

**Numerical Optimization with Applications to Magnetic  
Resonance Imaging Reconstruction**

by

**Alex McManus**

B.S., Rochester Institute of Technology, 2017

M.S., University of Colorado Boulder, 2024

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Department of Applied Mathematics

2025

Committee Members:

Nicholas Dwork, Chair

Stephen Becker

Adrianna Gillman

Eduardo Corona

James Folberth

McManus, Alex (Ph.D., Applied Mathematics)

Numerical Optimization with Applications to Magnetic Resonance Imaging Reconstruction

Thesis directed by Dr. Nicholas Dwork

Magnetic Resonance Imaging (MRI) reconstruction is a challenging class of inverse problems. This dissertation contributes both theoretical and practical advancements at the intersection of convex optimization, monotone operator theory, and modern computational imaging.

The first contribution investigates a class of methods that reconstruct undersampled k-space data via parallel imaging using linear predictability. We demonstrate that the quality of image reconstruction is impacted by the direction in which undersampling occurred. We present a novel sufficient condition for successful image reconstruction and a metric to predict degradation.

Second, we develop a line search for the primal-dual hybrid gradient method, a first-order optimization method tailored for nonsmooth optimization problems common in MRI and other image reconstruction regimes. This method removes the need for parameter tuning, offering a practical solution with improved runtime in matrix-free settings.

Third, we investigate enforced data consistency in deep learning-based reconstruction. We introduce a proximal layer that enforces strict data fidelity and can be used by many existing architectures. We introduce a novel unrolled architecture that mimics the iterations of FISTA while enforcing data consistency. Finally, We prove that incorporating a data consistency layer mitigates localized spatial errors that may present as medical hallucinations.

Finally, we propose a novel class of triangular preconditioners for proximal methods. These non-symmetric operators generalize classical ideas from linear iteration theory to convex optimization, and we establish partial convergence results in the context of monotone operator splitting. Experimental results suggest their utility in accelerating convergence while maintaining theoretical tractability.

## **Dedication**

To my mother, who gave me a love for books and learning, and the encouragement to follow them. And to my father, whose own unquenchable thirst for knowledge continues to inspire me every day.

## Acknowledgements

No one makes it through graduate school alone, and I have immense gratitude for everyone who supported me through this journey.

I would first like to thank my advisor, Dr. Nicholas Dwork, for your guidance, patience, and steady encouragement over the years.

I am grateful for my co-advisor, Dr. Stephen Becker, who was my rock throughout this process. You have been constant source of strength, clarity, and support throughout all of graduate school.

I am deeply thankful for all my friends and colleagues in APPM. In particular Jacob Spainhour, my office-mate for several years, whose presence and humor kept me sane.

To Dr. James Folberth, Dr. Mike Clark, Adam Smith, and my other coworkers at iCR: thank you for always providing meaningful and engaging work during the summers, and for inspiring me to see this degree through to the end.

Thank you to Kurt Motekew, who guided my younger self onto the path I currently walk, both professionally and academically.

To my girlfriend Lauren: thank you for your endless patience, your strength, and for being a comforting place to land at the end of long days. And to Maia, my other office-mate, for reminding me to take breaks whether I wanted to or not.

And finally, to my family. Without the love and support of my parents and my brother, none of this would have been possible.

I owe everyone here a debt which I will never be able to repay. Thank you.

## Contents

### Chapter

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>                                  | <b>1</b> |
| 1.1      | Motivation and Scope . . . . .                       | 1        |
| 1.2      | Research Contributions . . . . .                     | 1        |
| 1.3      | Organization of the Thesis . . . . .                 | 2        |
| <b>2</b> | <b>Background &amp; Preliminaries</b>                | <b>3</b> |
| 2.1      | Functional Analysis Background . . . . .             | 3        |
| 2.1.1    | Normed Vector Spaces and Hilbert Spaces . . . . .    | 3        |
| 2.2      | Closed, Convex, and Proper Functions . . . . .       | 5        |
| 2.3      | Differentiability and Subdifferentiability . . . . . | 6        |
| 2.4      | Convex Analysis . . . . .                            | 8        |
| 2.4.1    | The Conjugate Function . . . . .                     | 8        |
| 2.4.2    | Infimal Convolution . . . . .                        | 8        |
| 2.5      | Set-Valued Operators and Monotonicity . . . . .      | 9        |
| 2.5.1    | Set-Valued Operators . . . . .                       | 9        |
| 2.5.2    | Monotonicity . . . . .                               | 10       |
| 2.5.3    | Resolvents and Reflected Resolvents . . . . .        | 11       |
| 2.6      | Proximal Operators and the Moreau Envelope . . . . . | 12       |
| 2.6.1    | Proximal Operator . . . . .                          | 12       |

|          |   |           |
|----------|---|-----------|
| 2.6.2    | Moreau Envelope . . . . .   | 14        |
| 2.7      | Optimization Algorithms . . . . .   | 16        |
| 2.7.1    | Proximal Point Method . . . . .   | 16        |
| 2.7.2    | Forward-Backward Splitting . . . . .  | 16        |
| 2.7.3    | Douglas-Rachford Splitting . . . . .  | 17        |
| 2.7.4    | Primal-Dual Douglas-Rachford . . . . .  | 17        |
| 2.7.5    | Primal-Dual Hybrid Gradient (PDHG) . . . . .  | 18        |
| 2.7.6    | FISTA (Accelerated Forward-Backward) . . . . .  | 18        |
| <b>3</b> | <b>The Dependence of Parallel Imaging with Linear Predictability on the Undersampling Direction</b> | <b>20</b> |
| 3.1      | Introduction . . . . .  | 20        |
| 3.2      | Theory . . . . .  | 22        |
| 3.2.1    | AUTO-SMASH . . . . .  | 23        |
| 3.2.2    | GRAPPA . . . . .  | 24        |
| 3.2.3    | SPIRiT . . . . .  | 26        |
| 3.2.4    | Accuracy Metric . . . . .   | 27        |
| 3.3      | Experimental Setup . . . . .  | 28        |
| 3.4      | Results . . . . .   | 29        |
| 3.5      | Discussion . . . . .  | 32        |
| 3.6      | Conclusion . . . . .  | 33        |
| Appendix | . . . . .   | 33        |
| 3.6.1    | * . . . . .   | 33        |
| <b>4</b> | <b>A Relaxed Primal-Dual Hybrid Gradient Method with Line Search</b>                                | <b>42</b> |
| 4.1      | Introduction . . . . .  | 42        |
| 4.2      | Preliminaries . . . . .   | 43        |
| 4.2.1    | Notation and Definitions . . . . .  | 43        |

|          |  |           |
|----------|--|-----------|
| 4.2.2    | The Problem . . . . .  | 46        |
| 4.3      | Algorithms . . . . .   | 47        |
| 4.3.1    | Douglas-Rachford Splitting . . . . .                               | 47        |
| 4.3.2    | Primal-Dual Hybrid Gradient . . . . .                              | 49        |
| 4.3.3    | The Relationship Between PDDR and PDHG . . . . .                   | 50        |
| 4.3.4    | The Averaged Operator Iteration Line Search . . . . .              | 51        |
| 4.4      | The Line Search . . . . .  | 52        |
| 4.4.1    | Convergence . . . . .  | 53        |
| 4.4.2    | Line Search Activation . . . . .                                   | 54        |
| 4.4.3    | Limitations with Matrix Free Implementations . . . . .             | 56        |
| 4.5      | Numerical Experiments . . . . .                                    | 56        |
| 4.5.1    | LASSO . . . . .  | 57        |
| 4.5.2    | 1D-TV Denoising . . . . .  | 57        |
| 4.5.3    | 2D-TV Denoising . . . . .  | 59        |
| 4.6      | Application to Magnetic Resonance Imaging Reconstruction . . . . . | 60        |
| 4.6.1    | MRI Preliminaries . . . . .  | 61        |
| 4.6.2    | Compressed Sensing with Homodyne Detection . . . . .               | 61        |
| 4.6.3    | Results . . . . .  | 64        |
| 4.7      | Conclusion . . . . .   | 65        |
| <b>5</b> | <b>Data Consistent Deep Learning</b>                               | <b>72</b> |
| 5.1      | Introduction . . . . .   | 72        |
| 5.2      | Background . . . . .   | 74        |
| 5.2.1    | Parallel Imaging . . . . .   | 76        |
| 5.2.2    | Compressed Sensing . . . . .                                       | 76        |
| 5.2.3    | Deep Learning in MRI . . . . .                                     | 78        |
| 5.3      | Methods . . . . .  | 79        |

|          |   |           |
|----------|---|-----------|
| 5.3.1    | Proximal Layer . . . . .                              | 79        |
| 5.3.2    | Convolutional Neural Networks . . . . .               | 80        |
| 5.3.3    | Variational Network . . . . .                         | 81        |
| 5.3.4    | Unrolled Networks . . . . .                           | 82        |
| 5.3.5    | Zero-Shot Self Supervised Learning . . . . .          | 83        |
| 5.4      | Results . . . . .                                     | 84        |
| 5.4.1    | FastMRI U-Net . . . . .                               | 84        |
| 5.4.2    | Variational Network . . . . .                         | 85        |
| 5.4.3    | Zero-Shot Self-Supervised Learning . . . . .          | 87        |
| 5.4.4    | Unrolled Network . . . . .                            | 88        |
| 5.5      | Theoretical Guarantees . . . . .                      | 89        |
| 5.5.1    | Sparsity in Levels . . . . .                          | 91        |
| 5.6      | Conclusion . . . . .                                  | 93        |
| <b>6</b> | <b>Triangular Preconditioning</b>                     | <b>98</b> |
| 6.1      | Introduction . . . . .                                | 98        |
| 6.1.1    | Abstract . . . . .                                    | 98        |
| 6.1.2    | Motivation . . . . .                                  | 99        |
| 6.1.3    | Preconditioners . . . . .                             | 99        |
| 6.2      | Theoretical Foundation . . . . .                      | 100       |
| 6.2.1    | Classical Linear Iteration Analogy . . . . .          | 100       |
| 6.2.2    | Asymmetric-PPA . . . . .                              | 102       |
| 6.2.3    | Triangular matrices . . . . .                         | 103       |
| 6.3      | Monotonicity and Convergence Considerations . . . . . | 104       |
| 6.3.1    | Useful Theorems . . . . .                             | 105       |
| 6.4      | The Proximal Point Method . . . . .                   | 106       |
| 6.4.1    | Classical Setting . . . . .                           | 107       |

|  |     |
|--|-----|
| 6.4.2 Our Setting: $V$ Triangular, Non-Symmetric . . . . . | 107 |
| 6.4.3 Sufficient Conditions (Partial Results) . . . . .    | 107 |
| 6.4.4 On Firm Nonexpansiveness . . . . .                   | 108 |
| 6.4.5 Summary . . . . .                                    | 108 |
| 6.5 Forward Backward . . . . .                             | 109 |
| 6.5.1 Forward-backward algorithm . . . . .                 | 109 |
| 6.6 Implementation Details . . . . .                       | 110 |
| 6.6.1 Solving Eq. (6.22) when $V$ is triangular . . . . .  | 110 |
| 6.7 Choosing A Good $V$ . . . . .                          | 111 |
| 6.7.1 Selecting for $V^{-1}T$ To Be Monotone . . . . .     | 111 |
| 6.7.2 Desiderata for selecting $V^{-1}$ . . . . .          | 112 |
| 6.8 Summary and Outlook . . . . .                          | 113 |

|                     |            |
|---------------------|------------|
| <b>Bibliography</b> | <b>114</b> |
|---------------------|------------|

## Tables

### **Table**

|  |    |
|--|----|
| 3.1 Values of the accuracy metric of section 3.2.4 applied for vertical and horizontal undersampling. The metric values are large in undersampling directions that lead to poor quality images, and small in undersampling directions that lead to high quality images. . . . .                          | 31 |
| 4.1 An example of the total timing difference between the other algorithms tested and rPDHG with line search given the need for a parameter search for other algorithms. The time to 1000 iterations given a set of parameters is longer for rPDHG with line search, but the total time is less. . . . . | 65 |
| 5.1 Paired comparison of image quality metrics with and without data consistency. Reported values are median $\pm$ interquartile range (IQR). Statistical significance was assessed using the Wilcoxon signed-rank test. . . . .   | 85 |
| 5.2 Comparison of coronal reconstruction quality metrics with and without data consistency (DC). . . . .   | 88 |
| 5.3 Comparison of axial reconstruction quality metrics with and without data consistency (DC). . . . .   | 89 |
| 5.4 Ablation study conducted using the unrolled network trained in a supervised learning fashion. . . . .  | 97 |

## Figures

### Figure

|  |    |
|--|----|
| 3.1 In this depiction of parallel imaging with linear prediction, a portion of a sampling pattern is shown: filled in circles represent a $k$ -space location that was collected and unfilled circles represent a location that was not. A distance threshold of 1 and a metric of $\ \cdot\ _\infty$ are depicted with blue contours. The undersampling rate in each direction leads to different patterns of collected and uncollected points. The arrows represent displacements between collected data that lie within the threshold distance and a point an uncollected point that must be estimated. . . . . | 23 |
| 3.2 A depiction of accurate interpolation with SPIRiT. For this hypothetical example, the directions that the approximation of Eq. 3.5 is satisfied are down and downward-left. Without noise, the value of the blue circle can be estimated by two interpolations along the path indicated with the arrows. . . . .   | 28 |
| 3.3 Biot-Savart simulations of an 8-element birdcage coil oriented in perpendicular directions. (The top/bottom simulation would be the center axial/sagittal slice for a brain coil, respectively). The sensitivity maps of each coil are shown on the left. The plots on the right show the sensitivities of each coil for a single horizontal/vertical line through the center of the sensitivity maps. The numbers below each plot show the condition number of a matrix created by concatenating the sensitivities in the plots above. . . . .  | 36 |

|   |    |
|---|----|
| 3.4 Reconstructions of sagittal slices of a knee with different sampling patterns. The columns from left to right are a representation of the sampling indicating the undersampling direction, the GRAPPA reconstruction, the SPIRiT reconstruction, and the g-factor of the SPIRiT reconstructions. The top and bottom rows show undersampling in the horizontal and vertical directions, respectively. Both sampling masks used a reduction factor of 2. All reconstructions used a $31 \times 31$ ACR and a $3 \times 3$ kernel. . . . .   | 37 |
| 3.5 Approximations of the sensitivity maps for each coil in an 8-coil birdcage for the data of Fig. 3.4. Note that there are only estimates of the sensitivity in pixels where the magnitude of the corresponding image is sufficiently high for an accurate estimate. . . . .  | 37 |
| 3.6 SPIRiT reconstructions of a sagittal slice of a brain from data collected with an 8-channel birdcage coil using a horizontal (left) and vertical (right) undersampling mask. Top row: $3 \times 3$ kernel. Middle row: $5 \times 5$ kernel. Bottom row: $7 \times 7$ kernel. All reconstructions with an undersampling factor of two and a $31 \times 31$ ACR. . . . .  | 38 |
| 3.7 Axial slices of the knee. The columns, from left to right, are the reconstruction with fully sampled data, representations of the sampling masks used, the GRAPPA reconstructions, the SPIRiT reconstructions, and the g-factors of the SPIRiT reconstructions. All images are size $320 \times 256$ . The top and middle rows have undersampling factors of two in a single direction (horizontal and vertical, respectively); the bottom row has a reduction factor of approximately two in both directions (for a total reduction factor of approximately four). All reconstructions from GRAPPA and SPIRiT were made with a $31 \times 31$ ACR and a $7 \times 7$ kernel. . . . . | 39 |
| 3.8 SPIRiT reconstructions of retrospectively downsampled ankle data. Left: horizontal undersampling pattern. Right: vertical undersampling pattern. Top row: reduction factor of 2, $3 \times 3$ kernel. Middle row: reduction factor of 3, $3 \times 3$ kernel. Bottom row: reduction factor of 3, $5 \times 5$ kernel. All reconstructions with a $31 \times 31$ ACR. . . . .  | 40 |

|     |  |    |
|-----|--|----|
| 3.9 | Approximations to the individual sensitivity maps for the data of Fig. 3.8. Note that there are only estimates of the sensitivity in pixels where the magnitude of the corresponding image is sufficiently high for an accurate estimate. . . . .  | 41 |
| 4.1 | A comparison of the objective values from the different optimization algorithms as they solve the regularized least squares problem in (4.25). . . . .   | 58 |
| 4.2 | The ground truth and noisy signals used for the one-dimensional total variation denoising problem. . . . .   | 59 |
| 4.3 | (Left) A comparison of the objective values of different algorithms solving the one-dimensional total variation denoising problem. The line search proposed in this manuscript converges with fewer iterations than either constituent line search alone. The objective value jumps often as different step sizes are chosen. The raw objective value for each step is shown for academic purposes; in practice, one would choose the best objective value so far. The PDHG line performs better than standard PDHG. The AOI line search alone converges to a poor solution. (Right) A comparison of the best answers generated by the different optimization algorithms after 1000 iterations, plotted as the residual. . . . . | 67 |
| 4.4 | The cameraman image used for the two-dimensional total variation denoising problem. On the right is the image with white noise added. . . . .  | 67 |
| 4.5 | (Left) A comparison of the objective values of the different optimization algorithms used for the two-dimensional total variation denoising problem. The AOI line search converged to a poor solution. The line search proposed in this manuscript took fewer iterations to converge to any given level of error and did not require searching over any parameters. The comparison against PDHG and PDHG with line search is done against their best sets of parameters. (Right) The final output of rPDHG, the line search proposed in this manuscript, solving the two dimensional TV denoising problem as in (4.26). . . . .  | 68 |

|   |    |
|---|----|
| 4.6 Sampling masks for three different methods of acceleration, where white indicates collected samples. A) is the sampling mask associated with partial Fourier sampling, where the entire top half plus some number of lines around the origin are collected. B) is the sampling mask associated with compressed sensing using a variable density Poisson disc sampling mask [39]. C) is the sampling mask for the novel combination of partial Fourier and compressed sensing reconstruction presented in this manuscript. It is the intersection of masks A) and B). . . . .  | 68 |
| 4.7 Comparison of a fully-sampled reconstruction of a sagittal slice of an ankle to two methods of reconstruction from data undersampled according to sampling mask Fig. 4.6-C. (Center) Reconstruction of the undersampled ankle data using the formulation of the problem in (4.30), solved with the line search presented in this manuscript. (Right) This reconstruction was made using the formulation of the problem in (4.30), solved with the PDHG method. Bottom row shows 4 times the magnitude difference between the accelerated reconstructions and the fully-sampled reconstruction. . . . .  | 69 |
| 4.8 Comparison of a fully-sampled reconstruction of an axial slice of a brain to reconstruction from data undersampled according to sampling mask Fig. 4.6-C. (Center) Reconstruction of the undersampled ankle data using the formulation of the problem in (4.30), solved with the line search presented in this manuscript. (Right) Reconstruction of the undersampled ankle data using the formulation of the problem in (4.30), solved with the standard PDHG method given in (4.16). Bottom row shows 4 times the magnitude difference between the accelerated reconstructions and the fully-sampled reconstruction. The reconstruction using the new line search shows the same quality reconstruction without requiring a lengthy parameter search. . . . . | 70 |

|  |    |
|--|----|
| 4.9 Objective value plots for the reconstruction of the same ankle data as Fig. 4.7 (left) and the brain data as in Fig. 4.8 (right). This compares the performance of rPDHG algorithm with standard PDHG and PDHG with Malitsky’s line search. Over time, the line search finds a better solution. The line search jumps in objective value a lot as it chooses different step sizes. The objective value at each iteration is shown for academic purposes, though in practice at each iteration the best so far would be chosen. . . . . | 71 |
| 5.1 A comparison of the undersampled reconstruction (left), and reconstructions using the network without data consistency (center) and with data consistency (right) on in-distribution data. The network without data consistency is overall “smoother” and has fewer noise artifacts. . . . .   | 86 |
| 5.2 A comparison of the fully sampled reconstruction (left), and reconstructions using the network without data consistency (center) and with data consistency (right) on out-of-distribution data. The network with data consistency produces a high-quality image, while the network without data consistency produces a low-quality reconstruction that lacks clinical interpretability. . . . .  | 87 |
| 5.3 A reconstruction using the variational network of axial T2 knee data. a) shows the reconstruction with fully sampled data. b) shows the inverse Fourier transform of the zero-filled data. c) shows the reconstruction from the network trained without data consistency. d) shows the reconstruction from the network trained with data consistency. The white box highlights a medical anomaly produced in the reconstruction without data consistency that is corrected in the reconstruction with data consistency. 94             |    |

|  |    |
|--|----|
| 5.4 An example of a hallucination fixed by data consistency. All data is undersampled knee data using an unrolled variational network [66]. a) shows the fully sampled reconstruction with the area of the hallucination in the white box. b) shows the zero-filled reconstruction. c) shows the reconstruction from the network without data consistency, exhibiting a hallucination inside the white box. d) shows the reconstruction from the network trained with data consistency with the hallucination no longer present. . . . . | 95 |
| 5.5 A zoomed in view of the white boxes from Fig. 3. i) shows the white box from the fully sampled reconstruction (box a from Fig. 3). ii) shows the white box from the reconstruction from the network without data consistency exhibiting the hallucination (box c from Fig. 3). iii) shows the white box from the reconstruction from the network with data consistency (box d from Fig. 3). . . . .  | 95 |
| 5.6 An example reconstruction from our novel unrolled network architecture, with and without data consistency. These reconstructions are made with a 10% sampling burden. On the left is the ground truth, fully sampled reconstruction. In the middle is the reconstruction from the network trained with data consistency. On the right is the reconstruction from the network trained without data consistency. . . . .   | 96 |
| 5.7 An example reconstruction from our novel unrolled network architecture, with and without data consistency. These reconstructions are made with a 7.5% sampling burden. On the left is the ground truth, fully sampled reconstruction. In the middle is the reconstruction from the network trained with data consistency. On the right is the reconstruction from the network trained without data consistency. . . . .  | 96 |

# Chapter 1

## Introduction

### 1.1 Motivation and Scope

Magnetic Resonance Imaging (MRI) reconstruction presents a challenging class of inverse problems characterized by ill-posedness, high dimensionality, and incomplete data acquisition. Modern approaches increasingly leverage optimization theory and machine learning to improve reconstruction quality and efficiency.

This dissertation focuses on the intersection of convex optimization, monotone operator theory, and computational methods in MRI reconstruction. It comprises both mathematical contributions to optimization theory and applied techniques for improving image reconstruction quality.

### 1.2 Research Contributions

The core contributions of this thesis span both mathematical theory and applied algorithms for image reconstruction. They are presented through four self-contained research topics:

- (1) **Directionality of Undersampling in MRI:** This work investigates how the orientation of k-space sampling affects image quality, specifically in parallel imaging techniques like SPIRiT and GRAPPA.
- (2) **A Relaxed Primal-Dual Method with Line Search:** A novel line search for a primal-dual algorithm is developed and applied to compressed sensing MRI, yielding both theoretical guarantees and practical improvements.

- (3) **Data Consistency in Deep Learning:** A hybrid deep learning architecture is introduced that enforces strict data fidelity, addressing the risk of hallucinated structures in standard deep learning methods.
- (4) **Triangular Preconditioning for Proximal Methods:** This work investigates using a class of novel triangular preconditioners on proximal methods. This includes a look into their properties, convergence guarantees, and potential uses.

### 1.3 Organization of the Thesis

- Chapter 2 provides mathematical and application-specific background.
- Chapters 3–6 present the four research papers.

## Chapter 2

### Background & Preliminaries

This chapter will collect all of the relevant background material and notation for the remainder of the paper. For the most part, we will follow the standard notation for optimization and convex analysis literature. More information about this can be found in Bauschke and Combettes.

We briefly review essential background from functional analysis and convex optimization.

#### 2.1 Functional Analysis Background

##### 2.1.1 Normed Vector Spaces and Hilbert Spaces

Let  $X$  be a vector space. In the interest of keeping things general, we will allow  $X$  to have a complex scalar field. In the event that some notation is greatly simplified by assuming  $X$  is real we will make note of that.

###### 2.1.1.1 Normed Vector Spaces

The building blocks of functional analysis are normed vector spaces. A **norm** on a vector space  $X$  is a function  $\|\cdot\| : X \rightarrow \mathbb{R}_{\geq 0}$  that satisfies the following properties:

- (1) (Non-negativity)  $\|x\| \geq 0$  for all  $x \in X$ , and  $\|x\| = 0$  if and only if  $x = 0$ .
- (2) (Homogeneity)  $\|\alpha x\| = |\alpha| \|x\|$  for all  $\alpha \in \mathbb{R}$  (or  $\mathbb{C}$ ) and  $x \in X$ .
- (3) (Triangle Inequality)  $\|x + y\| \leq \|x\| + \|y\|$  for all  $x, y \in X$ .

A **normed vector space** is a vector space  $X$  equipped with a norm  $\|\cdot\|$ . The norm induces a **topology** on  $X$  through the collection of **open sets** defined by the norm. The topology allows us to discuss concepts like convergence, continuity, and compactness in the context of vector spaces.

A **Banach space** is a normed vector space that is **complete** with respect to the metric induced by the norm. A space is **complete** if every Cauchy sequence converges to a limit in the space.

### 2.1.1.2 Hilbert Spaces

The more common space in this dissertation is a **Hilbert space**. We start by defining an inner product.

**Definition 1** (Inner Product). Let  $\mathcal{H}$  be a normed vector space and let  $F$  be a field called the **scalar field**. The **inner product** or **scalar product** is a function  $\langle \cdot, \cdot \rangle \rightarrow F$  with the following properties:

(1) (Linearity) We take the convention that the inner product is linear in the second argument.

$$\langle x, \alpha y + \beta z \rangle = \alpha \langle x, y \rangle + \beta \langle x, z \rangle \text{ for } x, y, z \in \mathcal{H} \text{ and } \alpha, \beta \in F.$$

(2) (Hermitian symmetry)  $\langle x, y \rangle = \overline{\langle y, x \rangle}$ , where  $\bar{\cdot}$  denotes the complex conjugate of a scalar.

(3) (Positive definite)  $\langle x, x \rangle \geq 0$  with  $\langle x, x \rangle = 0$  if and only if  $x = 0$ .

A normed vector space with an inner product is called an **inner product space** or a **pre-Hilbert space**. The norm induced by the inner product is given by

$$\|x\| = \langle x, x \rangle^{1/2}.$$

Thus every inner product space is a normed vector space. It's natural to ask when a normed vector space is an inner product space. The answer comes from the **parallelogram identity**.

**Theorem 2.1** (H & N Theorem 6.9 [63]). A normed vector space  $X$  is an inner product space with a norm derived from the inner product if and only if

$$\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2, \quad \forall x, y \in X. \quad (2.1)$$

Conversely, if a norm satisfies (2.1), then the equation

$$\langle x, y \rangle = \frac{1}{4} \left\{ \|x + y\|^2 - \|x - y\|^2 - i\|x + iy\|^2 + i\|x - iy\|^2 \right\}$$

defines an inner product on  $X$ .

A Hilbert space is a complete inner product space. In this dissertation, the typical examples are  $\mathbb{R}^n$ ,  $\mathbb{C}^n$ , and  $L^2(\Omega)$ , both of which are used as domains for image functions and operators. Completeness guarantees the convergence of Cauchy sequences, which is essential when analyzing fixed-point iterations of proximal algorithms.

