathbf{x}_k\}$ *as*

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k\left(\mathbf{A}_{\tau(k)}^\top(\mathbf{A}_{\tau(k)}\mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)}) + \Lambda_k\mathbf{L}^\top\mathbf{L}\mathbf{x}_{k-1}\right), \tag{4.9}$$

where $\mathbf{B}_k = \left(\sum_{i=1}^{k}\Lambda_i\mathbf{L}^\top\mathbf{L} + \sum_{i=1}^{k}\mathbf{A}_{\tau(i)}^\top\mathbf{A}_{\tau(i)}\right)^{-1}$*. Then* $\mathbf{x}_k \xrightarrow{\text{a.s.}} \mathbf{x}(\lambda)$*.*

Proof of Theorem 4.2. 1. From Theorem 4.1 for any $k \in \mathbb{N}$ we have

$$\begin{aligned}\mathbf{y}_k &= \left(\lambda\mathbf{L}^\top\mathbf{L} + \sum_{i=1}^{k}\mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{A}\right)^{-1}\left(\sum_{i=1}^{k}\mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{b} + \lambda\mathbf{L}^\top\mathbf{L}\mathbf{y}_0\right)\\ &= \left(\frac{\lambda\mathbf{L}^\top\mathbf{L} + \sum_{i=1}^{k}\mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{A}}{k}\right)^{-1}\left(\frac{\sum_{i=1}^{k}\mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{b} + \lambda\mathbf{L}^\top\mathbf{L}\mathbf{y}_0}{k}\right).\end{aligned}$$

Using the fact that $\mathbb{E}\,\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top = \frac{\ell}{m}\mathbf{I}_m$ (see equation (4.7)), by the law of large numbers and Slutsky's theorem for a.s. convergence [154]

$$\frac{\sum_{i=1}^{k}\mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{b} + \lambda\mathbf{L}^\top\mathbf{L}\mathbf{y}_0}{k} \xrightarrow{\text{a.s.}} \frac{\ell}{m}\mathbf{A}^\top\mathbf{b},$$

and

$$\left(\frac{\lambda\mathbf{L}^\top\mathbf{L} + \sum_{i=1}^{k}\mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{A}}{k}\right)^{-1} \xrightarrow{\text{a.s.}} \frac{m}{\ell}\left(\mathbf{A}^\top\mathbf{A}\right)^{-1}.$$

and therefore

$$\mathbf{y}_k \xrightarrow{\text{a.s.}} \left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{b} = \mathbf{x}(0).$$

2. In a similar fashion, for any $k \in \mathbb{N}$ we have

$$\mathbf{x}_k = \left(\frac{\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A}}{k} \right)^{-1} \left(\frac{\sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{b}}{k} \right).$$

Using the fact that $\mathbb{E}\, \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top = \frac{\ell}{m} \mathbf{I}_m$ and $\lim_{k\to\infty} \frac{\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L}}{k} = \frac{\ell}{m} \lambda \mathbf{L}^\top \mathbf{L}$, we have

$$\frac{\sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_i \mathbf{W}_i^\top \mathbf{b}}{k} \xrightarrow{\text{a.s.}} \frac{\ell}{m} \mathbf{A}^\top \mathbf{b}$$

and

$$\left(\frac{\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{A}^\top \mathbf{W}_i \mathbf{W}_i^\top \mathbf{A}}{k} \right)^{-1} \xrightarrow{\text{a.s.}} \frac{m}{\ell} \left(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{L}^\top \mathbf{L} \right)^{-1},$$

and thus we conclude that

$$\mathbf{x}_k \xrightarrow{\text{a.s.}} \left(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{L}^\top \mathbf{L} \right)^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{x}(\lambda).$$

□

The significance of Theorem 4.2 is that the `rrls` iterates converge asymptotically to the *unregularized* LS solution, $(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$, which is undesirable for ill-posed inverse problems. On the other hand, the `sTik` iterates converge asymptotically to a Tikhonov-regularized solution. Note that for a given λ, convergence to $\mathbf{x}(\lambda)$ is ensured by setting $\Lambda_k = \lambda / M$. A more realistic scenario would be to adapt Λ_k as data become available since the desired regularization parameter is typically not known before the data is received. Hence, parameter selection strategies for selecting Λ_k are addressed in Section 4.3.

4.2.2 Random Cyclic Sampling

Next we investigate `rrls` and `sTik` with random cyclic sampling. In addition to proving asymptotic convergence in this case, we can also describe the iterates as Tikhonov solutions after each epoch, where an epoch is defined as a sweep through all the data.

Theorem 4.3. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{L} \in \mathbb{R}^{s\times n}$ have full column rank and $\{\mathbf{W}_i\}_{i=1}^M$ be a set of real valued $m \times \ell$ matrices with the property that $\sum_{i=1}^M \mathbf{W}_i\mathbf{W}_i^\top = \mathbf{I}_m$, and let $\tau(k)$ be a random variable such that for $j \in \mathbb{N}$, $\{\tau(k)\}_{jM+1}^{(j+1)M}$ is a random permutation on the set $\{1,\dots,M\}$.*

1. *If $\lambda > 0$, $\mathbf{y}_0 = \mathbf{0}$, and the sequence $\{\mathbf{y}_k\}$ is defined as (4.8) with*

$$\mathbf{B}_k = \left(\lambda\mathbf{L}^\top\mathbf{L} + \sum_{i=1}^k \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1},$$

   then the `rrls` iterate at the jth epoch is $\mathbf{y}_{jM} = \mathbf{x}\left(\frac{1}{j}\lambda\right)$.

2. *Let $\{\Lambda_k\}$ be an infinite sequence with the property that $\lambda_k = \sum_{i=1}^k \Lambda_i > 0$. If $\mathbf{x}_0$ is arbitrary and the sequence $\{\mathbf{x}_k\}$ is defined as (4.9) with*

$$\mathbf{B}_k = \left(\sum_{i=1}^k \Lambda_i\mathbf{L}^\top\mathbf{L} + \sum_{i=1}^k \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1},$$

   then the `sTik` iterate at the jth epoch is $\mathbf{x}_{jM} = \mathbf{x}\left(\frac{1}{j}\lambda_{jM}\right)$.

Proof of Theorem 4.3. Notice that for random cyclic sampling schemes and for any iteration jM, $\sum_{i=1}^{jM} \mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{A} = j\mathbf{A}^\top\mathbf{A}$ and $\sum_{i=1}^{jM} \mathbf{A}^\top\mathbf{W}_{\tau(i)}\mathbf{W}_{\tau(i)}^\top\mathbf{b} = j\mathbf{A}^\top\mathbf{b}$ are deterministic. Hence

$$\mathbf{y}_{jM} = j\left(\lambda\mathbf{L}^\top\mathbf{L} + j\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{b} = \left(\frac{\lambda}{j}\mathbf{L}^\top\mathbf{L} + \mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{b} = \mathbf{x}\left(\tfrac{1}{j}\lambda\right)$$

and

$$\mathbf{x}_{jM} = j\left(\lambda_{jM}\mathbf{L}^\top\mathbf{L} + j\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{b} = \left(\frac{\lambda_{jM}}{j}\mathbf{L}^\top\mathbf{L} + \mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{b} = \mathbf{x}\left(\tfrac{1}{j}\lambda_{jM}\right).$$

□

Notice that at every epoch, the effective regularization parameter for `rrls`, i.e., λ/j, is reduced. Also, if $\mathbf{A}$ has full column rank, we have $\lim_{j\to\infty}\mathbf{y}_{jM} = \mathbf{x}(0)$. On the other hand, the `sTik` iterates converge to a Tikhonov-regularized solution, since at each epoch $j = k/M$ and we have $\mathbf{x}_{jM} = \mathbf{x}_k = \mathbf{x}\left(\frac{M}{k}\lambda_k\right)$ and $\frac{M}{k}\lambda_k > 0$. In Section 4.2.3 we illustrate the convergence behavior of the `rrls` and `sTik` iterates, but first we make some connections to existing optimization methods.

4.2.3 An Illustration

In the following illustration, we use a small toy example to highlight the convergence behavior of `rrls` and `sTik` iterates. We investigate both random sampling and random cyclic sampling, and we demonstrate convergence by plotting solutions after multiple epochs of the data. The example we use is a Tikhonov problem of the form (4.1), where

$$\mathbf{A} = \begin{bmatrix} \mathbf{1} & \boldsymbol{\delta}_{\mathbf{A}} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{10\times 2}, \qquad \mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\delta}_{\mathbf{b}}, \qquad \text{and} \qquad \mathbf{x}_{\text{true}} = \mathbf{1}.$$

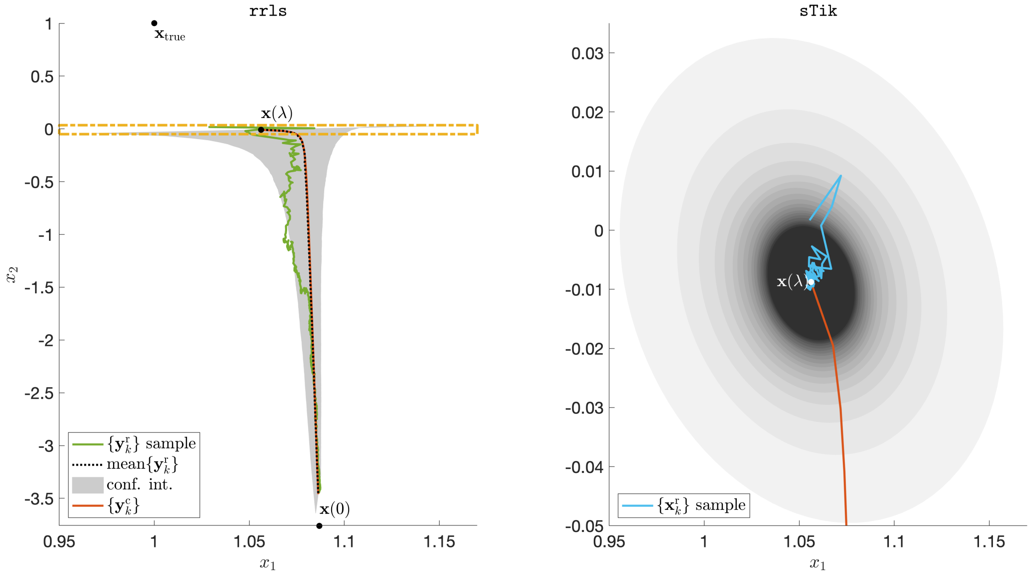

Figure 4.1: Illustration of convergence behaviors of **rrls** and **sTik** iterates. Shown in the left panel are the true solution $\mathbf{x}_{\text{true}}$, the unregularized solution $\mathbf{x}(0)$, the Tikhonov solution $\mathbf{x}(\lambda)$, and **rrls** iterates after multiple epochs. Both **rrls** with random sampling iterates $\{\mathbf{y}_k^{\text{r}}\}$ and **rrls** with random cyclic sampling iterates $\{\mathbf{y}_k^{\text{c}}\}$ converge asymptotically to the unregularized solution. In the right panel, we provide **sTik** with random sampling iterates $\{\mathbf{x}_k^{\text{r}}\}$ and confidence bounds. These iterates stay close to the Tikhonov solution. The axis for the right figure corresponds to the rectangular box in the left figure. The concentric gray circles represent the 95% confidence interval for these iterates after subsequent epochs.

The vectors $\boldsymbol{\delta}_{\mathbf{A}}$ and $\boldsymbol{\delta}_{\mathbf{b}}$ are realizations from the normal distributions $\mathcal{N}(\mathbf{0}, 0.005\,\mathbf{I}_9)$ and $\mathcal{N}(\mathbf{0}, 0.1\,\mathbf{I}_{10})$ respectively, and $\mathbf{1}$ is the vector of ones of appropriate length. We further choose $\mathbf{L} = \mathbf{I}_2$ and fix $\lambda = 0.2$ for the **rrls** iterates $\mathbf{y}_k$. For **sTik** iterates $\mathbf{x}_k$, we choose the parameters Λ_k such that the regularization is constant at each epoch, i.e., $\frac{10}{k}\sum_{i=1}^{k}\Lambda_i = 0.2$. With this setup we have $\mathbf{x}(0) = [1.0869, -1.3799]^\top$ and $\mathbf{x}(\lambda) = [1.0698, -0.0271]^\top$. We let $\mathbf{W}_{\tau(i)}$ be the $\tau(i)$th column of the identity matrix, and set $\mathbf{x}_0 = \mathbf{y}_0 = \mathbf{0}$.

In Figure 4.1, we provide two illustrations. In the left panel, we provide the true solution $\mathbf{x}_{\text{true}}$, the unregularized solution $\mathbf{x}(0)$, the Tikhonov solution $\mathbf{x}(\lambda)$, and the **rrls** iterates after each epoch. The **rrls** iterates with random sampling with replacement are denoted by $\mathbf{y}_k^{\text{r}}$, and the rrls iterates with random cyclic sampling are denoted by $\mathbf{y}_k^{\mathrm{c}}$. Notice that by Theorem 4.3, $\mathbf{y}_k^{\mathrm{c}}$ at each epoch is a Tikhonov solution, i.e., after the jth epoch $\mathbf{y}_{jM}^{\mathrm{c}} = \mathbf{x}\left(\frac{1}{j}\lambda\right)$. Thus, we get a set of Tikhonov solutions with vanishing regularization parameters, and these iterates asymptotically converge to the unregularized solution. For rrls with random sampling, we run 1,000 simulations and provide one sample path, along with the mean (dotted line) and region of the 95th percentile shaded in grey. We note that the mean of $\{\mathbf{y}_k^{\mathrm{r}}\}$ is almost identical to the random cyclic sequence $\{\mathbf{y}_k^{\mathrm{c}}\}$ (red line) suggesting that the random sequence $\{\mathbf{y}_k^{\mathrm{r}}\}$ is an unbiased estimator of the deterministic sequence $\{\mathbf{y}_k^{\mathrm{c}}\}$ (at each epoch). In the right panel of Figure 4.1, we provide the sTik iterates with random sampling, which are denoted by $\mathbf{x}_k^{\mathrm{r}}$. Again, we run 1,000 simulations and provide one simulation along with the shaded percentiles. It is evident that with more epochs, the iterates approach the desired Tikhonov solution. To aid with visual scaling, the axis for the right figure corresponds to the dotted rectangular box in the left figure. The sTik iterates with random cyclic sampling are omitted since $\mathbf{x}_{jM}^{\mathrm{c}} = \mathbf{x}(\lambda)$ (i.e., we get the Tikhonov solution after each epoch).

We observe that for random sampling, both rrls and sTik iterates contain undesirable uncertainties in the estimates. Although rrls iterates provide approximations to the Tikhonov solution, the main disadvantages are that the regularization parameter cannot be updated during the process and the iterates converge asymptotically to the unregularized solution. Hence, we disregard the rrls method and focus on sTik with *random cyclic sampling*, where λ can be updated via Λ_k.

4.3 Sampled Regularization Parameter Selection Methods

The ability to update the regularization parameter without sacrificing favorable convergence properties make the `sTik` and `slimTik` methods appealing for massive inverse problems. However, sampled regularization parameter selection methods must be developed to enable proper updates Λ_k. Adapting regularization parameters during iterative processes is not a new concept; however, much of the previous work in this area utilize projected systems, see, e.g., [104, 136], or are specialized to applications such as denoising [82]. Another common approach is to consider the unregularized problem and to terminate the iterative process before noise contaminates the solution. This phenomenon is called semi-convergence, and selecting a good stopping iteration can be very difficult. There have been investigations into semi-convergence behavior of iterative methods such as Kaczmarz, e.g., [55].

Unfortunately, standard regularization parameter selection methods are not feasible in this setting because many of them require access to the full residual vector, $\mathbf{r}(\lambda) = \mathbf{A}\mathbf{x}(\lambda) - \mathbf{b}$, which is not available. In this section, we investigate variants of existing regularization parameter selection methods [4, 8, 154] that are based on the sample residual. In the following we assume that at the kth iteration, Λ_i for $i = 1, \ldots, k-1$ have been computed. Then the goal is to determine an appropriate update parameter Λ_k. From Theorems 4.1 and 4.3, the kth `sTik` iterate can be represented as

$$\mathbf{x}_k(\lambda) = \mathbf{C}_k(\lambda)\mathbf{b}, \quad \text{where} \tag{4.10}$$

$$\mathbf{C}_k(\lambda) = \left(\left(\lambda + \sum_{i=1}^{k-1} \Lambda_i\right) \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A}\right)^{-1} \sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top .$$

Similar to standard regularization parameter selection methods, we assume that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$.

For methods that require estimates of σ^2, there are various ways that one can obtain such an estimate, see e.g., [49, 154].

4.3.1 Sampled Discrepancy Principle

The basic idea of the *sampled discrepancy principle (sDP)* is that at the kth iteration, the goal is to select the parameter Λ_k so that the sum of squared residuals for the current sample $\left\|\mathbf{W}_{\tau(k)}^\top(\mathbf{A}\mathbf{x}_k - \mathbf{b})\right\|_2^2$ is equal to $\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^\top\boldsymbol{\epsilon}\right\|_2^2$. Using properties of conditional expectation, we find

$$\begin{aligned}\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^\top\left(\mathbf{A}\mathbf{x}_{\text{true}} - \mathbf{b}\right)\right\|_2^2 &= \mathbb{E}\left\|\mathbf{W}_{\tau(k)}^\top\boldsymbol{\epsilon}\right\|_2^2\\ &= \mathbb{E}\,\mathbb{E}\left[\boldsymbol{\epsilon}^\top\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^\top\boldsymbol{\epsilon}\,|\,\boldsymbol{\epsilon}\right]\\ &= \sigma^2\text{tr}\left(\mathbb{E}\,\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^\top\right)\\ &= \sigma^2\ell,\end{aligned}$$

where $\text{tr}(\cdot)$ corresponds to the matrix trace function. Thus, at the kth iteration and for a given realization, we select λ such that

$$\left\|\mathbf{W}_{\tau(k)}^\top\left(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\right)\right\|_2^2 \approx \gamma\sigma^2\ell\,,$$

where $\gamma > 1$ is a predetermined real number. For the sampled methods, we select λ_k that solves the optimization problem,

$$\min_{\lambda} \quad \left(\|\mathbf{W}_{\tau(k)}^\top\left(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\right)\|_2^2 - \gamma\sigma^2\ell\right)^2, \tag{4.11}$$

where $\gamma = 4$ as suggested in [81, 154] and σ^2 is the true noise variance.

4.3.2 Sampled Unbiased Predictive Risk Estimator

Next, we describe a method to select Λ_k based on a *sampled unbiased predictive risk estimator (sUPRE)*. The basic idea is to find Λ_k to minimize the sampled predictive risk,

$$\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{A}\mathbf{x}_{\text{true}})\right\|_2^2 ,$$

which is equivalent to

$$\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}\left(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\right)\right\|_2^2 + 2\sigma^2\,\mathbb{E}\,\mathrm{tr}\big(\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\big) - \sigma^2\ell\,.$$

See B.0.1 for details of the derivation. Then, similar to the approach used in the standard UPRE derivation, the parameter Λ_k is selected by finding a minimizer of the unbiased estimator for the sampled predictive risk,

$$U_k(\lambda) = \left\|\mathbf{W}_{\tau(k)}^{\top}\left(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\right)\right\|_2^2 + 2\sigma^2\mathrm{tr}\big(\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\big) - \sigma^2\ell\,, \tag{4.12}$$

for a given realization.

4.3.3 Sampled Generalized Cross Validation

Lastly, we describe the *sampled generalized cross validation (sGCV)* method for selecting Λ_k and point the interested reader to B.0.2 for details of the derivation. The basic idea is to use a "leave-one-out" cross validation approach to find a value of Λ_k, but the main differences compared to the standard GCV method are that at the kth iteration, we only have access to the sample residual and the iterates only correspond to Tikhonov solutions with only partial data. The parameter λ_k is selected by finding a minimizer of the sGCV function,

$$G_k(\lambda) = \frac{\ell \left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}) \right\|_2^2}{\mathrm{tr}\Big(\mathbf{I}_\ell - \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\Big)^2} = \frac{\ell \left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}) \right\|_2^2}{\Big(\ell - \mathrm{tr}\Big(\mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\Big)\Big)^2}. \tag{4.13}$$

4.3.4 A Second Illustration

In this example we investigate the behavior of the previously discussed sampled regularization parameter update strategies, i.e., sDP, sUPRE, and sGCV, for multiple ill-posed inverse problems from the Matlab matrix gallery and from P. C. Hansens' Regularization Tools toolbox [79, 115]. For simplicity, we set $m = n = 100$ and use the true solutions $\mathbf{x}_{\text{true}}$ that are provided by the toolbox. If no true solution is provided, we set $\mathbf{x}_{\text{true}} = \mathbf{1}$. We let $\mathbf{L} = \mathbf{I}_{100}$, and set $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.01\,\mathbf{I}_{100})$. Sampling matrices $\mathbf{W}_j \in \mathbb{R}^{100\times 10}$ are given as $\mathbf{W}_j = [\mathbf{0}_{10(j-1)\times 10}; \mathbf{I}_{10}; \mathbf{0}_{10(10-j)\times 10}]$ for $j = 1, \ldots, 10$, such that $\mathbf{A}$ and $\mathbf{b}$ are sampled in 10 consecutive blocks. Here, we sample $\mathbf{W}$ in a random cyclic fashion and let σ^2 be the true noise variance for sDP and sUPRE.