Hilbert spaces also allow us to define notions of orthogonality and projections.

**Definition 2** (Orthogonality). Let  $x, y \in \mathcal{H}$ , a Hilbert space. We say that  $x$  and  $y$  are **orthogonal** if  $\langle x, y \rangle = 0$ .

**Definition 3** (Projection). Let  $\mathcal{M}$  be a closed linear subspace of  $\mathcal{H}$ .

(a) For each  $x \in \mathcal{H}$  there is a unique closest point  $y \in \mathcal{M}$  such that

$$\|x - y\| = \min_{z \in \mathcal{H}} \|x - z\|. \quad (2.2)$$

(b) The point  $y \in \mathcal{M}$  closest to  $x \in \mathcal{H}$  is the unique element of  $\mathcal{M}$  with the property that  $(x - y) \perp \mathcal{M}$ .

## 2.2 Closed, Convex, and Proper Functions

There are certain assumptions that we will place on functions that will be used repeatedly throughout this work. First we introduce the graph and epigraph of a function.

**Definition 4** (Graph). Let  $f : \mathcal{H} \rightarrow [-\infty, +\infty]$ . The **graph** of  $f$  is the set

$$\text{gra } f = \{(x, \nu) \in \mathcal{H} \times \mathbb{R} \mid f(x) = \nu\}. \quad (2.3)$$

**Definition 5** (Epigraph). Let  $f : \mathcal{H} \rightarrow [-\infty, +\infty]$ . The **epigraph** of  $f$  is the set

$$\text{epi } f = \{(x, \nu) \in \mathcal{H} \times \mathbb{R} \mid f(x) \leq \nu\}. \quad (2.4)$$

In essence, this is the set of all points “above” the graph of  $f$ . This is useful because the properties we want to define on functions can be defined in terms of their epigraphs.

**Definition 6** (Domain). Let  $f : \mathcal{H} \rightarrow [-\infty, +\infty]$ . The **domain** of  $f$  is the set

$$\text{dom } f = \{x \in \mathcal{H} \mid f(x) < +\infty\}. \quad (2.5)$$

**Definition 7** (Convex Function). Let  $f : \mathcal{H} \rightarrow [-\infty, +\infty]$ . We say  $f$  is **convex** if its epigraph is a convex set.

**Definition 8** (Closed Function). Let  $f : \mathcal{H} \rightarrow [-\infty, +\infty]$ . We say  $f$  is **closed** if its epigraph is a closed set.

In particular, continuity implies closedness of the function.

**Definition 9** (Proper Function). Let  $f : \mathcal{H} \rightarrow [-\infty, +\infty]$ . We say  $f$  is **proper** if it never attains the value  $-\infty$ , and if  $\text{dom } f \neq \emptyset$ .

**Definition 10.** The set of **closed, convex, and proper** functions  $f : \mathcal{H} \rightarrow (-\infty, +\infty)$  is denoted  $\Gamma_0(\mathcal{H})$ .

### 2.3 Differentiability and Subdifferentiability

There are notions of differentiability for Hilbert spaces, especially infinite dimensional spaces.

Let  $\mathcal{K}$  be a **Banach space** for this section.

**Definition 11** (Gâteaux Differentiability). Let  $C$  be a nonempty subset of  $\mathcal{H}$ , let  $T : C \rightarrow \mathcal{K}$ , and suppose that  $x \in C$  is such that there exists  $\alpha \in \mathbb{R}_{++}$  for all  $y \in \mathcal{H}$  such that  $[x, x + \alpha y] \subset C$ . Then  $T$  is **Gâteaux differentiable** at  $x$  if there exists an operator  $DT(x) \in \mathcal{B}(\mathcal{H}, \mathcal{K})$ , called the **Gâteaux derivative** of  $T$  at  $x$ , such that

$$DT(x)y = \lim_{\alpha \rightarrow 0} \frac{T(x + \alpha y) - T(x)}{\alpha}, \quad \forall y \in \mathcal{H}. \quad (2.6)$$

We say  $T$  is Gâteaux differentiable on  $C$  if it is Gâteaux differentiable at every point in  $C$ . Higher order Gâteaux derivatives are defined inductively. Furthermore, the Gâteaux derivative is unique whenever it exists [9].

**Definition 12** (Fréchet Differentiability). Let  $x \in \mathcal{H}$ , let  $C$  be a neighborhood of  $x$ , and let  $T : C \rightarrow \mathcal{K}$ . Then  $T$  is **Fréchet differentiable** at  $x$  if there exists an operator  $DT(x) \in \mathcal{B}(\mathcal{H}, \mathcal{K})$ , called the **Fréchet derivative** of  $T$  at  $x$ , such that

$$\lim_{\substack{y \rightarrow 0 \\ 0 \neq \|y\|}} \frac{\|T(x+y) - T(x) - DT(x)y\|}{\|y\|} = 0. \quad (2.7)$$

We say  $T$  is Fréchet differentiable on  $C$  if it is Fréchet differentiable at every point in  $C$ .

These two notions of differentiability are related. They completely coincide if  $\mathcal{H}$  is finite dimensional. Furthermore there is the following lemma:

**Lemma 1** (B&C Lemma 2.61 [9]). Let  $x \in \mathcal{H}$ , let  $C$  be a neighborhood of  $x$ , and let  $T : C \rightarrow \mathcal{K}$ . Suppose that  $T$  is Fréchet differentiable at  $x$ . Then the following hold:

- (1)  $T$  is Gâteaux differentiable at  $x$  and the two derivatives coincide.
- (2)  $T$  is continuous at  $x$ .

Many functions of interest in optimization are Fréchet differentiable, such as the Euclidean norm. In this thesis, if we say a function is “differentiable” we mean it is Fréchet differentiable. However there are useful functions that are neither Gâteaux nor Fréchet differentiable, such as the indicator function of a closed convex set. In this case, we can use the notion of subdifferentiability.

**Definition 13** (Subdifferential). Let  $f : \mathcal{H} \rightarrow (-\infty, \infty]$  be proper. The **subdifferential** of  $f$  is the set-valued operator  $\partial f : \mathcal{H} \rightrightarrows \mathcal{H}$  defined by

$$\partial f(x) = \{u \in \mathcal{H} : \langle y - x, u \rangle + f(x) \leq f(y), \forall y \in \mathcal{H}\}.$$

For  $x \in \mathcal{H}$ ,  $f$  is said to be **subdifferentiable** at  $x$  if  $\partial f(x) \neq \emptyset$ , and the elements of  $\partial f(x)$  are the **subgradients** of  $f$  at  $x$ .

If  $f$  is differentiable everywhere, then the subdifferential is single-valued everywhere and is the gradient everywhere. In this case,

$$\partial f(x) = \{\nabla f(x)\}, \quad \forall x \in \mathcal{H}.$$

## 2.4 Convex Analysis

### 2.4.1 The Conjugate Function

**Definition 14** (Conjugate Function). The **conjugate** or **Legendre-Fenchel conjugate** of a function  $f$  is defined as

$$f^*(y) = \sup_{x \in \mathcal{H}} \{\langle x, y \rangle - f(x)\}. \quad (2.8)$$

Regardless of  $f$ , its conjugate is **convex** and closed. If  $f \in \Gamma_0(\mathcal{H})$ , then the **bidual**  $f^{**} = f$ —this is half of the celebrated **Fenchel-Moreau theorem**. The Fenchel-Moreau theorem makes this an if and only if relationship: the set of functions that are equal to their bidual are exactly the closed, convex, and proper functions on  $\mathcal{H}$ .

The epigraph of  $f^*$  is the intersection of the epigraphs of the linear functions of  $y$

$$\langle x, y \rangle - f(x)$$

as  $x$  ranges over  $\mathcal{H}$ . Figure 13.1 in B&C shows that  $f^*(u)$  is the supremum of the signed vertical distance between the graph of  $f$  and that of the continuous linear function  $\langle \cdot, u \rangle$ .

The conjugate function of a norm is the indicator function for the dual norm unit ball. This also applies to the composition of norms, e.g. the  $\ell_{2,1}$  norm.

**Theorem 2.2** (Fermat's Rule). Let  $f : \mathcal{H} \rightarrow (-\infty, \infty]$  be proper. Then

$$\operatorname{argmin} f = \operatorname{zer} \partial f = \{x \in \mathcal{H} : 0 \in \partial f\}. \quad (2.9)$$

*Proof.* Let  $x \in \mathcal{H}$ . Then  $x \in \operatorname{argmin} f \Leftrightarrow \langle y - x, 0 \rangle + f(x) \leq f(y) \forall y \in \mathcal{H} \Leftrightarrow 0 \in \partial f(x)$ .  $\square$

**Theorem 2.3** (B&C Prop. 16.30 [9]). Let  $f \in \Gamma_0(\mathcal{H})$ . Then  $(\partial f)^{-1} = \partial f^*$ .

### 2.4.2 Infimal Convolution

**Definition 15** (Infimal Convolution). Let  $f$  and  $g$  be functions from  $\mathcal{H} \rightarrow (-\infty, \infty]$ . The **infimal convolution** of  $f$  and  $g$  is  $f \square g : \mathcal{H} \rightarrow (-\infty, \infty]$  defined by

$$(f \square g)(x) = \inf_{y \in \mathcal{H}} (f(y) + g(x - y)). \quad (2.10)$$

Infimal convolution will come up later, particularly in the context of proximal operators and proximal smoothing. It is called infimal convolution because it resembles the classic formula of integral convolution. It is intuitively understood as the dual of addition of functions [104, §16], in the sense that

$$(f \square g)^* = f^* + g^*.$$

## 2.5 Set-Valued Operators and Monotonicity

### 2.5.1 Set-Valued Operators

In this section, let  $\mathcal{H}$  be a (possibly complex) Hilbert space. Let  $2^{\mathcal{H}}$  denote the **power set** of  $\mathcal{H}$ . A **set-valued operator**  $A : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  is a function that assigns to each  $x \in \mathcal{H}$  a subset  $A(x) \subseteq \mathcal{H}$ . We will sometimes write  $A : \mathcal{H} \rightrightarrows \mathcal{H}$  to denote this.

**Definition 16** (Graph). The **graph** of an operator  $A$  is the set

$$\text{gr}(A) = \{(x, y) \in \mathcal{H} \times \mathcal{H} \mid y \in A(x)\}. \quad (2.11)$$

The graph makes it easy to define relationships between operators.

**Definition 17** (Inverse). The **inverse** of an operator  $A$  is the operator  $A^{-1} : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  defined with graph

$$\text{gr}(A^{-1}) = \{(y, x) \in \mathcal{H} \times \mathcal{H} \mid (x, y) \in \text{gr}(A)\}. \quad (2.12)$$

This relationship can also be written between the graphs themselves (with slight abuse of notation) as

$$\text{gr}(A^{-1}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{gr}(A), \quad (2.13)$$

where we interpret the matrix multiplication as

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \forall (x, y) \in \text{gr}(A).$$

For operator  $A$ , define left- and right-scalar multiplication as

$$(\lambda A)(x) = \{\lambda y : y \in A(x)\}, \quad (A\mu)(x) = \{y : y \in A(\mu x)\}. \quad (2.14)$$

The following relationships hold between the graph of  $A$  and the graphs of its inverse and scalar multiples:

$$\text{gr}(A^{-1}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{gr}(A), \quad \text{gr}(\lambda A \mu) = \begin{bmatrix} \mu^{-1} \mathcal{I} & 0 \\ 0 & \lambda \mathcal{I} \end{bmatrix} \text{gr}(A). \quad (2.15)$$

One particular relationship that will be useful later is  $(\lambda A)^{-1}$ . Using (4.1) we can write

$$\begin{aligned} \text{gr}(\lambda A)^{-1} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{gr}(\lambda A) \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix} \text{gr}(A) \\ &= \begin{bmatrix} 0 & \lambda \\ 1 & 0 \end{bmatrix} \text{gr}(A) \\ &= \begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{gr}(A) \\ &= \begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} \text{gr} A^{-1} \\ &= \text{gr}(A^{-1} \lambda^{-1}). \end{aligned}$$

This yields the relationship  $(\lambda A)^{-1} = A^{-1} \lambda^{-1}$ .

### 2.5.2 Monotonicity

Operator  $A$  is **monotone** if

$$\langle x - x', y - y' \rangle \geq 0, \quad \forall (x, y), (x', y') \in \text{gr}(A).$$

An operator  $A$  is **maximally monotone** if it is both monotone and there are no monotone operators  $B$  such that  $\text{gr}(A) \subset \text{gr}(B)$ . Monotone operators are related to both convex functions and positive definite matrices. For  $L \in \mathcal{B}(\mathcal{H})$ ,  $L$  is monotone if and only if it is positive semidefinite, i.e.  $\langle x, Lx \rangle \geq 0$  for all  $x \in \mathcal{H}$ .

The following definitions relate to notions of nonexpansiveness

**Definition 18** (Firmly Nonexpansive Operator). Let  $D$  be a nonempty subset of  $\mathcal{H}$  and  $T : D \rightarrow \mathcal{H}$ . Then  $T$  is **firmly nonexpansive** if

$$\|Tx - Ty\|^2 + \|(\mathcal{I} - T)x - (\mathcal{I} - T)y\|^2 \leq \|x - y\|^2, \quad \forall x, y \in \mathcal{H}. \quad (2.16)$$

**Definition 19** (Cocoercive Operator). Let  $D$  be a nonempty subset of  $\mathcal{H}$ ,  $T : D \rightarrow \mathcal{H}$ , and let  $\beta > 0$ . Then  $T$  is  **$\beta$ -cocoercive** (or  **$\beta$ -inverse strongly monotone**) if  $\beta T$  is firmly nonexpansive:

$$\langle x - y, Tx - Ty \rangle \geq \beta \|x - y\|^2 \quad \forall x, y \in \mathcal{H}. \quad (2.17)$$

**Theorem 2.4** (B&C Example 20.31 [9]). Let  $T : \mathcal{H} \rightarrow \mathcal{H}$  be  $\beta$ -cocoercive with  $\beta > 0$ . Then  $T$  is maximally monotone.

The following result is fundamental in the field, and useful for proving, e.g., convergence rates of forward-backward splitting algorithms.

**Theorem 2.5** (Baillon-Haddad [8]). Let  $f : \mathcal{H} \rightarrow \mathbb{R}$  be convex, Frechet differentiable on  $\mathcal{H}$ , and such that  $\nabla f$  is  $\beta$ -Lipschitz continuous for some  $\beta > 0$ . Then  $\nabla f$  is  $1/\beta$ -cocoercive.

### 2.5.3 Resolvents and Reflected Resolvents

We first introduce the **resolvent** of an operator.

**Definition 20** (Resolvent). Let  $A : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  be a set-valued operator and  $\lambda > 0$ . The **resolvent** of  $A$  is defined as

$$J_{\lambda A} = (\mathcal{I} + \lambda A)^{-1}. \quad (2.18)$$

The value of the resolvent at  $x \in \mathcal{H}$  is the set of all  $y \in \mathcal{H}$  such that

$$x - y \in A(y).$$

If  $A$  is maximally monotone, then  $J_{\lambda A}$  is single-valued, firmly non-expansive, and has full domain [43, Theorem 2].

The **reflected resolvent** of  $A$  is

$$R_{\lambda A} = 2J_{\lambda A} - \mathcal{I}. \quad (2.19)$$

If  $A$  is maximally monotone, then  $R_{\lambda A}$  is nonexpansive [9, Corollary 23.11].

The resolvents of  $A$  and its inverse satisfy the following relationships:

$$J_{\lambda A} + J_{(\lambda A)^{-1}} = \mathcal{I}, \quad R_{\lambda A} + R_{(\lambda A)^{-1}} = 0. \quad (2.20)$$

Using  $(\lambda A)^{-1} = A^{-1}\lambda^{-1}$  as shown above gives

$$J_{\lambda A} + \lambda J_{\lambda^{-1}A^{-1}} \circ \lambda^{-1} = \mathcal{I}, \quad R_{\lambda A} + \lambda R_{\lambda^{-1}A^{-1}} \circ \lambda^{-1} = 0. \quad (2.21)$$

## 2.6 Proximal Operators and the Moreau Envelope

### 2.6.1 Proximal Operator

**Definition 21** (Proximal Operator). Let  $f \in \Gamma_0(\mathcal{H})$ . The **proximal operator** associated with  $f$  is the (single valued) operator given by:

$$\text{prox}_{\lambda f}(x) = \underset{y \in \mathcal{H}}{\operatorname{argmin}} \left\{ f(y) + \frac{1}{2\lambda} \|x - y\|^2 \right\}. \quad (2.22)$$

Since  $f \in \Gamma_0(\mathcal{H})$ , the objective is strongly convex and the proximal operator is well-defined and single valued. The proximal operator is also the resolvent of the subdifferential of  $f$  [9, Example 23.3]:

$$\text{prox}_{\lambda f} = J_{\lambda \partial f}. \quad (2.23)$$

In some primal dual algorithms, one may need access to the proximal operator of  $f^*$ , the conjugate function of  $f \in \Gamma_0(\mathcal{H})$  (see section 2.4.1). This is made simple by the following result, which is applying the first relationship in (2.21) to (2.23):

**Theorem 2.6** (Moreau Decomposition). Let  $f \in \Gamma_0(\mathcal{H})$  and  $\gamma > 0$ . Then

$$x = \text{prox}_{\gamma f}(x) + \gamma \text{prox}_{\gamma^{-1}f^*}(\gamma^{-1}x), \quad \forall x \in \mathcal{H}. \quad (2.24)$$

**Theorem 2.7** (B&C Prop. 16.44). Let  $f \in \Gamma_0(\mathcal{H})$ , and let  $x$  and  $p$  be in  $\mathcal{H}$ . Then

$$p = \text{prox}_f x \iff x - p \in \partial f(p).$$

In other words,

$$\text{prox}_f = (\mathcal{I} + \partial f)^{-1}.$$

**Theorem 2.8** (B&C Prop. 12.29). Let  $f \in \Gamma_0(\mathcal{H})$ . Then

$$\text{Fix prox}_f = \text{Argmin } f,$$

where  $\text{Fix prox}_f$  is the set of **fixed points** of  $\text{prox}_f$

The connection between **Fermat's rule** and Thm. 2.8 shows the deep connection between monotone inclusion problems and convex optimization.

Following some inspiration from [95], there is an interpretation of the proximal operator as a **backwards** or **implicit** step of a discretization of the gradient flow. For a function  $f$ , define its **gradient flow** as

$$\frac{d}{dt}x(t) = -\nabla f(x(t)). \quad (2.25)$$

Here  $t \in \mathbb{R}^+$  and variable  $x : \mathbb{R}^+ \rightarrow \mathcal{H}$ . The equilibrium points of the gradient flow are exactly the minimizers of  $f$ .

If we approach this from a numerical analysis perspective, we can numerically solve (2.25) using discretization techniques. Let  $x_k = x(kh)$  be our approximation to the true solution at time step  $k$  for step size  $h$ . A **forward** Euler step to discretize the gradient flow would be

$$\frac{x_{k+1} - x_k}{h} = -\nabla f(x_k). \quad (2.26)$$

Rearranging yields the update as

$$x_{k+1} = x_k - h \nabla f(x_k), \quad (2.27)$$

which we recognize as a gradient descent step with step size  $h$ . Therefore, the gradient descent optimization method can be interpreted as a forward Euler step on the gradient flow of the function to be minimized.

We can similarly analyze the **backward** (or **implicit**) Euler step. The implicit Euler discretization of (2.25) is

$$\frac{x_{k+1} - x_k}{h} = -\nabla f(x_{k+1}). \quad (2.28)$$

Rearranging yields

$$x_{k+1} + h \nabla f(x_{k+1}) = x_k, \quad (2.29)$$

where we can rewrite the left side as

$$x_{k+1} + h \nabla f(x_{k+1}) = (\mathcal{I} + h \nabla f) x_{k+1}. \quad (2.30)$$

Then we can rewrite (2.29) as

$$x_{k+1} = (\mathcal{I} + h \nabla f)^{-1} x_k. \quad (2.31)$$

### 2.6.2 Moreau Envelope

**Definition 22** (Moreau Envelope). Let  $f : \mathcal{H} \rightarrow (-\infty, \infty]$  and let  $\gamma > 0$ . The **Moreau envelope** of  $f$  of parameter  $\gamma$  is

$${}^\gamma f = f \square \left( \frac{1}{2\gamma} \|\cdot\|^2 \right). \quad (2.32)$$

The Moreau envelope is a sort of regularized version of  $f$ . It has domain  $\mathcal{R}^n$ , even when  $f$  does not. In fact, it is Fréchet differentiable as shown in the next result.

**Theorem 2.9** (B&C Proposition 12.30 [9]). Let  $f \in \Gamma_0(\mathcal{H})$  and let  $\gamma > 0$ . Then  ${}^\gamma f : \mathcal{H} \rightarrow \mathbb{R}$  is Fréchet differentiable on  $\mathcal{H}$ , and its gradient

$$\nabla({}^\gamma f) = \gamma^{-1} \left( \mathcal{I} - \text{prox}_{\gamma f} \right) \quad (2.33)$$

is  $\gamma^{-1}$ -Lipschitz continuous.

This can be rewritten as

$$\text{prox}_{\gamma f}(x) = x - \nabla(\gamma f)x, \quad (2.34)$$

so the proximal operator has an interpretation as a gradient descent step on the Moreau envelope.

Given the interpretation of the infimal convolution as the dual of addition as in Def. 15, the Moreau envelope can be thought of as the conjugate of the conjugate of  $f$  regularized with a quadratic. That is, since

$$(f \square g)^* = f^* + g^*, \quad (2.35)$$

as discussed above,  $(\gamma f)^{**} = \gamma f$ , and since  $\frac{1}{2}\|\cdot\|_2^2$  is self-dual, then (2.32) yields

$$\gamma f = \left( f^* + \frac{1}{2}\|\cdot\|_2^2 \right)^*. \quad (2.36)$$

Hence this is a type of regularization on the conjugate of  $f$ . For example, applying this to  $f(x) = |x|$  yields the **Huber function**:

$$\phi_{\text{Huber}} = \begin{cases} x^2 & |x| \leq 1 \\ 2|x| - 1 & |x| > 1 \end{cases}. \quad (2.37)$$

A point  $x \in \mathcal{H}$  is in the argmin of a function  $f \in \Gamma_0(\mathcal{H})$  if and only if  $x$  is a fixed point of the proximal operator  $\text{prox}_{\lambda f}$  for  $\lambda > 0$ . That is,

$$\text{Fix } \text{prox}_{\lambda f} = \text{argmin } f. \quad (2.38)$$

## 2.7 Optimization Algorithms

This section briefly introduces a few core algorithms in convex optimization. These are iterative methods built from the operator-theoretic objects defined in previous sections, such as the proximal operator and resolvents of monotone operators.

### 2.7.1 Proximal Point Method

#### Proximal Point Method

**Problem.** Minimize  $f(x)$ , where  $f \in \Gamma_0(\mathcal{H})$ .

**Iteration.**  $x_{k+1} = \text{prox}_{\lambda f}(x_k)$

This method is a fixed-point iteration using the resolvent of  $\partial f$ . It weakly converges to a minimizer of  $f$  under general conditions. The proximal point method is conceptually foundational and can be viewed as the root of many other operator-splitting methods.

### 2.7.2 Forward-Backward Splitting

#### Forward-Backward Splitting (Proximal Gradient)

**Problem.** Minimize  $f(x) + g(x)$ , where  $f$  is convex and differentiable with  $\nabla f$  Lipschitz, and  $g \in \Gamma_0(H)$ .

**Iteration.**  $x_{k+1} = \text{prox}_{\lambda g}(x_k - \lambda \nabla f(x_k))$

This algorithm alternates a forward (gradient) step on  $f$  and a backward (proximal) step on  $g$ . When  $\nabla f$  is  $\beta$ -Lipschitz and  $\lambda \in (0, 2/\beta)$ , the method achieves  $\mathcal{O}(1/k)$  ergodic convergence in function values and weak convergence of iterates. See [13, 9].

### 2.7.3 Douglas-Rachford Splitting

#### Douglas-Rachford Splitting (DRS)

**Problem.** Minimize  $F + G$ , where  $F, G : \mathcal{H} \rightarrow 2^{\mathcal{H}}$ .

**Fixed Point Iteration.**  $y_{k+1} = (1 - \alpha_k)y_k + \alpha_k R_{\gamma G} R_{\gamma F} y_k$

**Practical Formulation.**

$$x_{k+1} = J_{\gamma F}(y_k)$$

$$w_{k+1} = J_{\gamma G}(2x_{k+1} - y_k)$$

$$y_{k+1} = y_k + \rho_k(w_{k+1} - x_{k+1}),$$

Here  $R_{\lambda F} = 2J_{\lambda F} - \mathcal{I}$  is the reflected resolvent, and  $\rho_k = 2\alpha_k$ . Setting  $F = \partial f$  and  $G = \partial g$  for  $f, g \in \Gamma_0(\mathcal{H})$  recovers the standard convex optimization method. This method solves monotone inclusions of the form  $0 \in \partial f(x) + \partial g(x)$ , and weak convergence to a solution is guaranteed under standard assumptions. For convex optimization, Douglas-Rachford can be interpreted as a splitting method for  $\text{prox}_{\lambda f}$  and  $\text{prox}_{\lambda g}$ . See [72, 9].

### 2.7.4 Primal-Dual Douglas-Rachford

#### Primal-Dual Douglas-Rachford (PDDR)

**Problem.** Minimize  $F + G$ , where  $F, G : \mathcal{H} \rightarrow 2^{\mathcal{H}}$ .

**Fixed Point Iteration.**  $y_{k+1} = (1 - \alpha_k)y_k - \alpha_k R_{(\gamma G)^{-1}} R_{\gamma F} y_k$

**Practical Formulation.**

$$x_{k+1} = J_{\gamma F}(y_k)$$

$$w_{k+1} = J_{(\gamma G)^{-1}}(2x_{k+1} - y_k)$$

$$y_{k+1} = (1 - \rho_k)y_k + \rho_k(x_{k+1} - w_{k+1}).$$

This primal-dual variant of Douglas-Rachford arises by using the Moreau decomposition to rewrite one step of the iteration. Alternatively, use the relation between reflected resolvents in (2.20)

in the fixed-point formulation of DRS.

### 2.7.5 Primal-Dual Hybrid Gradient (PDHG)

#### Primal-Dual Hybrid Gradient (Chambolle-Pock)

**Problem.** Solve

$$\min_{x \in H} f(x) + g(Ax) \quad \text{or equivalently} \quad \min_x \max_z \langle Ax, z \rangle + g(x) - f^*(z)$$

**Iteration.**

$$\bar{x}_{k+1} = J_{\tau_k F}(x_k - \tau_k A^* z_k)$$

$$\bar{z}_{k+1} = J_{\sigma_k G^{-1}}(z_k + \sigma_k A(2\bar{x}_{k+1} - x_k))$$

$$x_{k+1} = x_k + \alpha_k(\bar{x}_{k+1} - x_k)$$

$$z_{k+1} = z_k + \alpha_k(\bar{z}_{k+1} - z_k),$$

The PDHG algorithm solves convex-concave saddle-point problems with linear coupling. Step sizes  $\tau$ ,  $\sigma$ , and overrelaxation parameter  $\alpha_k$  must satisfy  $\tau\sigma\|A\|^2 \leq 1$  for convergence. Under convexity assumptions, the iterates converge weakly and the duality gap converges to zero. See [27].