We first consider the `prolate` example where $\mathbf{A}$ is an ill-conditioned Toeplitz matrix from Matlab's matrix gallery. In Figure 4.2 we illustrate the asymptotic behavior of the sampled parameter selection strategies by plotting the number of epochs against the value of λ for sDP, sUPRE, and sGCV. For comparison, we provide the regularization parameter for the full problem corresponding to DP, UPRE, and GCV. DP and UPRE use the true noise variance, and γ is as above for DP. For comparison, we also provide the optimal parameter λ_{opt} for the full problem, which is the parameter that minimizes the 2-norm of the error between the reconstruction and the true solution. This last approach is not possible in practice. We observe that with more iterations, the sampled regularization parameter selection methods tend to "stabilize" in that after some point, they do not change much. The sDP regular-

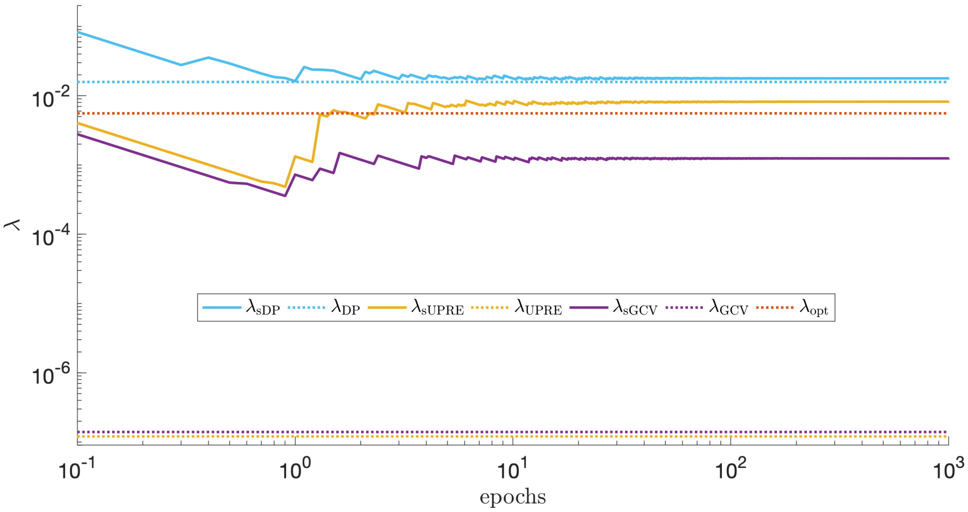

Figure 4.2: "Asymptotic" behavior of the sampled regularization parameter selection methods for the `prolate` example. Corresponding regularization parameters computed using the full data are provided as horizontal lines for comparison.

ization parameter stabilizes near the DP parameter for the full problem, but both sUPRE and sGCV stabilize closer to the optimal regularization parameter. While we observe similar results for other test problems (results not shown), the sampled regularization parameters may not necessarily be close to the corresponding parameter for the full system. Nevertheless, the sampled regularization parameter selection methods often lead to appropriate reconstructions $\mathbf{x}_k(\lambda)$ after a moderate number of iterations. Next, we investigate the relative reconstruction error $\left\|\mathbf{x}_k(\lambda)-\mathbf{x}_{\rm true}\right\|_2 / \left\|\mathbf{x}_{\rm true}\right\|_2$ of sampled regularization methods after *one* epoch (corresponding to $k = 10$). Figure 4.3 illustrates results from four test problems (`prolate`, `baart`, `shaw`, and `gravity`). First note that by Theorem 4.3, all solutions are Tikhonov solutions for a λ determined by the method, hence all relative reconstruction errors lie on a curve of relative errors for Tikhonov solutions.

We note that the above regularization parameter selection methods (including the standard DP, UPRE, and GCV) can only provide empirical estimations. However, we observe that in terms of relative reconstruction errors, our sampled regularization parameter selection methods perform reasonably well on the test problems.

As we have shown, our sampled regularization parameter selection methods can be used to update the regularization parameter in the sTik method, where the main benefit is the favorable convergence property. In the next section, we turn our attention to problems where it may be infeasible to construct or work with the $n \times n$ matrix $\mathbf{B}_k$. Although reduced models or subspace projection methods may be used to reduce the number of unknowns, obtaining a realistic basis for the solution may be difficult.

4.4 The slimTik Method

For the massive Tikhonov least squares problem in (4.1), the sTik method is not feasible due to the size of n. In this section we first introduce the slimTik algorithm as a limited memory variant of sTik, then we show how slimTik with random uniform sampling is nothing more that the slimLS algorithm introduced in Chapter 3 applied to the Tikhonov LS problem in (4.1). The connection between slimTik and slimLS allows us to prove almost sure convergence of the slimTik algorithm under random uniform sampling.

4.4.1 Limited Memory sTik

To avoid construction of the $n \times n$ $\mathbf{B}_k$, iterates of sTik iterates defined in (4.4) with a sampling method τ, can be equivalently defined as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{s}_k, \tag{4.14}$$

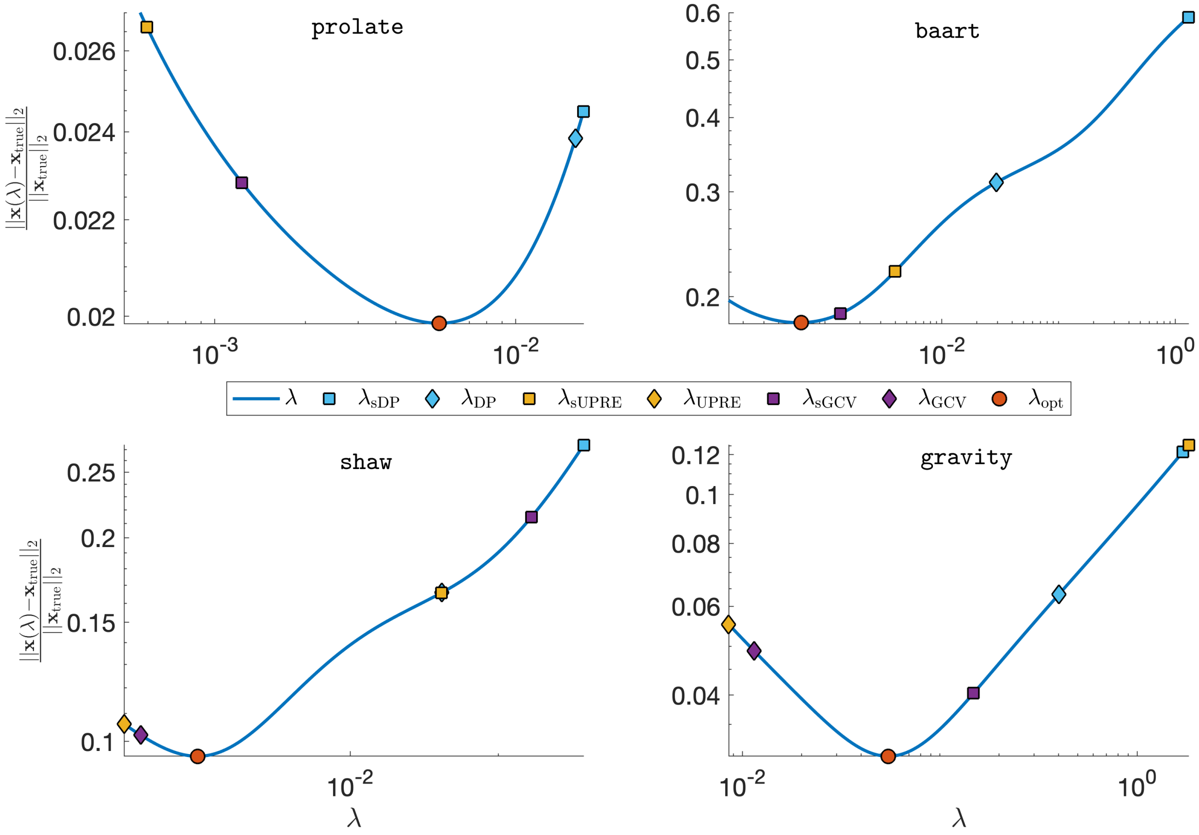

Figure 4.3: Relative reconstruction errors of the sampled and full regularization methods for four test problems `prolate`, `baart`, `shaw`, and `gravity`. All solutions lie on the solid line, which corresponds to relative errors for Tikhonov solutions. Note that the UPRE and GCV estimation in the `prolate` and `baart` test problem underperform significantly and are therefore omitted. The relative errors for λ_{sUPRE} and λ_{DP} coincide in the `shaw` example.

where

$$\mathbf{s}_k = \arg\min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{A}_{\tau(1)} \\ \vdots \\ \mathbf{A}_{\tau(k-1)} \\ \mathbf{A}_{\tau(k)} \\ \sqrt{\sum_{i=1}^{k} \Lambda_i}\mathbf{L} \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{A}_{\tau(k)}\mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)} \\ \frac{\Lambda_k}{\sqrt{\sum_{i=1}^{k} \Lambda_i}}\mathbf{L}\mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2 .$$

With this reformulation, we must solve a LS problem with matrix $\begin{bmatrix} \mathbf{A}_{\tau(k)}^\top & \cdots & \mathbf{A}_{\tau(k)}^\top \end{bmatrix}^\top$, which grows with each iteration. To avoid this computation we select a memory parameter $r \in \mathbb{N}_0$ and define $\mathbf{M}_k = \begin{bmatrix} \mathbf{A}_{\tau(k-r)}^\top & \cdots & \mathbf{A}_{\tau(k)}^\top \end{bmatrix}^\top \in \mathbb{R}^{r\ell \times n}$ and $\mathbf{A}_{\tau(k-r)} = \mathbf{0}$ for non-positive integers $k - r$. The `slimTik` iterates are given as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \tilde{\mathbf{s}}_k, \tag{4.15}$$

where

$$\tilde{\mathbf{s}}_k = \arg\min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{M}_k \\ \sqrt{\sum_{i=1}^{k} \Lambda_i}\mathbf{L} \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_{\tau(k)}\mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)} \\ \frac{\Lambda_k}{\sqrt{\sum_{i=1}^{k} \Lambda_i}}\mathbf{L}\mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2 . \tag{4.16}$$

Notice that the LS problem in (4.16) with the matrix $\mathbf{M}_k$ does not get arbitrarily large. By sacrificing past samples, `slimTik` is a computational feasible limited memory variant of `sTik`.

4.4.2 Connection to `slimLS`

The iterates of `slimTik` defined in (4.15) can be seen as the iterates of `slimLS` defined in (3.11) applied to the Tikhonov LS problem (4.1). The Tikhonov LS problem in (4.1) may be re-written as

$$\mathbf{x}_{\text{Tik}} = \arg\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 . \tag{4.17}$$

Let $\mathbf{W} \in \mathbb{R}^{m \times \ell}$ be a random variable defined as in Section 3.2.1 that has a uniform distribution across $M \in \mathbb{N}$ realizations $\{\mathbf{W}^{(i)}\}_{i=1}^{M}$. (for example, the generalized Kaczmarz matrices satisfy this condition). We can now define $\widehat{\mathbf{W}} \in \mathbb{R}^{m+n \times \ell+n}$ as

$$\widehat{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & \mathbf{0}_{m \times n} \\ \mathbf{0}_{n \times \ell} & \frac{1}{\sqrt{M}}\mathbf{I}_n \end{bmatrix} . \tag{4.18}$$

The random variable $\widehat{\mathbf{W}}$ has the property that $\mathbb{E}\left[\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top\right] = \beta \mathbf{I}_{m+n}$. The Tikhonov least square problem may be reformulated as the stochastic optimization problem

$$\begin{aligned} \arg\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 &= \arg\min_{\mathbf{x}} \mathbb{E} \left\| \widehat{\mathbf{W}}^\top \left(\begin{bmatrix} \mathbf{A} \\ \lambda\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right) \right\|_2^2 \\ &= \arg\min_{\mathbf{x}} \mathbb{E} \left\| \mathbf{W}^\top \left(\mathbf{A}\mathbf{x} - \mathbf{b} \right) \right\|_2^2 + \frac{\lambda}{M} \left\| \mathbf{L}\mathbf{x} \right\|_2^2 . \end{aligned} \tag{4.19}$$

For the random variable $\mathbf{W}$, $\mathbf{W}^\top \mathbf{A} = \mathbf{A}_{\tau(k)}$, where $\tau(k)$ is the random uniform variable on the set $\{1, \ldots, M\}$ and $\mathbf{A}_k = \left(\mathbf{W}^{(k)}\right)^\top \mathbf{A}$. For $\mathbf{C}_k = \mathbf{L}^\top \mathbf{L}$ and step size $\hat{\alpha}_k^{-1} = \frac{r+1}{M}\left(k\lambda - r + 1\right)$

applying `slimLS` to (4.19) defines iterates

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \left(\frac{k\lambda}{M} \mathbf{L}^\top \mathbf{L} + \mathbf{M}_k^\top \mathbf{M}_k \right)^{-1} \left(\mathbf{A}_{\tau(k)}^\top \left(\mathbf{A}_{\tau(k)} \mathbf{x}_k - \mathbf{b}_{\tau(k)} \right) + \frac{\lambda}{M} \mathbf{x}_{k-1} \right)$$

this is precisely the same as the `slimTik` iterates defined in (4.15), where the regularization parameter Λ_i is assumed to be fixed $\Lambda_i = \frac{\lambda}{M}$ for all $i \in \mathbb{N}$. Using the analysis from Section 3.4 we can show `slimTik` will almost surely converge to the Tikonov solution (1.1).

Theorem 4.4. *Let* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{b} \in \mathbb{R}^m$*. Let* $\mathbf{L} \in \mathbb{R}^{s \times n}$ *have full column rank and define* $\mathbf{x}(\lambda)$ *as in* (4.2)*. Let* $\{\mathbf{W}_i\}_{i=1}^M$ *be a set of real valued* $m \times \ell$ *matrices with the property that* $\sum_{i=1}^M \mathbf{W}_i \mathbf{W}_i^\top = \mathbf{I}_m$*, and let* $\tau(k)$ *be a uniform random variable on the set* $\{1, \dots M\}$*. For* $\lambda > 0$ *and* $\mathbf{x}_0 \in \mathbb{R}^n$*, define the sequence* $\{\mathbf{x}_k\}$ *as*

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \left(\frac{k\lambda}{M} \mathbf{L}^\top \mathbf{L} + \mathbf{M}_k^\top \mathbf{M}_k \right)^{-1} \left(\mathbf{A}_{\tau(k)}^\top \left(\mathbf{A}_{\tau(k)} \mathbf{x}_k - \mathbf{b}_{\tau(k)} \right) + \frac{\lambda}{M} \mathbf{x}_{k-1} \right)$$

with

$$\mathbf{M}_k = \begin{bmatrix} \mathbf{A}_{\tau(k-r)}, & \dots, & \mathbf{A}_{\tau(k)} \end{bmatrix}^\top,$$

where

- $\sum \alpha_k = \infty$ *and* $\sum \alpha_k^2$ *converges and*
- $\left\| \mathbf{A}_{\tau(k)}^\top \left(\mathbf{A}_{\tau(k)} \mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)} \right) + \frac{\lambda}{M} \mathbf{x}_{k-1} \right\|_2 \leq g$ *for* $g \geq 0$ *and all* $k \in \mathbb{N}$

Then $\mathbf{x}_k \xrightarrow{\text{a.s.}} \mathbf{x}(\lambda)$.

Proof. This result follows from Theorem 3.3, using the $\widehat{\mathbf{W}}$ in (4.18) as the choice of $\mathbf{W}$, applied to the LS problem (4.17). □

Theorem 4.4 is the first convergence result for a limited memory variant of the recursive LS algorithm. However, the connection between the full recursive LS and stochastic approximation methods was noted in [105]. The connection between slimTik and slimLS also reveals that the regularization parameter at the kth iteration Λ_k can also be considered a step size. Sampled methods for finding a good Λ_k discussed in Section 4.3 can be seen as a search for an optimal step size.

4.4.3 A Third Illustration

We illustrate convergence of slimTik for an example from the *Regularization Tools* toolbox [78]. We use the gravity example which provides a matrix $\mathbf{A} \in \mathbb{R}^{1{,}000\times 1{,}000}$ and a vector $\mathbf{x}_{\text{true}}$. We partition $\mathbf{A}$ into $M = 100$ blocks with $\ell = 10$ and let $\lambda = 0.0196$. We simulate observed data by adding Gaussian white noise with zero mean such that the noise level is 0.01, i.e., $\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}$ where $\frac{\|\boldsymbol{\epsilon}\|_2}{\|\mathbf{A}\mathbf{x}_{\text{true}}\|_2} = 0.01$. First, we run slimTik for one epoch under cyclic sampling with memory levels $r = 0, \ldots, M-1$, and we report the relative error between the reconstructions $\mathbf{x}_M$ and the Tikhonov solution $\mathbf{x}_{\text{Tik}}$ in the left panel of Fig. 4.4. Note that for full memory (i.e., $r = M-1$), the relative error is within machine precision. Also, for lower memory levels, the reconstructions $\mathbf{x}_M$ are close to the Tikhonov solution. The right panel of Fig. 4.4 illustrates the asymptotic convergence of slimTik for memory levels $r = 0, 2, 4, 6,$ and 8, where we also compare to a standard sampled gradient (sg) method without regularization. Errors are plotted after each full epoch. Empirically, we observe that the iterates $\mathbf{x}_k$ converge to $\mathbf{x}_{\text{Tik}}$ as $k \to \infty$

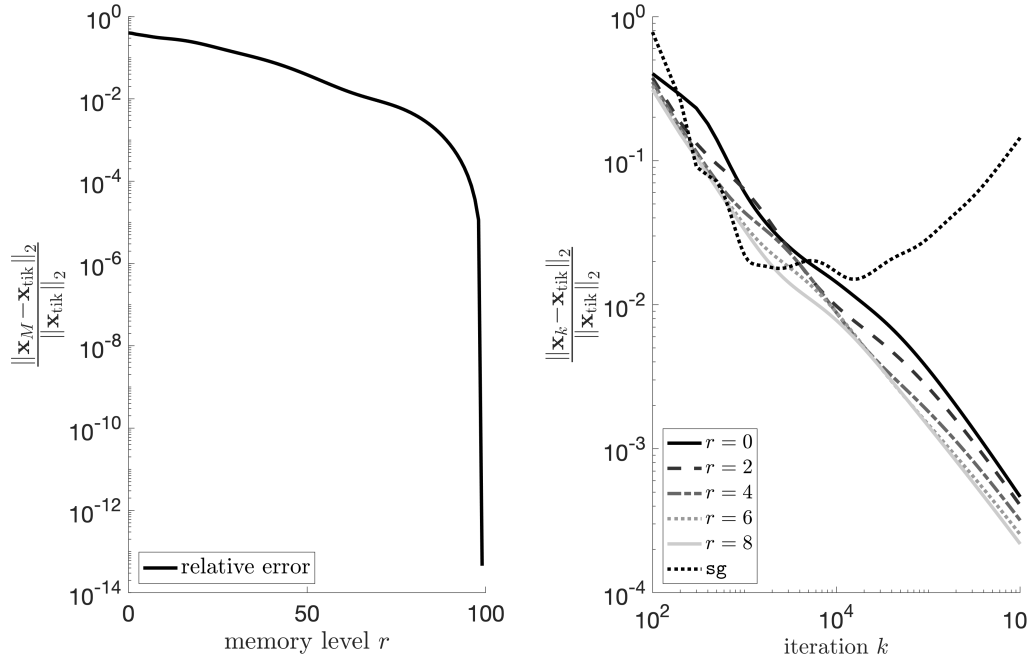

Figure 4.4: Convergence of the slimTik method. The plot in the left panel contains the relative errors between the iterates after one epoch and the Tikhonov solution, for different memory levels. The plot in the right panel illustrates asymptotic convergence of the slimTik method for memory levels $r = 0, 2, 4, 6$, and 8. For comparison we include relative errors for a sample gradient method.

4.5 Numerical Experiments

In this section we present three numerical experiments. Experiment 4 in Section 4.5.1 illustrates the convergence of `slimTik` compared to other methods *that do not contain regularization* for a large scale tomography problem. Experiment 5 in Section 4.5.2 considers convergence of the sampled regularized parameter selection methods for a small test problem. Experiment 6 in Section 4.5.3 applies `slimTik` to a massive super resolution problem coupled with sampled regularization parameter selection methods, showing that these methods can efficiently be applied to massive inverse problems.

4.5.1 Experiment 4: `slimTik` Applied to X-ray Tomography

Here we illustrate the benefits of `slimTik` with a large scale tomography test problem. Tomography is an imaging method where waves are detected after they transmit through and object of interest, see Figure 4.5 Here we look at an example of 2D parallel-beam x-ray tomography where we wish to reconstruct the interior densities of an object using projections of 362 parallel waves at 1,790 different angles. This problem comes from the AIRtools toolbox [80]. The true image is a 512 × 512 Shepp-Logan phantom. We have $\theta = 0 : 0.1 : 179$ as our angles, with 362 rays for each angle. Figure 4.6 shows the noisy sinogram. This produces a linear system where the $\mathbf{A}$ matrix is 916,480 × 262,144. We assume that the measurements are quite noisy with $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.6\mathbf{I}_m)$. Here $\{\mathbf{W}_i\}_{i=1}^{1,179}$ are Kaczmarz blocks of size $362 \times 262{,}144$ such that $\mathbf{A}_i$ and $\mathbf{b}_i$ are the model and corresponding projection for the angle $\theta_i = 0.1 * i$. We sample random and uniformly from these blocks. We compare `slimTik` applied to the Tikhonov problem with $\lambda = 0.007$ with memory level 3 and step size $\alpha_k = \frac{1}{0.7k}$ to the stochastic gradient method defined in Section 3.2, the Block Kaczmarz method defined in Section 2.5, and a stochastic LBFGS method. The stochastic LBFGS method is a stochastic optimization method that updates $\mathbf{B}_k$ with past iterates to approximate the inverse Hessian $\left(\mathbf{A}^\top \mathbf{A}\right)^{-1}$, for details see [26, 70, 120]. To allow the stochastic LBFGS method the same memory allocation as `slimTik` we set the memory level to $362 * 3 = 1{,}086$. The stochastic gradient method and stochastic LBFGS method required a small initial step size $\alpha_k = \frac{1}{120+0.7k}$ to prevent iterates from getting very large.

In Fig. 4.7, we provide the absolute errors images of the reconstructions, computed as $|\mathbf{x}_k - \mathbf{x}_{\text{true}}|$ for different values of k. The relative error, calculated by $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\| / \mathbf{x}_{\text{true}}$ is shown in Figure 4.8 After 5,372 iterations, we see that `slimTik` out preforms the other methods, with a relative error of .13. Additionally, the reconstruction from `slimTik` appears smoother because that the other reconstructions, see Figure 4.7.

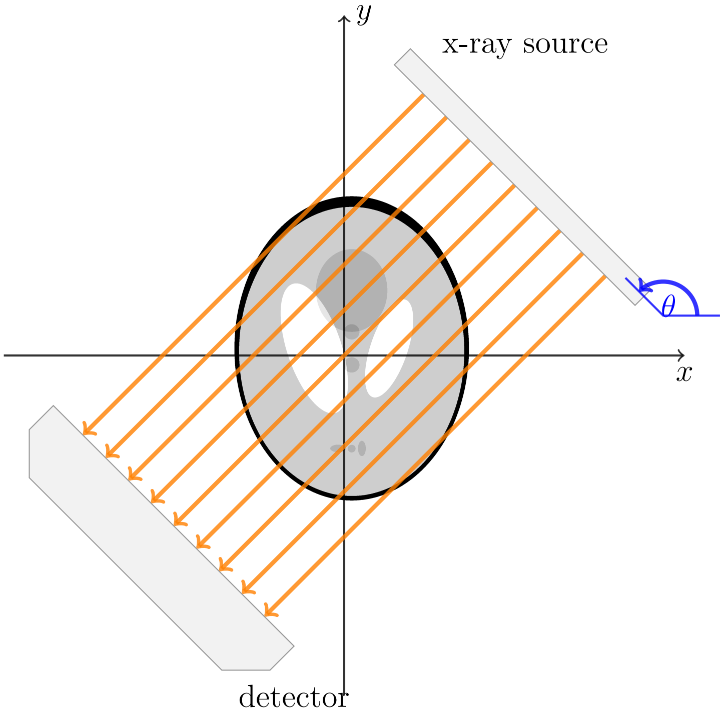

Figure 4.5: Experiment 4: Illustration of a 2-d parallel beam tomography set-up.