### 2.7.6 FISTA (Accelerated Forward-Backward)

#### FISTA (Fast Iterative Shrinkage-Thresholding Algorithm)

**Problem.** Minimize  $f(x) + g(x)$ , where  $f$  is convex, differentiable,  $\nabla f$  is  $\beta$ -Lipschitz, and  $g \in \Gamma_0(H)$ .

**Iteration.**

$$x_{k+1} = \text{prox}_{\lambda g}(y_k - \lambda \nabla f(y_k))$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4(t_k)^2}}{2}$$

$$y_{k+1} = x_{k+1} + \left( \frac{t_k - 1}{t_{k+1}} \right) (x_{k+1} - x_k)$$

FISTA is an accelerated variant of forward-backward splitting. Under convexity and smoothness assumptions, the objective values converge at rate  $\mathcal{O}(1/k^2)$ . See [13].

## Chapter 3

# The Dependence of Parallel Imaging with Linear Predictability on the Undersampling Direction

### 3.1 Introduction

Magnetic resonance imaging (MRI) is a ubiquitously used and highly versatile gross medical imaging modality with abundant natural contrast; it is relatively slow due to the length of time required for longitudinal magnetization recovery, parameterized by  $T_1$ . Producing images of diagnostic quality requires patients to remain still for a long time. For a given sequence and contrast, scan time is directly proportional to the amount of data collected—therefore, it is advantageous to reconstruct good quality images with fewer samples.

Several methods to accelerate MRI have been developed, including parallel imaging [35], compressed sensing (CS) [76, 6, 41, 40], partial Fourier acquisition [90], and, more recently, the use of convolutional neural networks and other deep learning solutions [110, 59, 68]. Parallel imaging, which is ubiquitously used in the clinic, takes advantage of multiple sensing coils to reduce the amount of data required for a high-quality image. There are two distinct approaches to parallel imaging: model based reconstruction (which requires estimates of the sensitivity maps), and linear predictability (which uses a calibration region to estimate linear interpolation coefficients). The quintessential algorithms for parallel imaging with linear predictability are AUTO-SMASH [65], GRAPPA [56], and SPIRiT [78]. In [58], Haldar et al. show that a sufficient condition for parallel imaging with linear predictability is that the support of the imaged object is a strict subset of the field-of-view of the image. That is, if there are regions of the field-of-view that do not image the

subject and just image air, then parallel imaging with linear predictability is possible. Notably, this is not a sufficient condition generally; it may not be sufficient for a specific instance of linear predictability based on specific user parameters (i.e., the size of the convolution kernel used). Thus, even when this condition is met, parallel imaging algorithms may fail to yield a high-quality image, as we will show.

It is well known the amount of undersampling in parallel imaging algorithms can affect image quality. There is a vague notion that the coil sensitivity maps must vary in the direction of undersampling [16], but it has not been specified what form the spatial variations must take. In [119], the sampling pattern for a SENSE reconstruction is altered to optimize a function of the geometry factor (g-factor) and image acceleration. In [20], Bruer et al. shows that g-factor analysis can be done for parallel imaging with linear predictability. G-factor analysis, though, confounds errors in the reconstruction from the sensing array with that of the object to be imaged [119]. Moreover, when using parallel imaging with linear predictability, it has been suggested that an interpolation operator could yield accurate values for arbitrary displacements in the Fourier domain [55]. We provide evidence that this is not the case. Indeed, we show that for some coil arrangements, some directions of undersampling will lead to a poor image quality, regardless of the subject to be imaged.

In this manuscript, we will present a novel and precise sufficient condition for accurate parallel imaging with linear predictability. We will show that parallel imaging reconstruction algorithms based on linear predictability, such as GRAPPA [56] and SPIRiT [78], have an innate dependence on the direction of undersampling, which is based on the arrangement of the sensing coils. We explore this dependence by showing that GRAPPA (and by extension, SPIRiT) is based on similar assumptions as those required by AUTO-SMASH [65]. We show results where the quality of the reconstruction differs dramatically when the direction of undersampling is changed. We also show cases where the quality is impervious to the direction of undersampling and explain why this is the case. Lastly, we propose a metric based on the fully sampled auto-calibration region that identifies which undersampling direction (horizontal or vertical) will lead to a low-quality image.

This metric could be used to adaptively alter the sampling pattern of the remaining samples to ensure a high-quality image is reconstructed.

For simplicity, the discussion of this manuscript will be restricted to Cartesian sampling; the extension to parallel imaging with linear predictability when sampling with a non-Cartesian trajectory is as expected based on prior work [112, 121, 75].

### 3.2 Theory

In this section, we start by reviewing the main points of AUTO-SMASH [65] and show that the generalization of its theoretical basis justifies GRAPPA [56] and SPIRiT [78]. Note that a complete review of the theory of AUTO-SMASH using the notation of this manuscript can be found in Appendix 3.6.1. For the purposes of this discussion, we will assume that processing is performed in two-dimensions. These sampling patterns may be generated with a spin-warp trajectory using two dimensions of phase encodes and one dimension of readout; after inverse Fourier transforming along the readout direction, the data are placed in a hybrid space and the reconstruction of each slice may be considered independent of every other slice [11].

For this discussion, the MRI signal with spin density  $\rho : \mathbb{R}^2 \rightarrow [0, \infty)$  for coil  $j$  is

$$\begin{aligned} S_j(k_x, k_y) &= \iint dx dy C_j(x, y) \rho(x, y) \exp(-i k_x x - i k_y y) \\ &= \mathcal{F}\{C_j \rho\}(k_x, k_y), \end{aligned} \tag{3.1}$$

where  $C_j$  is the sensitivity function for the  $j^{\text{th}}$  coil,  $k_x = \gamma \int_0^{t_x} G_x(\tau) d\tau$ ,  $k_y = \gamma \int_0^{t_y} G_y(\tau) d\tau$ ,  $\gamma$  is the gyromagnetic ratio,  $G_x$  and  $G_y$  are the x and y gradient waveforms,  $t_x$  and  $t_y$  are the lengths of time that the respective gradient fields are turned on, and  $\mathcal{F}\{C_j \rho\}(k_x, k_y)$  denotes the Fourier transform of  $C_j \rho$  evaluated at  $(k_x, k_y)$ . For simplicity, we are ignoring the effects of relaxation and recovery.

The MRI machine samples the Fourier domain at a set of finite points according to a sampling pattern that is achieved through control of the gradient waveforms. An example sampling pattern for parallel imaging with linear predictability is depicted in Fig. 3.1. In the example of this figure,

the data are undersampled by a factor of 2 in both the horizontal and vertical directions (for a total undersampling factor of 4). Note that with  $J$  coils, each sampled point consists of  $J$  complex values, one for each coil.

The user supplies a metric and (at least one) threshold, perhaps one per direction. Points that were not collected are synthesized as linear combinations of those values that lie within the threshold. For the example provided in Fig. 3.1, the user has specified the  $\|\cdot\|_\infty$  as the metric and a single threshold of 1. Thus, for any uncollected Fourier location, linear coefficient are found to interpolate from all collected points that lie within a threshold's distance.

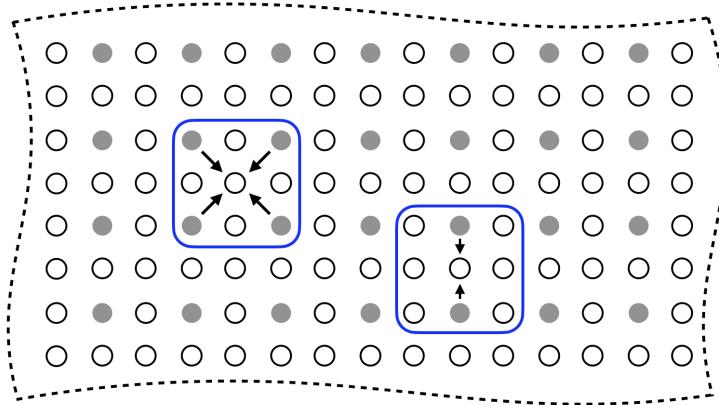


Figure 3.1: In this depiction of parallel imaging with linear prediction, a portion of a sampling pattern is shown: filled in circles represent a  $k$ -space location that was collected and unfilled circles represent a location that was not. A distance threshold of 1 and a metric of  $\|\cdot\|_\infty$  are depicted with blue contours. The undersampling rate in each direction leads to different patterns of collected and uncollected points. The arrows represent displacements between collected data that lie within the threshold distance and a point an uncollected point that must be estimated.

### 3.2.1 AUTO-SMASH

AUTO-SMASH is a seminal work of parallel imaging with linear predictability; the algorithm uses a two-dimensional spin-warp trajectory [44] (e.g., phase encode and readout in  $k_y$  and  $k_x$  dimensions, respectively). Whereas the Nyquist sampling theorem would dictate that the phase encoding lines be separated by  $\Delta k_y$ , AUTO-SMASH separates lines by  $M\Delta k_y$  for some  $M > 1$ . Consider the case where there are  $J$  coils, each with sensitivity map  $C_j$ . AUTO-SMASH defines a

composite sensitivity map,  $C_0^{\text{comp}}$ , created by a linear combination of the individual coil sensitivity maps with linear coefficients  $n^{(0)} \in \mathbb{C}^J$ . That is,  $C_0^{\text{comp}} = \sum_j n_j^{(0)} C_j$ . AUTO-SMASH further assumes that for each  $m = 1, \dots, M - 1$ ; there exist unknown linear coefficients  $n^{(m)} \in \mathbb{C}^J$  such that  $\sum_j n_j^{(m)} C_j \approx C_0^{\text{comp}} \exp(i m \Delta k_y y)$ . Then, by the Fourier shift theorem (as detailed in Appendix 3.6.1),  $\sum_j n_j^{(m)} S_j(k_x, k_y) = \mathcal{F}\{C_0^{\text{comp}} \rho\}(k_x, k_y - m \Delta k_y)$ . In [44], the coils were designed to satisfy the above requirements.

This illustrates that the linear coefficients  $n^{(m)}$  permit interpolation from known values to unknown values located  $m \Delta k_y$  distance away in the  $k_y$  direction of the Fourier domain. It is assumed that these same coefficients are valid across the entire Fourier domain. With AUTO-SMASH, the coils were designed so that  $C_0^{\text{comp}} \approx \kappa$ , for some constant  $\kappa$ , across the field-of-view of the image. To determine the linear coefficients  $n^{(m)}$ , a set of lines in the Fourier domain (called the auto-calibration signal – ACS) are collected and a linear system  $\mathcal{S} n^{(m)} = s_{\text{acs}}$  is numerically solved, where  $s_{\text{acs}}$  is comprised of the values of the ACS. (See Appendix 3.6.1 for details.)

### 3.2.2 GRAPPA

GRAPPA was developed heuristically; this prevents an understanding of its implicit assumptions and limitations. We have found that GRAPPA can be explained with a theoretical basis very similar to that of AUTO-SMASH, as we will now do.

GRAPPA eliminates the assumption of known  $n^{(0)}$  and the existence of an approximately constant composite sensitivity map. Moreover, instead of searching for coefficients that correspond to a single predetermined displacement, GRAPPA attempts to find linear coefficients for multiple displacements. The combination of distance threshold and undersampling rate gives rise to different patterns of collected and uncollected points used for interpolation, as depicted in Fig. 3.1. Each unique pattern of collected points surrounding an uncollected point is called a kernel; The blue contours in Fig. 3.1 shows two different kernels.

For a given kernel, as with AUTO-SMASH, the linear interpolation coefficients can be determined by solving a linear system. The following is a least-squares problem that simultaneously

identifies the interpolation coefficients for all coils:

$$\underset{N}{\text{minimize}} \quad \|\mathcal{S} N - s_{\text{acr}}\|_2^2, \quad (3.2)$$

where  $\mathcal{S}$  is comprised of the appropriate values from the auto-calibration region,  $N$  is the matrix of weights we solve for, and  $s_{\text{acr}}$  is a matrix of points from the auto-calibration region, a region which has rectangular size  $n_x \times n_y$ . For a specific uncollected Fourier location  $k$ , let  $D_k$  be the number of nonzero points of the relevant kernel. The matrix  $\mathcal{S}$  will be of size  $[\eta_x \eta_y \times J D_k]$ ,  $N$  is a matrix of size  $[J D_k \times J]$ ,  $s_{\text{acr}}$  is a matrix of size  $[\eta_x \eta_y \times J]$ , and  $\eta_x$  and  $\eta_y$  are the number of times that the kernel fits inside the auto-calibration region in the  $k_x$  and  $k_y$  directions, respectively.

We will now present a novel explicit relationship between problem (3.2) and an assumption similar to that of AUTO-SMASH. An equivalent form of problem (3.2) is

$$\underset{N}{\text{minimize}} \quad \sum_{\mathbf{k}} |\mathcal{S}^{(k)} N - s_{\text{acr}}^{(k)}|^2, \quad (3.3)$$

where  $k \in \mathbf{k}$  is an individual location inside the auto-calibration region,  $\mathcal{S}^{(k)}$  is the row of  $\mathcal{S}$  that corresponds to location  $k$ , and  $s_{\text{acr}}^{(k)}$  is the  $k^{\text{th}}$  row of  $s_{\text{acr}}$ . We recognize further that we can write  $\mathcal{S}^{(k)} N$  as

$$\mathcal{S}^{(k)} N = \sum_{d \in \mathcal{K}_k} \sum_j S_j(k + d) n_j^{(d)} \quad (3.4)$$

where  $S_j(k + d)$  is the signal collected at point  $k + d$  from coil  $j$ ,  $n_j^{(d)}$  is the appropriate weight, the inner sum is over the coils, and the outer sum is over the collected sample points that lie within the kernel  $\mathcal{K}_k$ .

For a given displacement  $(d_x, d_y)$ , let us assume that

$$\sum_j n_j^{(d)} C_j \approx \exp \left( i (u d_x x + v d_y y) \right) C_\ell. \quad (3.5)$$

Then Eq. (3.4) becomes

$$\begin{aligned} \mathcal{S}^{(k)} N &= \sum_{d \in \mathcal{K}_k} \iint dx dy C_\ell(x, y) \rho(x, y) \exp(-i(k_x - d_x)x - i(k_y - d_y)y) \\ &= \mathcal{F}\{C_\ell \rho\}(k_x + d_x, k_y + d_y) \end{aligned} \quad (3.6)$$

This novel reformulation shows that, with the assumption of Eq. (3.5), for a given displacement, GRAPPA seeks a set of linear coefficients  $n^{(d)} \in \mathbb{C}^J$  such that the coil sensitivity maps approximate a complex exponential to best take advantage of the Fourier shift theorem. The notable difference is that the linear combination yields a complex exponential weighted by an individual coil's sensitivity rather than a composite sensitivity. Using analogous mathematics as presented in App. 3.6.1, performing a linear combination of the collected points from all coils linearly interpolates missing values of a specific coil's data. After the missing data of all coils is interpolated, the data from multiple coils can be combined into a single image [105].

Thus, we can now present a novel sufficient condition for estimation without error when the approximation of 3.5 is perfectly satisfied and without any noise. The formulation above shows that each point of the GRAPPA kernel corresponds to a set of weights (one for each coil). Each interpolated point can be thought of as a linear combination of sensitivity maps approximating a complex exponential of a specific frequency in accordance with Eq. (3.5). ***For a given location  $k$  and kernel, if at least one displacement vector to a collected point within the kernel satisfies Eq. (3.5), then the interpolation will be accurate.*** If more than one displacement vector satisfies Eq. (3.5), then GRAPPA finds the set of linear coefficients that interpolate from multiple points in a least-squares optimal sense.

### 3.2.3 SPIRiT

SPIRiT [78] is an extension of GRAPPA. With a fixed kernel size, rather than just interpolating from points that were collected, SPIRiT will use *every point* in the kernel, regardless of whether or not it was collected.

SPIRiT interpolates all values (even those that were collected) by solving the following constrained least-squares problem:

$$\underset{\theta}{\text{minimize}} \quad \frac{1}{2} \|G\theta - \theta\|_2^2 \quad \text{subject to} \quad \|D\theta - y\|_2^2 \leq \varepsilon, \quad (3.7)$$

where  $\|\cdot\|_2$  denotes the  $l_2$  norm,  $G$  represents linear interpolation from all values that lie within

the kernel (even those that were not collected),  $D$  is the linear transformation that isolates the sample points that were collected and,  $y$  is the values of the collected data, and  $\varepsilon$  is a bound on the noise power. In general, problem (3.7) can be solved with the Fast Iterative Shrinkage-Thresholding algorithm (FISTA) [13]. In [78], Lustig et al. set  $\varepsilon = 0$  and only solve for the values of the uncollected data, which somewhat simplifies the implementation of the optimization algorithm, but this is not necessary.

Here, again, we present a novel sufficient condition for estimation without error when the approximation of Eq. 3.5 is perfectly satisfied and without any noise. This novel sufficient condition for an accurate SPIRiT interpolation is similar to that which was developed for GRAPPA, but more general. Consider the set of collected data as a directed graph where the location of each Fourier value is a node and the displacement vectors from each point in the kernel to that location are the directed edges. ***Accurate interpolation at a location  $k$  is possible when there is a path from a collected data point to  $k$  such that all edges of that path satisfy the approximation of Eq. (3.5) well.*** Practically, any error in the approximation and noise in the values are amplified with each edge of the path; so shorter paths lead to more accurate interpolations.

Figure 3.2 depicts an example that explains this sufficient condition. In this example, we would like to interpolate the value of the blue circle. The directions where condition Eq. 3.5 is satisfied is down and downward-left. We could perfectly interpolate the value of the blue circle in two steps: 1) estimate all values below all sampled values, and 2) estimating all values downward and left of all known or estimated values. If this process were iterated until all unknown values were estimated, then it would solve problem Eq. 3.7 with an objective function value of 0 (the minimum possible value).

### 3.2.4 Accuracy Metric

In the results section, we will provide examples where the quality of the reconstruction is significantly degraded when the sufficient conditions identified are not met. Here, we present a metric that can be calculated from the ACR that identifies any undersampling directions that would

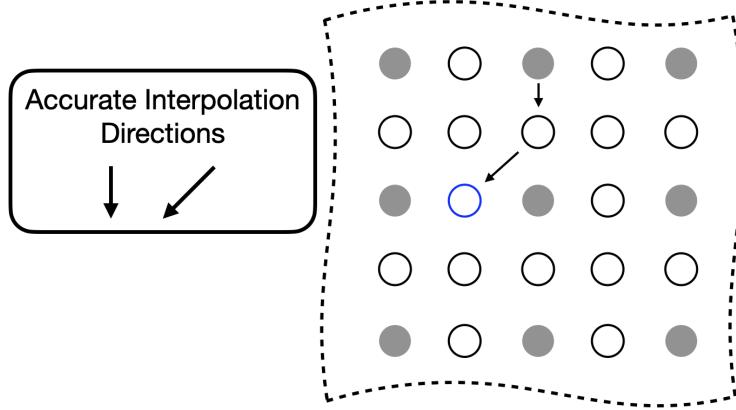


Figure 3.2: A depiction of accurate interpolation with SPIRiT. For this hypothetical example, the directions that the approximation of Eq. 3.5 is satisfied are down and downward-left. Without noise, the value of the blue circle can be estimated by two interpolations along the path indicated with the arrows.

not meet the sufficient conditions for accurate interpolation. Given the fully sampled ACR and a kernel size, we solve for the interpolation coefficients using two kernels—one solely in the horizontal direction and one in the vertical direction—of the same size as the kernel used for reconstruction. The kernel has a 0 in the center and all other values are 1. E.g., if the desired kernel is  $3 \times 3$ , the test kernels look like  $\mathbf{k}_h = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$  and  $\mathbf{k}_v = \mathbf{k}_h^T$ , where  $.^T$  denotes transpose. Denote the solutions to problem (3.2) with each of these kernels as  $N_h^*$  and  $N_v^*$ , respectively. The metric for accuracy is the relative norm error of the linear system:

$$\|\mathcal{S} N^* - s_{acr}\| / \|s_{acr}\| \quad (3.8)$$

If the value of the relative error is high, then there is not a consistent set of interpolation coefficients for the auto-calibration region in the direction specified by the kernel. If the value of the relative error is low, then there is and the data can be undersampled in that direction while retaining a high-quality reconstruction.

### 3.3 Experimental Setup

In this manuscript, we first use Biot-Savart simulations to examine an 8-element birdcage coil. We then analyze three different datasets: a knee, a brain, and an ankle. All datasets were

fully-sampled three-dimensional Cartesian data with two dimensions of phase encodes and one dimension of readout. The data of the knee was taken from the publicly available MRIData [93]. Data of the brain and ankle were collected with a 3 Tesla General Electric MR750 clinical scanner (GE Healthcare, Waukesha, WI). All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. MR data of humans were gathered with institutional review board (IRB) approval and Health Insurance Portability and Accountability Act (HIPAA) compliance. Informed consent was obtained from all individual participants included in the study.

Data was retrospectively downsampled for processing. The data was inverse transformed along the readout direction and placed into a hybrid space of  $(k_x, k_y, z)$ . Then, individual slices at specific  $z$  locations were isolated for further processing.

### 3.4 Results

Figure 3.3 shows results from Biot-Savart simulations for two perpendicular orientations of an 8-element birdcage coil according to [51]. The top/bottom of the figure shows simulations for an axial/sagittal plane that lies at the birdcage coil, respectively. The plots isolate single horizontal and vertical lines that lie at the center of the simulations; the number below each plot is the condition number of the matrix made by concatenating the vectors of the plots above it. This condition number is a metric that indicates how much variation there is between the sensitivity maps across space. For both the horizontal and vertical lines of the axial simulations, the condition number is on the order of  $10^3$ . With a high condition number, the solution to problem (3.2) will be heavily dependent on the noise and will not yield good results when the coefficients are later used for interpolation. While this remains the case for the horizontal line of the sagittal simulation, the condition number for the vertical line is much higher: on the order of  $10^8$ . By looking at the corresponding plot, it becomes obvious that the sensitivities are approximately scaled versions of each other, and that the problem of finding coefficients to linearly combine the sensitivity maps so

that they approximate a complex exponential is ill-conditioned.

Figure 3.4 shows reconstructions with GRAPPA and SPIRiT using two separate sampling patterns. Both data were retrospectively undersampled at the same reduction factor of 2; the only difference is the direction of undersampling. This dependence of quality on undersampling direction is present with both GRAPPA [56] and SPIRiT [78]. Note the higher values of the g-factor with the sampling pattern corresponding to horizontal stripes corresponds with the lower image quality.

Figure 3.5 illuminates why this happens based on the understanding presented in Sec. 3.2—for any specific horizontal location, the coil sensitivity functions as a function of vertical location are approximately scaled versions of each other. When undersampled by every other row, GRAPPA and SPIRiT will attempt to interpolate unknown values from collected data that lies above and below in the Fourier domain. With relatively little spatial variation in the coil sensitivities in those directions, there does not exist a linear combination of coil sensitivities such that they approximate a complex exponential and the interpolation coefficients do not yield an accurate estimate.

This data was collected with an 8-channel birdcage coil. Therefore, each coil extends from the most inferior to the most superior portions of the image. Thus, in the SI (vertical) direction, there is not enough variation to approximate a complex exponential well. This same phenomenon happens when imaging other anatomy with a similar coil arrangement; e.g., imaging the brain with a birdcage coil, as shown in Fig. 3.6.

Figure 3.7 shows MR images of an axial slice of the knee; this is the same dataset as Fig. 3.4. In this case, the quality of the reconstruction is independent of the undersampling direction. Moreover, when simultaneously undersampling in both directions, the quality remains high; though the signal-to-noise ratio has been reduced due to the reduced scan time [79, 88]. Good quality reconstructions in both directions indicates that the corresponding optimization problems to find the interpolation coefficients for GRAPPA and SPIRiT found good solutions. Owing to our understanding that the optimization problem is attempting to find linear coefficients for the coil sensitivities such that they linearly combine to a weighted complex exponential, the high quality indicates that this is true for at least one direction identified with the GRAPPA and SPIRiT kernels.

| Data                 | Kernel | Vertical      | Horizontal    |
|----------------------|--------|---------------|---------------|
| Knee—sagittal slice  | 3x3    | 55.0% (large) | 17.1% (small) |
| Knee—axial slice     | 3x3    | 25.2% (small) | 26.6% (small) |
| Ankle—sagittal slice | 3x3    | 8.1% (small)  | 5.0% (small)  |
| Brain—sagittal slice | 3x3    | 61.7% (large) | 5.0% (small)  |
| Brain—sagittal slice | 5x5    | 52.6% (large) | 3.9% (small)  |
| Brain—sagittal slice | 7x7    | 47.3% (large) | 3.6% (small)  |

Table 3.1: Values of the accuracy metric of section 3.2.4 applied for vertical and horizontal undersampling. The metric values are large in undersampling directions that lead to poor quality images, and small in undersampling directions that lead to high quality images.

Figure 3.8 shows reconstructions of a sagittal slice of an ankle from data collected with an 8-channel dedicated ankle coil arrangement. The SPIRiT reconstructions of retrospectively down-sampled data results in a high-quality reconstruction independent of the direction of undersampling. Fig. 3.9 shows the sensitivities of each coil. In contrast to the sensitivity maps of Fig. 3.5, each coil shows good spatial variation in both directions.

We varied reconstruction parameters for GRAPPA and SPIRiT reconstructions of the knee and brain to include kernel sizes of  $5 \times 5$  and  $7 \times 7$  and to include an undersampling factor of approximately 3. In all cases, the same trend was observed: the quality of the reconstruction is highly dependent on the undersampling direction. (Results are not shown.)

Table 3.1 shows the horizontal and vertical accuracy metrics for the data studied in this manuscript. All results conform to the expectations that undersampling directions which yield a high accuracy metric correspond to a poor quality image. The relative errors for those undersampling directions that were predicted to yield a poor quality image based on the accuracy metric are larger than those that were predicted to yield a high-quality image. The axial slice of the knee, which yielded a high-quality reconstruction independent of the undersampling direction, has a small relative error in both directions. For this dataset, a threshold on the relative error of 0.4 would identify directions that would yield a poor-quality image. This, however, is too small a dataset for us to make that a conclusion. Instead, we present this as a preliminary result and hope to pursue it further in the future.

### 3.5 Discussion

We have presented a rigorous physical reasoning that explains why the direction of undersampling can impact reconstruction quality.

Having the requisite spatial variation in the coil sensitivities is a *sufficient* condition for quality reconstruction based on linear predictability. There are other sufficient conditions that may be satisfied for linear predictability which may preclude this difficulty of GRAPPA and SPIRiT. One such sufficient condition is that the support of the imaged object be less than the field-of-view of the image [58]. Note further that this condition of support is met by the data studied in this manuscript; however, the quality remains dependent on the direction of undersampling. This indicates that, for the data studied, the more important consideration is the direction in which the coil sensitivities can be linearly combined to approximate a complex exponential.

Furthermore, as detailed in Eq. A5 of Appendix 3.6.1, the linear combination of the sensitivities only needs to approximate a complex exponential over the support of the image. Generally, the support of the image is a strict subset of the field of view. Consider the reconstruction of the ankle; there is little coil sensitivity in the superior-anterior quadrant of the image. However, there is plenty of coil variation over the other quadrants, which is where the ankle and leg are located. This is why the reconstruction of the ankle is robust regardless of the direction of undersampling. The coils are placed along the ankle and leg and oriented in similar directions (the normal vectors of the coils are all approximately parallel), allowing for local approximations of the sensitivities to complex exponentials.