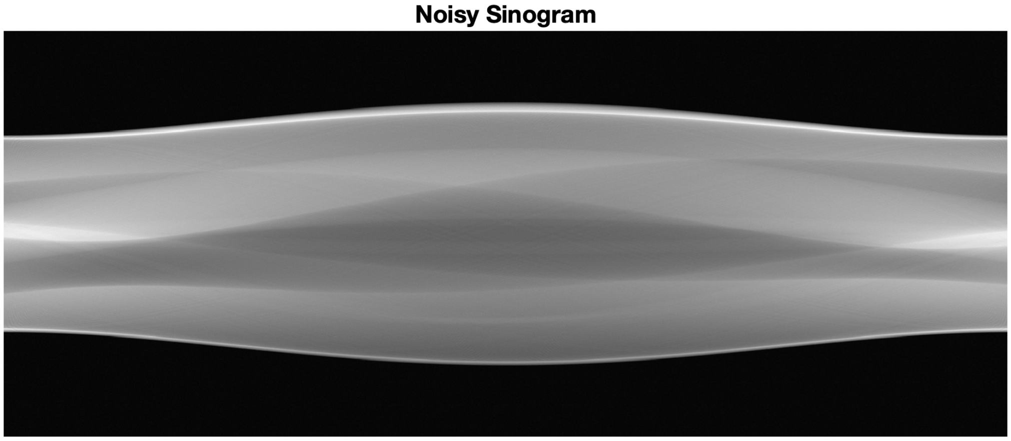

Figure 4.6: Experiment 4: The noisy sinogram from the parallel beam tomography problem.

4.5.2 Experiment 5: Convergence of Sampled Regularization Methods

In this chapter, we addressed some of the computational concerns and demonstrated our methods on a large imaging problem. First, we reformulate the updates as solutions to LS problems so that iterative methods can be used to compute approximations efficiently. In addition to being computationally feasible, these methods can take advantage of the adaptive regularization parameter selection methods described in Section 4.3.

These methods are based on the `sTik` method. In particular, we consider a sampled gradient (`sg`) method where the iterates are defined as (4.4) where $\mathbf{B}_k = \left(\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{I}_n\right)^{-1}$ and a sampled block Kaczmarz (`sbK`) method where the iterates are defined as (4.4) with $\mathbf{B}_k = \left(\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1}$. We also consider `slimTik`, which we described in 4.4. In the case where $r = 0$, `slimTik` and `sbK` iterates are identical. First, we investigate the performance of `sg`, `sbK`, and `slimTik` while taking advantage of the regularization parameter update described in Section 4.3. We use the `gravity` example from Regularization Tools,

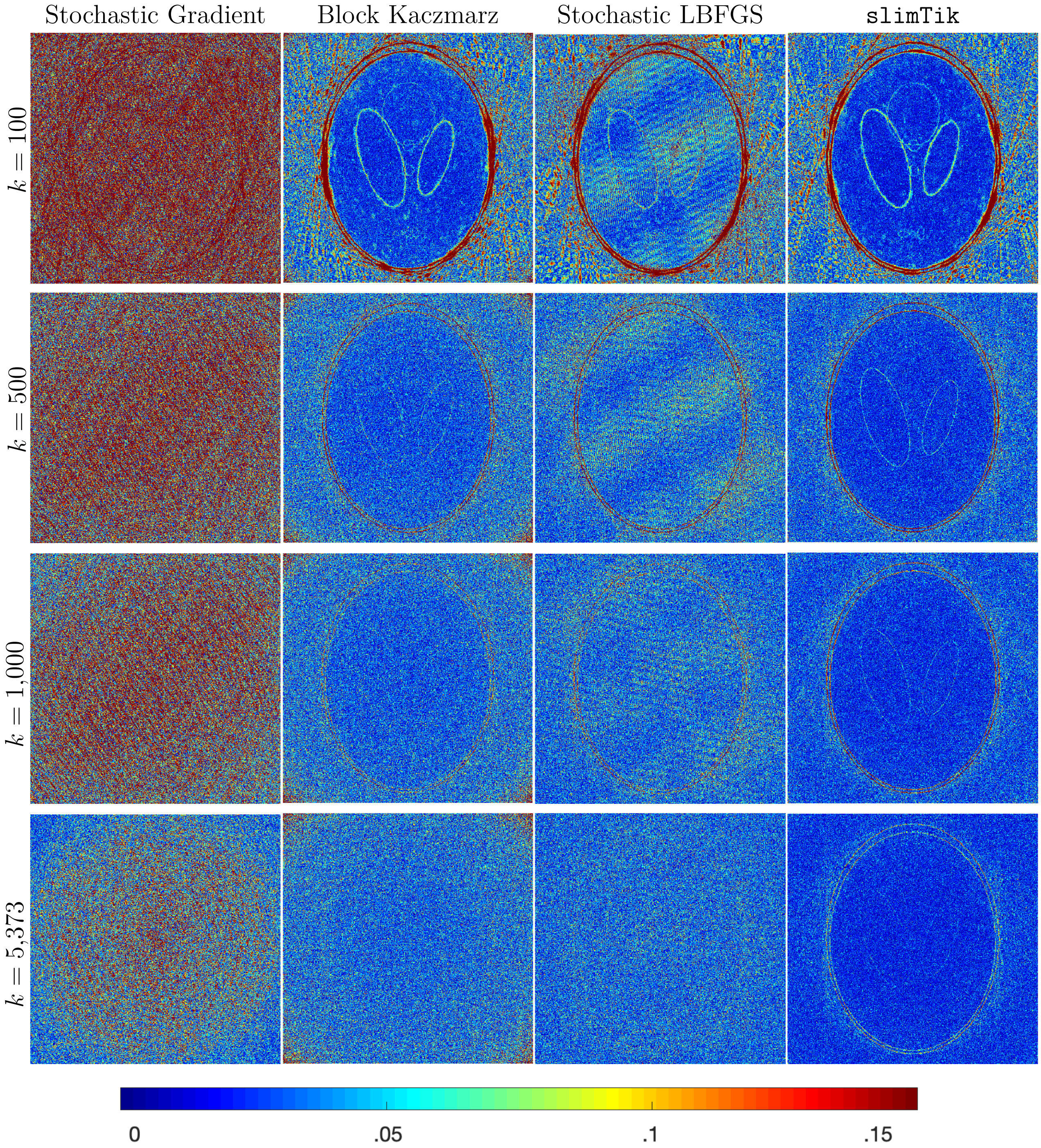

Figure 4.7: Experiment 4: Error images the reconstructed images for the 2-D tomography example. Reconstructions correspond to `sg`, `sbK`, `sLBFGS` and `slimTik`. For comparisons we provide the error image after 100, 500, 1000, and 5,373 iterations.

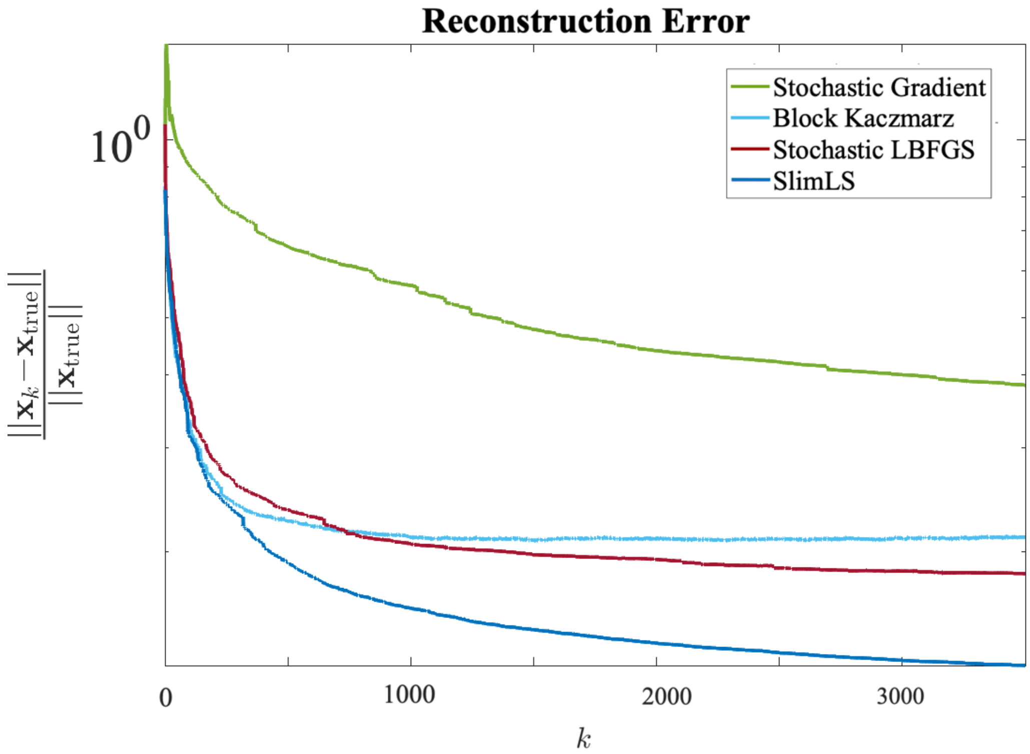

Figure 4.8: Experiment 4: Comparing the relative error to the true images at each iteration of the `sg`, `sbK`, `sLBFGS` and `slimTik` algorithms.

where $\mathbf{A} \in \mathbb{R}^{1{,}000 \times 1{,}000}$, $\mathbf{L} = \mathbf{I}_{1{,}000}$, and the noise level defined as $\|\boldsymbol{\epsilon}\|_2 \,/\, \|\mathbf{A}\mathbf{x}_{\text{true}}\|_2$ is 0.01. The samples consist of 10 blocks, each comprised of 100 consecutive rows of $\mathbf{A}$. The initial guess for the regularization parameter is chosen to be 0.1 (the optimal overall regularization parameter in this example is approximately 0.0196), and we iterate for one epoch.

In Figure 4.9 we provide the relative reconstruction errors per iteration for `sg`, `sbK`, `slimTik`, and `sTik`. Overall, we notice a correspondence between the amount of curvature information used to approximate the Hessian and an improvement in the relative reconstruction error. Including more curvature results in greater computational costs and storage requirements, e.g., `sTik` may be infeasible for very large problems, but the number of row accesses is the same for each method. In terms of regularization parameter selection methods, sGCV performs better than sUPRE and sDP for this example. The relative reconstruction error corresponding to the best overall Tikhonov solution is provided as the horizontal line. Although the results are not shown here, we note that without regularization, the relative reconstruction errors will become very large for all of these methods due to semi-convergence.

4.5.3 Experiment 6: Super Resolution

Having demonstrated that regularization parameter update methods can be incorporated in a variety of stochastic optimization methods, we investigate the performance of these limited-memory methods for super-resolution image reconstruction. The basic goal of super-resolution imaging is to reconstruct an $n \times n$ high-resolution image represented by a vector $\mathbf{x}_{\text{true}} \in \mathbb{R}^{n^2}$ given M low-resolution images of size $\ell \times \ell$ represented by $\mathbf{b}_1, \ldots, \mathbf{b}_M$, where $\mathbf{b}_i \in \mathbb{R}^{\ell^2}$. The forward model for each low-resolution image is given as

$$\mathbf{b}_i = \mathbf{R}\mathbf{S}_i\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}_i \,,$$

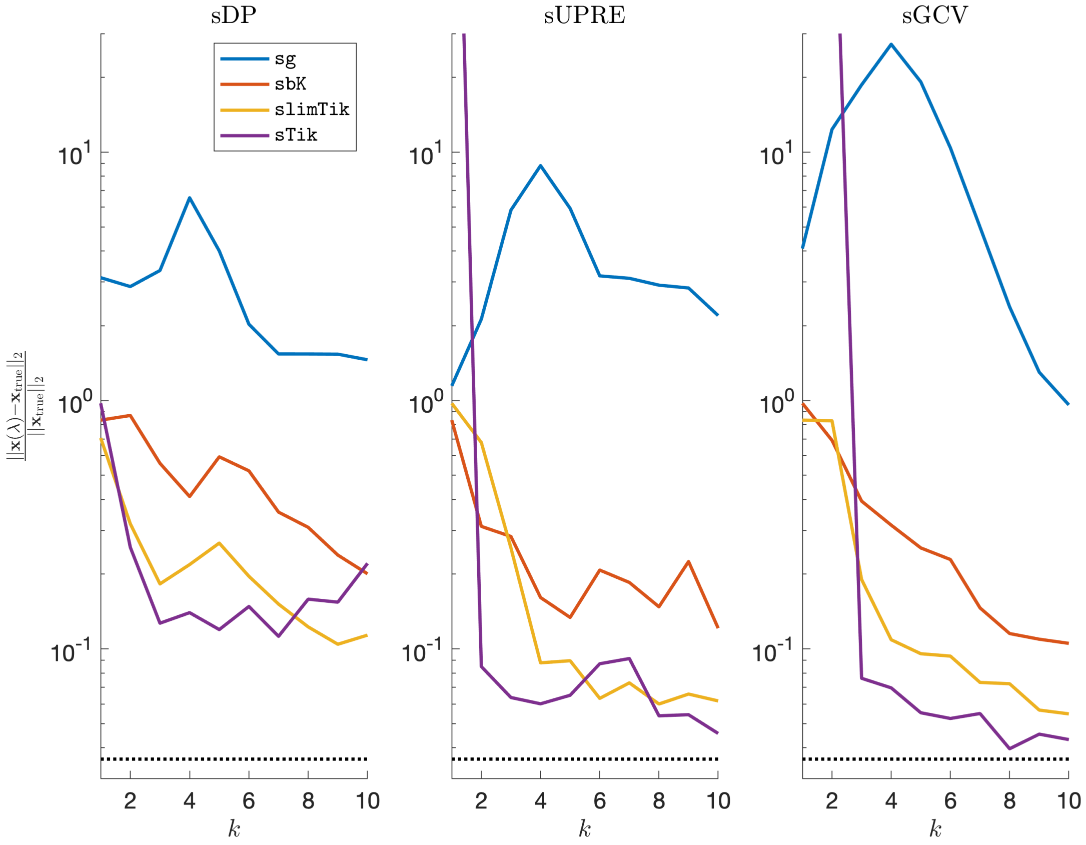

Figure 4.9: Experiment 5: Comparison of relative reconstruction errors for `sg`, `sbK`, `slimTik`, and `sTik` iterates for `gravity` using various sampled regularization parameter selection methods for the first 10 iterations, i.e., one epoch. We compare sDP, sUPRE, and sGCV. The horizontal black line is the relative error corresponding to the optimal regularization parameter for the full problem, which is not feasible to obtain in practice.

where $\mathbf{R} \in \mathbb{R}^{\ell^2 \times n^2}$ is a restriction matrix, $\mathbf{S}_i \in \mathbb{R}^{n^2 \times n^2}$ represents an affine transformation that may account for shifts, rotations, and scalar multiplications, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\mathbf{0}_{\ell^2}, \sigma^2 \mathbf{I}_{\ell^2}\right)$. To reconstruct a high-resolution image, we solve the Tikhonov problem,

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{R}\mathbf{S}_1 \\ \vdots \\ \mathbf{R}\mathbf{S}_M \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{bmatrix} \right\|_2^2 + \lambda \left\| \mathbf{L}\mathbf{x} \right\|_2^2 .$$

For cases where the low-resolution images are being streamed or where the number of low-resolution images is very large, standard iterative methods may not be feasible. Furthermore, it can be very challenging to determine a good choice of λ prior to solution computation [41, 91, 134].

For our example, we have 30 images of size 128 × 128, and we wish to reconstruct a high-resolution image of size 2,048 × 2,048. In Figure 4.10, we provide the true high-resolution image of the moon [34] and three of the low-resolution images. Here, $\mathbf{A}_i = \mathbf{R}\mathbf{S}_i \in \mathbb{R}^{128^2 \times 2{,}048^2}$. Due to the inherent partitioning of the problem, we take $\mathbf{W}_i^\top \in \mathbb{R}^{128^2 \times 30 \cdot 128^2}$ to be a matrix such that $\mathbf{W}_i^\top \mathbf{A} = \mathbf{A}_i$; these $\mathbf{W}_i$ matrices are never computed. For the simulated low-resolution images, Gaussian white noise is added such that the noise level for each image is 0.01 and take $\mathbf{L} = \mathbf{I}_{2{,}048^2}$. Notice that the size of the matrix $\mathbf{A}$ is 491,520 × 4,194,304, and holding $\mathbf{A}$ in computer memory is impractical despite its sparse structure.

We compare the performances of `sg`, `sbK`, and `slimTik`, including our sampled regularization parameter update methods sDP, sUPRE, and sGCV. The true noise variance is used for sDP and sUPRE, and the memory parameter for `slimTik` is $r = 2$. Each iteration of `sbK` and `slimTik` requires a linear solve, which can be handled efficiently by reformulating the problem as a LS problem as in equation (4.16), and using standard techniques such as LSQR [131, 132]. These iterative methods can also be used to update the regularization parameter.

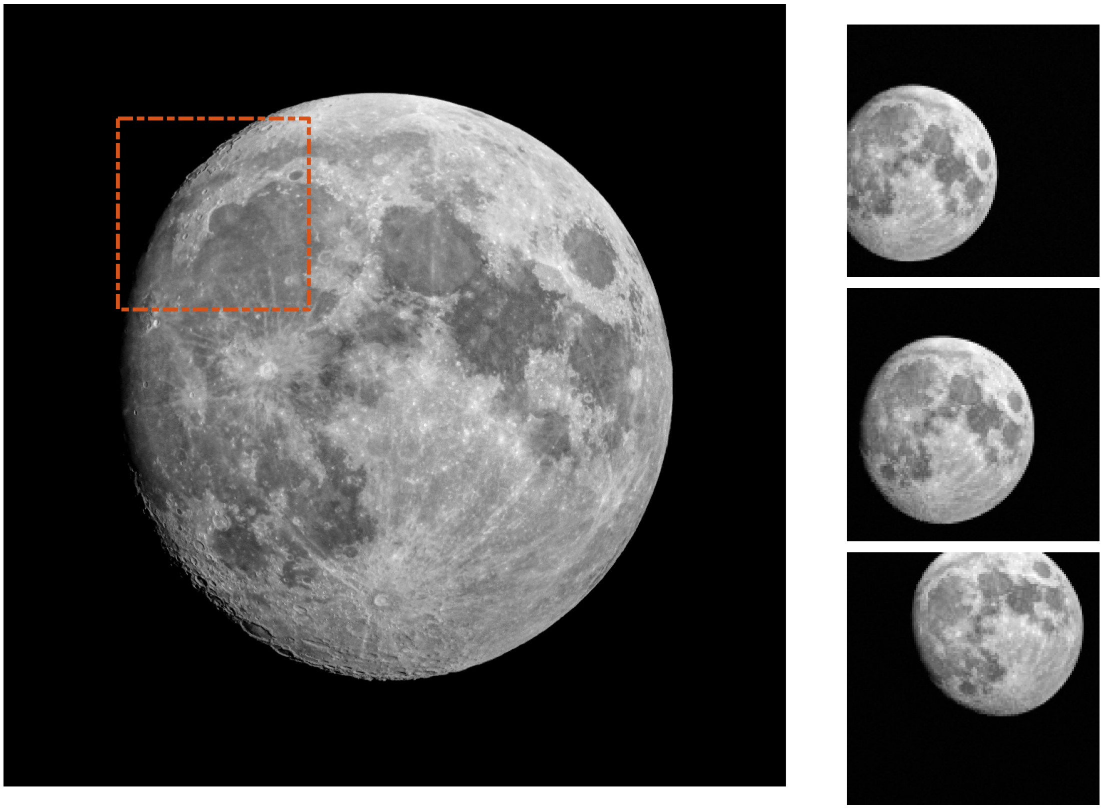

Figure 4.10: Experiment 6: Super-resolution imaging example. On the left is the true high-resolution image, and on the right are three sample low-resolution images. The red-box corresponds to sub-images shown in Figure 4.12.

Furthermore, we use the Hutchison trace estimator to efficiently evaluate the trace term in sGCV and sUPRE, see (4.12) and (4.13). More specifically, rather than compute 128^2 linear solves, we note that if $\mathbf{v}$ is a random variable such that $\mathbb{E}\left(\mathbf{v}\mathbf{v}^\top\right) = \mathbf{I}_{128^2}$, then

$$\operatorname{tr}\left(\mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\right) = \mathbb{E}\mathbf{v}^\top \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\mathbf{v}.$$

Here we use the Rademacher distribution where the i.i.d. entries of $\mathbf{v}$ are $v_i = \pm 1$ with equal probability. We use a single realization of $\mathbf{v}$ to approximate the trace, hence resulting in just one linear solve [5, 74, 138].

Relative reconstruction errors are provided in Figure 4.11, and sub-images of the reconstructions are provided in Figure 4.12. We observe that, in general, sDP errors are more erratic than sUPRE and sGCV errors. Notice that for sUPRE and sGCV, `sbK` produces higher reconstruction errors compared to `sg`, which may be attributed to insufficient global curvature information. Furthermore, we observe that `slimTik` reconstructions contain more details than `sg` and `sbK` reconstructions.

4.6 Remarks and Future Directions

In this chapter, we described row-action methods for solving ill-posed inverse problems for which it is not feasible to access the data all-at-once and regularization must be introduced. Such methods are necessary when handling data sets that do not fit in memory and also can naturally handle streaming data problems.

We investigate two iterative methods, `rrls` and `sTik`, and show that under various sampling schemes, `rrls` iterates converge asymptotically to the unregularized solution while `sTik` iterates converge to a Tikhonov-regularized solution. Although the sampling mechanisms we

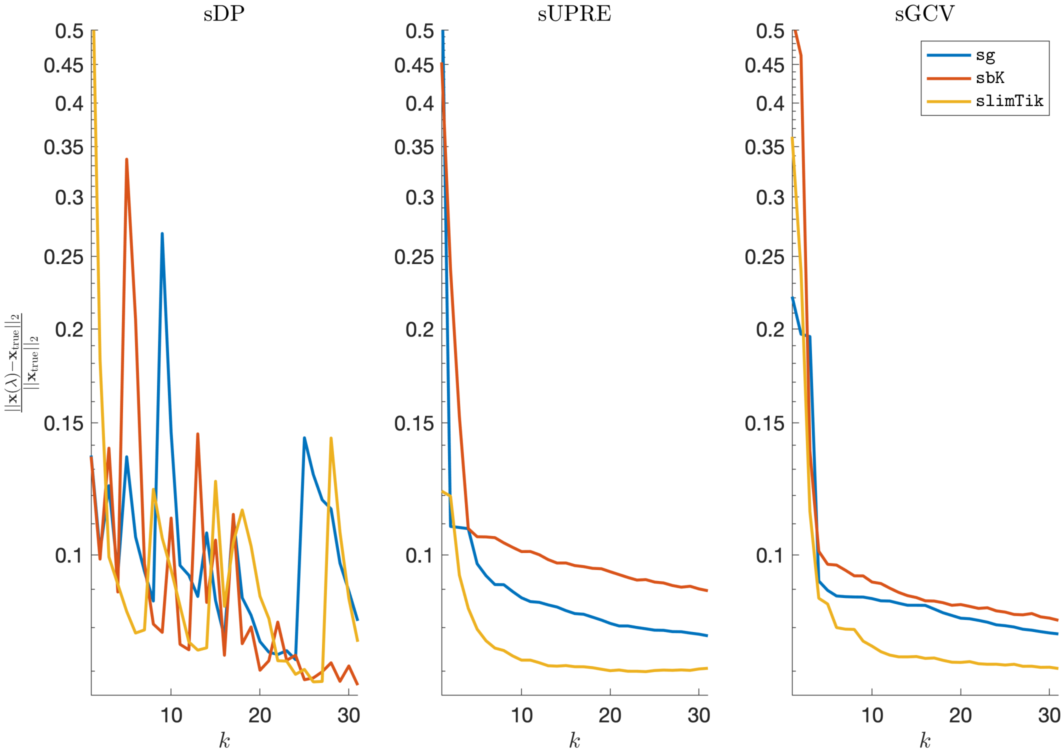

Figure 4.11: Experiment 6: Relative reconstruction errors for the super-resolution imaging example for one epoch. We note that sUPRE and sGCV produce good reconstructions. Additionally, `slimTik` produces a smaller relative reconstruction error, since it is using more curvature information.

discuss do not play a role in the asymptotic convergence, they do allow for interesting interpretations. In particular, for random cyclic sampling, we can characterize the iterates as Tikhonov solutions after every epoch, providing insight into the path that the iterates take towards the solution. For iterative methods where the regularization parameter can be updated during the iterative process (e.g., `sTik`), we describe sampled variants of existing regularization parameter selection methods to update the parameter. Using several well-known data sets, we show empirically that sampled Tikhonov methods with automatic regularization parameter updates can be competitive. For very large inverse problems, we describe a limited-memory version of `sTik`, and we demonstrate the efficacy of the limited-memory

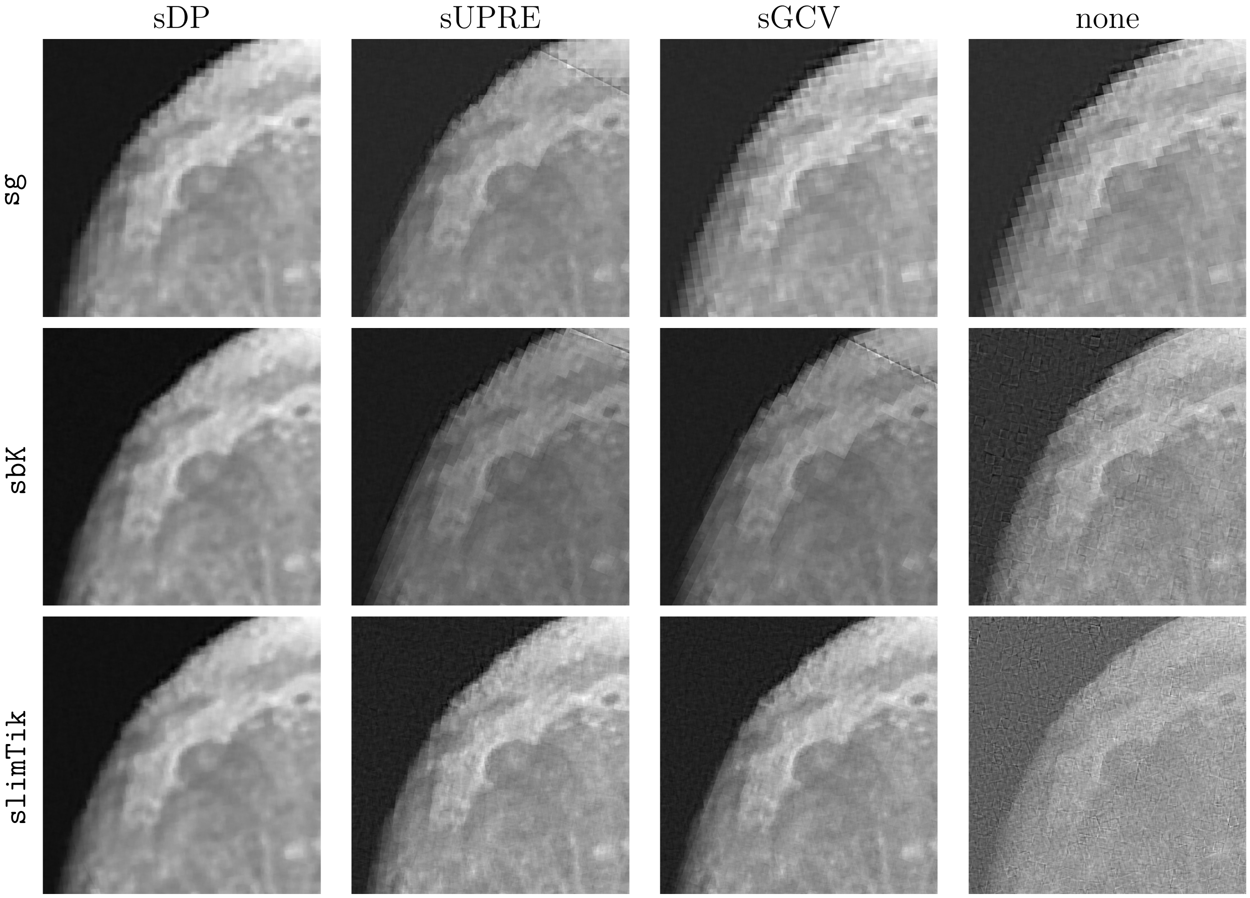

Figure 4.12: Experiment 6: Sub-images of the reconstructed images for the super-resolution imaging example. Reconstructions correspond to `sg`, `sbK`, and `slimTik` with regularization parameter updates computed using sDP, sUPRE, and sGCV. For comparison, we provide reconstructions corresponding to no regularization, i.e., $\lambda = 0$.

approach on a standard benchmark dataset as well as on a streaming super-resolution image reconstruction problem.

Future directions of research include developing an asymptotic analysis of `slimTik` for the case of varying regularization parameter, since the almost sure convergence shown in 4.4 assumed a constant regularization parameter. This analysis would aid in understanding `slimTik` when it is coupled with a sampled regularization parameter selection method. Convergence analysis for cyclic and random cyclic sampling would also be beneficial.

Finally, extensions to nonlinear inverse problems would require more advanced convergence analyses and further algorithmic developments. We begin this, by extending `slimTik` to solve the separable nonlinear inverse problem in Chapter 5.

Chapter 5

Extension to Separable, Non-linear Inverse Problems

This chapter focuses on extending the row-action methods described in Chapter 4 to the massive nonlinear separable Tikhonov LS problem

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) = \|\mathbf{A}(\mathbf{y})\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \,, \tag{5.1}$$

where $\lambda > 0$ is a regularization parameter that balances the data-fit and the regularization term. When (5.1) is not massive, numerical optimization methods to solve the problem have been investigated and range from fully decoupled approaches (e.g., alternating optimization) to fully coupled (e.g, nonlinear) approaches [40]. A popular alternative is the *variable projection* method [64, 130], where the linear parameters are mathematically eliminated and a nonlinear optimization scheme is used to solve the reduced optimization problem. These methods have been investigated for various image processing applications, see e.g., [12, 44, 88]. However, all of these methods require all-at-once access to the data to perform full matrix-vector multiplications with $\mathbf{A}(\mathbf{y})$, and hence they cannot be used for massive problems.

In this chapter, we develop an iterative sampled method to estimate a solution for (5.1) in the case of massive or streaming data. The method follows a variable projection approach by first mathematically eliminating the linear variables. However, to address massive or streaming data, we use `slimTik` defined in Chapter 4 to approximate the regularized linear problem and use a sampled Gauss-Newton method to approximate the nonlinear variables.

An outline of this chapter is as follows. In Section 5.1 we describe iterative sampled methods for separable nonlinear inverse problems, where the sampled Tikhonov methods from Chapter 4 are integrated within a nonlinear optimization framework for updating estimates of $\mathbf{x}_{\text{true}}$ and $\mathbf{y}_{\text{true}}$. Numerical results from super-resolution imaging are presented in Section 5.2, and conclusions and future work are presented in Section 5.3.

5.1 Row-action Methods for Separable, Non-linear Inverse Problems

Next for separable nonlinear inverse problems of the form (5.1), we describe an iterative sampled approach that integrates `slimTik` within a nonlinear optimization framework so that both sets of parameters can be updated as data become available.

In this chapter we assume that $\mathbf{A}\left(\,\cdot\,\right)$ and $\mathbf{b}$ can be partitioned into M blocks,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(1)}\left(\,\cdot\,\right) \\ \vdots \\ \mathbf{A}^{(M)}\left(\,\cdot\,\right) \end{bmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}^{(1)} \\ \vdots \\ \mathbf{b}^{(M)} \end{bmatrix}. \tag{5.2}$$

For simplicity we assume that all blocks have the same dimension, i.e., $\mathbf{A}^{(i)}(\,\cdot\,) : \mathbb{R}^p \to \mathbb{R}^{\ell \times n}$, $i = 1, \ldots, M$, and $\mathbf{b}^{(i)} \in \mathbb{R}^\ell$, $i = 1, \ldots, M$, with $\ell = m/M$. We also consider these methods only with cyclic sampling.

For an initial guess of the linear parameters $\mathbf{x}_0 \in \mathbb{R}^n$, nonlinear parameters $\mathbf{y}_0 \in \mathbb{R}^p$, and

$\lambda > 0$, the kth iterate of the separable nonlinear `slimTik` (`sn-slimTik`) method can be written as

$$\begin{aligned}\mathbf{x}_k &= \mathbf{x}_{k-1} - \mathbf{s}_k \\ \mathbf{y}_k &= \mathbf{y}_{k-1} - \alpha_k \left(\mathbf{J}_k^\top \mathbf{J}_k\right)^\dagger \mathbf{J}_k^\top \mathbf{r}_k\left(\mathbf{y}_{k-1}\right) \qquad (5.3)\end{aligned}$$

with

$$\mathbf{s}_k = \arg\min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{M}_k\left(\mathbf{y}_{k-1}\right) \\ \mathbf{A}_k\left(\mathbf{y}_{k-1}\right) \\ \sqrt{\frac{k\lambda}{M}}\mathbf{I}_n \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_k\left(\mathbf{y}_{k-1}\right)\mathbf{x}_{k-1} - \mathbf{b}_k \\ \sqrt{\frac{\lambda}{kM}}\mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2,$$

where $\mathbf{A}_k\left(\,\cdot\,\right) = \mathbf{A}^{((\mod (k-1)M)+1)}\left(\,\cdot\,\right)$, $\mathbf{b}_k = \mathbf{b}^{((\mod (k-1)M)+1)}$, and

$$\mathbf{M}_k\left(\,\cdot\,\right) = \left[\mathbf{A}_{k-r}\left(\,\cdot\,\right)^\top, \ldots, \mathbf{A}_{k-1}\left(\,\cdot\,\right)^\top\right]^\top$$

for chosen memory level $r \in \mathbb{N}$. The blocks of $\mathbf{A}$ and $\mathbf{b}$ with negative indices are set to the zero function and zero vector, respectively. Here $\mathbf{r}_k(\cdot) : \mathbb{R}^p \to \mathbb{R}^{\ell(r+1)}$ is the sample residual function defined as

$$\mathbf{r}_k\left(\mathbf{y}\right) = \begin{bmatrix} \mathbf{A}_{k-r}\left(\mathbf{y}\right) \\ \vdots \\ \mathbf{A}_{k-1}\left(\mathbf{y}\right) \\ \mathbf{A}_k\left(\mathbf{y}\right) \end{bmatrix} \mathbf{x}_k - \begin{bmatrix} \mathbf{b}_{k-r} \\ \vdots \\ \mathbf{b}_{k-1} \\ \mathbf{b}_k \end{bmatrix},$$

$\mathbf{J}_k$ is the Jacobian of $\mathbf{r}_k$ evaluated at $\mathbf{y}_{k-1}$, and α_k is the step size determined by a line search method [129]. The Jacobian can be approximated with finite differences or found analytically. Note that † represents the pseudo-inverse in (5.3) and is required since $\mathbf{J}_k$ might not have full column rank. Also, as with any nonlinear, nonconvex optimization method, the initial guess must be within the basin of attraction of the desired minimizer. A summary of the sn-slimTik algorithm is provided below.

Algorithm 1 sn-slimTik

```
Inputs: $\mathbf{x}_0$, $\mathbf{y}_0$, $r$, $\lambda$, $M$
for $k = 1, 2, \ldots$ do
    Get $\mathbf{A}_k(\mathbf{y}_{k-1})$, $\mathbf{b}_k$, and $\mathbf{M}_k(\mathbf{y}_{k-1})$
    $\mathbf{s}_k = \arg\min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{M}_k(\mathbf{y}_{k-1}) \\ \mathbf{A}_k(\mathbf{y}_{k-1}) \\ \sqrt{\frac{k\lambda}{M}}\mathbf{I}_n \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_k(\mathbf{y}_{k-1})\mathbf{x}_{k-1} - \mathbf{b}_k \\ \sqrt{\frac{\lambda}{kM}}\mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2$
    $\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{s}_k$
    $\mathbf{y}_k = \mathbf{y}_{k-1} - \alpha_k \left(\mathbf{J}_k^\top \mathbf{J}_k\right)^\dagger \mathbf{J}_k^\top \mathbf{r}_k(\mathbf{y}_{k-1})$
end for
```

5.2 Numerical Results

In this section, we provide numerical results for super-resolution image reconstruction, which can be represented as a separable nonlinear inverse problem [41]. Suppose we have M low-resolution images. The underlying model for super-resolution imaging can be represented as (1.2), where $\mathbf{x}_{\text{true}}$ contains the high-resolution (HR) image, and $\mathbf{b}$ and $\mathbf{A}(\mathbf{y}_{\text{true}})$ can be partitioned as in (5.2), where $\mathbf{b}^{(i)}$ contains the ith low-resolution (LR) image and $\mathbf{A}^{(i)}(\cdot) : \mathbb{R}^p \to \mathbb{R}^{\ell \times n}$. More specifically, if we assume that the deformation for each LR image is affine (e.g., can be described with at most 6 parameters) and independent of the parameters for the other images, then we can partition $\mathbf{y}$ as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}^{(1)} \\ \vdots \\ \mathbf{y}^{(M)} \end{bmatrix}$$

and have $\mathbf{A}^{(i)}(\mathbf{y}) = \mathbf{R}\mathbf{S}(\mathbf{y}^{(i)})$ where $\mathbf{R}$ is a restriction matrix that takes a HR image to a LR one and $\mathbf{S}(\mathbf{y}^{(i)})$ represents an affine transformation defined by parameters in $\mathbf{y}^{(i)}$. Then the goal is to solve (5.1) to estimate the HR image as well as update the transformation parameters.

We will investigate iterative sampled methods for super-resolution problems with massive or streaming data, but first we investigate a smaller problem where all of the data can be accessed at once. In Experiment 7, we compare our proposed `sn-slimTik` method with different memory levels to the results from the variable projection method. We show that with relatively modest memory levels, our approaches can achieve reconstructions with similar quality to full-memory reconstructions in comparable time. Then in Experiment 8, we consider a very large streaming super-resolution problem, where both the resolution of the images as well as the number of LR images present a computational bottleneck.

In both experiments, we initialize $\mathbf{x}_0 = \mathbf{0}$, and $\mathbf{y}_0$ is obtained by adding Gaussian white noise with zero mean to $\mathbf{y}_{\text{true}}$ where the variance is $2.45 \cdot 10^{-3}$ in Experiment 1 and $4.48 \cdot 10^{-4}$ in Experiment 2. We set the regularization parameter in advance, but mention that methods for updating the regularization parameter can be found in [143].

Algorithm 2 `variable projection`

```
Inputs: $\mathbf{y}_0$, $\lambda$
for $k = 1, 2, \ldots$ do
    $\mathbf{x}_k = \arg\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{y}_{k-1}) \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2$
    $\tilde{\mathbf{r}}_k(\mathbf{y}_{k-1}) = \mathbf{A}(\mathbf{y}_{k-1})\mathbf{x}_k - \mathbf{b}$
    $\mathbf{y}_k = \mathbf{y}_{k-1} - \alpha_k \left(\tilde{\mathbf{J}}_k^\top \tilde{\mathbf{J}}_k\right)^\dagger \tilde{\mathbf{J}}_k^\top \tilde{\mathbf{r}}_k(\mathbf{y}_{k-1})$
end for
```

5.2.1 Comparing `sn-slimTik` to `variable projection`

Both `sn-slimTik` and variable projection are iterative methods that update $\mathbf{x}$ and $\mathbf{y}$. However, the variable projection method requires access to all data at once and thus may be infeasible for massive or streaming problems. The goal of this experiment is to show that we can achieve similar reconstructions as existing methods, but without the need to access all data and matrices at once.

For completeness, we provide in Algorithm 2 the basic variable projection algorithm [64, 130], which is a Gauss-Newton algorithm applied to the problem,

$$\min_{\mathbf{y}} f(\mathbf{x}(\mathbf{y}), \mathbf{y}).$$

Here $\tilde{\mathbf{J}}_k$ is the Jacobian of $\mathbf{A}(\mathbf{y})\mathbf{x}_k - \mathbf{b}$ with respect to $\mathbf{y}$ at $\mathbf{y}_{k-1}$, and α_k is a line search parameter. Analytical methods can be used to obtain the Jacobian, see [41]. Notice that each iteration of the variable projection algorithm requires access to the entire data set $\mathbf{b}$ as well as matrix $\mathbf{A}(\mathbf{y})$ in order to solve the linear LS problem in step 3. For our experiments, we use the LSQR method to solve the linear Tikhonov problem, where each *iteration* of LSQR

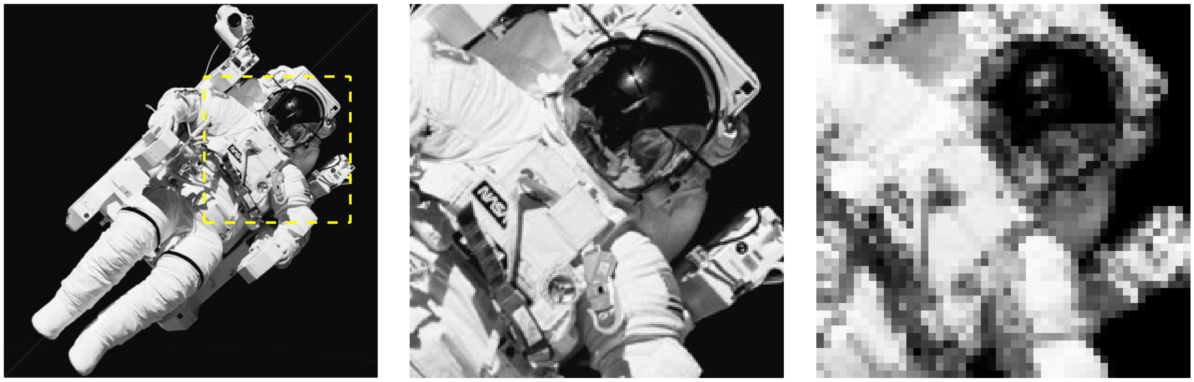
(a) HR image (b) HR subimage (c) LR subimage

Figure 5.1: Experiment 7: Super-resolution imaging example. The high-resolution (HR) image and a subimage corresponding to the yellow box are provided in (a) and (b) respectively. The subimage of one of the low-resolution (LR) images is provided in (c).

requires a matrix-vector multiplication with $\mathbf{A}(\mathbf{y}_{k-1})$ and $\mathbf{A}(\mathbf{y}_{k-1})^\top$. Each multiplication requires access to all of the data, and thus, in terms of data access, is equivalent to one epoch of `slimTik`.

For this experiment, the goal is to recover a HR image that contains 512^2 pixels from a set of $M = 100$ LR images, each containing 128^2 pixels, i.e., $\mathbf{A}(\mathbf{y}) \in \mathbb{R}^{100 \cdot 128^2 \times 512^2}$. The HR image is of an astronaut and was obtained from NASA's website [122]. The HR image and three of the simulated LR images are provided in Fig. 5.1. The noise level for each LR image was set to 0.01, and the regularization parameter was set to $\lambda = 8 \cdot 10^{-2}$.

In Fig. 5.2, we provide relative error norms for the reconstructions and relative error norms for the affine parameters,

$$\frac{\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2}{\|\mathbf{x}_{\text{true}}\|_2} \quad \text{and} \quad \frac{\|\mathbf{y}_k - \mathbf{y}_{\text{true}}\|_2}{\|\mathbf{y}_{\text{true}}\|_2}, \tag{5.4}$$

respectively. We compare the `sn-slimTik` method with memory levels $r = 0, 1$, and 5 for 5 epochs (100 iterations correspond to one epoch), and provide results for 4 iterations of the

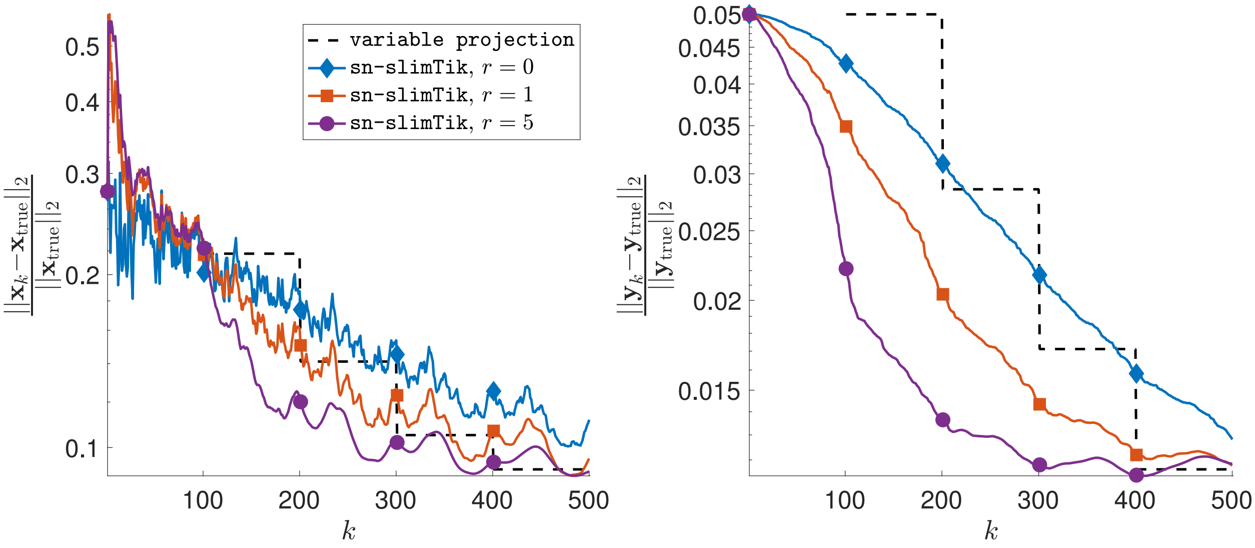

Figure 5.2: Experiment 7: Relative reconstruction error norms for the image $\mathbf{x}_k$ (left) and the nonlinear parameters $\mathbf{y}_k$ (right) for variable projection and `sn-slimTik` for various memory levels. Note that variable projection errors are only provided after every 100 iterations of `sn-slimTik`.

variable projection method for comparison.