Results indicate that the accuracy metric can be used to determine undersampling directions that will yield a poor-quality reconstruction, so that those undersampling directions can be avoided during the scan. In the future, we hope to test this metric on a much larger dataset to ensure that it is reliable. If so, we hope to create an adaptive algorithm that uses the auto-calibration region to identify good undersampling directions prior to collecting the outer portion of the Fourier domain. By doing so, parallel imaging with linear predictability would become robust to an inappropriately

selected undersampling direction.

Existing methods for combining parallel imaging with compressed sensing often utilize a variable density poisson disc sampling pattern [39] based on the intuition that samples should not be placed too close together since parallel imaging permits interpolation between samples [117]. Based on the work of this manuscript, parallel imaging with linear predictability may not be able to accurately interpolate between samples that are spaced in a given direction. If the parameter that governs variable density for the sampling pattern is radially symmetric, this indicates that the reconstruction relies more heavily on the sparsity assumptions of compressed sensing in one direction over another and suggests that there may be a sampling pattern that yields a higher quality when combining parallel imaging with compressed sensing.

### 3.6 Conclusion

We showed that the quality of parallel imaging reconstruction algorithms based on linear predictability, such as GRAPPA and SPIRiT, has a dependence on the direction of undersampling. This directional dependence is related to the amount of spatial variance in the individual coil sensitivities. The impact to the quality of reconstructions is the difference between having a diagnostic image or not. We note that this happens for specific coil configurations and body slices, whereas other configurations are robust to this problem.

## Appendix

### 3.6.1 AUTO-SMASH

In a coil array with  $J$  elements, the  $j^{\text{th}}$  coil has a distinct sensitivity function  $C_j : \mathbb{R}^2 \rightarrow \mathbb{C}$ . A composite sensitivity is generated as a linear combination of individual coil sensitivities with linear coefficients  $n_j^{(0)}$  as follows:

$$C_0^{\text{comp}}(x, y) = \sum_{j=1}^J n_j^{(0)} C_j(x, y). \quad (\text{A1})$$

The composite two-dimensional MR signal takes the form,

$$\begin{aligned} S^{\text{comp}}(k_x, k_y) &= \sum_{j=1}^J n_j^{(0)} S_j(k_x, k_y) \\ &= \iint dx dy \sum_{j=1}^J n_j^{(0)} C_j(x, y) \rho(x, y) \exp\{-ik_x x - ik_y y\} \\ &= \iint dx dy C_0^{\text{comp}}(x, y) \rho(x, y) \exp\{-ik_x x - ik_y y\} \end{aligned} \quad (\text{A2})$$

$$= \mathcal{F}\{C_0^{\text{(comp)}} \rho\}(k_x, k_y), \quad (\text{A3})$$

the Fourier transformation of  $C_0^{\text{comp}} \rho$ .

Suppose that there is another set of complex weights  $\{n_j^{(m)}\}$  such that the linear combination of coil sensitivities yields the following composite sensitivity:

$$C_m^{\text{comp}}(x, y) = \sum_{j=1}^J n_j^{(m)} C_j(x, y) \approx C_0^{\text{comp}} \exp(i m \Delta k_y y). \quad (\text{A4})$$

Importantly, this approximation only needs to be valid over the support of the image. With these linear coefficients, the composite MR signal becomes

$$\begin{aligned} \sum_{j=1}^J n_j^{(m)} S_j(k_x, k_y) &= \sum_{j=1}^J n_j^{(m)} \iint_{-\infty}^{\infty} dx dy C_j(x, y) \rho(x, y) e^{-i(k_x x + k_y y)} \\ &= \iint_{\Omega} dx dy \left[ \sum_{j=1}^J n_j^{(m)} C_j(x, y) \right] \rho(x, y) e^{-i(k_x x + k_y y)} \\ &\approx \iint_{\Omega} dx dy C_0^{\text{comp}} \exp(i m \Delta k_y y) \rho(x, y) e^{-i(k_x x + k_y y)} \\ &= \iint_{-\infty}^{\infty} dx dy C_0^{\text{comp}} \rho(x, y) e^{-i(k_x x + (k_y - m \Delta k_y) y)}, \\ &= \mathcal{F}\{C_0^{\text{comp}} \rho\}(k_x, k_y - m \Delta k_y), \end{aligned} \quad (\text{A5})$$

where  $\Omega$  is the support of  $\rho$ . The  $n^{(m)}$  coefficients serve to interpolate Fourier values at a distance of  $m \Delta k_y$ . Note that for the special case where  $C_0^{\text{comp}} \approx 1$ , the Fourier coefficients are those of  $\rho$ .

The innovation of AUTO-SMASH is to use collected auto-calibration signal (ACS) lines to estimate the weights  $\{n_j^{(m)}\}$  for Eq. A4. These ACS data,  $S_j^{ACR}$  are exactly shifted by the amount  $m \Delta k_y$ . The composite signal generated using weights  $\{n_j^{(0)}\}$  according to Eq. A2 yields

$$S^{\text{comp}}(k_x, k_y - m \Delta k_y) = \sum_{j=1}^J n_j^{(0)} S_j^{ACS}(k_x, k_y - m \Delta k_y). \quad (\text{A6})$$

Alternatively, following Eq. A5, we can write

$$S^{\text{comp}}(k_x, k_y - m\Delta k_y) = \sum_{j=1}^J n_j^{(m)} S_j(k_x, k_y). \quad (\text{A7})$$

Equating Eq. A6 and Eq. A7 yields

$$\sum_{j=1}^J n_j^{(m)} S_j(k_x, k_y) = \sum_{j=1}^J n_j^{(0)} S_j^{\text{ACS}}(k_x, k_y - m\Delta k_y) \quad (\text{A8})$$

Write the right-hand side of Eq. A8 simply as  $S^{\text{comp}}(k_x, k_y - m\Delta k_y)$  to reinforce that is the final, combined image produced using the original weights. For each  $k_x$  this is a (complex) scalar, and the left-hand side is a linear combination of the collected MR signals. We are attempting to solve for  $n_j^{(m)}$ .

To write this as a linear system, denote  $\Sigma$ ,  $n^{(m)}$  and  $b$  as follows:

$$\Sigma = \underbrace{\begin{bmatrix} S_1(k_{x_1}, k_y) & S_2(k_{x_1}, k_y) & \dots & S_J(k_{x_1}, k_y) \\ S_1(k_{x_2}, k_y) & S_2(k_{x_2}, k_y) & \dots & S_J(k_{x_2}, k_y) \\ \vdots & \ddots & & \vdots \\ S_1(k_{x_{n_x}}, k_y) & S_2(k_{x_{n_x}}, k_y) & \dots & S_J(k_{x_{n_x}}, k_y) \end{bmatrix}}_{n_x \times J},$$

$$n^{(m)} = \underbrace{\begin{bmatrix} n_1^{(m)} \\ n_2^{(m)} \\ \vdots \\ n_J^{(m)} \end{bmatrix}}_{J \times 1}, \text{ and } b = \underbrace{\begin{bmatrix} S^{\text{comp}}(k_{x_1}, k_y - m\Delta k_y) \\ S^{\text{comp}}(k_{x_2}, k_y - m\Delta k_y) \\ \vdots \\ S^{\text{comp}}(k_{x_{n_x}}, k_y - m\Delta k_y) \end{bmatrix}}_{n_x \times 1}.$$

To find the interpolation coefficients  $n^{(m)}$ , one can minimize  $\|\Sigma n^{(m)} - b\|_2$ .

Once the weights  $n^{(m)}$  are determined, the matrix-vector multiplication  $\Sigma n^{(m)}$  for  $\Sigma$  constructed for a specific  $(k_x, k_y)$  will estimate the *composite* Fourier value at  $(k_x, k_y - m\Delta k_y)$ .

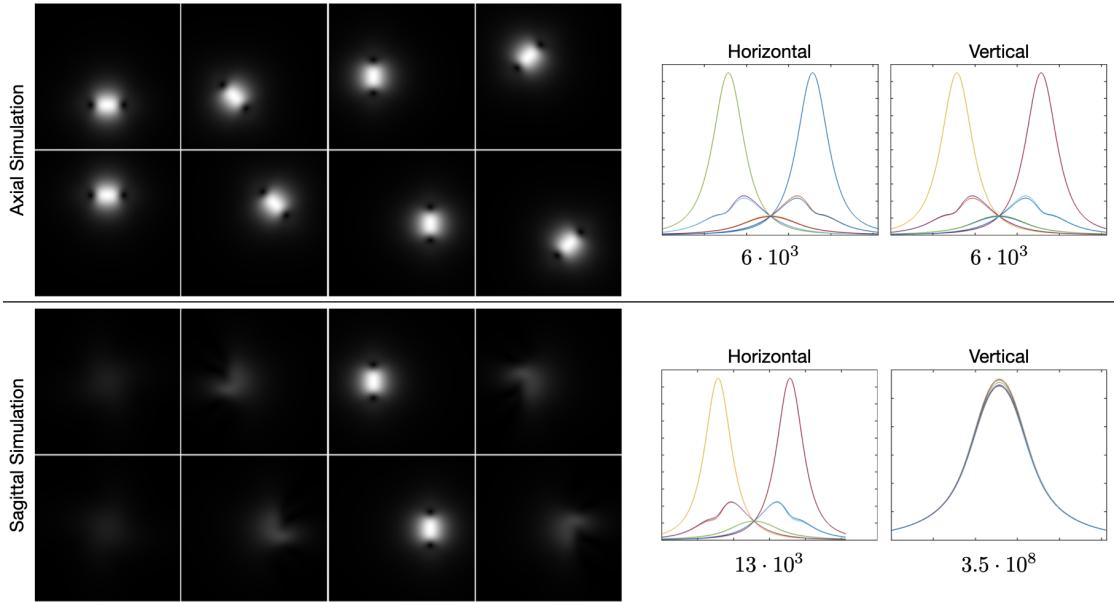


Figure 3.3: Biot-Savart simulations of an 8-element birdcage coil oriented in perpendicular directions. (The top/bottom simulation would be the center axial/sagittal slice for a brain coil, respectively). The sensitivity maps of each coil are shown on the left. The plots on the right show the sensitivities of each coil for a single horizontal/vertical line through the center of the sensitivity maps. The numbers below each plot show the condition number of a matrix created by concatenating the sensitivities in the plots above.



Figure 3.4: Reconstructions of sagittal slices of a knee with different sampling patterns. The columns from left to right are a representation of the sampling indicating the undersampling direction, the GRAPPA reconstruction, the SPIRiT reconstruction, and the g-factor of the SPIRiT reconstructions. The top and bottom rows show undersampling in the horizontal and vertical directions, respectively. Both sampling masks used a reduction factor of 2. All reconstructions used a  $31 \times 31$  ACR and a  $3 \times 3$  kernel.

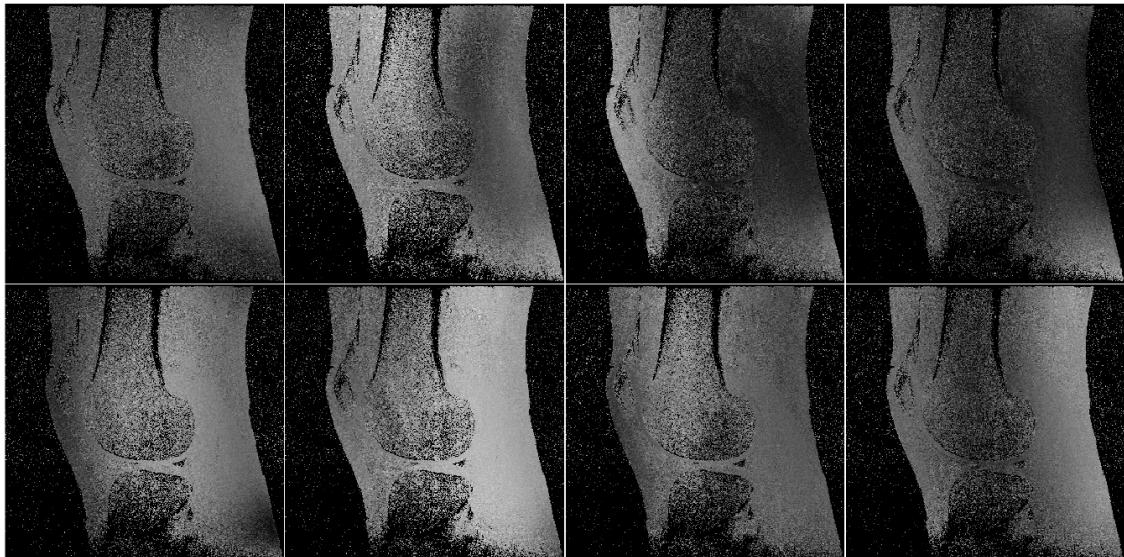


Figure 3.5: Approximations of the sensitivity maps for each coil in an 8-coil birdcage for the data of Fig. 3.4. Note that there are only estimates of the sensitivity in pixels where the magnitude of the corresponding image is sufficiently high for an accurate estimate.

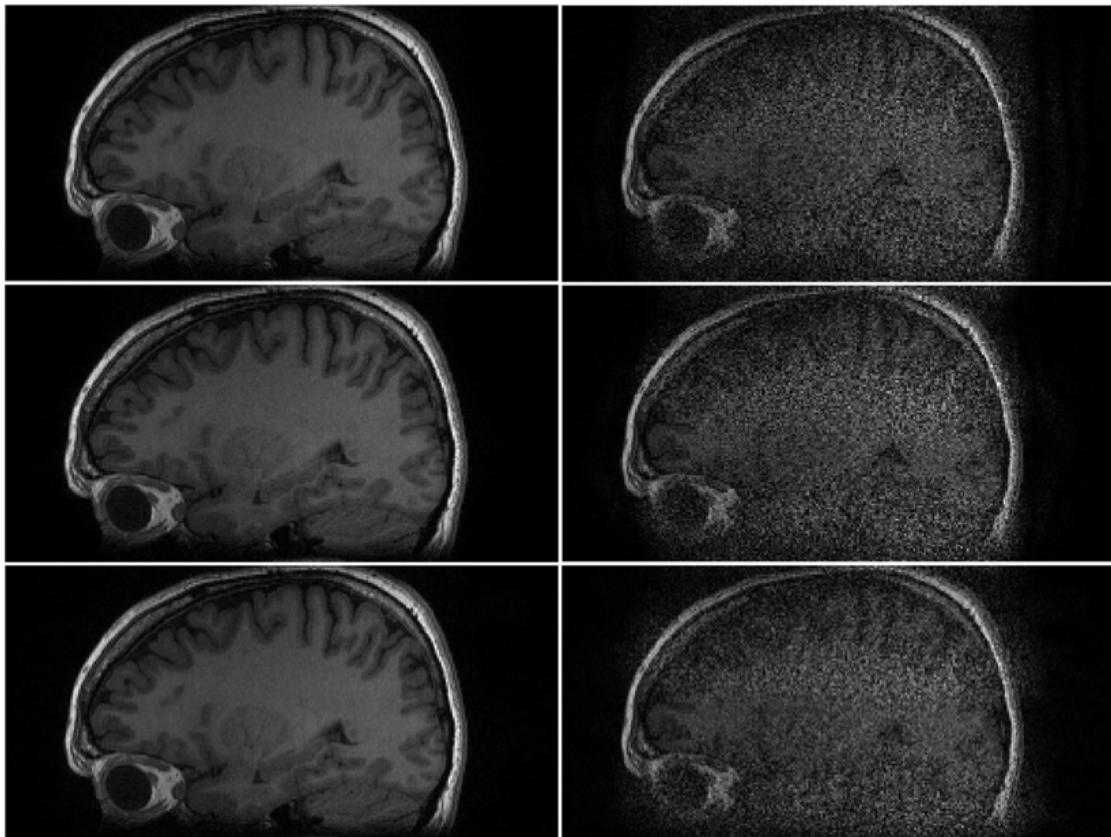


Figure 3.6: SPIRiT reconstructions of a sagittal slice of a brain from data collected with an 8-channel birdcage coil using a horizontal (left) and vertical (right) undersampling mask. Top row:  $3 \times 3$  kernel. Middle row:  $5 \times 5$  kernel. Bottom row:  $7 \times 7$  kernel. All reconstructions with an undersampling factor of two and a  $31 \times 31$  ACR.

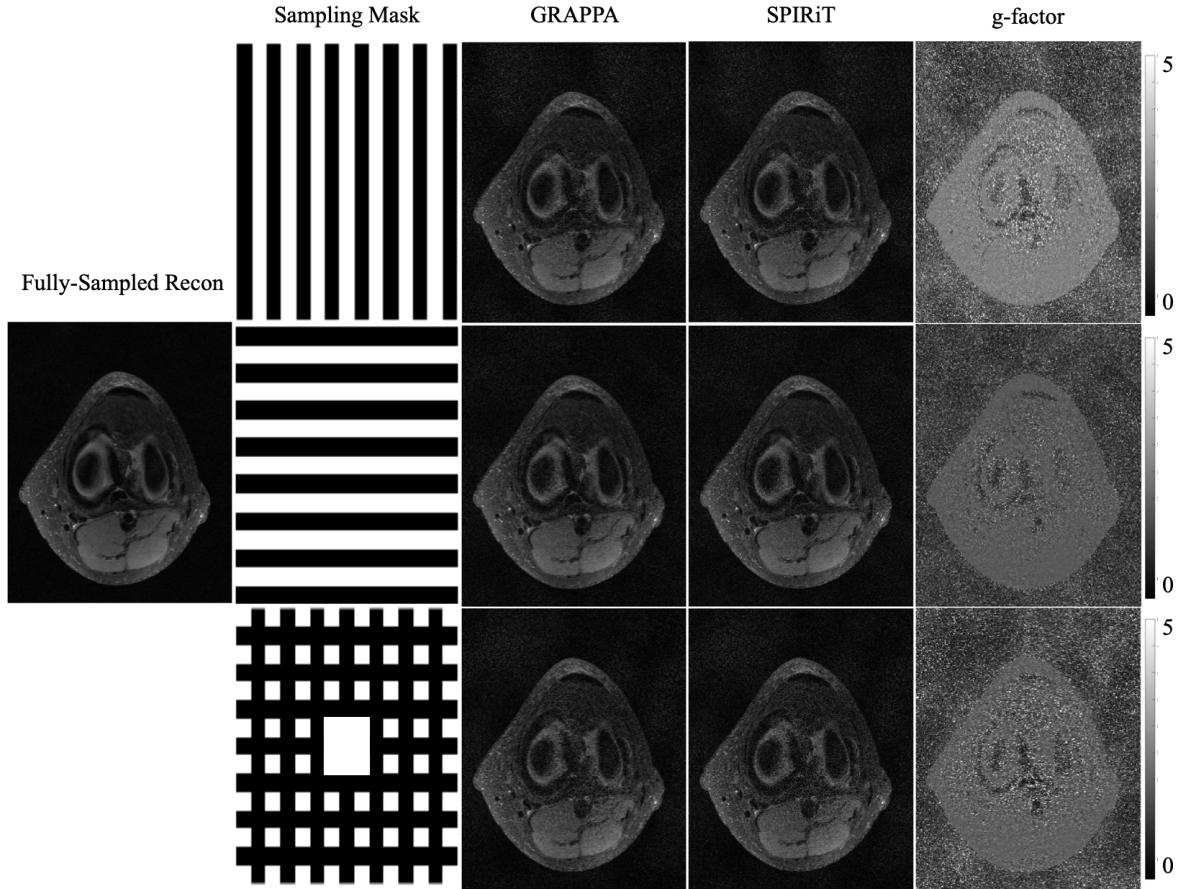


Figure 3.7: Axial slices of the knee. The columns, from left to right, are the reconstruction with fully sampled data, representations of the sampling masks used, the GRAPPA reconstructions, and the g-factors of the SPIRiT reconstructions. All images are size  $320 \times 256$ . The top and middle rows have undersampling factors of two in a single direction (horizontal and vertical, respectively); the bottom row has a reduction factor of approximately two in both directions (for a total reduction factor of approximately four). All reconstructions from GRAPPA and SPIRiT were made with a  $31 \times 31$  ACR and a  $7 \times 7$  kernel.

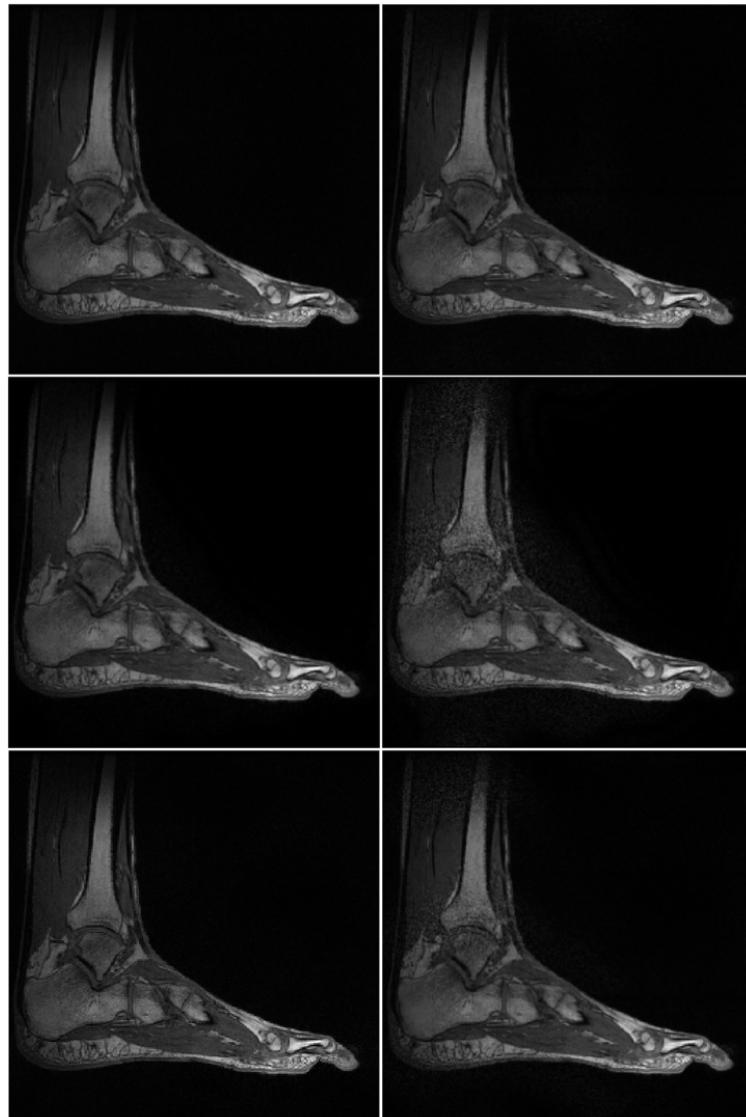


Figure 3.8: SPIRiT reconstructions of retrospectively downsampled ankle data. Left: horizontal undersampling pattern. Right: vertical undersampling pattern. Top row: reduction factor of 2,  $3 \times 3$  kernel. Middle row: reduction factor of 3,  $3 \times 3$  kernel. Bottom row: reduction factor of 3,  $5 \times 5$  kernel. All reconstructions with a  $31 \times 31$  ACR.

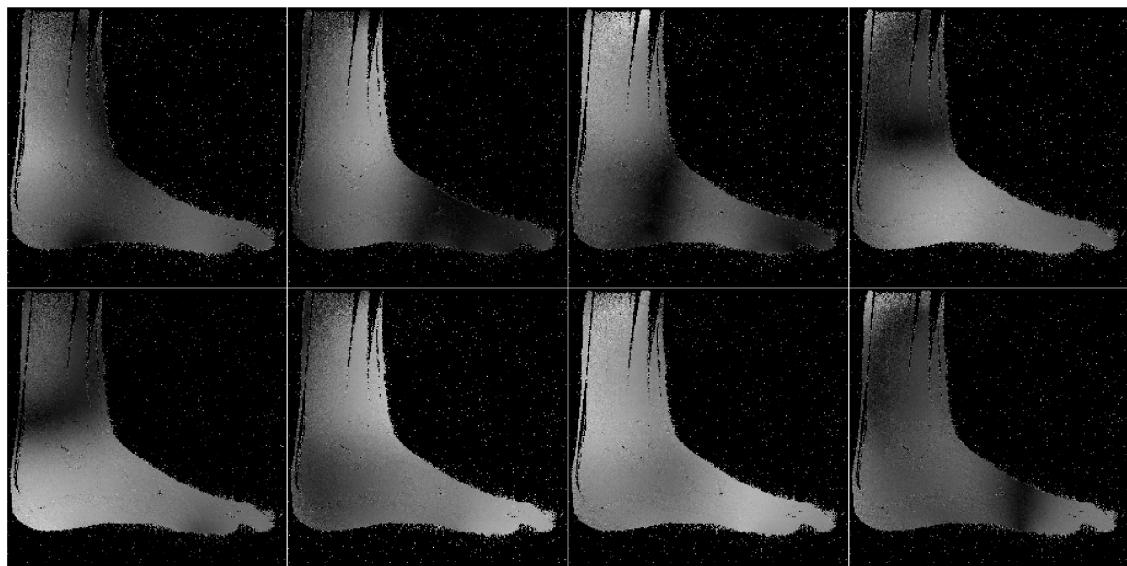


Figure 3.9: Approximations to the individual sensitivity maps for the data of Fig. 3.8. Note that there are only estimates of the sensitivity in pixels where the magnitude of the corresponding image is sufficiently high for an accurate estimate.

## Chapter 4

### A Relaxed Primal-Dual Hybrid Gradient Method with Line Search

#### 4.1 Introduction

Convex optimization problems of functions that are not smooth are common in image reconstruction. Proximal algorithms are often used to solve these problems because they are capable of processing optimization variables with many components. This paper is focused on the primal-dual hybrid gradient method (PDHG) by Chambolle and Pock, presented in [27], which is a first-order proximal algorithm to solve non-smooth problems of the form

$$x^* \in \operatorname{argmin}_{x \in \mathcal{H}} \{f(x) + g(Ax)\},$$

where  $\mathcal{H}$  is a Hilbert space,  $A$  is a linear operator, and  $f$  and  $g$  are closed convex proper functions.

PDHG can efficiently solve problems that are more general than those that can be solved with other proximal splitting methods (e.g., Douglas-Rachford) because it only requires the proximal operator of  $g$ , rather than the proximal operator of  $g \circ A$  which generally only has a closed form solution when  $AA^T$  is a scaled identity.

A downside of PDHG is that, when relaxed, there are three parameters the user needs to set: two step sizes  $\tau$  and  $\sigma$ , and a relaxation parameter  $\theta$ . Convergence is guaranteed when  $\tau\sigma\|A\|^2 \leq 1$ . Therefore, if the spectral norm of  $A$  is known or can be estimated, the number of parameters that must be chosen by the user can be reduced to two parameters. The rate of convergence is highly dependent on the parameter values which makes a line search that eliminates user-defined parameters desirable.