Following the discussion above, it is difficult to provide a fair comparison since each variable projection iteration requires a linear solve and here we use 20 LSQR iterations for each outer iteration. Performing one LSQR iteration requires the same memory access as 100 iterations of `sn-slimTik` with any memory level. Thus, in Fig. 5.2 we plot the relative reconstruction error norms for variable projection only after every 100 iterations of `sn-slimTik`. We see that for both parameters sets, `sn-slimTik` produces relative reconstruction errors that are comparable to the variable projection method. For this experiment variable projection took 644 seconds, `sn-slimTik` took 366 seconds with memory 0, 800 seconds with memory 1, and 2,570 seconds with memory 5.

Sub-images of `sn-slimTik` reconstructions at iterations $k = 100$ and 200 with memory parameters 0, 1, and 5 are provided in Fig. 5.3. We note that for $k = 1$ all three reconstructions

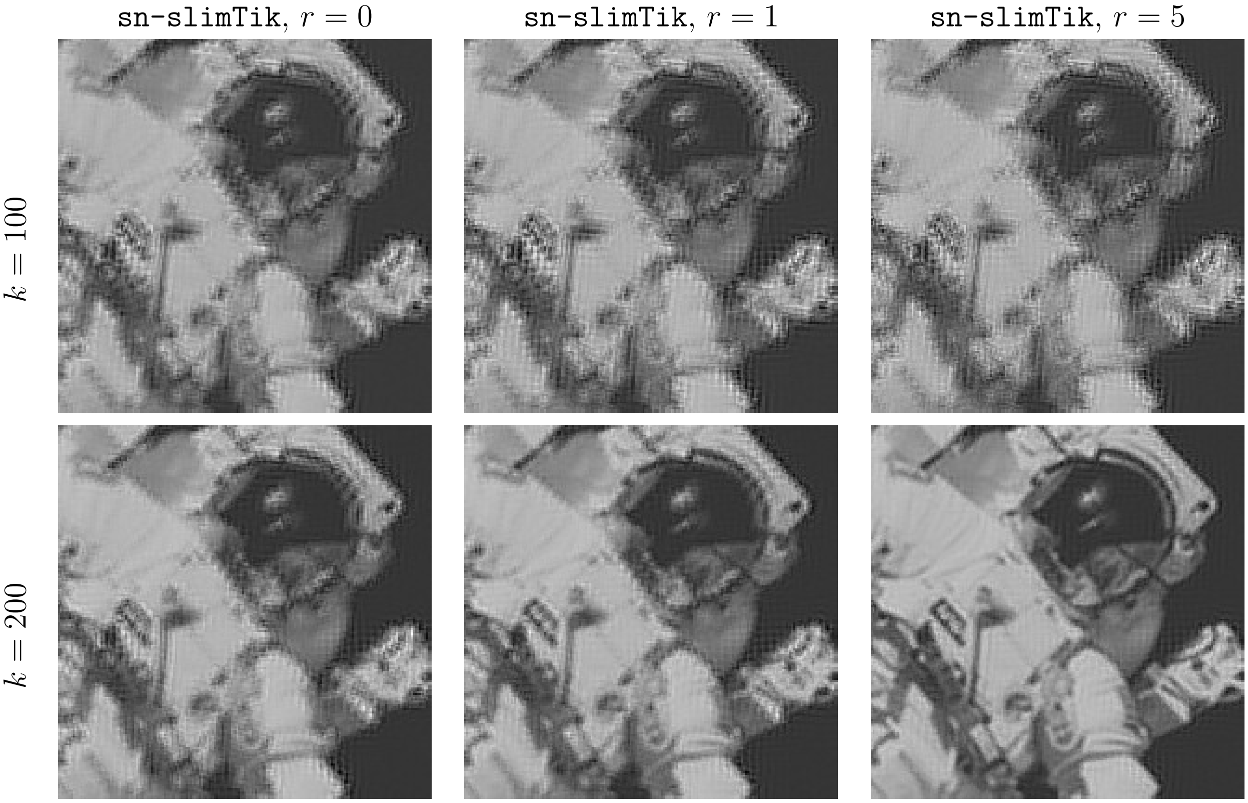

Figure 5.3: Experiment 7: Sub-images of **sn-slimTik** reconstructions for memory levels $r = 0, 1$, and 5 for iterates within the first two epochs of data access.

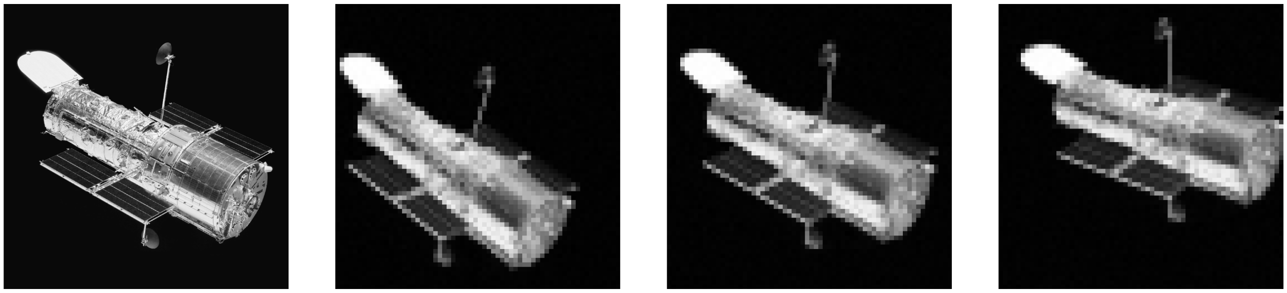
(a) HR image (b) LR image (c) LR image (d) LR image

Figure 5.4: Experiment 8: Streaming super-resolution imaging example. The high-resolution (1,024 × 1,024) image is provided in (a), along with three of the low-resolution (64 × 64) images in (b)–(d).

are identical since all of them only have access to the first LR image. Reconstructions after 100 iterations are also similar, but after 200 iterations, we see that `sn-slimTik` with memory level 5 produces a better reconstruction. These results show that including memory in the `slimTik` algorithm may be beneficial, and results are comparable to those of variable projection.

5.2.2 Experiment 8: `sn-slimTik` for a Streaming Problem

Next we consider a very large *streaming* super-resolution problem, where the goal is to reconstruct a HR image of $1,024^2$ pixels from 300 LR images of 64^2 pixels that are being observed in time. The HR image comes from NASA [122] and is depicted, along with three of the LR images, in Fig. 5.4. For this example, once all data has been accessed, $\mathbf{A} \in \mathbb{R}^{300 \cdot 64^2 \times 1,024^2}$ is too large to store in memory. Furthermore, in many streaming scenarios, we would like to be able to compute partial image reconstructions and update the nonlinear parameters during the data acquisition process, e.g., while LR images are still being streamed. Notice that the variable projection method requires us to wait until all LR images are observed, and even then it may be too costly to access all of $\mathbf{A}$ at once.

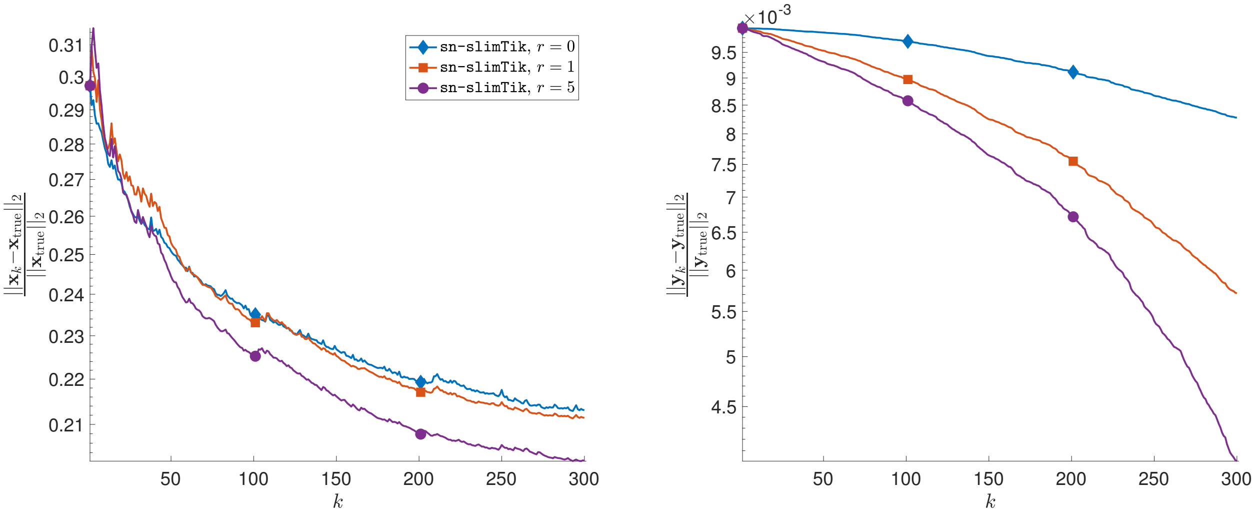

Figure 5.5: Experiment 8: Relative reconstruction errors for both the linear (left) and nonlinear (right) parameters for the streaming data super-resolution problem.

Thus, in this experiment, we consider the **sn-slimTik** method with memory levels $r = 0, 1,$ and 5. We run 300 iterations (e.g., accessing one epoch of the data) and set the noise level for each LR image to be 0.01 and $\lambda = 5 \cdot 10^{-3}$. In Fig. 5.5 we provide the relative reconstruction errors for $\mathbf{x}_k$ and $\mathbf{y}_k$. We observe that a higher memory level corresponds to improved estimates of the nonlinear parameters and the reconstructions. In Fig. 5.6, we provide sub-images of absolute errors images of the reconstructions, computed as $|\mathbf{x}_{300} - \mathbf{x}_{\text{true}}|$,

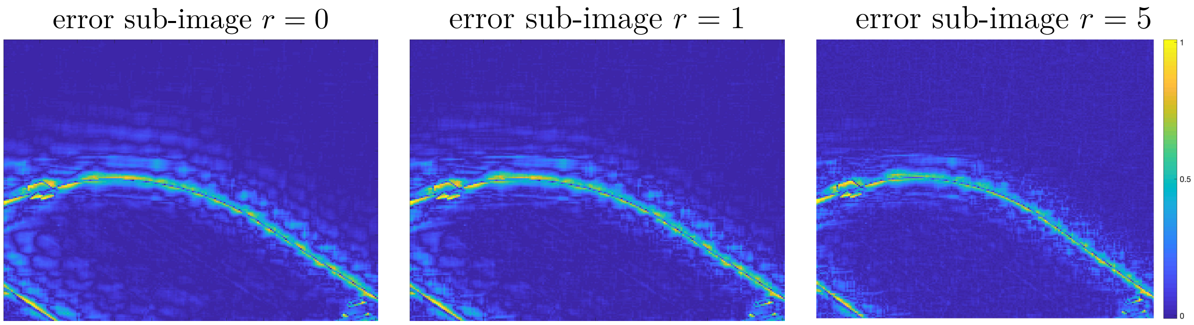

Figure 5.6: Experiment 8: Sub-image of absolute error images for **sn-slimTik** reconstructions with different memory levels.

in inverted colormap so that white corresponds to small absolute errors. These images show that `sn-slimTik` methods produce better reconstructions with increased memory level, but an increased memory level comes with an increase in computation time. For this example, the CPU times for `sn-slimTik` are 1,035, 1,954, and 5,858 seconds for memory levels of 0, 1, and 5, respectively.

5.3 Remarks and Future Directions

This chapter extended introduced the `sn-slimTik` method, which is an extension of the `slimTik` algorithm that the solution of a separable nonlinear inverse problem, for the case where the data cannot be accessed all-at-once. The method combines limited-memory sampled Tikhonov methods, which were developed for linear inverse problems, within a nonlinear optimization framework. Numerical results on massive super-resolution problems show that results are comparable to those from variable projection, when all data can be accessed at once. When this is not the case (e.g., streaming or massive data), the `sn-slimTik` method can effectively and efficiently update both sets of parameters.

A future area of research is to develop a theoretical analysis of the convergence properties of `sn-slimTik` similar to the convergence theory that has been done for `slimTik`. This would include asymptotic analysis and bounds on the mean square error. Additionally, `sn-slimTik` was only studied under cyclic sampling in this section, but the randomized variant should be analyzed and implemented. Finally, `sn-slimTik` should be generalized to nonlinear separable functions that are massive not only in the linear parameters, but the nonlinear parameters as well.

Chapter 6

Conclusion

This dissertation focused on constructing numerical solutions to massive inverse problems. We introduced row-action methods that are tailored for solving LS problems with massive data sets, and we focused on massive inverse problems for which we have developed regularization parameter selection techniques that can work with only samples of the data at any given time. By addressing the linear and nonlinear separable inverse problem, we were able to apply the algorithms developed to a wide range of scientific applications.

This work presented several mathematical contributions. Row-action methods were connected to stochastic approximation methods by exploiting a stochastic reformulation of the LS problem. Analysis of row-action methods under this context presented the first almost sure convergence results for the Kaczmarz and block Kaczmarz methods applied to the general LS problem. Additionally, the `slimLS` method was developed and investigated, and we showed asymptotic convergence to the LS solution. Additionally, in a particular case, we showed favorable expected linear convergence rate of the `slimLS` algorithm, making a note of the trade-off between precision of iterations and convergence rate, that depends on step size.

We showed that `slimLS` could be interpreted as a limited memory variant of a recursive LS algorithm. This connection allowed new insights into sampling with and sampling without replacement. Additionally, this connection helped to determine how to choose the regularization parameter and displayed a connection between the regularization parameter and step size in the iterates. To choose the correct Tikhonov regularization parameter, sampled variants of the discrepancy principle, the unbiased predictive risk estimator, and the generalized cross validation method were developed.

In addition to the mathematical contributions of this work, a significant amount of work was done to ensure computational efficiency of these methods and broader scientific impacts. Efficient implementations of row-action methods and sampled regularization parameter selection methods were applied to massive problems related to super resolution, data science, and tomography. A generalization of `slimLS` to the nonlinear, separable inverse problem that arises in super-resolution imaging shows the potential of row-action methods to be applied to a wide range of inverse problems.

Bibliography

[1] D Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[2] M Altman. On the approximate solution of linear algebraic equations. *Bulletin de l'Académie Polonaise des Sciences Cl*, 3(3):365–370, 1957.

[3] M S Andersen and P C Hansen. Generalized row-action methods for tomographic imaging. *Numerical Algorithms*, 67(1):121–144, 2014.

[4] R C Aster, B Borchers, and C H Thurber. *Parameter Estimation and Inverse Problems*. Elsevier, New York, 2018.

[5] H Avron and S Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8, 2011.

[6] Z Bai, G B Huang, D Wang, H Wang, and M B Westover. Sparse extreme learning machine for classification. *IEEE Transactions on Cybernetics*, 44(10):1858–1870, 2014.

[7] Z Z Bai and X G Liu. On the meany inequality with applications to convergence analysis of several row-action iteration methods. *Numerische Mathematik*, 124(2): 215–236, 2013.

[8] J M Bardsley. *Computational Uncertainty Quantification for Inverse Problems*. SIAM, Philadelphia, 2018.

[9] S Bartels, J Cockayne, I C F Ipsen, M Girolami, and P Hennig. Probabilistic linear solvers: A unifying view. *arXiv preprint arXiv:1810.03398*, 2018.

[10] E C Beckmann. Ct scanning the early days. *The British journal of radiology*, 79(937): 5–8, 2006.

[11] A Benveniste, S S Wilson, M Metivier, and P Priouret. *Adaptive Algorithms and Stochastic Approximations*. Stochastic Modelling and Applied Probability. Springer, New York, 2012. ISBN 9783642758942.

[12] S Berisha, J G Nagy, and R J Plemmons. Estimation of atmospheric PSF parameters for hyperspectral imaging. *Numerical Linear Algebra with Applications*, 2015.

[13] D P Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM Journal on Optimization*, 7(4):913–926, 1997.

[14] D P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

[15] A Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[16] A Björck and T Elfving. Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations. *BIT Numerical Mathematics*, 19(2):145–163, 1979.

[17] E Bodewig. *Bericht über die verschiedenen Methoden zur Lösung eines Systems linearer Gleichungen mit reellen Koeffizienten*. North-Holland Publishing Company, 1948.

[18] E Bodewig. *Matrix calculus*. North-Holland, 1956.

[19] L Bottou. *Online Learning in Neural Networks*, chapter 2. Online learning and stochastic approximations, pages 9–42. Cambridge University Press, 1998.

[20] L Bottou and Y L Cun. Large scale online learning. In *Advances in Neural Information Processing Systems*, pages 217–224, 2004.

[21] L Bottou and Y Le Cun. On-line learning for very large data sets. *Applied stochastic models in business and industry*, 21(2):137–151, 2005.

[22] L Bottou, F E Curtis, and J Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[23] C Boutsidis and A Gittens. Improved matrix algorithms via the subsampled randomized Hadamard transform. *arXiv preprint arXiv:1204.0062*, 2012.

[24] C Brezinski. *Projection methods for systems of equations*, volume 7. North-Holland, 1997.

[25] A Buccini, M Donatelli, and L Reichel. Iterated Tikhonov regularization with a general penalty term. *Numerical Linear Algebra with Applications*, 24(4):e2089, 2017.

[26] R H Byrd, S L Hansen, J Nocedal, and Y Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[27] D Calvetti and E Somersalo. *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. Springer, New York, 2007. ISBN 0387733930.

[28] A Cegielski. *Iterative methods for fixed point problems in Hilbert spaces*, volume 2057. Springer, 2012.

[29] A Cegielski. Bibliography on the kaczmarz method (up to 2010), 2016.

[30] Y Censor. Row-action methods for huge and sparse systems and their applications. *SIAM review*, 23(4):444–466, 1981.

[31] Y Censor and S A Zenios. *Parallel optimization: Theory, algorithms, and applications.* Oxford University Press on Demand, 1997.

[32] Y Censor, P P B Eggermont, and D Gordon. Strong underrelaxation in kaczmarz's method for inconsistent systems. *Numerische Mathematik*, 41(1):83–92, 1983.

[33] Y Censor, G T Herman, and M Jiang. A note on the behavior of the randomized kaczmarz algorithm of strohmer and vershynin. *Journal of Fourier Analysis and Applications*, 15(4):431–436, 2009.

[34] NASA Goddard Space Flight Center. Image from NASA's lunar reconnaissance orbitor. https://lunar.gsfc.nasa.gov/imagesandmultimedia.html, 2014.

[35] X Chen and A M Powell. Almost sure convergence of the kaczmarz algorithm with random measurements. *Journal of Fourier Analysis and Applications*, 18(6):1195–1214, 2012.

[36] J T Chi and I C F Ipsen. Randomized least squares regression: Combining model-and algorithm-induced uncertainties. *arXiv preprint arXiv:1808.05924*, 2018.

[37] Y Chi and Y M Lu. Kaczmarz method for solving quadratic equations. *IEEE Signal Processing Letters*, 23(9):1183–1187, 2016.

[38] J Chung. *Numerical approaches for large-scale ill-posed inverse problems.* PhD thesis, Emory University, 2009.

[39] J Chung and J Nagy. Nonlinear least squares and super resolution. In *Journal of Physics: Conference Series*, volume 124, page 012019. IOP Publishing, 2008.

[40] J Chung and J G Nagy. An efficient iterative approach for large-scale separable nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 31(6):4654–4674, 2010.

[41] J Chung, E Haber, and J Nagy. Numerical methods for coupled super-resolution. *Inverse Problems*, 22(4):1261, 2006.

[42] J Chung, M Chung, J T Slagel, and L Tenorio. Stochastic Newton and quasi-Newton methods for large linear least-squares problems. *arXiv preprint arXiv:1702.07367*, 2017.

[43] J Chung, M Chung, and J T Slagel. Iterative sampled methods for massive and separable nonlinear inverse problems. In *[To Appear In] Seventh International Conference on Scale Space and Variational Methods in Computer Vision*, 2019.

[44] A Cornelio, E L Piccolomini, and J G Nagy. Constrained variable projection method for blind deconvolution. In *Journal of Physics: Conference Series*, volume 386, page 012005. IOP Publishing, 2012.

[45] L Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[46] P S R Diniz. *Adaptive filtering.* Springer, New York, 1997.

[47] T N Do, V H Nguyen, and F Poulet. Speed up svm algorithm for massive classification tasks. In *International conference on advanced data mining and applications*, pages 147–157. Springer, 2008.

[48] C Dong and Y Jin. Mimo nonlinear ultrasonic tomography by propagation and backpropagation method. *IEEE Transactions on Image Processing*, 22(3):1056–1069, 2013.

[49] D L Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[50] P Drineas, M W Mahoney, S Muthukrishnan, and T Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2011.

[51] P Drineas, M Magdon-Ismail, M W Mahoney, and D P Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(Dec):3475–3506, 2012.

[52] A Dumitraşc and C Popa. Parallel solution of large sparse linear least squares problems. *arXiv preprint arXiv:1708.07693*, 2017.

[53] P P B Eggermont, G T Herman, and A Lent. Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear algebra and its applications*, 40:37–67, 1981.

[54] T Elfving. Block-iterative methods for consistent and inconsistent linear equations. *Numerische Mathematik*, 35(1):1–12, 1980.

[55] T Elfving, P C Hansen, and T Nikazad. Semi-convergence properties of Kaczmarz's method. *Inverse Problems*, 30(5):055007, 2014.

[56] H W Engl. On the choice of the regularization parameter for iterated Tikhonov regularization of ill-posed problems. *Journal of Approximation Theory*, 49(1):55–63, 1987.

[57] H W Engl, M Hanke, and A Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[58] R Escalante and M Raydan. *Alternating Projection Methods*. SIAM, Philadelphia, 2011.

[59] H G Feichtinger, C Cenker, M Mayer, H Steier, and T Strohmer. New variants of the pocs method using affine subspaces of finite codimension with applications to irregular sampling. In *Visual Communications and Image Processing'92*, volume 1818, pages 299–311. International Society for Optics and Photonics, 1992.

[60] D L R Fisk. *Quasi-Martingales and Stochastic Integrals.* Department of Mathematics Research Monograph. Kent State University, Department of Mathematics, 1963.

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

[62] A Galántai. On the rate of convergence of the alternating projection method in finite dimensional spaces. *Journal of mathematical analysis and applications*, 310(1):30–44, 2005.

[63] N Gastinel. *Linear numerical analysis.* Academic Press, 1970.

[64] G Golub and V Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19:R1–R26, 2003.

[65] G H Golub and C F Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

[66] G H Golub, M Heath, and G Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[67] R Gordon, R Bender, and G T Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of theoretical Biology*, 29(3):471–481, 1970.

[68] R M Gower. Sketch and project: randomized iterative methods for linear systems and inverting matrices. *arXiv preprint arXiv:1612.06013*, 2016.

[69] R M Gower and P Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.

[70] R M Gower and P Richtárik. Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms. *arXiv preprint arXiv:1602.01768*, 2016.

[71] R M Gower, D Goldfarb, and P Richtárik. Stochastic block BFGS: Squeezing more curvature out of data. *arXiv preprint arXiv:1603.09649*, 2016.

[72] C W Groetsch. *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind*. Pitman, 1984.

[73] G B H, H Zhou, X Ding, and R Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.

[74] E Haber, M Chung, and F Herrmann. An effective method for parameter estimation with PDE constraints with multiple right-hand sides. *SIAM Journal on Optimization*, 22(3):739–757, 2012.

[75] J Hadamard. *Lectures on Cauchy's problem in linear partial differential equations*, volume 37. Yale University Press, 1923.

[76] M Hanke and C W Groetsch. Nonstationary iterated tikhonov regularization. *Journal of Optimization Theory and Applications*, 98(1):37–53, 1998.

[77] P C Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, volume 4. Siam, 2005.

[78] P C Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, 2010.

[79] P C Hansen. Regularization Tools Version 4.1 (for Matlab). http://www.imm.dtu.dk/~pcha/Regutools/, 2018. Accessed: 2018-11-16.

[80] P C Hansen and M Saxild-Hansen. Air tools, a matlab package of algebraic iterative reconstruction methods. *Journal of Computational and Applied Mathematics*, 236(8):2167–2178, 2012.

[81] P C Hansen, J G Nagy, and D P O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, Philadelphia, 2006.

[82] S Hashemi, S Beheshti, R Cobbold, and N Paul. Adaptive updating of regularization parameters. *Signal Processing*, 113:228–233, 2015.

[83] A Hedayat and W D Wallis. Hadamard matrices and their applications. *The Annals of Statistics*, 6(6):1184–1238, 1978.

[84] G T Herman. *Fundamentals of computerized tomography: image reconstruction from projections*. Springer Science & Business Media, 2009.

[85] G T Herman and L B Meyer. Algebraic reconstruction techniques can be made computationally efficient (positron emission tomography application). *IEEE transactions on medical imaging*, 12(3):600–609, 1993.

[86] G T Herman, H Hurwitz, A Lent, and H P Lung. On the bayesian approach to image reconstruction. *Information and Control*, 42(1):60–71, 1979.

[87] G T Herman, A Lent, and H Hurwitz. A storage-efficient algorithm for finding the regularized solution of a large, inconsistent system of equations. *IMA Journal of Applied Mathematics*, 25(4):361–366, 1980.

[88] J Herring, J Nagy, and L Ruthotto. LAP: a linearize and project method for solving inverse problems with coupled variables. *Sampling Theory in Signal and Image Processing*, 17(2):127–151, 2018.

[89] G N Hounsfield. Computerized transverse axial scanning (tomography): Part 1. description of system. *The British journal of radiology*, 46(552):1016–1022, 1973.

[90] A S Householder. *The theory of matrices in numerical analysis*. Courier Corporation, 2013.

[91] B Huang, W Wang, M Bates, and X Zhuang. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science*, 319(5864):810–813, 2008.

[92] G Huang, G B Huang, S Song, and K You. Trends in extreme learning machines: A review. *Neural Networks*, 61:32–48, 2015.

[93] G B Huang, Q Y Zhu, and C K Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.

[94] G B Huang, Q Y Zhu, and C K Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[95] G B Huang, D H Wang, and Y Lan. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, 2(2):107–122, 2011.

[96] G B Huang, Z Bai, L L C Kasun, and C M Vong. Local receptive fields based extreme learning machine. *IEEE Computational Intelligence Magazine*, 10(2):18–29, 2015.

[97] N Jamil, D Needell, J Muller, C Lutteroth, and G Weber. Kaczmarz algorithm with soft constraints for user interface layout. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 818–824. IEEE, 2013.

[98] M Jiang and G Wang. Convergence studies on iterative algorithms for image reconstruction. *IEEE Transactions on Medical Imaging*, 22(5):569–579, 2003.

[99] A K S Johan et al. *Least squares support vector machines.* World Scientific, 2002.

[100] S Kaczmarz. Angenäherte Auflösung linearer Gleichungssysteme. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques*, pages 355–357, 1937.

[101] S Kaczmarz. Approximate solution of systems of linear equations. *International Journal of Control*, 57(6):1269–1271, 1993. doi: 10.1080/00207179308934446. URL https://doi.org/10.1080/00207179308934446.

[102] J Kaipio and E Somersalo. *Statistical and Computational Inverse Problems*, volume 160. Springer Science & Business Media, 2006.

[103] G Kamath, Pa Ramanan, and W Z Song. Distributed randomized kaczmarz and applications to seismic imaging in sensor network. In *2015 International Conference on Distributed Computing in Sensor Systems*, pages 169–178. IEEE, 2015.

[104] M E Kilmer and D P O'Leary. Choosing regularization parameters in iterative methods for ill-posed problems. *SIAM Journal on Matrix Analysis and Applications*, 22(4): 1204–1221, 2001.

[105] H J Kushner and G G Yin. *Stochastic Approximation Algorithms and Applications.* Springer, New York, 1997.

[106] Ha J Kushner and D S Clark. *Stochastic approximation methods for constrained and unconstrained systems*, volume 26. Springer, 2012.

[107] T L Lai, H Robbins, and C Z Wei. Strong consistency of least squares estimates in multiple regression. *Proceedings of the National Academy of Sciences of the United States of America*, 75(7):3034, 1978.

[108] E B Le, A Myers, T Bui-Thanh, and Q P Nguyen. A data-scalable randomized misfit approach for solving large-scale PDE-constrained inverse problems. *Inverse Problems*, 33(6):065003, 2017.

[109] E Levin, T Bendory, N Boumal, J Kileel, and A Singer. 3d ab initio modeling in

cryo-em by autocorrelation analysis. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1569–1573. IEEE, 2018.

[110] J Luo, C M Vong, and P K Wong. Sparse Bayesian extreme learning machine for multi-classification. *IEEE Transactions on Neural Networks and Learning Systems*, 25 (4):836–843, 2014.

[111] L Maligranda. Stefan kaczmarz (1895-1939). *Antiquitates Mathematicae*, 1:15–61, 2007.

[112] P Maponi. The solution of linear systems by using the sherman-morrison formula. *Linear Algebra and its Applications*, 420(2-3):276–294, January 2007.

[113] S Marchesini, H Krishnan, B J Daurer, D A Shapiro, T Perciano, J A Sethian, and F R N C Maia. Sharp: a distributed gpu-based ptychographic solver. *Journal of applied crystallography*, 49(4):1245–1252, 2016.

[114] J T Marti. On the convergence of the discrete art algorithm for the reconstruction of digital pictures from their projections. *Computing*, 21(2):105–111, 1979.

[115] MathWorks. Matlab Test Matrices Gallery. https://www.mathworks.com/help/matlab/ref/gallery.html, 2018. Accessed: 2018-11-16.

[116] J Matoušek. On variants of the Johnson–Lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156, 2008.

[117] S F McCormick. An iterative procedure for the solution of constrained nonlinear equations with application to optimization problems. *Numerische Mathematik*, 23(5): 371–385, 1975.

[118] S F McCormick. 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, 1977.

[119] M Métivier. *Semimartingales: A Course on Stochastic Processes*. De Gruyter Studies in Mathematics. XI, 1982. ISBN 9783110086744.

[120] A Mokhtari and A Ribeiro. Global convergence of online limited memory BFGS. *The Journal of Machine Learning Research*, 16(1):3151–3181, 2015.

[121] J L Mueller and S Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, 2012.

[122] NASA. Images from NASA webpage. `https://www.nasa.gov`, 2019. Accessed: 2019-01-10.

[123] F Natterer. *The mathematics of computerized tomography*. SIAM, 2001.

[124] D Needell. Topics in compressed sensing. *arXiv preprint arXiv:0905.4482*, 2009.

[125] D Needell, N Srebro, and R Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming*, 155(1-2):549–573, 2016.

[126] A Nemirovski, A Juditsky, G Lan, and A Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[127] J Ni, X Li, T He, and G Wang. Review of parallel computing techniques for computed tomography image reconstruction. *Current Medical Imaging Reviews*, 2(4):405–414, 2006.

[128] D Nikolayev, A Shmyrin, and S Blyumin. Tropical neighborhood and neural models for modelling of greedy organizational systems. *Global Journal of Pure and Applied Mathematics*, 12(6):4741–4747, 2016.

[129] J Nocedal and S J Wright. *Numerical Optimization*. Springer, New York, second edition, 2006.

[130] D P O'Leary and B W Rust. Variable projection for nonlinear least squares problems. *Computational Optimization and Applications*, 54(3):579–593, 2013.

[131] C C Paige and M A Saunders. Algorithm 583, LSQR: Sparse linear equations and least-squares problems. *ACM Transactions on Mathematical Software*, 8(2):195–209, 1982.

[132] C C Paige and M A Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[133] Y H Pao and F Khatibi. Neural network with non-linear transformations, December 18 1990. US Patent 4,979,126.

[134] S Park, M Park, and M Kang. Super-resolution image reconstruction: A technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003.

[135] M Pilanci and M J Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research*, 17(53):1–38, 2016.

[136] R A Renaut, S Vatankhah, and V E Ardestani. Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems. *SIAM Journal on Scientific Computing*, 39(2):B221–B243, 2017. doi: 10.1137/15M1037925.

[137] H Robbins and D Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Herbert Robbins Selected Papers*, pages 111–135. Springer, 1985.

[138] A K Saibaba, A l Alexanderian, and I C F Ipsen. Randomized matrix-free trace and log-determinant estimators. *Numerische Mathematik*, 137(2):353–395, 2017.

[139] T Schaul, S Zhang, and Ya LeCun. No more pesky learning rates. In *International Conference on Machine Learning*, pages 343–351, 2013.

[140] A Shapiro, D Dentcheva, and A Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*, volume 16. SIAM, Philadelphia, 2014.

[141] J Sjoberg and M Viberg. Separable non-linear least-squares minimization-possible improvements for neural net fitting. In *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop*, pages 345–354. IEEE, 1997.

[142] J T Slagel. The sherman morrison iteration. Master's thesis, Virginia Tech, 2015.

[143] J T Slagel, J Chung, M Chung, D Kozak, and L Tenorio. Sampled Tikhonov regularization for large linear inverse problems. *arXiv preprint arXiv:1812.06165*, 2018.

[144] J T Slagel, J Chung, and M Chung. Sample limited memory methods for least squares problems with massive data. *[in preperation]*, 2019.

[145] W Spakman and G Nolet. Imaging algorithms, accuracy and resolution in delay time tomography. In *Mathematical geophysics*, pages 155–187. Springer, 1988.

[146] J C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, Hoboken, 2005.

[147] S U Stich. *Convex optimization with random pursuit.* PhD thesis, ETH Zurich, 2014.

[148] T Strohmer and R Vershynin. Comments on the randomized kaczmarz method. *Journal of Fourier Analysis and Applications*, 15(4):437–440, 2009.

[149] T Strohmer and R Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

[150] R Sznajder. Kaczmarz algorithm revisited. *Czasopismo Techniczne*, 2016.

[151] K Tanabe. Projection method for solving a singular system of linear equations and its applications. *Numerische Mathematik*, 17(3):203–214, 1971.

[152] K Tanabe. Characterization of linear stationary iterative processes for solving a singular system of linear equations. *Numerische Mathematik*, 22(5):349–359, 1974.

[153] L Tenorio. Statistical regularization of inverse problems. *SIAM review*, 43(2):347–366, 2001.

[154] L Tenorio. *An Introduction to Data Analysis and Uncertainty Quantification for Inverse Problems.* SIAM, Philadelphia, 2017.

[155] R P Tewarson. Projection methods for solving sparse linear systems. *The Computer Journal*, 12(1):77–80, 1969.

[156] C Tompkins. Projection methods in calculation of some linear problems. In *BULLETIN OF THE AMERICAN MATHEMATICAL SOCIETY*, volume 55, pages 520–520. AMER MATHEMATICAL SOC 201 CHARLES ST, PROVIDENCE, RI 02940-2213, 1949.

[157] F Toutounian and S Karimi. Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides. *Applied Mathematics and Computation*, 178(2):452–460, 2006.

[158] B Triggs, P F McLauchlan, R I Hartley, and A W Fitzgibbon. Bundle adjustment, a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[159] M R Trummer. Reconstructing pictures from projections: on the convergence of the art algorithm with relaxation. *Computing*, 26(3):189–195, 1981.

[160] C R Vogel. *Computational methods for inverse problems*, volume 23. Siam, 2002.

[161] T Wallace and A Sekmen. Kaczmarz iterative projection and nonuniform sampling with complexity estimates. *Journal of medical engineering*, 2014, 2014.

[162] D T Westwick and R E Kearney. Separable least squares identification of nonlinear hammerstein models: Application to stretch reflex dynamics. *Annals of Biomedical Engineering*, 29(8):707–718, 2001.

[163] D P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–158, 2014.

[164] P Xu, J Yang, F Roosta-Khorasani, C Ré, and M W Mahoney. Sub-sampled Newton methods with non-uniform sampling. *arXiv preprint arXiv:1607.00559*, 2016.

[165] J F Yin and K Hayami. Preconditioned gmres methods with incomplete givens orthogonalization method for large sparse least-squares problems. *Journal of Computational and Applied Mathematics*, 226(1):177–186, 2009.

[166] M D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[167] J Zhao, Z Wang, and D S Park. Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing*, 87:79–89, 2012.

Appendices

Appendix A

Proofs for Chapter 3

This chapter contains the proofs for Chapter 3. First, we introduce the quasi-martingale convergence theorem in Section A.1. Almost sure convergence of stochastic Newton and quasi-Newton methods will be shown in Section A.2, and convergence rates of `slimLS` are shown in Section A.3.

A.1 The Quasi-martingale Convergence Theorem

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space, and let $\{\mathcal{F}_k\}$ be a sequence of sub-σ-algebras of $\mathcal{A}$. Then $\{\mathcal{F}_k\}$ is a called a filtration if $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ for all $k \in \mathbb{N}$. A sequence of random variables $\{u_k\}$ on $(\Omega, \mathcal{A})$ is said to be *adapted to* $\{\mathcal{F}_k\}$ if u_k is $\mathcal{F}_k$-measurable for all $k \in \mathbb{N}$. The σ-algebra generated by the random variables $\{u_i\}_{i=1}^{k}$ is denoted by $\sigma\left(u_i : i < k\right)$. We use $\mathbb{I}_A$ to denote the indicator function of the set A. The quasi-martingale convergence theorem is stated below. For details and proof, see [60, 105, 119].

Theorem A.1 (quasi-martingale convergence theorem)**.** *Let $\{X_k\}$ be a sequence of non-negative random variables adapted to a filtration $\{\mathcal{F}_k\}$ such that $\mathbb{E}X_k < \infty$ for all k. Let $Z_{k-1} = \mathbb{E}\left[X_k - X_{k-1} \,|\, \mathcal{F}_{k-1}\right]$. Then, if*

$$\sum_{k=1}^{\infty} \mathbb{E}\left[\mathbb{I}_{Z_{k-1} \geq 0}\left(X_k - X_{k-1}\right)\right] < \infty$$

then there is an non-negative random variable X with finite expectation such that $X_k \xrightarrow{\text{a.s.}} X$.

A.2 Almost Sure Convergence

Proof of Theorem 3.1. Define $\mathbf{H}_k = (\mathbf{A}_k)^\dagger \mathbf{W}_k^\top$, $\mathbf{C} = \mathbb{E}(\,\mathbf{H}_k^\top \mathbf{H}_k\,)$, $\mathbf{P} = \mathbb{E}\,\mathbf{H}_k$, and $\mathcal{F}_k = \sigma(\,\mathbf{W}_i;\, i < k\,)$. The matrices $\mathbf{C}$, and $\mathbf{P}$ are finite since $\mathbf{W}$ takes on only finitely many values. We split the proof into three parts

(i) The matrices $\mathbf{C}$ and $\mathbf{A}\mathbf{C}\mathbf{A}^\top$ are symmetric positive semi-definite, and $\mathbf{P}\mathbf{A}$ is symmetric positive definite (SPD).

(ii) If $e_k = \|\,\mathbf{x}_k - \widetilde{\mathbf{x}}\,\|^2$, then $\mathbb{E}\,e_k < \infty$ and $\mathbb{E}(\,\|\,\mathbf{s}_k\,\|^2\,) < \infty$ for all k.

(iii) $\mathbf{x}_k \xrightarrow{\text{a.s.}} \widetilde{\mathbf{x}}$.

(i) Let $\mathbf{v} \in \mathbb{R}^m$. Since $\mathbf{v}^\top \mathbf{C} \mathbf{v} = \mathbb{E}(\,\|\mathbf{H}_k \mathbf{v}\|^2\,) \geq 0$, it follows that $\mathbf{C}$ is semi-positive definite, and therefore so is $\mathbf{A}^\top \mathbf{C} \mathbf{A}$. Since $(\mathbf{W}_k^\top \mathbf{A})^\dagger \mathbf{W}_k^\top \mathbf{A}$ is symmetric, $\mathbf{P}\mathbf{A}$ is symmetric. Let $\mathbf{v} \in \mathbb{R}^n$. Using properties of the pseudoinverse gives

$$\mathbf{v}^\top \mathbf{P} \mathbf{A} \mathbf{v} = \mathbf{v}^\top \mathbb{E}(\,\mathbf{H}_k \mathbf{A}\,)\mathbf{v} = \mathbb{E}(\,\|\,\mathbf{H}_k \mathbf{A} \mathbf{v}\,\|^2\,) \geq 0,$$

where equality holds iff $\mathbf{H}_k \mathbf{A} \mathbf{v} = \mathbf{0}$ a.s., and since $\mathbb{E}(\mathbf{W}_k \mathbf{W}_k^\top) = \beta \mathbf{I}_m$ and $\mathbf{A}$ is full column-rank, it follows that $\mathbf{v} = \mathbf{0}$. Hence $\mathbf{P}\mathbf{A}$ is SPD.

(ii) Note that

$$\mathbf{s}_k = -\mathbf{H}_k \mathbf{A}(\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}) - \mathbf{H}_k\,\mathbf{r}, \tag{A.1}$$

where $\mathbf{r} = \mathbf{A}\widetilde{\mathbf{x}} - \mathbf{b}$. Therefore

$$\mathbf{x}_k - \widetilde{\mathbf{x}} = \mathbf{x}_{k-1} - \widetilde{\mathbf{x}} - \alpha_k \, \mathbf{H}_k \mathbf{A}(\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}) - \alpha_k \, \mathbf{H}_k \, \mathbf{r},$$

and since $\mathbf{H}_k\mathbf{A} = \left(\mathbf{W}_k^\top \mathbf{A}\right)^\dagger \mathbf{W}_k^\top \mathbf{A}$ is an orthogonal projection matrix, it follows that

$$e_{k+1} \leq 4(1+\alpha_k^2)\, e_k + 4\alpha_k^2 \, \|\mathbf{H}_k \mathbf{r}\|^2. \tag{A.2}$$

Using equation (A.2) and properties of the expected value leads to

$$\mathbb{E}\, e_{k+1} \leq 4(1+\alpha_k^2)\, \mathbb{E}\, e_k + 4\alpha_k^2 \, \|\mathbf{C}\mathbf{r}\|^2,$$

which implies that $\mathbb{E}\, \mathrm{e}_k < \infty$ and $\mathbb{E}(\, \|\mathbf{s}_k\|^2 \,) < \infty$ for all k.

(iii) The idea is to use the quasi-martingale convergence theorem. Since $\mathbf{W}_k$ and $\mathcal{F}_k$ are independent,

$$\begin{aligned}\mathbb{E}(\, e_{k+1} - e_k \mid \mathcal{F}_k \,) &= 2\alpha_k \, (\mathbf{x}_{k-1} - \widetilde{\mathbf{x}})^\top \mathbb{E}(\, \mathbf{s}_k \mid \mathcal{F}_k \,) + \alpha_k^2 \, \mathbb{E}(\, \|\mathbf{s}_k\|^2 \mid \mathcal{F}_k \,) \\ &= -2\alpha_k \, \|\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}\|_{\mathbf{PA}}^2 + \alpha_k^2 \, \mathbb{E}(\, \|\mathbf{s}_k\|^2 \mid \mathcal{F}_k \,). \end{aligned} \tag{A.3}$$

To put a bound on the second term of (A.3) we use again the fact that $\mathbf{H}_k\mathbf{A}$ is a projection matrix to obtain:

$$\mathbb{E}(\, \|\mathbf{s}_k\|^2 \mid \mathcal{F}_k \,) \leq 2(\mathbf{A}\widetilde{\mathbf{x}} - \mathbf{b})^\top \mathbf{C}(\mathbf{A}\widetilde{\mathbf{x}} - \mathbf{b}) + 2\|\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}\|^2 \leq c_1 + 2\|\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}\|^2, \tag{A.4}$$

where $c_1 = \lambda_{\max}(\mathbf{C}) \, \|\mathbf{A}\widetilde{\mathbf{x}} - \mathbf{b}\|^2$. Therefore, equation (A.3) can be bounded as

$$\mathbb{E}(\, e_{k+1} - e_k \mid \mathcal{F}_k \,) \leq -2\alpha_k \, \|\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}\|_{\mathbf{PA}}^2 + c_1\alpha_k^2 + 2\alpha_k^2 \, e_k,$$

which yields

$$\mathbb{E}(\, e_{k+1} - e_k(1+\alpha_k^2) \mid \mathcal{F}_k \,) \leq -2\alpha_k \, \|\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}\|^2_{\mathbf{PA}} + c_1\alpha_k^2 \leq c_1\alpha_k^2. \tag{A.5}$$

Let $\nu_k = \prod_{i=1}^{k-1}(\,1+\alpha_i^2\,)^{-1} < 1$. The sequence $\{\nu_k\}$ converges to some $\nu > 0$ because $\sum_{k=1}^{\infty}\alpha_k^2 < \infty$. Define $\widetilde{e}_k = \nu_k e_k$, and multiply both sides of (A.5) by ν_{k+1}. We obtain

$$\mathbb{E}(\, \widetilde{e}_{k+1} - \widetilde{e}_k \mid \mathcal{F}_k \,) \leq -2\alpha_k \, \nu_{k+1} \, \|\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}\|^2_{\mathbf{PA}} + c_1\alpha_k^2 \, \nu_{k+1} \leq c_1\alpha_k^2 \, \nu_{k+1}. \tag{A.6}$$

Define $Z_k = \mathbb{E}(\, \widetilde{e}_{k+1} - \widetilde{e}_k \mid \mathcal{F}_k \,)$. Then,

$$\mathbb{E}[\, \mathbb{I}_{Z_k \geq 0} \, (\widetilde{e}_{k+1} - \widetilde{e}_k) \,] = \mathbb{E}(\, \mathbb{I}_{Z_k \geq 0} \, \mathbb{E}[\, \widetilde{e}_{k+1} - \widetilde{e}_k \mid \mathcal{F}_k \,] \,) \leq c_1\alpha_k^2 \, \nu_{k+1}.$$

Since $\sum_{k=1}^{\infty}\alpha_k^2\nu_{k+1} < \infty$ the series $\sum_{k=1}^{\infty}\mathbb{E}(\, \mathbb{I}_{X_k \geq 0} \, (\tilde{e}_{k+1} - \tilde{e}_k) \,)$ converges and therefore $\{\tilde{e}_k\}$ converges a.s. by Theorem A.1. But since ν_k converges to a nonzero value, it also follows that $\{e_k\}$ converges a.s. The final step is to show that $\{e_k\}$ in fact converges to zero. Rearranging the terms and taking the expected value of both sides of (A.6) yields

$$\sum_{k=1}^{\infty} \alpha_k \nu_{k+1} \mathbb{E}(\, \|\mathbf{x}_{k-1} - \widetilde{\mathbf{x}}\|^2_{\mathbf{PA}} \,) < \infty.$$

Since $\sum_{k=1}^{\infty}\alpha_k = \infty$ and $\nu_k \to \nu > 0$, it follows that $\mathbb{E}(\, \|\mathbf{x}_{n_k} - \widetilde{\mathbf{x}}\|^2_{\mathbf{PA}} \,) \to 0$ for some subsequence (n_k), and therefore we also have $\mathbb{E}(\, \|\mathbf{x}_{n_k} - \widetilde{\mathbf{x}}\|^2 \,) = \mathbb{E}\, e_{n_k} \to 0$. By Fatou's lemma:

$$0 \leq \mathbb{E} \liminf e_{n_k} = \mathbb{E} \lim e_{n_k} \leq \liminf \mathbb{E}\, e_{n_k} = 0.$$

It follows that $\lim e_{n_k} = 0$ a.s. and since $\{e_k\}$ converges a.s., this implies $e_k \xrightarrow{\text{a.s.}} 0$ and therefore $\mathbf{x}_k \xrightarrow{\text{a.s.}} \widetilde{\mathbf{x}}$. □

Proof of Theorem 3.2. The proof of 3.2 does not require the quasi-martingale convergence theorem and is nearly identical to the proof of 4.2 part (i), in Chapter 4, Section 4.2.1. □

Proof of Theorem 3.3. To prove Theorem 3.3 first we introduce some useful notation. Denote

- $F(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}_{\text{LS}}\|_2^2$,
- $F_k = F(\mathbf{x}_k)$,
- $\nabla F_k = \nabla F(\mathbf{x}_k)$,
- $\mathbf{s}_k = \left(\mathbf{C}_k + \alpha_k \mathbf{M}_k^\top \mathbf{M}_k\right)^{-1} \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)$,
- $\mathcal{F}_k = \sigma\left(\mathbf{W}_i | i = 1, \dots k\right)$,
- $D = \lambda_{\max}\left(\mathbb{E}\mathbf{W}\mathbf{W}^\top\mathbf{W}\mathbf{W}^\top\right)$, and
- $\eta = \frac{\lambda_{\max}\left(\mathbf{A}^\top\mathbf{A}\right)^2 D}{\eta_{\min}}$.