Toward that goal, this manuscript presents a combination of two line search techniques over two sets of parameters. Malitsky et al. introduced a line search over the proximal step sizes in [81]. By using the connection between PDHG and the primal-dual formulation of the Douglas-Rachford splitting method (PDDR) [92], we incorporate the theory of *averaged operator iterations* (AOI). AOI admits a line search over the relaxation parameter [53]. The remaining user-set parameter is the ratio between the two scalings for the proximal steps in PDHG. In general, this can be left as unity, so there are no user-defined parameters remaining.

This paper is organized as follows. Section 4.2 contains notation and relevant definitions and facts. Section 4.3 contains a review of relevant algorithms including the two line search algorithms. Section 4.4 is about the combined line search; its subsection 4.4.1 details convergence thereof. Section 4.5 shows numerical results on several example problems. Section 4.6 details an application to magnetic resonance imaging (MRI) reconstruction, a novel combination of two existing MRI reconstruction methods—compressed sensing and partial Fourier reconstruction with homodyne detection. The formulation of the optimization problem for this type of MRI reconstruction is inconvenient to solve with the standard algorithms for compressed sensing reconstruction, which motivated this study of the primal-dual hybrid gradient algorithm.

## 4.2 Preliminaries

### 4.2.1 Notation and Definitions

We follow standard optimization literature notation conventions; see [9] for further background. Let  $\mathcal{H}$  be a real Hilbert space equipped with inner product  $\langle x, y \rangle$  and norm  $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$ . Let  $\Gamma_0(\mathcal{H})$  be the set of proper, lower semicontinuous convex functions from  $\mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ . Take  $f \in \Gamma_0(\mathcal{H})$ . The domain of  $f$  is  $\text{dom}(f) = \{x \in \mathcal{H} \mid f(x) < +\infty\}$ . Define as  $2^{\mathcal{H}}$  the *power set* of  $\mathcal{H}$ , or the set of all subsets of the Hilbert space. Let  $\mathcal{I}$  denote the identity operator.

The *Legendre-Fenchel conjugate* of function  $f$  is  $f^* \in \Gamma_0(\mathcal{H})$ , given by

$$f^*(x) = \sup_{y \in \mathcal{H}} \{ \langle x, y \rangle - f(y) \}.$$

Note that if  $f \in \Gamma_0(\mathcal{H})$ , then  $f^* \in \Gamma_0(\mathcal{H})$  and  $f^{**} = f$  [9]. The *proximal operator* of  $f$  is defined as:

$$\text{prox}_{\lambda f}(x) = \underset{u \in \mathcal{H}}{\operatorname{argmin}} \left\{ f(u) + \frac{1}{2\lambda} \|u - x\|^2 \right\},$$

for any  $\lambda > 0$ . A *set-valued* operator  $A : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  assigns to each  $x \in \mathcal{H}$  a subset of  $\mathcal{H}$  (or an element in  $2^{\mathcal{H}}$ ). The image of  $x$  under  $A$  is denoted  $A(x)$ . If  $A(x)$  is a singleton, we write  $A(x) = y$  instead of  $A(x) = \{y\}$ . The *graph* of operator  $A$  is denoted  $\text{gr}(A)$  and defined as

$$\text{gr}(A) = \{(x, y) \mid y \in A(x)\}.$$

The inverse of operator  $A$ , denoted  $A^{-1}$ , is defined with graph

$$\text{gr}(A^{-1}) = \{(y, x) \mid (x, y) \in \text{gr}(A)\}.$$

For operator  $A$ , define left- and right-scalar multiplication as

$$(\lambda A)(x) = \{\lambda y : y \in A(x)\}, \quad (A\mu)(x) = \{y : y \in A(\mu x)\}.$$

The following relationships hold between the graph of  $A$  and the graphs of its inverse and scalar multiples:

$$\text{gr}(A^{-1}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{gr}(A), \quad \text{gr}(\lambda A \mu) = \begin{bmatrix} \mu^{-1} \mathcal{I} & 0 \\ 0 & \lambda \mathcal{I} \end{bmatrix} \text{gr}(A). \quad (4.1)$$

Operator  $A$  is *monotone* if

$$\langle x - x', y - y' \rangle \geq 0, \quad \forall (x, y), (x', y') \in \text{gr}(A).$$

An operator  $A$  is *maximally monotone* if it is both monotone and there are no monotone operators  $B$  such that  $\text{gr}(A) \subset \text{gr}(B)$ .

Let  $\mathcal{D}$  be a nonempty subset of  $\mathcal{H}$  and let  $A : \mathcal{D} \rightarrow \mathcal{H}$ .  $A$  is *nonexpansive* if it is Lipschitz continuous with constant 1, or

$$\|Ax - Ay\| \leq \|x - y\|, \quad \forall x \in \mathcal{D}, \forall y \in \mathcal{D}.$$

$A$  is *firmly nonexpansive* if

$$\|Ax - Ay\|^2 + \|(\mathcal{I} - A)x - (\mathcal{I} - A)y\|^2 \leq \|x - y\|^2, \quad \forall x \in \mathcal{D}, \forall y \in \mathcal{D}.$$

We say operator  $A$  is *averaged* with constant  $\alpha$  if there exists a nonexpansive operator  $R : \mathcal{D} \rightarrow \mathcal{H}$  such that  $A = (1 - \alpha)\mathcal{I} + \alpha R$ .  $A$  is firmly nonexpansive if and only if it is  $1/2$ -averaged [9, Remark 4.34].

For a set-valued operator  $A$ , define its *resolvent* as:

$$J_{\lambda A} = (\mathcal{I} + \lambda A)^{-1},$$

where  $\mathcal{I}$  is the identity. If  $A$  is a monotone operator, its resolvent is firmly nonexpansive. Define the *reflected resolvent* of  $A$  as:

$$R_{\lambda A} = 2J_{\lambda A} - \mathcal{I}.$$

Importantly, if  $A$  is maximally monotone then its reflected resolvent is nonexpansive [9, Corollary 23.11]. The resolvent and reflected resolvents of  $A$  satisfy the following identities:

$$J_A(x) + J_{A^{-1}}(x) = x, \quad \text{and} \quad R_A(x) + R_{A^{-1}}(x) = 0. \quad (4.2)$$

Since  $(\gamma A)^{-1} = A^{-1}\gamma^{-1}$  for scalar  $\gamma$  (which can be shown directly from the relationships in (4.1)),

$$J_{\gamma A}(x) + \gamma J_{\gamma^{-1}A^{-1}}(x/\gamma) = x. \quad (4.3)$$

Let  $A : \mathcal{H} \rightarrow \mathcal{H}$  be such that  $AA^* = \alpha\mathcal{I}$ , where  $\cdot^*$  denotes the operator's adjoint. Let  $f \in \Gamma_0(\mathcal{H})$  and let  $g(x) = f(Ax)$ . Then

$$\text{prox}_{\gamma g}(x) = x + \frac{1}{\alpha}A^* \left( \text{prox}_{\alpha\gamma f}(Ax) - Ax \right).$$

The *subdifferential* of  $f \in \Gamma_0(\mathcal{H})$  is a set-valued operator  $\partial f$  defined as:

$$\partial f(x) = \{\zeta \in \mathcal{H} \mid f(y) \geq f(x) + \langle \zeta, y - x \rangle \forall y \in \mathcal{H}\}.$$

By the definition of the subdifferential, we have  $x^* \in \underset{x}{\operatorname{argmin}} f(x)$  if and only if  $0 \in \partial f(x^*)$ —this is known as *Fermat's law* [9, Theorem 16.3]. Therefore, for suitable functions  $f \in \Gamma_0(\mathcal{H})$ , solving a minimization problem is the same as solving a *monotone inclusion problem*.

Maximal monotone operators are useful for convex optimization because the subdifferential of  $f \in \Gamma_0(\mathcal{H})$  is maximally monotone. Furthermore, the inverse of the subdifferential is the subdifferential of the conjugate function:

$$(\partial f)^{-1} = \partial f^*.$$

An important fact for the convergence of proximal algorithms is that the proximal operator of a proper function  $f \in \Gamma_0(\mathcal{H})$  is the resolvent of the subdifferential:

$$\text{prox}_{\gamma f}(x) = (\mathcal{I} + \gamma \partial f)^{-1} x. \quad (4.4)$$

The subdifferential of  $f \in \Gamma_0(\mathcal{H})$  is maximally monotone, so (4.4) is single valued. Letting  $A = \partial f$  for  $f \in \Gamma_0(\mathcal{H})$  in (4.3) yields the **Moreau identity**:

$$\text{prox}_{\gamma f}(x) + \gamma \text{prox}_{\gamma^{-1} f^*}(x/\gamma) = x.$$

#### 4.2.2 The Problem

For the remainder of this manuscript, let  $\mathcal{X}$  and  $\mathcal{Y}$  be real Hilbert spaces. We are interested in solving the following (*primal*) optimization problem:

$$x^* \in \operatorname{argmin}_{x \in \mathcal{X}} f(x) + g(Ax), \quad (4.5)$$

where  $f \in \Gamma_0(\mathcal{X})$  and  $g \in \Gamma_0(\mathcal{Y})$  have computationally tractable forms of their respective proximal operators and  $A : \mathcal{X} \rightarrow \mathcal{Y}$  is a linear operator with known adjoint  $A^*$  and finite operator norm:

$$\|A\| = \sup \{\|Ax\| : x \in \mathcal{X} \text{ with } \|x\| \leq 1\} < +\infty.$$

The corresponding dual problem to (4.5) is:

$$y^* \in \operatorname{argmax}_{y \in \mathcal{Y}} - (f^*(-A^*y) + g^*(y)). \quad (4.6)$$

The algorithm that is the focus of this manuscript is *primal-dual*, meaning it is applied to the saddle point interpretation of the primal (4.5) and dual (4.6) problems:

$$(\hat{x}, \hat{y}) \in \operatorname{argmin}_{x \in \mathcal{X}} \operatorname{argmax}_{y \in \mathcal{Y}} \{ \langle Ax, y \rangle + f(x) - g^*(y) \}. \quad (4.7)$$

More details can be found in, e.g., [27, 32, 9].

The saddle point formulation yields the following optimality conditions, which are the starting point for development of primal-dual algorithms to solve the problem:

$$(\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y} \text{ such that } \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} 0 & A^* \\ -A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \partial f(x) \\ \partial g^*(y) \end{pmatrix}. \quad (4.8)$$

The Karush-Kuhn-Tucker conditions state that if  $(\hat{x}, \hat{y})$  is a solution to (4.8), then  $\hat{x}$  is a solution to (4.5) and  $\hat{y}$  is a solution to (4.6) [9, Theorem 19.1]. Furthermore,  $(\hat{x}, \hat{y})$  is a solution to (4.7) [9, Proposition 19.20]. Note that the converse is not true in general and the set of solutions to (4.8) may be empty. Under certain constraint qualifications such as

$$0 \in \text{sri}(\text{dom}(g) - A(\text{dom}(f))), \quad (4.9)$$

then the duality gap is zero (i.e. strong duality holds) and the set of solutions to (4.6) is nonempty [9, Theorem 15.23]. Here  $\text{sri}(C)$  is the *strong relative interior* of a set  $C$  (see [9, Section 6.2] for more details). In many applications of interest,  $\mathcal{X}$  and  $\mathcal{Y}$  are finite-dimensional. The condition (4.9) is then satisfied if

$$(\text{ri dom}(g)) \cap A(\text{ri dom}(f)) \neq \emptyset, \quad (4.10)$$

where  $\text{ri}$  is the *relative interior* of a set [9, Proposition 15.24]. The condition (4.10) is easy to show if either  $f$  or  $g$  is full domain. Under this condition, whenever  $\hat{x}$  is a solution to (4.5) and  $\hat{y}$  is a solution to (4.6), then  $(\hat{x}, \hat{y})$  is a solution to (4.8).

### 4.3 Algorithms

#### 4.3.1 Douglas-Rachford Splitting

The standard form of Douglas-Rachford splitting (DR) was originally given by Lions and Mercier [72] to find zeros of a sum of maximally monotone operators  $F$  and  $G$ ; i.e., to find  $x$  such

that

$$0 \in F(x) + G(x).$$

The standard version of the algorithm is the fixed point iteration

$$y_{k+1} = \frac{1}{2}y_k + \frac{1}{2}R_{\gamma G}R_{\gamma F}y_k, \quad (4.11)$$

where  $\gamma$  is a positive constant.

This is commonly presented as the following steps:

$$\begin{aligned} x_{k+1} &= J_{\gamma F}(y_k) \\ w_{k+1} &= J_{\gamma G}(2x_{k+1} - y_k) \\ y_{k+1} &= y_k + w_{k+1} - x_{k+1}. \end{aligned}$$

If we allow for a relaxation parameter  $\alpha_k \in (0, 1)$ , then (4.11) is altered as follows:

$$y_{k+1} = (1 - \alpha_k)y_k + \alpha_k R_{\gamma G}R_{\gamma F}y_k. \quad (4.12)$$

Equivalently,

$$\begin{aligned} x_{k+1} &= J_{\gamma F}(y_k) \\ w_{k+1} &= J_{\gamma G}(2x_{k+1} - y_k) \\ y_{k+1} &= y_k + \rho_k(w_{k+1} - x_{k+1}), \end{aligned} \quad (4.13)$$

where  $\rho_k = 2\alpha_k$ .

Results on the convergence of DR are given in, e.g., [9, 72, 10, 43]. The standard assumptions for convergence are the maximal monotonicity of  $F$  and  $G$  and the existence of a solution. Using relationship (4.2) on the second step of the iteration (4.13) yields the iterations for *primal dual Douglas-Rachford* (PDDR):

$$\begin{aligned} x_{k+1} &= J_{\gamma F}(y_k) \\ w_{k+1} &= J_{(\gamma G)^{-1}}(2x_{k+1} - y_k) \\ y_{k+1} &= (1 - \rho_k)y_k + \rho_k(x_{k+1} - w_{k+1}). \end{aligned} \quad (4.14)$$

By using relationship (4.2) applied to the fixed point iteration version of DR in (4.12), we get the fixed point iteration form of PDDR:

$$y_{k+1} = (1 - \alpha_k)y_k - \alpha_k R_{(\gamma G)^{-1}} R_{\gamma F} y_k. \quad (4.15)$$

#### 4.3.2 Primal-Dual Hybrid Gradient

The Primal-Dual Hybrid Gradient (PDHG) method that solves problems of the form of (4.8) was originally proposed by Pock et al. in [98]. Convergence of the algorithm was formalized by Chambolle and Pock in [27], so it is alternatively called the *primal-dual hybrid gradient* method (PDHG) or the *Chambolle-Pock* algorithm. The method has the following iterations:

$$\begin{aligned} \bar{x}_{k+1} &= J_{\tau_k F}(x_k - \tau_k A^* z_k) \\ \bar{z}_{k+1} &= J_{\sigma_k G^{-1}}(z_k + \sigma_k A(2\bar{x}_{k+1} - x_k)) \\ x_{k+1} &= x_k + \alpha_k(\bar{x}_{k+1} - x_k) \\ z_{k+1} &= z_k + \alpha_k(\bar{z}_{k+1} - z_k), \end{aligned} \quad (4.16)$$

where  $F = \partial f$ ,  $G^{-1} = \partial g^*$  for proper, lower semicontinuous convex functions  $f \in \Gamma_0(\mathcal{X})$  and  $g \in \Gamma_0(\mathcal{Y})$ , and  $A : \mathcal{X} \rightarrow \mathcal{Y}$  is a bounded linear operator with known adjoint. Since  $g \in \Gamma_0(\mathcal{Y})$ ,  $g^* \in \Gamma_0(\mathcal{Y})$  as well. The resolvents  $J_F$  and  $J_{G^{-1}}$  are proximal operators of the functions  $f$  and  $g$ , as in (4.4).

As written, the algorithm has three parameters for each iteration:  $\tau$  and  $\sigma$ , which must satisfy  $\tau\sigma\|A\|^2 \leq 1$  to guarantee convergence [32], and  $\alpha$ , which is a *relaxation parameter* and typically takes values in  $(0, 2)$  [9]. Commonly, parameter values are held constant for all iterations. I.e.,  $\tau_k = \tau$ ,  $\sigma_k = \sigma$ , and  $\alpha_k = \alpha$  for all  $k$ .

A line search for PDHG was introduced in [81] by Malitsky et al. The pseudocode for the line search is show in Alg. 1. Line searches for primal-dual methods are difficult in general. The conditions for a standard line search, e.g., Armijo backtracking line search [5], cannot always be evaluated (e.g., when primal-dual methods are applied to problems where  $f$  or  $g$  are an indicator function).

**Algorithm 1** *PDHG Line Search* [81]

---

**Initialization:** Given  $x_{k-1}, z_{k-1} \in \mathcal{H}$ ,  $\text{prox}_f$ ,  $\text{prox}_{g^*}$ ,  $\tau_{k-1} > 0$ ,  $\theta_{k-1} > 0$ ,  $\mu \in (0, 1)$ ,  $\beta > 0$ ,  $\delta \in (0, 1)$ .

**Main iteration:**

1. Compute

$$x_k = \text{prox}_{\tau_{k-1} f}(x_{k-1} - \tau_{k-1} A^* z_{k-1})$$

2. Choose  $\tau_k \in [\tau_{k-1}, \tau_{k-1}\sqrt{1 + \theta_{k-1}}]$  and run {By default we choose  $\tau_k = \tau_{k-1}\sqrt{1 + \theta_{k-1}}$ }

**Linesearch:**

- 2.a. Compute

$$\theta_k = \frac{\tau_k}{\tau_{k-1}}$$

$$\bar{x}_k = x_k + \theta_k(x_k - x_{k-1})$$

$$z_k = \text{prox}_{\beta \tau_k g^*}(z_{k-1} + \beta \tau_k A \bar{x}_k)$$

- 2.b. Break linesearch if

$$\sqrt{\beta} \tau_{k+1} \|A^* z_k - A^* z_{k-1}\| \leq \delta \|z_k - z_{k-1}\|$$

and return

$$z_k, \bar{x}_k$$

Otherwise, set  $\tau_k \leftarrow \mu \tau_k$  and go to 2.a.

**End of linesearch**

---

### 4.3.3 The Relationship Between PDDR and PDHG

It has been shown in [92] that the PDDR iterations (4.14) applied to the following *modified problem* of (4.17) yield the same iterations as PDHG applied to the original problem (4.5):

$$(\hat{x}, \hat{y}) \in \underset{x \in \mathcal{X}, y \in \mathcal{Y}}{\operatorname{argmin}} \tilde{f}(x, y) + \tilde{g}(x, y) \quad (4.17)$$

with modified functions

$$\tilde{f}(x, y) = f(x) + \delta_{\{0\}}(y), \quad \tilde{g}(x, y) = g(Ax + By),$$

where  $\delta_{\{0\}}$  is an indicator function specifying that the argument is 0 and  $B$  is chosen to satisfy  $AA^* + BB^* = \frac{1}{\theta} \mathcal{I}$  for some  $\theta > 0$ . (Note that an indicator takes on a value of 0 when its argument satisfies its condition; otherwise, it takes on a value of  $\infty$ .) As an example, if  $A$  is an  $n \times n$  matrix with  $\frac{1}{\theta} \geq \|A\|_2^2$ , let  $B = \left(\frac{1}{\theta} \mathcal{I} - AA^*\right)^{1/2}$  where  $(\cdot)^{1/2}$  is the Cholesky factorization. Defining these

modified functions and applying the iterations of PDDR (4.14) to problem (4.17) yields the PDHG iterations of (4.16) [92]. In [92], the choice of  $\theta$  is left to the user. The relationship between  $\theta$  and the  $\tau, \sigma$  parameters from PDHG is  $\tau\sigma = \theta$  [92].

As can be seen in [92], the relationship between the variables of PDDR in (4.14) when applied to (4.17) and PDHG in (4.16) when applied to (4.5) is

$$y_k = (x_k - \tau_k A^* z_k, -\tau_k B^* z_k). \quad (4.18)$$

This relationship is what allows us to tie the two existing line searches together. An iteration of PDHG is the same as one iteration of PDDR with the relationship above.

#### 4.3.4 The Averaged Operator Iteration Line Search

An *averaged operator iteration* (AOI) for a nonexpansive operator  $S$  is an iteration of the form

$$y_{k+1} = (1 - \alpha)y_k + \alpha S y_k = y_k + \alpha(Sy_k - y_k). \quad (4.19)$$

All of the algorithms discussed in this section so far can be written as an AOI. The AOI form of Douglas-Rachford is given in (4.11) where  $S = R_{\gamma G} R_{\gamma F}$ . The AOI form of PDDR is given in (4.15) with  $S = -R_{(\gamma G)^{-1}} R_{\gamma F}$ . In both cases,  $S$  is nonexpansive as it is a composition of reflected resolvents, which are themselves nonexpansive [9, Proposition 4.4].

A line search for averaged operator iterations was introduced by Gisselson et al. in [53]. The pseudocode for the algorithm is presented in Alg. 2.

AOI with line search requires the user to set a nominal step size  $\bar{\alpha}$ . This can be set to  $\frac{1}{2}$  to ensure the *nominal* step corresponds to a fixed-point iteration of a firmly nonexpansive operator. In order, the steps of Alg. 2 compute the current residual, compute the *nominal* next step with the nominal step size, and then the residual with the nominal step size. The line search then iterates over the values of  $\alpha_k$  in the final step. Either the step size is  $\alpha_k = \bar{\alpha}$ , the nominal step, or  $\alpha_k \in (\bar{\alpha}, \alpha_{\max}]$  is chosen such that

$$\|r_{k+1}\|_2 = \|Sy_{k+1} - y_{k+1}\|_2 \leq (1 - \varepsilon)\|\bar{r}_k\|_2, \quad (4.20)$$

---

**Algorithm 2** AOI Line Search [53]

---

**Initialization:** Choose  $y_0 \in \mathcal{H}$ ,  $\alpha_{\max} > 0$ ,  $\bar{\alpha} \in (0, \alpha_{\max})$ ,  $\varepsilon > 0$ ,  $\mu \in (0, 1)$ .

**Main iteration:**

1. Compute

$$\begin{aligned} r_k &= Sy_k - y_k \\ \bar{y}_k &= y_k + \bar{\alpha}r_k \\ \bar{r}_k &= S\bar{y}_k - \bar{y}_k \end{aligned}$$

2. Set  $\alpha_k = \alpha_{\max}$  and run

**Linesearch:**

- 2.a. Compute

$$\begin{aligned} y_{k+1} &= y_k + \alpha_k r_k \\ r_{k+1} &= Sy_{k+1} - y_{k+1} \end{aligned}$$

- 2.b. Break linesearch if

$$\|r_{k+1}\| = \|Sy_{k+1} - y_{k+1}\| \leq (1 - \varepsilon)\|\bar{r}_k\|$$

Otherwise, set  $\alpha_k \leftarrow \mu\alpha_k$  and go to 2.a. If  $\alpha_k = \bar{\alpha}$ , terminate the line search.

---

**End of linesearch**

---

where  $\varepsilon > 0$  and  $\alpha_{\max} > \bar{\alpha}$  are fixed algorithm parameters.

The AOI iterations of (4.19) can be interpreted as steps of length  $\alpha$  in the direction of the residual  $Sx_k - x_k$ . Taking a large step in this direction may be beneficial and that is what this line search aims to accomplish. In practice,  $\alpha_{\max}$  is set much larger than  $\bar{\alpha}$  and progressively shrunk to test if a large step in the residual direction reduces the norm of the residual more than the nominal step.

#### 4.4 The Line Search

The connection between PDDR as an AOI has been shown and the relationship between the variables of PDDR and PDHG are presented in (4.18). Thus the strategy of the proposed line search:

- Use the AOI line search to find an advantageous relaxation parameter, where for each proposed relaxation parameter we

- convert the variables to those of PDHG and use the line search of Malitsky et al. to find proximal step sizes  $\tau$  and  $\sigma$ .

We refer to these steps, respectively, as the “outer” and “inner” line searches, as the AOI line search calls the PDHG line search at each step. We propose a *relaxed primal-dual hybrid gradient method with line search*, or rPDHG. The line search executes the Malitsky line search for PDHG [81] and uses the relationship between PDHG and the primal-dual form of Douglas-Rachford to create a line search over the relaxation parameter. Pseudocode for the Outer Line Search is shown in Alg. 3, and pseudocode for the Inner Line Search (which is equivalent to the Maltisky PDHG line search) is shown in Alg. 4.

The proposed line search inherits some parameters from the AOI and PDHG line searches. The  $\beta$  parameter is the ratio between the two proximal step sizes  $\tau_k$  and  $\sigma_k$ , which we leave as unity in experiments but include in the algorithm for completeness.

#### 4.4.1 Convergence

Convergence of the outer (AOI) line search is detailed in the original paper [53]. To briefly summarize, let  $k_0$  denote the number of times the outer line search activates (that is, chooses  $\alpha_k > \bar{\alpha}$ ). If  $k_0$  is finite, let  $k_{\max}$  denote the final iteration on which  $\alpha_k > \bar{\alpha}$ . For all  $k > k_{\max}$ , the iteration is

$$y_{k+1} = y_k + \bar{\alpha} (S_k y_k - y_k).$$

For  $\bar{\alpha} \in (0, 1)$  this corresponds exactly to an iteration of PDHG with Malitsky’s line search. As shown in [81, Lemma 1], this line search always terminates. Convergence is equivalent to convergence of PDHG with line search.

For each iteration  $k$  that  $\alpha_k > \bar{\alpha}$  the line search condition requires that (for  $\varepsilon \in (0, 1)$ )

$$\|r_{k+1}\| = \|S_k y_k - y_k\| \leq (1 - \varepsilon) \|\bar{r}_k\| \leq (1 - \varepsilon) \|r_k\|,$$

which yields

$$\|r_{k+1}\| \leq (1 - \varepsilon)^{k_0} \|r_0\|.$$

**Algorithm 3** Relaxed PDHG with Line Search - Outer Line Search

**Initialization:** Choose  $x_0, z_0 \in \mathcal{H}$ ,  $\alpha_{\max} > 0$ ,  $\bar{\alpha} \in (0, \alpha_{\max})$ ,  $\varepsilon > 0$ ,  $\mu \in (0, 1)$ ,  $\tau_0 > 0$ ,  $A, B$  such that  $AA^* + BB^* = \frac{1}{\theta}\mathcal{I}$ .