Since F is twice continuously differentiable, by Taylor's theorem

$$F_k = F_{k-1} + \left(\mathbf{x}_k - \mathbf{x}_{k-1}\right)^\top \nabla F_{k-1} + \left(\mathbf{x}_k - \mathbf{x}_{k-1}\right)^\top \mathbf{A}^\top\mathbf{A}\left(\mathbf{x}_k - \mathbf{x}_{k-1}\right). \tag{A.7}$$

Since $\alpha_k \mathbf{s}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ we may re-write and bound equation (A.7)

$$\begin{aligned} F_k &= F_{k-1} + \alpha_k \mathbf{s}_k^\top \nabla F_{k-1} + \alpha_k^2 \left\|\mathbf{A}\mathbf{s}_k\right\|_2^2 && \text{(A.8)} \\ &\le F_{k-1} + \alpha_k \mathbf{s}_k^\top \nabla F_{k-1} + \frac{\alpha_k^2 \lambda_{\max}\left(\mathbf{A}^\top\mathbf{A}\right)^2}{\eta_{\min}} \left\|\mathbf{W}_k \mathbf{W}_k^\top \left(\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b}\right)\right\|_2^2 && \text{(A.9)} \\ &\le F_{k-1} + \alpha_k \mathbf{s}_k^\top \nabla F_{k-1} + \frac{\alpha_k^2 \lambda_{\max}\left(\mathbf{A}^\top\mathbf{A}\right)^2}{\eta_{\min}} \left\|\mathbf{W}_k \mathbf{W}_k^\top \left(\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b}\right)\right\|_2^2. && \text{(A.10)} \end{aligned}$$

Subtracting F_{k-1} from both sides and taking the expectation conditioned on $\mathcal{F}_{k-1}$ yields

$$\mathbb{E}\left[F_k - F_{k-1} \,|\, \mathcal{F}_{k-1}\right] \leq \alpha_k \mathbb{E}\left[\mathbf{s}_k^\top \,|\, \mathcal{F}_{k-1}\right] \nabla F_{k-1} + \alpha_k^2 \eta \left(F_{k-1} + \|\mathbf{A}\mathbf{x}_{\mathrm{LS}} - \mathbf{b}\|_2^2\right). \tag{A.11}$$

Subtracting $\alpha_k^2 \eta F_{k-1}$ from both sides

$$\mathbb{E}\left[F_k - (1+\alpha_k^2\eta) F_{k-1} \,|\, \mathcal{F}_{k-1}\right] \leq \alpha_k \mathbb{E}\left[\mathbf{s}_k^\top \,|\, \mathcal{F}_{k-1}\right] \nabla F_{k-1} + \alpha_k^2 \eta \|\mathbf{A}\mathbf{x}_{\mathrm{LS}} - \mathbf{b}\|_2^2. \tag{A.12}$$

Define $\gamma_k = \prod_{i=1}^k \left(1+\alpha_k^2\eta\right)^{-1} \leq 1$. Since

$$0 \leq -\log\left(\gamma_k\right) \leq \eta \sum_{i=1}^{\infty} \alpha_i^2 < \infty \tag{A.13}$$

for any k, it follow that $\{\gamma_k\}$ is a decreasing and convergence sequence to some $\gamma > 0$. Define $\tilde{F}_k = \gamma_k F_k$. Then

$$\mathbb{E}\left[\tilde{F}_k - \tilde{F}_{k-1} \,|\mathcal{F}_{k-1}\right] \leq \alpha_k \gamma_k \mathbb{E}\left[\mathbf{s}_k^\top \,|\, \mathcal{F}_{k-1}\right] \nabla F_{k-1} + \alpha_k^2 \gamma_k \eta \|\mathbf{A}\mathbf{x}_{\mathrm{LS}} - \mathbf{b}\|_2^2 \tag{A.14}$$

Now using the Woodbury formula

$$\begin{aligned}\mathbb{E}\left[\mathbf{s}_k^\top \,|\, \mathcal{F}_{k-1}\right] \nabla F_k = & -\left(\mathbf{A}^\top(\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b})\right)^\top \mathbf{C}_k^{-1} \mathbb{E}\left[\mathbf{A}_k^\top(\mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k) \,|\, \mathcal{F}_{k-1}\right] \\ & + \alpha_k \left(\mathbf{A}^\top(\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b})\right)^\top \mathbb{E}\left[\mathbf{C}_k^{-1}\mathbf{M}_k^\top \left(\mathbf{I} + \alpha_k \mathbf{M}_k \mathbf{C}_k^{-1} \mathbf{M}_k^\top\right)^{-1} \mathbf{M}_k \mathbf{C}_k^{-1} \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k)\right]\end{aligned}$$

Using the fact that $\mathbb{E}\left[\mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right) \,|\, \mathcal{F}_{k-1}\right] = \beta \mathbf{A}^\top \left(\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b}\right)$ and the Cauchy Schwarz inequality

$$
\begin{aligned}
\mathbb{E}\left[\mathbf{s}_k^\top \mid \mathcal{F}_{k-1}\right]\nabla F_{k-1} \leq & -\lambda_{\min}\beta\left\|\mathbf{A}^\top(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b})\right\|_2^2 \\
& +\alpha_k\left\|\mathbf{A}^\top(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b})\right\|_2\left\|\mathbb{E}\left[\mathbf{C}^{-1}\mathbf{M}_k^\top\left(\mathbf{I}+\alpha_k\mathbf{M}_k\mathbf{C}^{-1}\mathbf{M}_k^\top\right)^{-1}\mathbf{M}_k\mathbf{C}^{-1}\mathbf{A}_k^\top(\mathbf{A}_k\mathbf{x}_{k-1}-\mathbf{b}_k)\right]\right\|_2
\end{aligned}
$$

Using Jenson's inequality and the fact that the 2-norm is a convex function

$$
\begin{aligned}
\mathbb{E}\left[\mathbf{s}_k^\top \mid \mathcal{F}_k\right]\nabla F_k \leq & -\lambda_{\min}\left\|\mathbf{A}^\top\left(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b}\right)\right\|_2^2 \\
& +\alpha_k\left\|\mathbf{A}^\top(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b})\right\|_2\mathbb{E}\left[\left\|\mathbf{C}^{-1}\mathbf{M}_k^\top\left(\mathbf{I}+\alpha_k\mathbf{M}_k\mathbf{C}^{-1}\mathbf{M}_k^\top\right)^{-1}\mathbf{M}_k\mathbf{C}^{-1}\right\|_2\left\|\mathbf{A}_k^\top(\mathbf{A}_k\mathbf{x}_{k-1}-\mathbf{b}_k)\right\|_2\right] \\
\leq & -\lambda_{\min}\left(\mathbf{C}\right)\left\|\mathbf{A}^\top\left(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b}\right)\right\|_2^2+\alpha_k c^3 d
\end{aligned}
$$

where $\left\|\mathbf{C}^{-1}\mathbf{M}_k^\top\mathbf{M}_k\mathbf{C}^{-1}\right\|_2^2 \leq d$. We know that this d exists because $\mathbf{W}$ is from a finite sample space, and the sequence $\{\mathbf{C}_k\}$ has eigenvalues bounded away from infinity. We also know that $\mathbb{E}\left[\left\|\mathbf{A}_k^\top\left(\mathbf{A}_k\mathbf{x}_k-\mathbf{b}_k\right)\right\|_2\right] \leq c$. and $\left\|\mathbf{A}^\top\left(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b}\right)\right\|_2^2 = \left\|\mathbb{E}\left[\mathbf{A}^\top\mathbf{W}\mathbf{W}^\top\left(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b}\right)\right]\right\|_2^2 \leq \mathbb{E}\left[\left\|\mathbf{A}^\top\mathbf{W}\mathbf{W}^\top\left(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{b}\right)\right\|_2^2\right] \leq c^2$.

$$
\begin{aligned}
\mathbb{E}\left[\tilde{F}_k-\tilde{F}_{k-1}\,|\mathcal{F}_k\right] &\leq \alpha_k\gamma_k\mathbb{E}\left[\mathbf{s}_k^\top \mid \mathcal{F}_k\right]\nabla F_{k-1}+\alpha_k^2\gamma_k\left\|\mathbf{A}\mathbf{x}_{\mathrm{LS}}-\mathbf{b}\right\|_2^2 \\
&\leq -\alpha_k\gamma_k\lambda_{\min}\left(\mathbf{C}\right)\left\|\nabla F_{k-1}\right\|_2^2+\alpha_k^2\gamma_k c^3 d+\alpha_k^2\gamma_k\eta\left\|\mathbf{A}\mathbf{x}_{\mathrm{LS}}-\mathbf{b}\right\|_2^2, xt
\end{aligned}
\tag{A.15}
$$

Now note that

$$
\sum_{k=1}^{\infty}\mathbb{E}\left[\mathbb{I}_{Z_k\geq 0}\left(\tilde{F}_k-\tilde{F}_{k-1}\right)\right] \leq \left(\eta\left\|\mathbf{A}\mathbf{x}_{\mathrm{LS}}-\mathbf{b}\right\|_2^2+c^3 d\right)\sum_{k=1}^{\infty}\alpha_k^2 < \infty
$$

By the quasi-martingale convergence theorem $\tilde{F}_k$ converges almost surely. since γ_k converges to a nonzero value, it follows that F_k converges almost surely. The final step is to show that $F_k \to 0$. It follows from (A.15) that

$$\sum_{k=1}^{n} \alpha_k \gamma_k \lambda_{\text{min}}\left(\mathbf{C}\right) \mathbb{E} \left\|\nabla F_{k-1}\right\|_2^2 \leq \left(\eta \left\|\mathbf{A}\mathbf{x}_{\text{LS}} - \mathbf{b}\right\|_2^2 + c^3 d\right) \sum_{k=1}^{n} \alpha_k^2 + \mathbb{E}\tilde{F}_1$$

Therefore there must be a subsequence such that

$$\lim_{k\to\infty} \mathbb{E} \left\|\mathbf{A}^\top \mathbf{A}(\mathbf{x}_{n_k} - \mathbf{x}_{\text{LS}})\right\|_2 = 0$$

by Fatou's lemma we can say that $\mathbf{x}_{n_k} \to \mathbf{x}_{\text{LS}}$ and so we are done.

□

A.3 Convergence Rates

We begin with a lemma proving a selection of properties that will be useful in the proofs of Theorems 3.4 and 3.5.

Lemma A.2. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be full column rank and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{W} \in \mathbb{R}^{m\times \ell}$ be a random variable with M realizations and with the property that $\mathbb{E}\left[\mathbf{W}\mathbf{W}^\top\right] = \beta\mathbf{I}$. Denote the realizations of $\mathbf{W}$ as $\{\mathbf{W}^{(i)}\}_{i=1}^M$ with probabilities $\{p^{(i)}\}_{i=1}^M$ and associated blocks $\mathbf{A}^{(i)} = \left(\mathbf{W}^{(i)}\right)^\top \mathbf{A}$ and $\mathbf{b}^{(i)} = \left(\mathbf{W}^{(i)}\right)^\top \mathbf{b}$. Define $p_{min} = \min_i p^{(i)}$ and $p_{max} = \max_i p^{(i)}$. Let A_{min} and A_{max} be the smallest non-zero and largest eigenvalue of all matrices $\{\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\}_{i=1}^M$.*

Define $\mathbf{Z}_k = \left(\mathbf{I} + \alpha \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1}$ *and* $\mathbf{Z} = \mathbb{E}\left[\mathbf{Z}_k \mathbf{A}_k^\top \mathbf{A}_k\right]$. *Then, the following statements are true:*

1. $\mathbf{B}$ *defined in* (3.15)*, is symmetric positive definite,*

2. $\|\mathbb{E}\mathbf{Z}_k\|_2 < 1$*,*

3. $\|\mathbf{Z}^{-1}\|_2 \leq \frac{1+\alpha A_{min}}{p_{min} A_{min}}$*, and*

4. $\left\|\alpha \mathbf{Z}_k \mathbf{A}_k^\top \mathbf{A}_k\right\|_2 \leq \frac{\alpha A_{max}}{1+\alpha A_{max}}$*,*

Proof. We prove this in parts.

1. $\mathbf{B}$ is the sum of symmetric semi-definite matrices, and therefore is symmetric semi-definite or symmetric positive definite . Let $\mathbf{y} \in \text{Null}\left(\mathbf{B}\right)$. This means that $\mathbf{y}^\top \mathbf{B} \mathbf{y} = \mathbf{0}$, which implies $\mathbf{y}^\top \left(\alpha \mathbf{I} + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \mathbf{A}_k^\top \mathbf{A}_k \mathbf{y} = 0$ for all realizations of $\mathbf{W}$. Since $\left(\alpha \mathbf{I} + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \mathbf{A}_k^\top \mathbf{A}_k$ is semi-positive definite. this means that

$$\mathbf{y} \in \text{Null}\left(\left(\alpha \mathbf{I} + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \mathbf{A}_k^\top \mathbf{A}_k\right) = \text{Null}\left(\mathbf{A}_k^\top \mathbf{A}_k\right).$$

   It follows that

$$\mathbf{y}^\top \mathbf{A}^\top \mathbf{A} \mathbf{y} = \frac{1}{\beta} \mathbf{y}^\top \mathbb{E}\left[\mathbf{A}_k^\top \mathbf{A}_k\right] \mathbf{y} = 0.$$

   Since $\mathbf{A}$ is full column rank, $\mathbf{A}^\top \mathbf{A}$ is invertible. Therefore, $\mathbf{y} = \mathbf{0}$ and we conclude that $\mathbf{B}$ is invertible.

2. It is true that $\left\|\left(\mathbf{I} + \alpha \left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1}\right\|_2 \leq 1$ for each i such that $1 \leq i \leq M$, and so

$\left\|\mathbb{E}\mathbf{Z}_k\right\|_2 \le 1$ since

$$\begin{aligned}\left\|\mathbb{E}\mathbf{Z}_k\right\|_2 &= \left\|\sum_{i=1}^{M} p_i \left(\mathbf{I} + \alpha\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1}\right\|_2 \\ &\le \sum_{i=1}^{M} p_i \left\|\left(\mathbf{I} + \alpha\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1}\right\|_2 \\ &\le \sum_{i=1}^{M} p_i = 1.\end{aligned}$$

Since $\mathbf{Z}_k$ is always symmetric semi-definite, $\mathbb{E}\mathbf{Z}_k$ is symmetric semi-definite. If $\left\|\mathbb{E}\mathbf{Z}_k\right\|_2 = 1$, then there is a vector $\mathbf{y}$ such that

$$\mathbb{E}\mathbf{y}^\top \mathbf{Z}_k \mathbf{y} = \sum_{i=1}^{M} p_i \mathbf{y}^\top \left(\mathbf{I} + \alpha\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1} \mathbf{y} = 1. \tag{A.16}$$

This means that $\mathbf{y}$ is the eigenvector associated with the eigenvalue 1 for $\left(\mathbf{I} + \alpha\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1} = 1$ for each i such that $1 \le i \le M$. Therefore bfy is an eigenvector associated with the eigenvalue 1 for $\left(\mathbf{I} + \alpha\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right) = 1$ for each i such that $1 \le i \le M$.

However this implies that $\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\mathbf{y} = \mathbf{0}$ for all i such that $1 \le i \le M$. This means that

$$\mathbf{A}^\top \mathbf{A}\mathbf{y} = \frac{1}{\beta}\sum_{i=1}^{M} p_i \left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\mathbf{y} = 0$$

Which is not possible since $\mathbf{A}$ is full column rank. Therefore. $\left\|\mathbb{E}\mathbf{Z}_k\right\|_2 < 1$

3. We know that $\left\|\mathbf{Z}^{-1}\right\|_2 = \frac{1}{\lambda_{\min}(\mathbf{Z})}$. Let $y \in \mathbb{R}^n$. Since $\mathbf{Z}$ is positive definite $\mathbf{y}^\top \mathbf{Z}\mathbf{y} > 0$. It follows that

$$\mathbf{y}^\top \mathbf{Z}\mathbf{y} = \sum_{i=1}^{M} p_i \mathbf{y}^\top \left(\mathbf{I} + \alpha\left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1} \left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\mathbf{y}.$$

It follows that $\mathbf{y}^\top \left(\mathbf{I} + \alpha \left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1} \left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\mathbf{y} > 0,$, for some i such that $1 \leq i \leq M$. The minimum eigenvalue of $\left(\mathbf{I} + \alpha \left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}\right)^{-1} \left(\mathbf{A}^{(i)}\right)^\top \mathbf{A}^{(i)}$ is bounded below by $\frac{A_{\min}}{1+\alpha A_{\min}}$. Therefore

$$\mathbf{y}^\top \mathbf{Z}\mathbf{y} \geq \frac{p_{\min} A_{\min}}{1 + \alpha A_{\min}}, \tag{A.17}$$

which means $\|\mathbf{Z}^{-1}\|_2 \leq \frac{1+\alpha A_{\min}}{p_{\min} A_{\min}}$.

4. Every eigenvalue of $\mathbf{Z}_k \alpha \mathbf{A}_k^\top \mathbf{A}_k$ take the form $\frac{\alpha\lambda}{1+\alpha\lambda}$, where λ is an eigenvalue of $\mathbf{A}_k^\top \mathbf{A}_k$. Therefore, $\frac{\alpha\lambda}{1+\alpha\lambda} \leq \frac{\alpha A_{\max}}{1+\alpha A_{\max}}$, and we have our result.

□

We now prove Theorem 3.4.

Proof of Theorem 3.4. Let $\mathbf{Z}_k = \left(\mathbf{I} + \alpha \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1}$ and $\mathbf{Z} = \mathbb{E}\left[\mathbf{Z}_k \mathbf{A}_k^\top \mathbf{A}_k\right]$. Using the definition of $\widehat{\mathbf{x}}$ and $\mathbf{x}_{\text{LS}}$ and the fact that $\mathbf{A}\mathbf{b} = \frac{1}{\beta}\mathbb{E}\left[\mathbf{A}_k^\top \mathbf{b}_k\right]$ we have

$$\|\widehat{\mathbf{x}} - \mathbf{x}_{\text{LS}}\|_2 = \left\|\mathbf{Z}^{-1}\mathbb{E}\left[\mathbf{Z}_k \mathbf{A}_k^\top \mathbf{b}_k\right] - \left(\beta \mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbb{E}\left[\mathbf{A}_k^\top \mathbf{b}_k\right]\right\|_2.$$

Adding and subtracting $\mathbf{Z}^{-1}\mathbb{E}\left[\mathbf{A}_k^\top \mathbf{b}\right]$, using the triangle inequality, and using submultiplicativity of the 2-norm gives us

$$\|\widehat{\mathbf{x}} - \mathbf{x}_{\text{LS}}\|_2 \leq \left\|\mathbf{Z}^{-1}\right\|_2 \left\|\mathbb{E}\left[\mathbf{Z}_k \mathbf{A}_k^\top \mathbf{b}_k - \mathbf{A}_k^\top \mathbf{b}_k\right]\right\|_2 + \left\|\mathbf{Z}^{-1} - \left(\beta \mathbf{A}^\top \mathbf{A}\right)^{-1}\right\|_2 \left\|\mathbb{E}\left[\mathbf{A}_k^\top \mathbf{b}_k\right]\right\|_2. \tag{A.18}$$

Using the inequality

$$\left\|\mathbf{Z}^{-1} - \left(\beta \mathbf{A}^\top \mathbf{A}\right)^{-1}\right\|_2 \leq \left\|\mathbf{Z}^{-1}\right\|_2 \left\|\mathbf{Z} - \beta \mathbf{A}^\top \mathbf{A}\right\|_2 \left\|\left(\beta \mathbf{A}^\top \mathbf{A}\right)^{-1}\right\|_2$$

in (A.18) yields

$$\|\widehat{\mathbf{x}}-\mathbf{x}_{\mathrm{LS}}\|_2\leq\left\|\mathbf{Z}^{-1}\right\|_2\Big(\left\|\mathbb{E}\left[\mathbf{Z}_k\mathbf{A}_k^\top\mathbf{b}_k-\mathbf{A}_k^\top\mathbf{b}_k\right]\right\|_2+\left\|\left(\beta\mathbf{A}^\top\mathbf{A}\right)^{-1}\right\|_2\left\|\beta\mathbf{A}^\top\mathbf{A}-\mathbf{Z}\right\|_2\left\|\mathbb{E}\left[\mathbf{A}_k^\top\mathbf{b}_k\right]\right\|_2\Big).$$

Using Lemma (A.2) part (1) we have

$$\left\|\mathbf{Z}^{-1}\right\|_2^2 \leq \frac{1+\alpha A_{\min}}{p_{\min} A_{\min}},$$

Using the fact that $\mathbf{Z}_k - \mathbf{I} = -\alpha \mathbf{Z}_k \mathbf{A}_k^\top \mathbf{A}_k$ and Jensen's inequality

$$\begin{aligned}\left\|\mathbb{E}\left[\mathbf{Z}_k\mathbf{A}_k^\top\mathbf{b}_k-\mathbf{A}_k^\top\mathbf{b}_k\right]\right\|_2 &= \left\|\mathbb{E}\left[\left(\mathbf{Z}_k-\mathbf{I}\right)\mathbf{A}_k^\top\mathbf{b}_k\right]\right\|_2\\ &= \left\|\mathbb{E}\left[-\alpha\mathbf{Z}_k\mathbf{A}_k^\top\mathbf{A}_k\mathbf{A}_k^\top\mathbf{b}_k\right]\right\|_2\\ &\leq \mathbb{E}\left\|\alpha\mathbf{Z}_k\mathbf{A}_k^\top\mathbf{A}_k\right\|_2\left\|\mathbf{A}_k^\top\mathbf{b}_k\right\|_2.\end{aligned}$$

Using Lemma (A.2) part (4) we have

$$\left\|\mathbb{E}\left[\mathbf{Z}_k\mathbf{A}_k^\top\mathbf{b}_k-\mathbf{A}_k^\top\mathbf{b}_k\right]\right\|_2 \leq \frac{\alpha A_{\max}}{1+\alpha A_{\max}}\left\|\mathbb{E}\left[\mathbf{A}_k^\top\mathbf{b}_k\right]\right\|_2.$$

Also

$$\left\|\mathbf{Z}-\beta\mathbf{A}^\top\mathbf{A}\right\|_2 = \left\|\mathbb{E}\left[\left(\mathbf{Z}_k-\mathbf{I}\right)\mathbf{A}_k^\top\mathbf{A}_k\right]\right\|_2$$

and so a similar argument gives us the bound

$$\left\|\mathbf{Z}-\beta\mathbf{A}^\top\mathbf{A}\right\|_2 \leq \frac{\alpha A_{\max}}{1+\alpha A_{\max}}\mathbb{E}\left[\left\|\mathbf{A}_k^\top\mathbf{A}_k\right\|_2\right],$$

we have our result. □

Now we present the proof of Theorem 3.5

Proof of Theorem 3.5. We prove each part separately

1. Using the recursive definition of $\mathbf{x}_k$, and the expected value conditioned on $\mathcal{F}_k$

$$\begin{aligned}
\mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}} \middle| \mathcal{F}_k\right] &= \mathbf{x}_{k-1} - \widehat{\mathbf{x}} - \mathbb{E}\left[\mathbf{B}_k\mathbf{A}_k^\top\left(\mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k\right)\middle| \mathcal{F}_k\right] \\
&= \mathbf{x}_{k-1} - \widehat{\mathbf{x}} - \mathbf{B}\mathbf{x}_{k-1} - \mathbb{E}\left[\mathbf{B}_k\mathbf{A}_k^\top\mathbf{b}_k\right] \\
&= \mathbf{x}_{k-1} - \widehat{\mathbf{x}} - \mathbf{B}\left(\mathbf{x}_{k-1} - \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{B}_k\mathbf{A}_k^\top\mathbf{b}_k\right]\right) \\
&= \mathbf{x}_{k-1} - \widehat{\mathbf{x}} - \mathbf{B}\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right) \\
&= \left(\mathbf{I} - \mathbf{B}\right)\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right) \\
&= \left(\mathbf{I} + \alpha\mathbf{A}_k^\top\mathbf{A}_k\right)^{-1}\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right).
\end{aligned}$$

This last equality follows from the fact that

$$\begin{aligned}
\left(\mathbf{I} - \mathbf{B}\right) &= \mathbb{E}\left[\mathbf{I} - \left(\alpha^{-1}\mathbf{I} + \mathbf{A}_k^\top\mathbf{A}_k\right)^{-1}\mathbf{A}_k^\top\mathbf{A}_k\right] \\
&= \mathbb{E}\left[\left(\alpha^{-1}\mathbf{I} + \mathbf{A}_k^\top\mathbf{A}_k\right)^{-1}\alpha^{-1}\right] \\
&= \mathbb{E}\left[\left(\mathbf{I} + \alpha_k\mathbf{A}_k^\top\mathbf{A}_k\right)^{-1}\right].
\end{aligned}$$

Now $\mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}}\right] = \mathbb{E}\,\mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}} \middle| \mathcal{F}_k\right]$ so taking full expectation of both sides

$$\mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}}\right] = \mathbb{E}\left(\mathbf{I} + \alpha\mathbf{A}_k^\top\mathbf{A}_k\right)^{-1}\mathbb{E}\left[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right].$$

Since $\mathbf{A}_k$ are all i.i.d., unrolling the recursion gives

$$\mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}}\right] = \left(\mathbb{E}\left(\mathbf{I} + \alpha\mathbf{A}_k^\top\mathbf{A}_k\right)^{-1}\right)^k\mathbb{E}\left[\mathbf{x}_0 - \widehat{\mathbf{x}}\right].$$

Taking the norm of both sides and using sub-multiplicitivity gives us

$$\left\| \mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}}\right] \right\|_2 \leq \left\| \mathbb{E}\left(\mathbf{I} + \alpha \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \right\|_2^k \left\| \mathbb{E}\left[\mathbf{x}_0 - \widehat{\mathbf{x}}\right] \right\|_2 .$$

The norm $\left\| \mathbb{E}\left(\mathbf{I} + \alpha \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1} \right\|_2 < 1$ since $\mathbf{A}$ is full column rank. we have that $\left\| \mathbb{E}\left[\mathbf{x}_k - \widehat{\mathbf{x}}\right] \right\|_2 \to 0$ and so $\mathbb{E}\left[\mathbf{x}_k\right] \to \widehat{\mathbf{x}}$ at a linear rate, which is the result we are trying to prove.

2. Let us show the iterates $\mathbf{x}_k$ converge linearly to a convergence horizon near $\widehat{\mathbf{x}}$. Using the definition of $\mathbf{x}_k$ gives

$$\begin{aligned}
\left\|\mathbf{x}_k - \widehat{\mathbf{x}}\right\|_2^2 &= \left\|\mathbf{x}_{k-1} - \widehat{\mathbf{x}} - \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k)\right\|_2^2 \\
&= \left\|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right\|_2^2 - 2\left\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}} \, , \, \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) \ \right\rangle + \left\|\mathbf{B}_k \mathbf{A}_k^\top \left(\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k\right)\right\|_2^2 \\
&= \left\|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right\|_2^2 - 2\left\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}} \, , \, \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) \ \right\rangle \\
&\quad + \left\|\mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k(\mathbf{x}_{k-1} - \widehat{\mathbf{x}})) + \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k(\widehat{\mathbf{x}} - \mathbf{b}_k))\right\|_2^2 .
\end{aligned}$$

Using the fact that $\left\|\mathbf{x} + \mathbf{y}\right\|_2^2 \leq 2\left\|\mathbf{x}\right\|_2^2 + 2\left\|\mathbf{y}\right\|_2^2$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and using sub-multiplicitivity gives us

$$\begin{aligned}
\left\|\mathbf{x}_k - \widehat{\mathbf{x}}\right\|_2^2 &\leq \left\|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right\|_2^2 - 2\left\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}} \, , \, \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) \ \right\rangle \\
&\quad + 2\left\|\mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}))\right\|_2^2 + 2\left\|\mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2 \\
&\leq \left\|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right\|_2^2 - 2\left\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}} \, , \, \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_k - \mathbf{b}_k) \ \right\rangle \\
&\quad + 2\left\|\mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}))\right\|_2^2 + 2\lambda_{\max}^2\left(\mathbf{B}_k\right)\left\|\mathbf{A}_k^\top (\mathbf{A}_k \widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2 .
\end{aligned}$$

Note that $\lambda^2_{\max}\left(\mathbf{B}_k\right) \leq \alpha^2$, therefore

$$\begin{aligned}\|\mathbf{x}_k - \widehat{\mathbf{x}}\|_2^2 \leq \|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\|_2^2 - 2\left\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}_k\mathbf{A}_k^\top(\mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k)\ \right\rangle \\ + 2\left\|\mathbf{B}_k\mathbf{A}_k^\top(\mathbf{A}_k(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}))\right\|_2^2 + 2\alpha^2\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2.\end{aligned}$$

$\mathbf{B}_k\mathbf{A}_k^\top\mathbf{A}_k$ is a symmetric semi-definite matrix, so it has a Cholesky factorization $\mathbf{B}_k\mathbf{A}_k^\top\mathbf{A}_k = \mathbf{L}_k^\top\mathbf{L}_k$ [65]. Therefore

$$\begin{aligned}\left\|\mathbf{B}_k\mathbf{A}_k^\top\mathbf{A}_k(\mathbf{x}_k - \widehat{\mathbf{x}})\right\|_2^2 &= \left\|\mathbf{L}_k^\top\mathbf{L}_k\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}\right)\right\|_2^2 \\ &\leq \|\mathbf{L}_k\|_2^2\,\|\mathbf{L}_k\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}\right)\|_2^2 \\ &= \|\mathbf{L}_k\|_2^2\,\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}_k\mathbf{A}_k^\top\mathbf{A}_k\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right)\rangle.\end{aligned}$$

Now $\|\mathbf{L}_k\|_2^2$ is the maximum eigenvalue of $\mathbf{L}_k^\top\mathbf{L}_k = \mathbf{B}_k\mathbf{A}_k^\top\mathbf{A}_k = \left(\mathbf{I} + \alpha\mathbf{A}_k^\top\mathbf{A}_k\right)^{-1}\alpha\mathbf{A}_k^\top\mathbf{A}_k$. Let $A_{\max}$ denote the maximum eigenvalue of $\mathbf{A}_k^\top\mathbf{A}_k$ for any realization of $\mathbf{W}$. Using the fact that $\left\|\mathbf{L}_k^\top\right\|_2^2 \leq \frac{\alpha A_{\max}}{1+\alpha A_{\max}}$ we get

$$\begin{aligned}\|\mathbf{x}_k - \widehat{\mathbf{x}}\|_2^2 \leq \|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\|_2^2 - 2\left\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}_k\mathbf{A}_k^\top(\mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k)\ \right\rangle \\ + 2\frac{\alpha A_{\max}}{1 + \alpha A_{\max}}\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}_k\mathbf{A}_k^\top\mathbf{A}_k\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right)\rangle + 2\alpha^2\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2.\end{aligned}$$

Taking the expectation of both sides conditioned on $\mathcal{F}_k$ gives

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}_k - \widehat{\mathbf{x}}\|_2^2\,|\,\mathcal{F}_k\right] &\leq \|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\|_2^2 - 2\left\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right)\ \right\rangle \\ &\quad + 2\frac{\alpha A_{\max}}{1 + \alpha A_{\max}}\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right)\rangle + 2\alpha^2\mathbb{E}\left[\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2\right] \\ &= \|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\|_2^2 - 2\left(1 - \tfrac{\alpha A_{\max}}{1+\alpha A_{\max}}\right)\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}(\mathbf{x}_{k-1} - \widehat{\mathbf{x}})\rangle + 2\alpha^2\mathbb{E}\left[\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2\right] \\ &= \|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\|_2^2 - \tfrac{2}{1+\alpha A_{\max}}\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,,\, \mathbf{B}(\mathbf{x}_{k-1} - \widehat{\mathbf{x}})\rangle + 2\alpha^2\mathbb{E}\left[\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2\right].\end{aligned}$$

Note that $\langle \mathbf{x}_{k-1} - \widehat{\mathbf{x}}\,, \mathbf{B}\left(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\right)\rangle \geq \lambda_{\min}\left(\mathbf{B}\right)$, therefore

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}_k - \widehat{\mathbf{x}}\|_2^2 \,|\, \mathcal{F}_k\right] &\leq \left(1 - 2\frac{\lambda_{\min}\left(\mathbf{B}\right)}{(1 + \alpha A_{\max})}\right)\|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\|_2^2 + 2\alpha^2\mathbb{E}\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2 \\ &= (1 - 2c)\|\mathbf{x}_{k-1} - \widehat{\mathbf{x}}\|_2^2 + 2\alpha^2\mathbb{E}\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2.\end{aligned}$$

Taking the expectation in of both sides, using the fact that $\mathbb{E}\,\mathbb{E}\left[\|\mathbf{x}_k - \widehat{\mathbf{x}}\|_2^2 \,|\, \mathcal{F}_k\right] = \mathbb{E}\left[\|\mathbf{x}_k - \widehat{\mathbf{x}}\|_2^2\right]$ and unrolling the recursion gives us

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}_k - \widehat{\mathbf{x}}\|_2^2\right] &\leq (1 - 2c)^k\,\mathbb{E}\left[\|\mathbf{x}_0 - \widehat{\mathbf{x}}\|_2^2\right] \\ &\quad + 2\alpha^2\mathbb{E}\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2\sum_{i=0}^{k-1}(1 - 2c)^i \\ &\leq (1 - 2c)^k\|\mathbf{x}_0 - \widehat{\mathbf{x}}\|_2^2 + \alpha^2c^{-1}\mathbb{E}\left\|\mathbf{A}_k^\top(\mathbf{A}_k\widehat{\mathbf{x}} - \mathbf{b}_k)\right\|_2^2.\end{aligned}$$

This last inequality is using the fact that

$$\sum_{i=0}^{k-1}(1 - 2c)^i \leq \sum_{i=0}^{\infty}(1 - 2c)^i = \frac{1}{2c},$$

given that $0 < 1 - 2c < 1$. The last step is to show that $0 < 1 - 2c < 1$.

Since $\mathbf{B}$ is invertible $\lambda_{\min}\left(\mathbf{B}\right) > 0$, which implies

$$\left(1 - 2\frac{\lambda_{\min}\left(\mathbf{B}\right)}{(1 + \alpha A_{\max})}\right) < 1.$$

Also

$$\lambda_{\min}\left(\mathbf{B}\right) \leq \lambda_{\max}\left(\mathbf{B}\right) \leq \frac{\alpha A_{\max}}{1 + \alpha A_{\max}},$$

which means

$$\left(1-2\frac{\lambda_{\min}\left(\mathbf{B}\right)}{(1+\alpha A_{\max})}\right) \geq \left(1-2\frac{\alpha A_{\max}}{(1+\alpha A_{\max})(1+\alpha A_{\max})}\right).$$

Since

$$2\frac{\alpha A_{\max}}{(1+\alpha A_{\max})(1+\alpha A_{\max})} \leq 1$$

we have $\left(1-2\frac{\lambda_{\min}(\mathbf{B})}{(1+\alpha A_{\max})}\right) > 0$. We have our result.

□

Appendix B

Derivation of the Sampled UPRE and GCV

B.0.1 Derivation of the Sampled UPRE

The basic idea is to find Λ_k by minimizing an estimate of the predictive error. Let the *sampled predictive error* be given by

$$P(\lambda) = \left\| \mathbf{W}_{\tau(k)}^{\top} (\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{A}\mathbf{x}_{\text{true}}) \right\|_2^2 .$$

Using the notation from (4.10), the expected sampled predictive error, $\mathbb{E}\, P(\lambda)$, can be written as

$$\mathbb{E} \left\| \mathbf{W}_{\tau(k)}^{\top} \left(\mathbf{A}\mathbf{C}_k(\lambda) - \mathbf{I}_m \right) \mathbf{A}\mathbf{x}_{\text{true}} \right\|_2^2 + \sigma^2 \mathbb{E} \operatorname{tr}\Big(\mathbf{C}_k(\lambda)^{\top} \mathbf{A}^{\top} \mathbf{W}_{\tau(k)} \mathbf{W}_{\tau(k)}^{\top} \mathbf{A}\mathbf{C}_k(\lambda) \Big) , \tag{B.1}$$

where the mixed term vanishes due to independence of $\mathbf{W}_{\tau(1)}, \ldots \mathbf{W}_{\tau(k)}$ and $\boldsymbol{\epsilon}$ and since $\mathbb{E}\boldsymbol{\epsilon} = \mathbf{0}$. Similar to the derivation for standard UPRE, the predictive error is not computable in practice since $\mathbf{x}_{\text{true}}$ is not available. Thus, we perform a similar calculation for the expected sampled residual norm,

$$\begin{aligned}\mathbb{E}\ \left\|\mathbf{W}_{\tau(k)}^{\top}\left(\mathbf{A}\mathbf{x}_k(\lambda)-\mathbf{b}\right)\right\|_2^2 &= \mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{C}_k(\lambda)-\mathbf{I}_m)\mathbf{b}\right\|_2^2\\ &= \mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{C}_k(\lambda)-\mathbf{I}_m)\mathbf{A}\mathbf{x}_{\text{true}}\right\|_2^2+\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{C}_k(\lambda)-\mathbf{I}_m)\boldsymbol{\epsilon}\right\|_2^2.\end{aligned} \tag{B.2}$$

Next, notice that using the trace lemma for a symmetric matrix [8], the second term in (B.2) can be written as

$$\sigma^2\Big(\mathbb{E}\,\mathrm{tr}\big(\mathbf{C}_k(\lambda)^{\top}\mathbf{A}^{\top}\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\big)-2\,\mathbb{E}\,\mathrm{tr}\big(\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\big)+\ell\Big). \tag{B.3}$$

Combining (B.1) with (B.2) and (B.3), we get

$$\mathbb{E}P(\lambda)=\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}\left(\mathbf{A}\mathbf{x}_k(\lambda)-\mathbf{b}\right)\right\|_2^2+2\sigma^2\,\mathbb{E}\,\mathrm{tr}\big(\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\big)-\sigma^2\ell.$$

Finally for a given realization, we get an estimator for the predictive risk

$$U_k(\lambda)=\left\|\mathbf{W}_{\tau(k)}^{\top}\left(\mathbf{A}\mathbf{x}_k(\lambda)-\mathbf{b}\right)\right\|_2^2+2\sigma^2\mathrm{tr}\big(\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\big)-\sigma^2\ell\,,$$

which is equivalent to (4.12).

B.0.2 Derivation of the Sampled GCV

Next, we derive the sampled generalized cross validation function, following a similar derivation of the cross validation and generalized cross validation function found in [66]. For notational simplicity, we denote $\mathbf{A}_{\tau(i)}=\mathbf{W}_{\tau(i)}^{\top}\mathbf{A}$ and $\mathbf{b}_{\tau(i)}=\mathbf{W}_{\tau(i)}^{\top}\mathbf{b}$. Then, notice that the kth iterate of `sTik`, which is given by $\mathbf{x}_k(\lambda)=\mathbf{C}_k(\lambda)\mathbf{b}$ is the solution to the following problem,

$$\min_{\mathbf{x}} \left\| \mathbf{A}_{\tau(k)}\mathbf{x} - \mathbf{b}_{\tau(k)} \right\|_2^2 + \lambda \left\| \mathbf{L}\mathbf{x} \right\|_2^2 + \left\| \begin{bmatrix} \mathbf{A}_{\tau(1)} \\ \vdots \\ \mathbf{A}_{\tau(k-1)} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b}_{\tau(1)} \\ \vdots \\ \mathbf{b}_{\tau(k-1)} \end{bmatrix} \right\|_2^2 .$$

To derive sampled GCV, at the kth iterate, define the $\ell \times \ell$ identity matrix with 0 is the jth entry, i.e.,

$$\mathbf{E}_j = \mathbf{I}_\ell - \mathbf{e}_j^\top \mathbf{e}_j,$$

here $\mathbf{e}_j$ is the jth column of the identity matrix. Our goal is to find $\mathbf{x}_{[j]}(\lambda)$, which is the solution to

$$\min_{\mathbf{x}} \left\| \mathbf{E}_j \left(\mathbf{A}_{\tau(k)}\mathbf{x} - \mathbf{b}_{\tau(k)} \right) \right\|_2^2 + \lambda \left\| \mathbf{L}\mathbf{x} \right\|_2^2 + \left\| \begin{bmatrix} \mathbf{A}_{\tau(1)} \\ \vdots \\ \mathbf{A}_{\tau(k-1)} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b}_{\tau(1)} \\ \vdots \\ \mathbf{b}_{\tau(k-1)} \end{bmatrix} \right\|_2^2 .$$

Then, the sampled cross-validation estimate for λ minimizes the average error,

$$V_k(\lambda) = \frac{1}{\ell} \sum_{j=1}^{\ell} \left(\mathbf{e}_j^\top \mathbf{b}_{\tau(k)} - \mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{x}_{[j]}(\lambda) \right)^2 .$$

Using the normal equations and the fact that $\mathbf{E}_j^\top \mathbf{E}_j = \mathbf{E}_j$, an explicit expression for $\mathbf{x}_{[j]}(\lambda)$ is given as

$$\begin{aligned} \mathbf{x}_{[j]}(\lambda) &= \left(\mathbf{A}_{\tau(k)}^\top \mathbf{E}_j^\top \mathbf{E}_j \mathbf{A}_{\tau(k)} + \lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k-1} \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)} \right)^{-1} \left(\mathbf{A}_{\tau(k)}^\top \mathbf{E}_j^\top \mathbf{E}_j \mathbf{b}_{\tau(k)} + \sum_{i=1}^{k-1} \mathbf{A}_{\tau(i)}^\top \mathbf{b}_{\tau(i)} \right) \\ &= \left(\mathbf{B}_k(\lambda)^{-1} - \mathbf{A}_{\tau(i)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{A}_{\tau(i)} \right)^{-1} \left(\sum_{i=1}^{k} \mathbf{A}_{\tau(i)}^\top \mathbf{b}_{\tau(i)} - \mathbf{A}_{\tau(k)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{b}_{\tau(k)} \right), \end{aligned}$$

where $\mathbf{B}_k(\lambda) = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1}$. Next defining $t_{jj} = \mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{B}_k(\lambda) \mathbf{A}_{\tau(k)}^\top \mathbf{e}_j$ and using the Sherman-Morrison-Woodbury formula, we get

$$\left(\mathbf{B}_k(\lambda)^{-1} - \mathbf{A}_{\tau(i)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{A}_{\tau(i)}\right)^{-1} = \tfrac{1}{1-t_{jj}} \left((1-t_{jj})\mathbf{B}_k(\lambda) + \mathbf{B}_k(\lambda)\mathbf{A}_{\tau(k)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{B}_k(\lambda)\right)$$

and after some algebraic manipulations, we arrive at

$$\mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{x}_{[j]}(\lambda) = \frac{1}{1-t_{jj}} \left(\mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{C}_k(\lambda)\mathbf{b} - t_{jj}\mathbf{e}_j^\top \mathbf{b}_{\tau(k)}\right).$$

Thus,

$$\mathbf{e}_j^\top \mathbf{b}_{\tau(k)} - \mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{x}_{[j]}(\lambda) = \frac{1}{1-t_{jj}} \mathbf{e}_j^\top \left(\mathbf{b}_{\tau(k)} - \mathbf{A}_{\tau(k)} \mathbf{x}_k(\lambda)\right)$$

and we can write the sampled cross-validation function as

$$V_k(\lambda) = \frac{1}{\ell} \left\|\mathbf{D}_k(\lambda)(\mathbf{b}_{\tau(k)} - \mathbf{A}_{\tau(k)}\mathbf{x}_k(\lambda))\right\|_2^2,$$

where $\mathbf{D}_k(\lambda) = \mathrm{diag}\left(\frac{1}{1-t_{11}}, \ldots, \frac{1}{1-t_{\ell\ell}}\right)$. Now the extension from the sampled cross-validation to the sampled generalized cross validation function is analogous to the generalization process from cross-validation to GCV provided in [66].