**Main iteration:**

1. Compute

$$[x_{k+1/2}, z_{k+1/2}, \tau_k] = \text{Inner Line Search}(x_{k-1}, z_{k-1}, \tau_{k-1}, \bar{\alpha}).$$

2. Compute

$$\begin{aligned}\bar{x}_{k+1} &= \text{prox}_{\tau_k f} \left( x_{k+1/2} - \tau_k A^* z_{k+1/2} \right), \\ \bar{z}_{k+1} &= \text{prox}_{\beta \tau_k g^*} \left( z_{k+1/2} + \beta \tau_k A (2 * \bar{x}_{k+1} - x_{k+1/2}) \right),\end{aligned}$$

3. Compute

$$\bar{r}_k = \begin{bmatrix} (\bar{x}_k - \tau_k A^* \bar{z}_k) - (x_{k+1/2} - \tau_{k+1/2} A^* z_{k+1/2}) \\ -\tau_k B^* \bar{z}_k - (-\tau_{k+1/2} B^* z_{k+1/2}) \end{bmatrix}.$$

4. Set  $\alpha_k = \alpha_{\max}$  and run

**Linesearch:**

4.a. Compute

$$[x_{k+1/2}, z_{k+1/2}, \tau_k] = \text{Inner Line Search}(x_{k-1}, z_{k-1}, \tau_{k-1}, \alpha_k), \quad (4.21)$$

$$\bar{x}_{k+1} = \text{prox}_{\tau_k f} \left( x_{k+1/2} - \tau_k A^* z_{k+1/2} \right), \quad (4.22)$$

$$\bar{z}_{k+1} = \text{prox}_{\beta \tau_k g^*} \left( z_{k+1/2} + \beta \tau_k A (2 * \bar{x}_{k+1} - x_{k+1/2}) \right), \quad (4.23)$$

$$r_{k+1} = \begin{bmatrix} (\bar{x}_{k+1} - \tau_k A^* \bar{z}_{k+1}) - (x_{k+1/2} - \tau_k A^* z_{k+1/2}) \\ -\tau_k B^* \bar{z}_{k+1} - (-\tau_k B^* z_{k+1/2}) \end{bmatrix}. \quad (4.24)$$

4.b. Break line search if

$$\|r_{k+1}\| \leq (1 - \varepsilon) \|\bar{r}_k\|$$

Otherwise, set  $\alpha_k \leftarrow \mu \alpha_k$  and go to 4.a. If  $\alpha_k = \bar{\alpha}$ , terminate the line search.

**End of line search**

---

If  $k_0 \rightarrow \infty$  as  $k \rightarrow \infty$ , then  $\|r_{k+1}\| \rightarrow 0$ .

#### 4.4.2 Line Search Activation

Conducting the outer line search is computationally costly. If there were an inexpensive test that could be run to determine whether or not line search would be beneficial, then one could activate line search only in these instances to save time. In [53], Gisselson et al. suggest a test

**Algorithm 4** Relaxed PDHG with Line Search - Inner Line Search

**Initialization:** Given  $x_{k-1}, z_{k-1} \in \mathcal{H}$ ,  $\text{prox}_f$ ,  $\text{prox}_{g^*}$ ,  $\alpha_k \in [\bar{\alpha}, \alpha_{\max}]$ ,  $\tau_{k-1} > 0$ ,  $\theta_{k-1} > 0$ ,  $A, B$  such that  $AA^* + BB^* = \frac{1}{\theta}\mathcal{I}$ ,  $\mu \in (0, 1)$ ,  $\beta > 0$ ,  $\delta \in (0, 1)$ .

**Main iteration:**

1. Compute

$$\hat{x}_k = \text{prox}_{\tau_{k-1}f}(x - \tau_{k-1}A^*z_{k-1})$$

2. Choose  $\tau_k \in [\tau_{k-1}, \tau_{k-1}\sqrt{1 + \theta_{k-1}}]$  and run {By default we choose  $\tau_k = \tau_{k-1}\sqrt{1 + \theta_{k-1}}$ }

**Line search:**

- 2.a. Compute

$$\begin{aligned}\theta_k &= \frac{\tau_k}{\tau_{k-1}} \\ \bar{x}_k &= \hat{x}_k + \theta_k(\hat{x}_k - x_{k-1}) \\ \bar{z}_k &= \text{prox}_{\beta\tau_k g^*}(z_{k-1} + \beta\tau_k K\bar{x}_k)\end{aligned}$$

- 2.b. Break line search if

$$\sqrt{\beta}\tau_k \|A^*\bar{z}_k - A^*z_{k-1}\| \leq \delta \|\bar{z}_k - z_{k-1}\|.$$

Otherwise, set  $\tau_k \leftarrow \mu\tau_k$  and go to 2.a.

**End of line search**

3. Compute

$$\begin{aligned}x_k &= (1 - 2\alpha_k)x_{k-1} + 2\alpha_k\bar{x}_k \\ z_k &= (1 - 2\alpha_k)z_{k-1} + 2\alpha_k\bar{z}_k.\end{aligned}$$

---

Return  $x_k, z_k, \tau_k, \theta_k$ .

---

based on the displacement vectors of the trajectory:  $v_k = y_k - y_{k-1}$ . They suggest activating the line search when  $v_k$  is approximately aligned with  $v_{k+1}$ ; i.e., when the following is true

$$\frac{v_k^T v_{k-1}}{\|v_k\|_2 \|v_{k-1}\|_2} > 1 - \hat{\varepsilon}$$

for some small  $\hat{\varepsilon} > 0$ .

We hypothesize that this heuristic was useful because the objective function of the problem in their experiments was a strongly-convex quadratic where the gradient becomes smaller quickly as one approaches the optimal point. The  $\ell_1$  norm used as the objective function in our application does not have this property, and the line search was almost always activated.

Instead, we propose the following: the line search will be activated if either 1) it is the first iteration, 2) line search was activated in the previous iteration and  $\alpha_k$  did not equal  $\bar{\alpha}$ , or 3)  $\|r_k\|_2/\|r_{k-1}\|_2 < 1 - \hat{\varepsilon}$  for some small  $\hat{\varepsilon} > 0$ . For the results presented in this manuscript,  $\hat{\varepsilon} = 0.05$ . In words, the third condition means that line search should be activated when the size of the residual for the current iteration is smaller than that of the previous iteration. We based this on (4.20), where we are using  $r_{k-1}$  as a surrogate for  $\bar{r}_k$ .

#### 4.4.3 Limitations with Matrix Free Implementations

The line search presented may not be suitable for every application. For example, in image processing applications  $A$  may operate on  $n \times n$  images, so  $A : \mathbb{R}^{n^2} \rightarrow \mathbb{R}^{n^2}$ . For modestly sized  $n$ , this can quickly exceed memory limitations for standard computers if  $A$  is stored in dense matrix form. Many of these operators work matrix-free, but finding a suitable  $B$  for the proposed line search may require storing  $B$  as a dense matrix where  $B$  is at least as large as  $A$ . An example of this phenomenon is shown in Section 4.5.3, where the image had to be scaled down so  $B$  could fit in memory.

When  $A$  is matrix free, it may be possible to find a matrix-free implementation of  $B$ ; an example is shown in Section 4.6. Alternatively, one may be able to evaluate  $B$  matrix-free for arbitrary  $A$  by using the Lanczos method [4], but it is unclear whether this would work in practice due to the tradeoff between accuracy and cost.

### 4.5 Numerical Experiments

We present numerical results for the following problems:

- A generalized LASSO-style regularized least squares problem,
- a one-dimensional (signal) total variation denoising problem,
- a two-dimensional (image) total variation denoising problem, and
- a novel magnetic resonance image (MRI) reconstruction problem using compressed sensing.

For the numerical experiments, we compare the proposed line search against:

- Standard PDHG iterations (4.16),
- PDHG with the line search of Malitsky et al. as in Alg. 1,
- PDDR applied to the modified problem using the AOI formulation and line search as in Alg. 2.

Extensive parameter searches were conducted to choose the optimal parameters for these algorithms, used in the comparison to the proposed method.

#### 4.5.1 LASSO

Consider the unconstrained generalized LASSO problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \frac{1}{2} \|x - b\|_2^2 + \lambda \|Ax\|_1. \quad (4.25)$$

Here  $A \in \mathbb{R}^{n \times n}$ ,  $x \in \mathbb{R}^n$ , and  $b \in \mathbb{R}^n$ . We write  $f(x) = \frac{1}{2}\|x - b\|_2^2$  and  $g(x) = \lambda\|x\|_1$ .  $A$  and  $b$  are randomly generated from a standard normal distribution.  $B$  (for the relaxed PDHG with line search) is chosen by setting  $\theta = 0.9/\|A\|_2^2$  and solving for  $B$  from  $BB^* = \frac{1}{\theta}\mathcal{I} - AA^*$  via Cholesky factorization. For the results presented, we set  $n = 1000$ . After searching over parameters, we compare the best convergence rate for each method. This is shown in Fig. 4.1. For rPDHG, however, we set  $\beta = 1$  and choose a  $\tau_0 > 0$ , avoiding a costly parameter search.

#### 4.5.2 1D-TV Denoising

Consider the one-dimensional total variation denoising problem (cf. [25]). We take a signal  $b$  of length  $n$  with additive white noise and solve the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \frac{1}{2} \|x - b\|_2^2 + \lambda \|\nabla x\|_1,$$

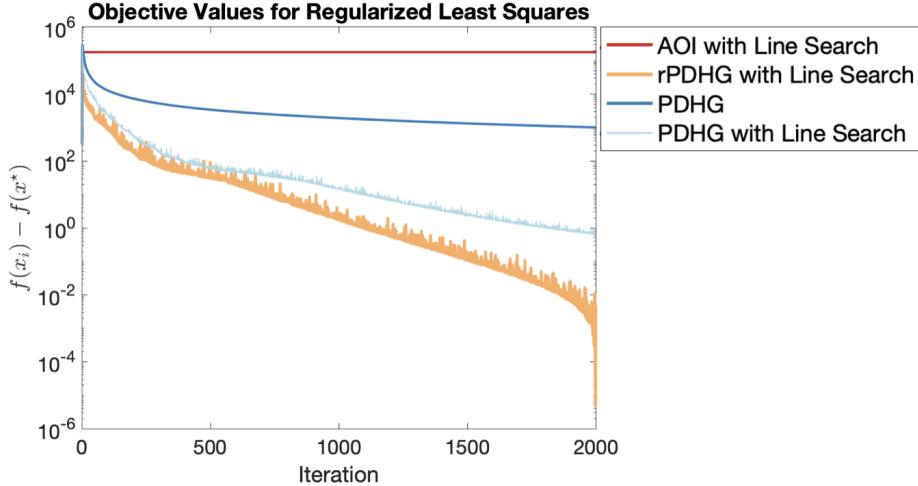


Figure 4.1: A comparison of the objective values from the different optimization algorithms as they solve the regularized least squares problem in (4.25).

where  $\lambda > 0$  is a chosen parameter and  $\nabla : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the one-dimensional finite difference operator:

$$\nabla x_i = \begin{cases} x_1 - x_n, & i = 1 \\ x_i - x_{i-1}, & i = 2, \dots, n. \end{cases}$$

The signal is created by specifying a number of segments  $num_{\text{segs}}$  and length of segments  $len_{\text{segs}}$  and letting  $n = num_{\text{segs}} * len_{\text{segs}}$ . Then  $b$  is created by taking  $num_{\text{segs}}$  random integers  $z \in [-5, 5]$  and creating a step function from  $len_{\text{segs}}$ . White noise is then added to this to create the measurements  $b$ . For this experiment, we chose  $\lambda = 1$ .

Here  $f(x) = \frac{1}{2}\|x - b\|_2^2$  and  $g(x) = \lambda\|x\|_1$ . The  $A$  operator for the algorithm is the finite difference operator  $\nabla$ . Creating  $B$  for the line search requires first creating a matrix representation of  $A$  and then creating  $B$  by choosing  $\theta > \frac{1}{\|A\|^2}$  and solving for  $B$  from

$$BB^* = \frac{1}{\theta}\mathcal{I} - AA^*,$$

via Cholesky factorization. Figure 4.2 shows the ground truth and the noisy function. Figure 4.3 shows the objective values for each iteration of the four different algorithms along with the residual of the final answer produced by the algorithms. Note that for the three algorithms that are not

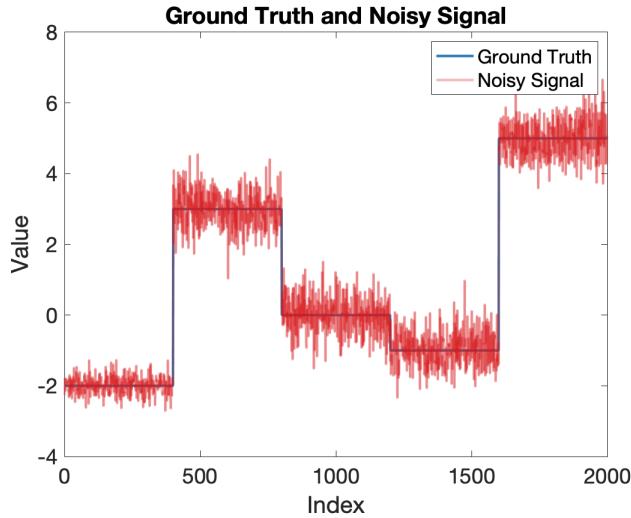


Figure 4.2: The ground truth and noisy signals used for the one-dimensional total variation denoising problem.

rPDHG, we did extensive parameter searches to find the best parameters for apt comparison. For rPDHG, however, we simply set  $\beta_0 = 1$  and choose a  $\tau_0 > 0$ , avoiding a costly parameter search.

#### 4.5.3 2D-TV Denoising

The two-dimensional total variation denoising problem follows the standard ROF model [106, 25]:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \frac{1}{2} \|x - b\|_2^2 + \|\nabla x\|_{2,1}. \quad (4.26)$$

The experiments conducted for this manuscript used the `cameraman.tif` image in MATLAB. The image was read in as a  $256 \times 256$  double and scaled so that its maximum value is 1. The image was then downsized to  $77 \times 77$  pixels for memory considerations. (As discussed previously in Section 4.4.3, for image processing tasks the  $B$  matrix can have a large memory footprint. Therefore, to run this example on a desktop computer the image was scaled down.) Noise was generated as a  $77 \times 77$  matrix drawn from a standard normal distribution and multiplied by 0.08. The original image and the noised image are shown in Fig. 4.4.

The discrete gradient operator  $\nabla$  is defined  $\nabla : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times m \times 2}$ :

$$\nabla x_{i,j}^1 = \begin{cases} x_{i+1,j} - x_{i,j}, & i = 1, \dots, m-1 \\ x_{1,j} - x_{n,j}, & i = m \end{cases} \quad (4.27)$$

$$\nabla x_{i,j}^2 = \begin{cases} x_{i,j+1} - x_{i,j}, & i = 1, \dots, n-1 \\ x_{i,n} - x_{i,1}, & i = n. \end{cases} \quad (4.28)$$

The  $\ell_{2,1}$  norm is defined as

$$\|\nabla x\|_{2,1} = \sum_{i=1}^m \sum_{j=1}^n \left| (\nabla x_{i,j}^1)^2 + (\nabla x_{i,j}^2)^2 \right|^{1/2}.$$

Similar to the one-dimensional problem, the discrete gradient operator is our  $A$  operator for the minimization problem. To calculate  $B$ , we first turn  $A$  into a matrix from its matrix-free form and then choose  $\theta > \frac{1}{\|A\|^2}$  and solve for  $B$  from

$$BB^* = \frac{1}{\theta} \mathcal{I} - AA^*$$

via Cholesky factorization. A comparison of the objective values for the different algorithms used is in Fig. 4.5 (left). PDHG and the line search for the AOI formulation of PDDR converge slowly. The line search proposed in this manuscript and the Malitsky line search both tend to “oscillate” around a solution. For a fixed number of iterations, our line search and the Malitsky line search have the lowest error. The final output of the proposed algorithm is Fig. 4.5 (right).

## 4.6 Application to Magnetic Resonance Imaging Reconstruction

The driving force behind this line search is an application to accelerated magnetic resonance imaging (MRI) reconstruction. We present a novel combination of acceleration methods, compressed sensing and partial Fourier with homodyne detection. This problem can be written in the form of (4.5).

#### 4.6.1 MRI Preliminaries

An MRI machine is a Fourier sensing device. A simple acquisition model the data from the machine  $b$  as

$$b = \mathcal{F}\{x\} + \text{noise} \quad (4.29)$$

where  $\mathcal{F}$  represents the 2 dimensional Fourier transform,  $x$  is the subject being imaged, and there is noise in the acquisition process [89]. If a sufficiently large amount of data is collected to satisfy the Nyquist-Shannon sampling theorem independently in each direction, the the image can be reconstructed with an inverse Discrete Fourier Transform:  $\hat{x} = F^{-1} b$ .

The scan time of MRI is directly proportional to the number of samples acquired. Acquiring a fully sampled grid takes a long time; to accelerate the acquisition, one can reduce the number of samples. That is, there is some data mask  $D$  (a matrix comprised of a subset of rows of the identity matrix) such that

$$b = D F x + \text{noise} \in \mathbb{C}^{nmp},$$

where the data is acquired on an  $n \times m$  Cartesian grid and  $p \in (0, 1]$  represents the undersampling factor corresponding to the number of 1s on the diagonal of  $D$ .

Reconstructing the image given the acquired samples is then a noisy, underdetermined inverse problem. Common approaches to solving this problem include compressed sensing [76, 24, 41] and machine learning techniques [110, 67].

Matrix-free implementations of the algorithm and problem statement are desirable due to memory limitations and to take advantage of the Fast Fourier Transform. A parameter-free version of PDHG is also desirable so the MRI technologist does not have to set a parameter by hand which can prevent the reconstruction of a high-quality image within a reasonable number of iterations.

#### 4.6.2 Compressed Sensing with Homodyne Detection

Compressed sensing incorporates the *a priori* knowledge that natural images are approximately sparse after an appropriate linear transformation (e.g., a discrete Wavelet transform ) to reconstruct

high-quality images with from fewer samples than those that would be required by the Nyquist-Shannon sampling limit [77, 24]. Partial Fourier reconstruction incorporates the fact that the Fourier transform of a real image is Hermitian symmetric into the reconstruction process [83, 22]. Though MR images are not generally real due to variances in the  $B_0$  magnetic field and non-linearities in the  $B_1$  magnetic field, the phase is almost all low-bandwidth. Partial Fourier with homodyne detection takes advantage of this fact [90] by estimating the phase from a low-pass filtered version of the image. Once estimated, phase correction makes the data real and the assumptions of partial Fourier reconstruction are satisfied.

Partial Fourier sampling collects data with a sampling pattern consisting of the top  $\nu$  portion, where  $\nu > 0.5$ . Common values of  $\nu$  are  $\nu = 5/8$  and  $\nu = 9/16$ . Homodyne detection uses a low-pass filtered version of the image to estimate the phase,  $\Phi$ . Once the phase is estimated, the image is reconstructed with homodyne detection using

$$x = P_\Phi b = \operatorname{Re} [\Phi F^{-1} R b],$$

where  $R$  is a weighting ramp that smoothly transitions from 2 to 0,  $\Phi = \exp(-i \operatorname{angle}(M_s))$ ,  $M_s = F^{-1} L b$ , and  $L$  is a low-pass rectangular filter [96].

The crucial realization is that once the phase  $\Phi$  is estimated, the  $P_\Phi$  operator—which implements reconstruction with homodyne detection—is a linear transformation with a computationally efficient adjoint. This allows us to combine the phase estimate  $\Phi$  into a compressed sensing image reconstruction as the following optimization problem:

$$\underset{\xi}{\text{minimize}} \quad \|\Psi P_\Phi \xi\|_1 \quad \text{subject to} \quad \|D F \xi - b\|_2 \leq \epsilon,$$

where  $\xi$  is a vector of the top  $\nu$  Fourier values for the reconstructed image,  $\Psi$  is a sparsifying transformation, and  $\epsilon > 0$  is a noise bound. A typical sparsifying transformation is a wavelet transformation. We write this as an unconstrained problem by using an indicator function to enforce data consistency:

$$\xi^* \in \underset{\xi}{\operatorname{argmin}} \quad \mathbb{I}(M \xi \in B_\epsilon[b]) + \|\Psi P_\Phi \xi\|_1, \quad (4.30)$$

where  $\mathbb{I}$  is an indicator function that requires the Fourier values  $\xi$  at the known, collected indices specified by sampling mask  $D$  to be within  $\varepsilon$  of the collected data  $b$ . Setting  $\varepsilon = 0$  enforces data consistency completely. The second term contains the sparsifying transform  $\Psi$  and the homodyne operator  $P_\Phi$ . By letting  $A = \Psi P_\Phi$ ,  $f(\xi) = \mathbb{I}(M\xi \in B_\varepsilon[b])$  and  $g(\xi) = \|\Psi P_\Phi \xi\|_1$ , we see that this problem is of the form (4.5) so we solve it via rPDHG with line search. Once  $f^*$  is determined by solving (4.30), the image is reconstructed according to  $x^* = P_\Phi \xi^*$ .

#### 4.6.2.1 Matrix-free implementation

To aid in analysis, it will help to write  $A = \Psi P_\Phi$  in its full, matrix-form implementation:

$$A = \Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ D^*. \quad (4.31)$$

Here  $D^* : \mathbb{C}^{nmp} \rightarrow \mathbb{C}^{n \times m}$  is a zero-filling operator. The linear transformation  $D$  selects only the indices at which data was collected (non-zero in the sampling mask) so  $D^*$  returns a zero-filled full-sized image. The linear transformation  $R : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$  applies a weighting matrix,  $\mathcal{F} : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$  is the two dimensional discrete Fourier transform,  $\Phi : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$  applies the phase estimate.  $\text{Re} : \mathbb{C}^{n \times m} \rightarrow \mathbb{R}^{n \times m}$  takes the real part, and  $\Psi : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times m}$  is the orthogonal discrete wavelet transform.

If  $A$  were implemented as a matrix, we could choose  $\frac{1}{\theta} \geq \|A\|^2$  and let  $B = \left( \frac{1}{\theta} \mathcal{I} - AA^* \right)^{1/2}$ , where  $(\cdot)^{1/2}$  indicates taking the Cholesky factorization of a matrix. When we choose  $\frac{1}{\theta}$  that satisfies  $\frac{1}{\theta} \geq \|A\|^2$ , then  $\frac{1}{\theta} \mathcal{I} - AA^*$  is symmetric and positive definite and the Cholesky factorization is well-defined. However, for this application, the implementation of  $A$  is matrix-free—we write it out in the form of (4.31) for analysis. Note that  $AA^* = \Psi P_\Phi P_\Phi^* \Psi^*$  is *not* equal to a scaled identity. Therefore, to use the line search presented above for an image of reasonable size we need a matrix-free implementation of  $B$  such that  $AA^* + BB^* = \frac{1}{\theta} \mathcal{I}$ .

Let  $B$  be defined as

$$B = \Psi \circ \text{Re} \circ \Phi \circ F^{-1} \circ R \circ Q. \quad (4.32)$$

$Q : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$  is defined as  $\left(\frac{1}{\theta}\mathcal{I} - RD^*DR\right)^{1/2}$ . The operator  $RD^*DR : \mathbb{C}^{n \times m} \rightarrow \mathbb{C}^{n \times m}$  is symmetric so  $Q$  is the symmetric square root. In practice, this square root is cheap to compute as the matrix is diagonal. This form of  $B$  was found by writing  $A$  in matrix form as in (4.31), forming  $AA^*$  and setting  $BB^* = \frac{1}{\theta}\mathcal{I} - AA^*$ . Note that  $R$  is symmetric. Similar to above, we write  $B$  out in its matrix form to aid in analysis, but  $B$  has a natural matrix-free implementation. With this definition of  $B$ , and understanding that we are using unitary Fourier and wavelet transformations, we can apply the primal-dual form of Douglas-Rachford to the problem (4.17) and recover the PDHG iterations.

#### 4.6.3 Results

We first compare our proposed line search method against standard PDHG and PDHG with line search [81]. A variable density sampling mask with a sampling burden of 0.3 is chosen. This corresponds to Fig. 4.6-B. Intersecting with the partial Fourier mask as in Fig. 4.6-C yields an *effective* sampling burden of approximately 0.17. (I.e., the number of samples of the reduced sampling pattern divided by those required to satisfy the Nyquist-Shannon sampling theorem was approximately 0.17.) We solve the problem (4.30) using each method with the same sampling mask. Each method is run for 4000 iterations and the image is reconstructed using the method of Roemer [105]. Comparing the reconstructions on ankle data in Fig. 4.7 center and right shows that the proposed line search yields the same quality reconstruction without a lengthy parameter search. The difference images show that the primary differences are down in the fine structure near the toes and along other areas where the phase is changing rapidly.

Fig. 4.8 is an example using brain data with the same undersampling pattern applied. On the right is the reconstruction using standard PDHG. In the center is the novel combination of compressed sensing with homodyne detection solve using the line search presented in this paper. Similar to the ankle data, the same quality reconstruction is achieved without a parameter search.

A comparison of the objective values for reconstruction of the ankle data with and without the line search is presented in Fig. 4.9. Let  $f_o$  be the objective function the algorithm is minimizing

(e.g. (4.30)). For these plots, let  $x^*$  be the optimal point obtained after letting the algorithm run for several thousand iterations. The objective value plotted in Fig. 4.9 is  $f_o(x) - f_o(x^*)$ , or the difference between the current iteration and optimal. As mentioned previously, there is no guarantee that each iterate is feasible. To calculate the objective value we first use the proximal operator of the indicator function for the constraint to make sure the iterate is feasible and then calculate the objective value. The data was run with three different methods: standard PDHG without the line search, PDHG with the Malitsky line search, and relaxed PDHG with the line search and the activation criterion discussed in Sec. 4.4.2. The objective of the relaxed PDHG algorithm jumps as different step sizes are accepted and converges to a solution better than the PDHG algorithm. The objective values for all iterations are shown for academic purposes; in practice, at any iteration, the result with the best objective value so far would be chosen. The best PDHG run is shown after a lengthy parameter search. The rPDHG run requires no such parameter search. Table 4.1 is an example of the timing difference given the parameter search. The time to 1000 iterations is longer for rPDHG but the pretraining time to find the correct set of parameters is zero.

| Algorithm              | Time of 1000 Iterations (s) | Time of Pretraining (s) | Total time (s) |
|------------------------|-----------------------------|-------------------------|----------------|
| PDHG                   | 38.44                       | 653.53                  | 691.97         |
| PDHG with Line Search  | 98.37                       | 1672.29                 | 1770.66        |
| AOI with Line Search   | 86.67                       | 1473.39                 | 1560.06        |
| rPDHG with Line Search | 417.8                       | 0                       | 417.8          |

Table 4.1: An example of the total timing difference between the other algorithms tested and rPDHG with line search given the need for a parameter search for other algorithms. The time to 1000 iterations given a set of parameters is longer for rPDHG with line search, but the total time is less.

## 4.7 Conclusion

A line search for the primal-dual hybrid gradient method was proposed. Motivated by the connection between the Douglas-Rachford algorithm and the primal-dual hybrid gradient method, this line search expresses PDHG as an averaged operator iteration. An existing line search for averaged operator iterations is then applied to the relaxation parameter. This is combined with an

existing line search over the proximal step sizes in PDHG. This new combination of line searches is essentially parameter free, allowing for quality solutions without a lengthy parameter search. Several numerical examples are demonstrated showing the convergence performance. The efficacy of the line search is presented on a novel combination of magnetic resonance imaging reconstruction techniques.

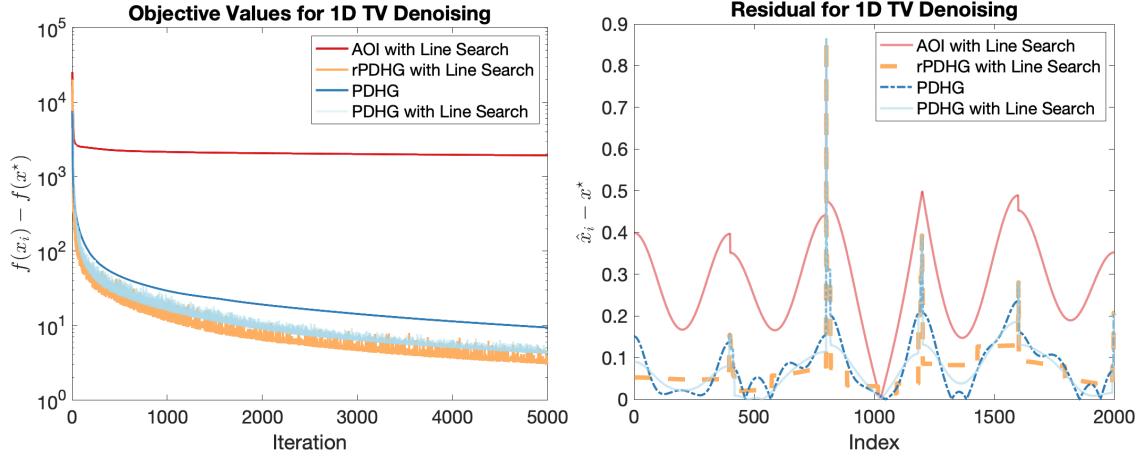


Figure 4.3: (Left) A comparison of the objective values of different algorithms solving the one-dimensional total variation denoising problem. The line search proposed in this manuscript converges with fewer iterations than either constituent line search alone. The objective value jumps often as different step sizes are chosen. The raw objective value for each step is shown for academic purposes; in practice, one would choose the best objective value so far. The PDHG line performs better than standard PDHG. The AOI line search alone converges to a poor solution. (Right) A comparison of the best answers generated by the different optimization algorithms after 1000 iterations, plotted as the residual.



Figure 4.4: The cameraman image used for the two-dimensional total variation denoising problem. On the right is the image with white noise added.

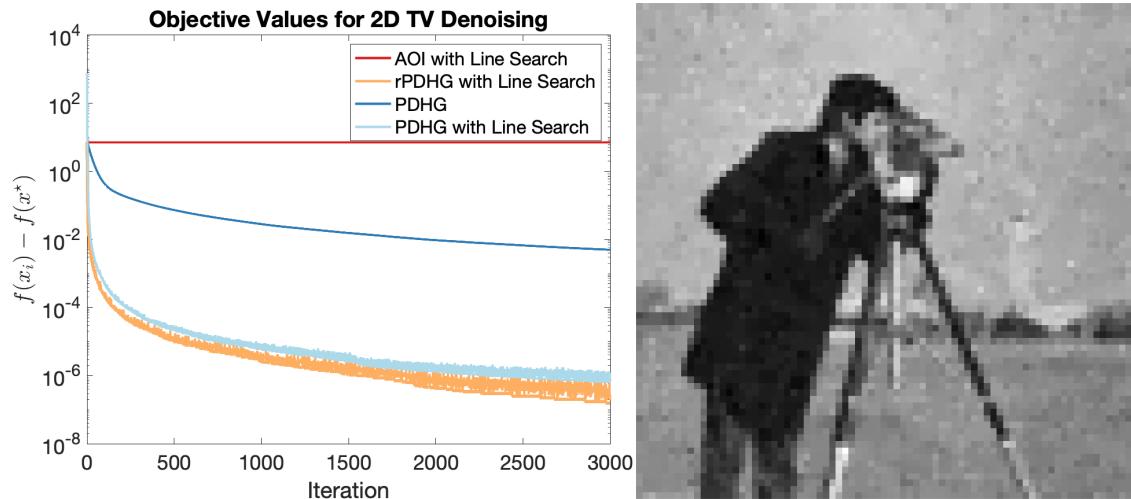


Figure 4.5: (Left) A comparison of the objective values of the different optimization algorithms used for the two-dimensional total variation denoising problem. The AOI line search converged to a poor solution. The line search proposed in this manuscript took fewer iterations to converge to any given level of error and did not require searching over any parameters. The comparison against PDHG and PDHG with line search is done against their best sets of parameters. (Right) The final output of rPDHG, the line search proposed in this manuscript, solving the two dimensional TV denoising problem as in (4.26).

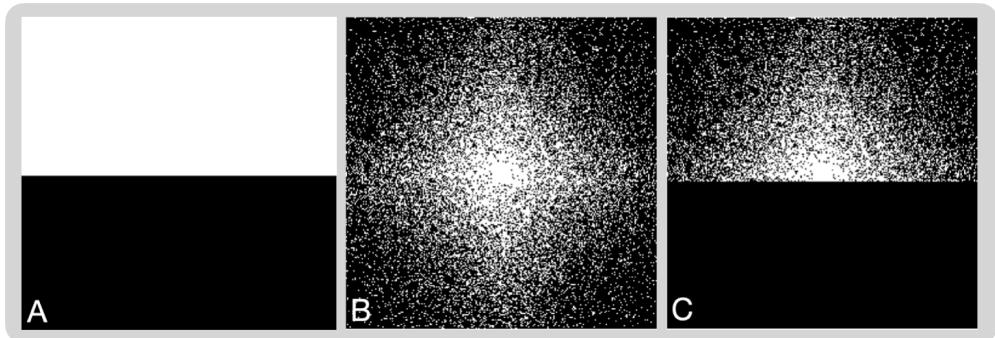


Figure 4.6: Sampling masks for three different methods of acceleration, where white indicates collected samples. A) is the sampling mask associated with partial Fourier sampling, where the entire top half plus some number of lines around the origin are collected. B) is the sampling mask associated with compressed sensing using a variable density Poisson disc sampling mask [39]. C) is the sampling mask for the novel combination of partial Fourier and compressed sensing reconstruction presented in this manuscript. It is the intersection of masks A) and B).

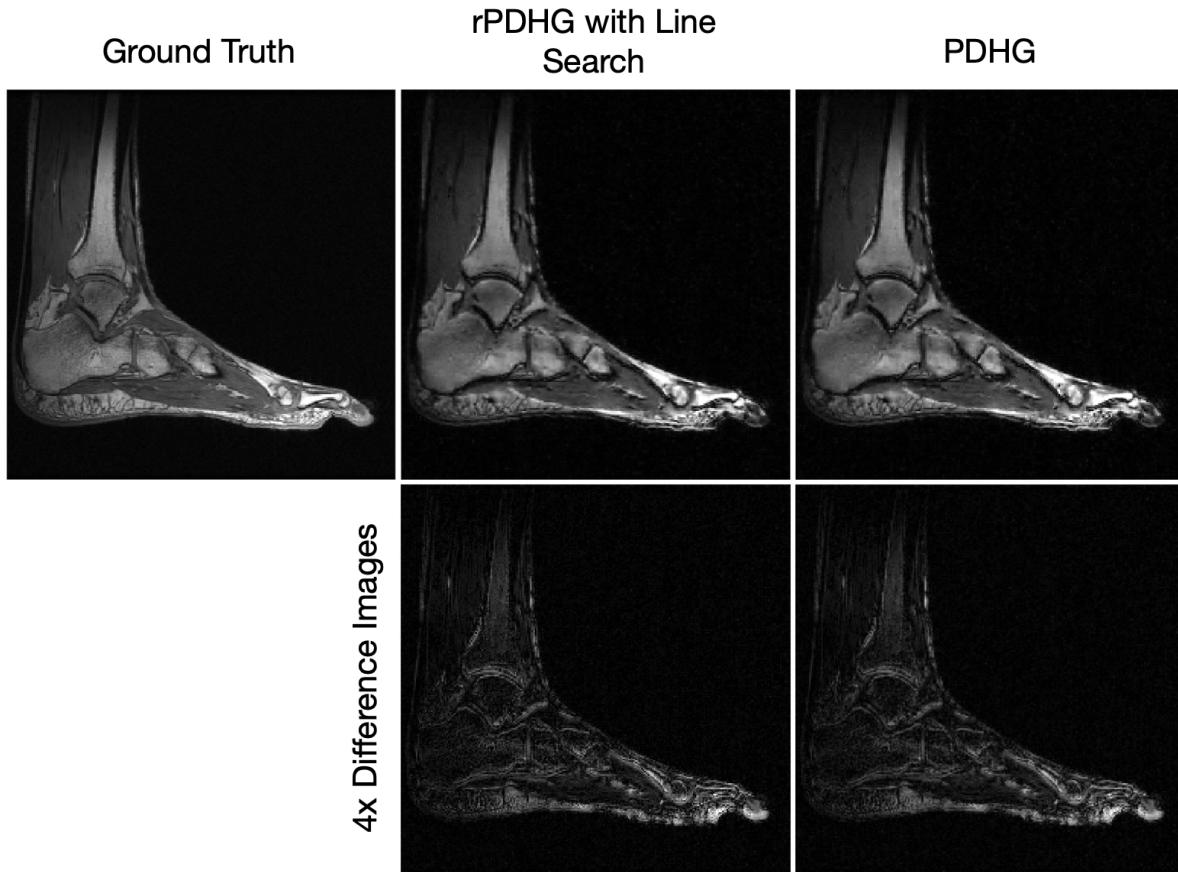


Figure 4.7: Comparison of a fully-sampled reconstruction of a sagittal slice of an ankle to two methods of reconstruction from data undersampled according to sampling mask Fig. 4.6-C. (Center) Reconstruction of the undersampled ankle data using the formulation of the problem in (4.30), solved with the line search presented in this manuscript. (Right) This reconstruction was made using the formulation of the problem in (4.30), solved with the PDHG method. Bottom row shows 4 times the magnitude difference between the accelerated reconstructions and the fully-sampled reconstruction.

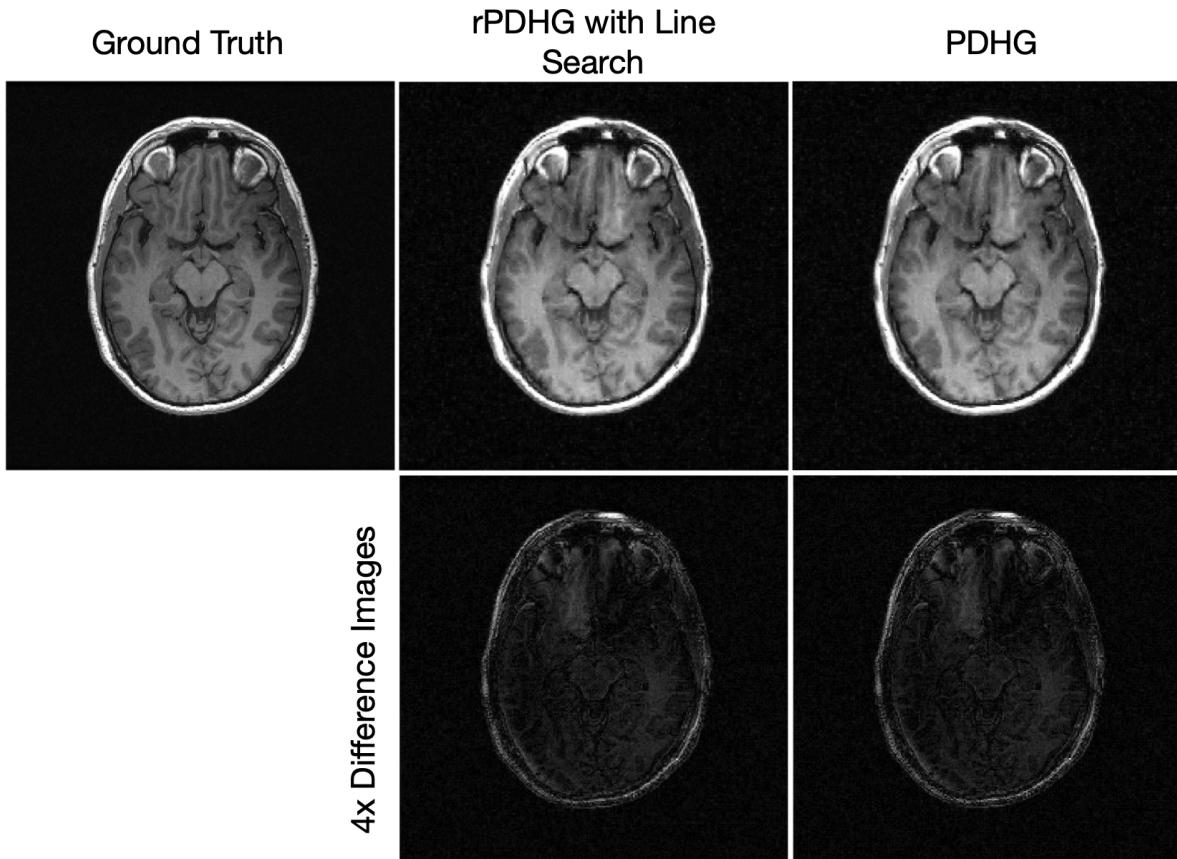


Figure 4.8: Comparison of a fully-sampled reconstruction of an axial slice of a brain to reconstruction from data undersampled according to sampling mask Fig. 4.6-C. (Center) Reconstruction of the undersampled ankle data using the formulation of the problem in (4.30), solved with the line search presented in this manuscript. (Right) Reconstruction of the undersampled ankle data using the formulation of the problem in (4.30), solved with the standard PDHG method given in (4.16). Bottom row shows 4 times the magnitude difference between the accelerated reconstructions and the fully-sampled reconstruction. The reconstruction using the new line search shows the same quality reconstruction without requiring a lengthy parameter search.

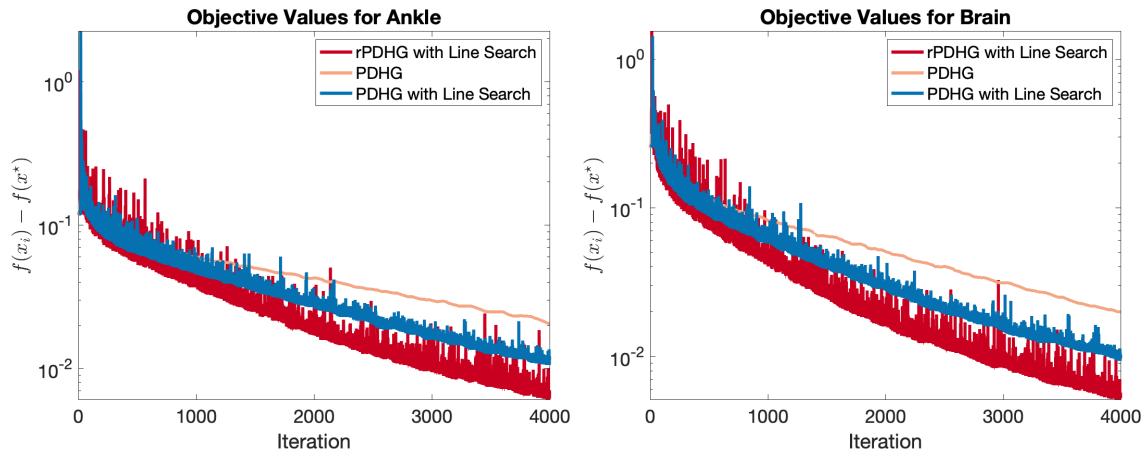


Figure 4.9: Objective value plots for the reconstruction of the same ankle data as Fig. 4.7 (left) and the brain data as in Fig. 4.8 (right). This compares the performance of rPDHG algorithm with standard PDHG and PDHG with Malitsky's line search. Over time, the line search finds a better solution. The line search jumps in objective value a lot as it chooses different step sizes. The objective value at each iteration is shown for academic purposes, though in practice at each iteration the best so far would be chosen.

## Chapter 5

### Data Consistent Deep Learning

#### 5.1 Introduction

Deep learning methods are now commonly used for interpolation and extrapolation applications, including image inpainting, audio restoration, and image reconstruction with Fourier sensing. Most methods use a data consistency penalty to encourage the output to resemble the input data but often lack specific guarantees that the output of the method is rigorously consistent with measurements from the system.

Magnetic resonance imaging (MRI) is a powerful, noninvasive medical imaging technique renowned for its diagnostic versatility. Unlike other medical imaging modalities, MRI avoids ionizing radiation, making it safer for vulnerable populations, such as children and pregnant patients, while its exceptional soft tissue contrast enables visualization of anatomical structures. Additionally, MRI's ability to generate detailed, multi-dimensional images with different contrasts supports accurate diagnosis across a wide range of conditions, from neurological disorders to musculoskeletal injuries. Its flexibility in imaging various body regions without harmful side effects further solidifies MRI as a cornerstone of modern medical diagnostics.

A challenge for MRI is long scan times. The scan time is directly proportional to the number of samples collected, so there is a natural tradeoff between scan time and quality of the resulting image. High resolution 3D scans which incur long scan times are more prone to motion artifacts; this effect is more pronounced in patients who have a hard time staying still (e.g. infants, fetuses, pregnant women).

There has been significant research effort aimed at accelerating MRI; one primary focus has been reconstructing diagnostic quality images from undersampled data. Initial steps in this direction were explored with the advent of **parallel imaging** in the 1990s. Parallel imaging uses multiple receive coils to collect redundant data, allowing undersampling by accurately estimating the missing data using Fourier theory [35, 99, 84]. In the early 2000s, **compressed sensing** came to the forefront of MRI research. Compressed sensing provides a new theory on how sampling at sub-Nyquist frequencies may still yield recovery of a quality image. Compressed sensing exploits sparsity in certain transform domains, such as wavelets, to enable even higher acceleration factors [42].

More recently, **deep learning** has come to the forefront of MR image reconstruction research. Deep learning, a subset of machine learning that is focused on computational neural networks, has enabled substantial progress on problems that were previously considered intractable. Deep learning-based reconstruction has quickly garnered lots of attention in the reconstruction community; review articles are available at, e.g., [62, 108, 91].

The challenges arising in deep learning are **generalizability** and **data consistency**. Supervised training methods rely on their training sets to reconstruct images and tend to introduce errors with data far outside of the distribution of the training set; e.g., a network trained only on data of the knee would inaccurately reconstruct an image of the brain. Data consistency is ensuring the output of the network is consistent with the measurements from the MRI machine itself that were input into the reconstruction system. There are several approaches to data consistency currently in the MRI reconstruction literature. Previous works often use **soft** data consistency, where a data consistency penalty is placed in the loss function but there is no guarantee of the output being consistent with the measurements [60].

This manuscript explores the idea of enforcing hard data consistency in deep learning reconstruction methods via a **proximal operator**. Hard data consistency, as opposed to soft data consistency, guarantees that the output of the reconstruction method is consistent with the measurements within some specified tolerance. In this manuscript, we present a formulation of the proximal

operator that is exact for single coil data and an approximation of the proximal operator for the multicoil parallel imaging setting.

We show examples of these proximal layers on several existing networks as well as our own networks. Furthermore, the primary theoretical contribution of this manuscript is a proof that, under certain conditions, hard data consistency of the type enforced by the proximal layer bounds the possible size of errors. The proof draws on the theory from compressive imaging and compressed sensing [1].

The outline of the paper is as follows. Section 5.2 contains background on the MRI reconstruction problem generally as well as a brief introduction to popular methods of solving the reconstruction problem. Section 5.3 introduces the proposed method of this manuscript and covers technical background of deep learning methods in MR image reconstruction. Section 5.4 contains results of experiments run on the networks described in section 5.3. Section 5.5 contains a technical proof showing that hard data consistency prevents reconstruction errors under certain conditions.

## 5.2 Background

The MRI machine is a **Fourier sensing device**. We model MRI as a linear sensing system, where  $x \in \mathbb{C}^N$  is the discretized image, and  $b \in \mathbb{C}^N$  are the measured **k**-space values. These are related via

$$b = Fx + \varepsilon,$$

where  $F \in \mathbb{C}^{N \times N}$  represents the discrete Fourier transform, and  $\varepsilon$  denotes measurement noise.

In the case where both  $x$  and  $b$  represent samples taken on a Cartesian grid and **k**-space was fully sampled (in the sense that the sampling frequency is chosen to satisfy the Shannon-Nyquist sampling theorem in each relevant direction),  $F$  is a square matrix in  $\mathbb{C}^{N \times N}$ . The image  $x$  can be recovered as

$$x = F^{-1}b,$$

where the inverse discrete Fourier transform  $F^{-1}$  can be computed via the Fast Fourier Transform

(FFT) algorithm. In accelerated MRI scanning, the data is undersampled. In the model this is represented by a sampling mask  $D$ , which is a matrix in  $\mathbb{R}^{m \times N}$  with  $m < N$  where the rows of  $D$  correspond to the samples of  $x$  collected. We write matrix  $A = DF$  and the model becomes

$$b = Ax + \varepsilon.$$

The matrix  $A$  has more columns than rows so the linear system is now **underdetermined** and the solution  $\hat{x} = A^{-1}b$  is no longer unique. Solving for  $x$  in this case is usually via iterative methods, for example posing it as a least-squares problem:

$$\underset{x}{\text{minimize}} \|Ax - b\|_2^2. \quad (5.1)$$

One widely used iterative method for this objective is gradient descent, which in this context yields the so-called **Landweber iteration** [29]:

$$x_{k+1} = x_k - \alpha A^* (Ax_k - b),$$

where  $\alpha > 0$  is an appropriately chosen step size and  $A^*$  represents the adjoint of the matrix  $A$ . It has been shown that stopping the iteration early may still lead to a good solution [?], though a more popular solution is to add a **regularization** term into the optimization problem

$$\underset{x}{\text{minimize}} \|Ax - b\|_2^2 + \lambda \mathcal{R}(x), \quad (5.2)$$

where  $\mathcal{R} : \mathbb{C}^N \rightarrow \mathbb{R}$  is some regularizer and  $\lambda > 0$  is the **regularization parameter**. Common choices of the regularizer include the  $\ell_1$  norm, the  $\ell_2$  norm, total variation, generalized total variation, or neural networks. [48, 42]

Consider a generic MR image reconstruction method with forward operator  $A$ . We will say that a reconstruction  $\hat{x}$  is **data consistent with parameter  $\varepsilon$**  if

$$\|A\hat{x} - b\|_2^2 \leq \varepsilon. \quad (5.3)$$

for some specified  $\varepsilon > 0$ , usually a known bound on the measurement noise. In words,  $\hat{x}$  is data consistent if the **k**-space values of the reconstruction at the measured locations do not differ too much from the measurements. With MRI,  $\varepsilon$  can be estimated by reconstructing an image from data acquired without any excitation.

### 5.2.1 Parallel Imaging

One of the pioneering methods of accelerating MRI acquisition was **parallel imaging**. Parallel imaging uses multiple receive coils to image the subject. The data collected from each of the receive coils is used to reconstruct the image from undersampled collection both in k-space [114, 65, 56, 78, 84] and image space [101].

Parallel imaging reconstruction with sensitivity encoding[99] uses **sensitivity maps** which specify how much the coils apodize a signal measured from each location in space. The sensitivity maps can be estimated from a smaller, fully-sampled collection at the center of k-space [116, 73]. The data for the  $i^{\text{th}}$  receive coil is then modeled as

$$b^{(i)} = FS_i x + \varepsilon, \quad (5.4)$$

where  $S_i \in \mathbb{C}^{N \times N}$  is a diagonal matrix containing the sensitivity profile for coil  $i$ . By applying the sampling mask  $D$ , the model is

$$b^{(i)} = DFS_i x + \varepsilon.$$

Aggregating this across coils yields the regularized objective

$$\underset{x}{\text{minimize}} \left\{ \sum_{i=1}^{n_c} \|DFS_i x - b_i\|_2^2 + \mathcal{R}(x) \right\}, \quad (5.5)$$

where  $n_c$  is the total number of receive coils.

### 5.2.2 Compressed Sensing

Compressed sensing uses ideas of sparsity to recover images sampled at sub-Nyquist levels. Early efforts in this vein include those of Donoho [37] and Candès and Tao [23] in the early 2000s. Compressed sensing was introduced to MRI by Lustig et al. [77], and recently extended by Dwork et al. [41, 40, ?]. A comprehensive overview of compressive imaging theory is provided in [1].

The general idea is, if there exists some basis  $\Psi$  in which a given signal is sparse or approximately sparse, it is possible to recover an accurate representation of the signal with fewer samples than is specified by the Shannon-Nyquist sampling theorem. Compressed sensing has found success in

MRI and is included in most clinical MRI scanners. However the difficulty in selecting appropriate regularization parameters and long reconstruction times can complicate the use of compressed sensing in the clinic.

In this work, we assume sparsity in the wavelet domain, which is widely used in compressed sensing MRI [80]. We will borrow from the ideas of sparse recovery for a proof later in this manuscript.

To reconstruct with compressed sensing, the regularizer in (5.2) is the  $\ell_1$  norm of the signal's representation in the sparse domain, such as wavelets. The problem to be solved is a modified basis pursuit denoising problem:

$$\underset{x}{\text{minimize}} \|Ax - b\|_2^2 + \lambda\|Wx\|_1, \quad (5.6)$$

where  $\lambda > 0$  is a regularization parameter to balance the solution between the data consistency and regularization terms. Overall, the development of both parallel imaging and compressed sensing have greatly increased the utility of the MRI machines in the clinic.

Solving (5.6) can be more challenging since the  $\ell_1$ -norm is not differentiable. The problem may be cast as a second order cone problem and solved via interior point methods [15], but the dimensionality of the MRI problem usually makes this approach intractable. The classic approach in imaging is the **iterative soft-thresholding algorithm** or ISTA, which is a gradient-based first order method (see [26]). This method alternates between a gradient step and a shrinkage or soft-thresholding step. This is a special case of proximal gradient descent, also known as forward-backward splitting [30].

The shrinkage operation is an example of a **proximal operator**, specifically the proximal operator of the  $\ell_1$ -norm.

**Definition 23** (Proximal Operator). Given a **closed, convex and proper** function  $f : \mathcal{H} \rightarrow \mathbb{R}$  for some Hilbert space  $\mathcal{H}$  and  $\tau > 0$ , the **proximal operator** or **proximal mapping** of  $f$  is given by

$$\text{prox}_{\tau f}(u) = \underset{y \in \mathcal{H}}{\text{argmin}} \left\{ f(y) + \frac{1}{2\tau} \|y - u\|_2^2 \right\}. \quad (5.7)$$

The proximal operator is a generalization of Euclidean projection [95] (in the sense that the proximal operator of an indicator function of a convex set is the projection onto that set) and is widely used in first-order optimization schemes of convex but nonsmooth functions [12].

### 5.2.3 Deep Learning in MRI

While compressed sensing and parallel imaging have improved acceleration in MRI, these methods rely on handcrafted regularizers with unknown values for the regularization parameters. Recent advances in deep learning offer a powerful alternative. The standard form of deep learning methods are **supervised** deep learning methods, where the weights of the neural network are learned from a training set of fully-sampled data [123, 111]. A challenge for such deep learning methods is the lack of publicly available datasets for training. There has been considerable effort in putting together datasets and challenges such as FastMRI [69]. Regulations and privacy concerns prohibit hospitals from sharing certain types of data, so truly generalized datasets do not yet exist. For supervised methods, this leads to concerns about generalizability of deep learning methods to data outside of the training set. In medical image reconstruction, this raises concerns about rare pathologies that are not present in the training set [103, 86].

There are several approaches to try to ensure data consistency with deep learning reconstruction systems. The most popular option is using a data consistency **penalty** term such as a least squares term:

$$p(x) = \frac{1}{2} \|Ax - b\|_2^2.$$

This term is desirable because it is differentiable and its gradient can easily and cheaply be evaluated in closed form. However this penalty term only **encourages** the solution to be data consistent. The solution at the end is not guaranteed to be data consistent; the recent work of [?] shows that even with this term included, the inconsistency can be significant enough for the data to be discarded.

A different approach to training is zero-shot self-supervised learning methods [122]. In this training paradigm, the network is trained only on the set of data that is being reconstructed. This network type offers complete generalizability in the sense that it is retrained for each new set of

data.

A popular type of network used with MRI reconstruction are called **unrolled networks**.

These networks start with the form of some iterative optimization algorithm that solves a problem like (5.2). They unroll the iterations and replace some part of it with a neural network. These networks have seen some success as in [66, 59].

## 5.3 Methods

### 5.3.1 Proximal Layer

Consider a generic MR image reconstruction method with forward operator  $A$ . An output image  $x$  is **data consistent** when

$$\|Ax - b\|_2 \leq \varepsilon, \quad (5.8)$$

for some specified  $\varepsilon > 0$ , usually a known bound on the measurement noise. With MRI,  $\varepsilon$  can be estimated by reconstructing an image from data acquired without any excitation. To enforce data consistency, introduce the function

$$\mathcal{I}(x) = \mathbb{I}_{B[Ax - b, \varepsilon]}(x),$$

where  $B[Ax - b, \varepsilon]$  is the set of all  $x$  such that  $\|Ax - b\|_2 \leq \varepsilon$  and  $\mathbb{I}$  is an indicator function for the set which equals 0 when  $x$  is inside the set and  $\infty$  otherwise. The proximal operator of this indicator function is [95]

$$\text{prox}_{\mathcal{I}}(v) = \operatorname{argmin}_x \left\{ \mathcal{I}(x) + \frac{1}{2} \|x - v\|_2^2 \right\}. \quad (5.9)$$

Given an image  $v$  as input to this operator, the output is the image closest to  $v$  that is also consistent with the data.

To enforce hard data consistency,  $\varepsilon = 0$ . The proximal operator performs a Euclidean projection onto the set of exactly data consistent images. We propose incorporating a proximal projection **layer** as part of the network itself. This proximal layer performs the projection described above and is included as a standalone layer in a deep neural network, appended onto the end of

the forward method. When implemented with data collected by a single coil this reduces to a closed-form projection, which is computationally inexpensive. In the parallel imaging setting, (5.9) is a convex optimization problem that may be infeasible due to inaccuracies in the estimates of the sensitivity maps. In [60], Hammernik et al. suggest that this problem could be solved with the conjugate gradient method. However, this would dramatically increase the computation required for training, rendering the inclusion of a full data consistency operator computationally prohibitive. Instead, we incorporate a computationally efficient approximation to the proximal operator that allows the inclusion of a data consistency layer in the parallel imaging setting. It appears Samsonov et al. [109] were the first to use this style of approximation to the proximal operator in the convex optimization setting. The algorithm is listed in Alg. 5.

---

**Algorithm 5** Computationally Efficient Approximation for Data Consistency

---

- 1: **Inputs:** Image  $x$
  - 2: **for**  $c = 1, 2, \dots, C$  **do**
  - 3:   Multiply the image by the  $c^{\text{th}}$  coil's sensitivity map:  $y^{(c)} = S^{(c)}x$
  - 4:   Project  $y^{(c)}$  onto the set of data consistent images by solving (5.9) for a single coil:  $z^{(c)} = \Pi(y^{(c)})$
  - 5: **end for**
  - 6: Combine  $\{z^{(c)} : c = 1, 2, \dots, C\}$  using the method of Roemer [105]
- 

To add this consistency layer to a network, Alg. 5 is placed in the forward method of the network. The proximal layer is invoked during training which allows the network to incorporate this information into parameter updates, and during inference. If this is the last step of the network, then the output is guaranteed to be data consistent as defined in (5.8).

### 5.3.2 Convolutional Neural Networks

A popular application of neural networks to MR image reconstruction is using convolutional neural networks. If the network is formulated to work in image space—in the sense that the input and output of the network are images as opposed to **k**-space values—the network is trained as an image denoiser. It takes the zero-filled reconstruction as input and outputs a clean image. If the network is formulated to take in **k**-space samples, the network is attempting to approximate the

inverse map  $A^{-1}$ . This is a supervised learning method that requires a database of fully sampled training examples.

One example of this is the U-Net included in the FastMRI GitHub [69]. As described above, to make this network data consistent, Alg. 5 added as a final layer onto the U-Net. Since this network operates only on single coil data,  $C = 1$  and we perform only one projection and no coil combination.

### 5.3.3 Variational Network

The variational network introduced by Hammernik et al. in [66] uses the **field of experts** as the regularizer. The field of experts is a generalization of the widely-used **total variation** regularization term [31]. The authors adopt this model to construct a learned regularization term that generalizes traditional hand-crafted priors such as total variation. Specifically, the regularizer is defined as

$$R(u) = \sum_{i=1}^{N_k} \langle \Phi_i(K_i u), \mathbf{1} \rangle,$$

where  $K_i$  denotes a learned linear operator (typically implemented as convolution with trainable filters), and  $\Phi_i$  is a learned non-linear potential function applied pointwise to the filter responses. The inner product with the all-ones vector sums the contributions spatially across the image.

Rather than relying on fixed assumptions about image sparsity or smoothness, the authors parameterize both the filters  $K_i$  and the non-linearities  $\Phi_i$ , and learn them directly from data. This results in a flexible and expressive prior capable of modeling complex structures present in MR images. When integrated into a variational reconstruction framework, this regularizer acts analogously to a learned proximal operator or gradient step, guiding the reconstruction away from undersampling artifacts and toward more plausible solutions.

By learning the regularization from data, the Fields of Experts approach allows the variational network to adapt to specific anatomical and artifact patterns, overcoming limitations of more rigid models like TV or TGV.

Data consistency is handled in the variational network as a gradient descent step on the data

consistency 2–norm term. This is *soft* data consistency as a penalty. To enforce data consistency, we appended the proximal layer onto the end of the forward method and retrained.

### 5.3.4 Unrolled Networks

Unrolled networks take the steps of an iterative optimization algorithm and replace some part of each step with a neural network. One method begins with the iterations for proximal gradient descent (see Alg. 6) and replaces the proximal operator with a neural network.

Our implementation of an unrolled network begins from the iterations of proximal gradient descent with momentum (see Alg. 7). This algorithm solves a problem of the form

$$\hat{x} \in \operatorname{argmin}_x f(x) + g(x), \quad (5.10)$$

where  $f$  is smooth and  $g$  is convex but nonsmooth and has a proximal operator that is computationally efficient to evaluate. If  $f = \frac{1}{2}\|A \cdot - b\|_2^2$  and  $g = \|\cdot\|_1$ , this is the popular FISTA algorithm [13].

To make this an unrolled network, we replace the **momentum** step with a neural network. This is applied to the somewhat pathological problem

$$\underset{x}{\text{minimize}} \|Ax - b\|_2^2 \quad \text{s.t. } Ax = b. \quad (5.11)$$

The network starts by taking a gradient descent step on the differentiable term. Next, the proximal step for the constraint is a projection which enforces hard data consistency. Finally the convolutional neural network is applied. This block is repeated 10 times, and the convolutional neural network is **shared** between steps.

**Definition 24** (Complex Sign). The sign of a complex scalar  $z \in \mathbb{C}$  is defined as

$$\operatorname{sign}(z) = \frac{z}{|z|}, \quad z \in \mathbb{C} \setminus \{0\}, \quad (5.12)$$

with  $\operatorname{sign}(0) = 0$ .

**Definition 25** (Soft Thresholding Operator). The scalar **soft thresholding operator**  $\mathcal{S}_\tau : \mathbb{R} \rightarrow \mathbb{R}$

is defined as

$$\mathcal{S}_\tau(x) = \begin{cases} \text{sign}(x) (|x| - \tau) & |x| > \tau \\ 0 & \text{otherwise.} \end{cases} \quad (5.13)$$

The vector soft thresholding operator is defined as applying the scalar operator component wise.

The complex soft thresholding operator is defined equivalently using the complex sign function (5.12).

---

**Algorithm 6** Proximal Gradient Descent

---

**Require:** Objective function  $f(x) + g(x)$ , initial point  $x_0$ , step size  $\eta > 0$

```

1:  $x_n \leftarrow x_0$ 
2: for  $n = 0, 1, \dots$  do
3:    $x_{n+1/2} \leftarrow x_n - \eta \nabla f(x_n)$ 
4:    $x_n \leftarrow \text{prox}_{\eta g}(x_{n+1/2})$ 
5: end for
6: return  $x$ 
```

---



---

**Algorithm 7** FISTA [13]

---

**Require:** Objective function  $f(x) + g(x)$ , initial point  $x_0$ , step size  $\eta > 0$

```

1:  $x_n \leftarrow x_0$ 
2: for  $n = 0, 1, \dots$  do
3:    $x_{n+1/2} \leftarrow x_n - \eta \nabla f(x_n)$ 
4:    $x_{n+1} \leftarrow \text{prox}_{\eta g}(x_{n+1/2})$ 
5:    $t_{n+1} \leftarrow \frac{1+\sqrt{1+4t_n^2}}{2}$ 
6:    $y_{n+1} \leftarrow x_n + \left( \frac{t_n-1}{t_{n+1}} \right) (x_n - x_{n-1})$ 
7: end for
8: return  $x$ 
```

---

### 5.3.5 Zero-Shot Self Supervised Learning

A difficulty with traditional supervised learning methods applied to MRI reconstruction is a lack of quality training data. There are regulations governing when and how hospitals can share patient data. One large effort to generate a sizeable training set is fastMRI [69] which has become a useful benchmark for research networks.

Another approach is called zero-shot learning, where there is no large training set [122]. The network is trained again on each new data to be reconstructed. Since the (undersampled) k-space is

the only source of data, this is also a self-supervised method.

Zero-shot learning works by taking random subsets of the collected data and partitioning them into training and loss masks. Each epoch iterates over the specified number of “sub masks”, which now takes the place of a typical mini-batch in machine learning. On each sub-iteration, the data from the sub training mask is selected and used as the input to the network. The loss is then calculated only on the loss mask. In the current implementation, the training and loss masks are applied in **k**-space.

## 5.4 Results

We use a combination of data from the FastMRI dataset [69] and [mridata.org](http://mridata.org) [93]. The data is 3D cartesian sampling with two dimensions of phase encodes ( $k_x, k_y$ ) and one dimension of readout ( $k_z$ ). After inverse Fourier transforming along the readout dimension, the data is placed into the  $(k_x, k_y, z)$  hybrid domain [19]. Since the data is fully-sampled along the readout dimension (along with the use of the anti-aliasing analog filter included in the MRI machines), there is negligible aliasing in the hybrid domain along the  $z$  dimension. Therefore, once in the hybrid domain, we will independently reconstruct each slice with two-dimensional (2D) processing.

### 5.4.1 FastMRI U-Net

We first incorporated the proximal layer into the U-Net from FastMRI’s GitHub [69]. We modified the U-Net so it operates on complex image space. We append the data consistency proximal layer to the end. Similar to the existing model, it was trained with the  $\ell_1$  loss.

For this experiment, the input to the network was the zero-filled reconstruction from retrospectively downsampled single coil data using a variable density sampling mask. We trained this network both with and without the data consistency layer on the single coil knee training data from the FastMRI challenge [69]. The networks then reconstruct knee data and brain data. The metrics calculated are the structural similarity index measure (SSIM), the peak signal-to-noise ratio (pSNR), and the mean-squared error (MSE). All three metrics require a ground truth image, so we

retrospectively downsample the fully sampled **k**-space for the input data.

Results shown from FastMRI [69] used a separable Laplacian variable-density sampling mask at a 5% sampling burden [42]. We compare reconstructions for data of the knee (in-distribution) as well as data of the brain (out-of-distribution). Fig. 1 shows a comparison of in-distribution data (data collected of a knee) reconstructed from both the consistent and inconsistent network. Fig. 2 shows a comparison of a reconstructed image from out-of-distribution data. Both figures show results for the network with and without the data consistency layer included, along with the fully sampled reconstruction for reference. Both networks reconstruct high-quality images for in-distribution knee data. However, the network without the data consistency layer fails to reconstruct an accurate image on the out-of-distribution brain data. Even though there was not any brain data included in the training set, the network with enforced data consistency was able to reconstruct a high-quality image of the brain.

We performed a paired Wilcoxon signed-rank test comparing reconstructions with and without data consistency. SSIM showed a significant improvement with data consistency (median 0.520 vs. 0.390,  $p < 1e - 24$ ,  $r = 0.36$ ). Similar trends were observed for PSNR (median 25.1 vs. 24.98,  $p < 1e - 17$ ,  $r = 0.30$ ) and MSE (median 0.0031 vs. 0.0032,  $p < 1e - 11$ ,  $r = 0.24$ ).

Table 5.1: Paired comparison of image quality metrics with and without data consistency. Reported values are median  $\pm$  interquartile range (IQR). Statistical significance was assessed using the Wilcoxon signed-rank test.

| Metric | With DC             | Without DC          | <i>p</i> -value / Effect size <i>r</i> |
|--------|---------------------|---------------------|--|
| SSIM   | $0.5201 \pm 0.3288$ | $0.3895 \pm 0.2709$ | $1.17 \times 10^{-25}$ / 0.3613        |
| PSNR   | $25.106 \pm 3.4538$ | $24.980 \pm 3.5231$ | $2.77 \times 10^{-18}$ / 0.3009        |
| MSE    | $0.0031 \pm 0.0026$ | $0.0032 \pm 0.0029$ | $2.17 \times 10^{-12}$ / 0.2423        |

#### 5.4.2 Variational Network

We next evaluated the impact of our proximal data consistency layer on a variational network architecture. Specifically, we implemented the network proposed by Hammernik et al. [66], which is based on unrolling a variational optimization framework with a learned regularization term modeled

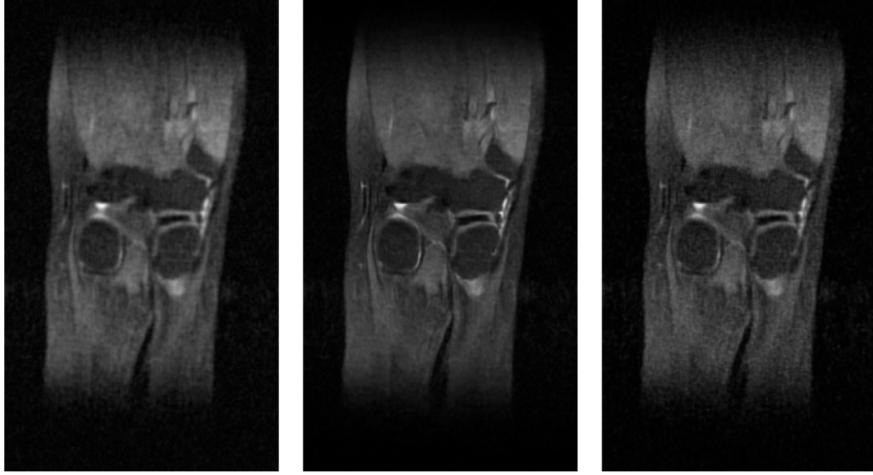


Figure 5.1: A comparison of the undersampled reconstruction (left), and reconstructions using the network without data consistency (center) and with data consistency (right) on in-distribution data. The network without data consistency is overall “smoother” and has fewer noise artifacts.

via Fields of Experts (FoE).

We modified their implementation to append our proximal data consistency layer at the final iteration. This ensures that the network’s output is strictly consistent with the measured k-space data, rather than relying on a soft penalty. The network was trained on knee MRI data using the same  $4\times$  Cartesian undersampling mask as in the original work.

Quantitative results are summarized in Tables 5.2 and 5.3. Although the variational network already produces high-quality reconstructions, adding the data consistency layer yields statistically significant improvements in SSIM, PSNR, and MSE across test slices. Notably, the effect size is large for SSIM, indicating perceptual improvements that are clinically meaningful.

Figure 5.3 shows example axial T2-weighted knee reconstructions. While the network without data consistency produces a visually plausible result, it introduces a hallucinated structure. This artifact is removed when the data consistency layer is used. The improvement is not only quantitative but also critical for clinical interpretability.

Results using the variational network [66] used their original code, which includes a  $4\times$  acceleration factor Cartesian undersampling pattern. Reconstruction results are seen in Fig. 3. The variational network trained without the data consistency layer produces reconstructions that may

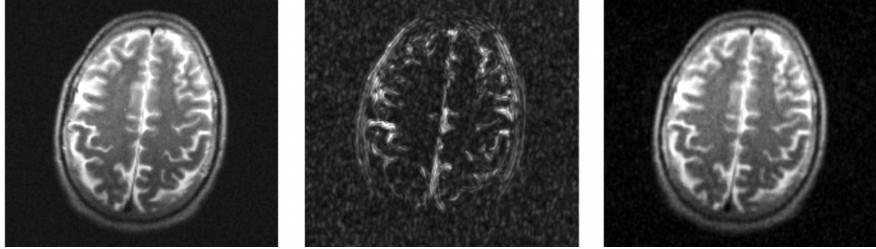


Figure 5.2: A comparison of the fully sampled reconstruction (left), and reconstructions using the network without data consistency (center) and with data consistency (right) on out-of-distribution data. The network with data consistency produces a high-quality image, while the network without data consistency produces a low-quality reconstruction that lacks clinical interpretability.

introduce hallucinated structures. The example in Fig. 3 shows that the hallucination from the “inconsistent” reconstruction is gone in the reconstruction of the same data from the network with the data consistency layer.

#### 5.4.3 Zero-Shot Self-Supervised Learning

We used the unrolled network described above, trained in a zero-shot self-supervised fashion, to reconstruct fetal data acquired from the University of Colorado Hospital. This data was collected using a 3D Cartesian acquisition with full sampling along the readout ( $k_z$ ) direction and variable-density undersampling in the  $(k_x, k_y)$  plane. After inverse Fourier transforming along the readout dimension, we performed 2D reconstruction in the hybrid  $(k_x, k_y, z)$  domain.

Zero-shot self-supervised learning requires a different training paradigm than supervised approaches. For each dataset, we define a set of sub-masks by partitioning the available k-space measurements into “training” and “loss” masks. On each sub-iteration, the network is presented only with the training mask as input, while the loss is computed solely over the disjoint loss mask. This allows for a self-supervised loss formulation without the need for fully sampled ground truth.

Because the masks change during training, enforcing data consistency must be applied per sub-mask. We incorporated the proximal layer within each forward pass, applied to the training mask only. This ensures consistency with the observed measurements even as the sampling mask varies during training.

| Metric | With DC (Median $\pm$ IQR) | Without DC (Median $\pm$ IQR) | p-value / Effect size |
|--------|----------------------------|-------------------------------|-----------------------|
| SSIM   | $0.9548 \pm 0.0153$        | $0.9450 \pm 0.0155$           | $1.1e - 22 / 0.693$   |
| PSNR   | $40.2913 \pm 2.0545$       | $40.2539 \pm 2.0841$          | $9.9e - 17 / 0.587$   |
| MSE    | $0.0002 \pm 0.0001$        | $0.0002 \pm 0.0001$           | $1.9e - 17 / 0.601$   |

Table 5.2: Comparison of coronal reconstruction quality metrics with and without data consistency (DC).

#### 5.4.4 Unrolled Network

We also evaluated the impact of the proximal data consistency layer within a custom unrolled network architecture. This network was based on proximal gradient descent with momentum (Alg. 6), where each iteration is unrolled as a separate block. The gradient and proximal steps are retained from classical optimization, while the momentum step is replaced with a convolutional neural network.

This is an extension of ISTA-Net [?]. In ISTA-Net, the authors took inspiration from ISTA, or proximal gradient descent (see Alg. 6). The authors unrolled iterations and replaced the proximal evaluations with learned convolutional layers. Without a proximal evaluation, the network does not enforce data consistency.

In our extension to this network, we maintain the proximal evaluations to enforce data consistency and instead learn the momentum layer. This is distinct from, e.g., [?] which replaces the proximal layer with a learned layer.

We trained this network in a supervised training fashion on brain data from the FastMRI dataset[69]. We performed an ablation study to see the performance of this unrolled network at different sampling burdens. To evaluate performance, we calculate the same three metrics as earlier for each element of the testing set after training for 200 epochs and taking the network that had the best validation performance.

At each sampling burden, the model with data consistency performs better on mean-squared error, peak signal-to-noise ratio (pSNR), and structured similarity index measure (SSIM). The model without data consistency overall consistently underperforms across all evaluated metrics.

| Metric | With DC (Median $\pm$ IQR) | Without DC (Median $\pm$ IQR) | p-value / Effect size |
|--------|----------------------------|-------------------------------|-----------------------|
| SSIM   | $0.9603 \pm 0.0141$        | $0.9583 \pm 0.0150$           | $1.4e - 34 / 0.867$   |
| PSNR   | $40.9991 \pm 2.2566$       | $40.9421 \pm 2.2571$          | $5.9e - 25 / 0.730$   |
| MSE    | $0.0002 \pm 0.0001$        | $0.0002 \pm 0.0001$           | $2.4e - 26 / 0.751$   |

Table 5.3: Comparison of axial reconstruction quality metrics with and without data consistency (DC).

Examples of reconstructions from the network trained with and without data consistency are in Figs. 5.4.4 and 5.4.4. The network with data consistency retains better quality as the sample burden drops.

## 5.5 Theoretical Guarantees

We desire a rigorous proof that enforced data consistency makes hallucinations of a certain size impossible. We adopt the notation and theory of compressed sensing literature. Most of the following theory is developed for the **sparse recovery** problem. The properties developed are useful for showing behavior of vectors in the null-space of a matrix  $A$ . This is relevant for two reasons:

- Under strict data consistency, any hallucination must lie in the null-space of the measurement matrix  $A$ , and
- there has been great work done to show that the measurement matrices for MRI problems satisfy these properties.

Let  $x \in \mathbb{C}^N$ . The **support** of  $x$  is

$$\text{supp}(x) = \{i : x_i \neq 0\} \subseteq \{1, \dots, N\}.$$

**Definition 26 (s-sparse vectors).** A vector  $x \in \mathbb{C}^N$  is **s-sparse** for some  $1 \leq s \leq N$  if

$$\|x\|_0 = |\text{supp}(x)| \leq s.$$

Denote as  $\Sigma_s$  the set of all **s-sparse** vectors.

**Definition 27** (Integer Partition). We define a **partition** of the natural numbers from  $1, \dots, N$  as

$$D_s = \{\Delta : \Delta \subseteq \{1, \dots, N\}, |\Delta| = s\}. \quad (5.14)$$

Define  $P_\Delta \in \mathbb{C}^{N \times N}$  as the matrix of the orthogonal projection with range  $\text{span}\{i : i \in \Delta\}$ . If  $x \in \mathbb{C}^N$  then

$$(P_\Delta x)_i = \begin{cases} x_i & i \in \Delta \\ 0 & \text{otherwise.} \end{cases}$$

We will write  $P^\perp$  for the orthogonal complement of an orthogonal projection  $P : \mathbb{C}^N \rightarrow \mathbb{C}^N$ , with  $P_\Delta^\perp = P_{\Delta^c}$  where  $\Delta^c = \{1, 2, \dots, N\} \setminus \Delta$  is the complement of  $\Delta$  in  $\{1, 2, \dots, N\}$ .

**Definition 28** (Null Space Property). A matrix  $A \in \mathbb{C}^{m \times N}$  satisfies the **null space property** of order  $1 \leq s \leq N$  if

$$\|P_\Delta x\|_1 < \|P_\Delta^\perp x\|_1 \quad (5.15)$$

for all  $x \in \text{Ker}(A) \setminus \{0\}$  and  $\Delta \in D_s$  where  $D_s$  is as in 5.14.

**Definition 29** (Robust Null Space Property). A matrix  $A \in \mathbb{C}^{m \times N}$  satisfies the **robust null space property (rNSP)** of order  $1 \leq s \leq N$  with constants  $0 < \rho < 1$  and  $\gamma > 0$  if

$$\|P_\Delta x\|_2 \leq \frac{\rho}{\sqrt{s}} \|P_\Delta^\perp x\|_1 + \gamma \|Ax\|_2, \quad (5.16)$$

for all  $x \in \mathbb{C}^N$  and  $\Delta \in D_s$ .

The robust Null Space Property of order  $s$  implies the Null Space Property of order  $s$  as

$$\|P_\Delta x\|_1 \leq \sqrt{s} \|P_\Delta x\|_2$$

by the Cauchy-Schwarz inequality, and therefore

$$\|P_\Delta x\|_1 \leq \rho \|P_\Delta x\|_1 + \gamma \sqrt{s} \|Ax\|_2 < \|P_\Delta x\|_1, \quad \forall x \in \text{Ker}(A) \setminus \{0\}.$$

The robust Null Space Property is more useful in general as it applies to all vectors, not just those in the null space of  $A$ .

**Definition 30** (Restricted Isometry Property). Let  $1 \leq s \leq N$ . The  $s^{\text{th}}$  **restricted isometry constant**  $\delta_s$  of a matrix  $A \in \mathbb{C}^{m \times N}$  is the smallest  $\delta > 0$  such that

$$(1 - \delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta)\|x\|_2^2, \quad \forall x \in \Sigma_s. \quad (5.17)$$

If  $0 < \delta_s < 1$  then the matrix  $A$  is said to have the **restricted isometry property** of order  $s$ .

The following lemma is the important one:

**Lemma 2.** Suppose that  $A \in \mathbb{C}^{m \times N}$  satisfies the RIP of order  $2s$  with constant

$$\delta_{2s} < \sqrt{2} - 1.$$

Then  $A$  satisfies the rNSP of order  $s$  with constants

$$\rho = \frac{\sqrt{2}\delta_{2s}}{1 - \delta_{2s}}, \quad \gamma = \frac{\sqrt{1 + \delta_{2s}}}{1 - \delta_{2s}}.$$

**Definition 31** (G-adjusted restricted isometry property). Let  $1 \leq s \leq N$  and  $G \in C^{N \times N}$  be nonsingular. The  $s^{\text{th}}$  **G-adjusted restricted isometry constant**  $\delta_{s,G}$  of matrix  $A \in \mathbb{C}^{m \times N}$  is the smallest  $\delta > 0$  such that

$$(1 - \delta)\|Gx\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta)\|Gx\|_2^2, \quad \forall x \in \Sigma_s. \quad (5.18)$$

If  $0 < \delta_{s,G} < 1$ , then  $A$  is said to have the **G-adjusted restricted isometry property** of order  $s$ .

### 5.5.1 Sparsity in Levels

**Definition 32** (Sparsity in Levels). Let  $1 \leq r \leq N$ ,  $M_0 = 0$ ,  $M = \{M_1, \dots, M_r\}$ , where  $1 \leq M_1 < \dots < M_r = N$ , and  $s = \{s_1, \dots, s_r\}$ , where  $s_k \leq M_k - M_{k-1}$  for  $k = 1, \dots, r$ . A vector  $x \in \mathbb{C}^N$  is  $(s, M)$ -sparse in levels if

$$|\text{supp}(x) \cap \{M_{k-1} + 1, \dots, M_k\}| \leq s_k, \quad k = 1, \dots, r.$$

The **total sparsity** of  $x$  is  $s = \sum_i s_i$ , and the set of **(s, M)-sparse** vectors is  $\Sigma_{s,M}$ .

**Definition 33.** Denote as  $D_{s,M}$ :

$$D_{s,M} = \{\Delta \subset \{1, \dots, N\} : |\Delta \cap \{M_{k-1} + 1, \dots, M_k\}| = s_k, k = 1, \dots, r\}. \quad (5.19)$$

This is the set of integer partitions of  $\{1, \dots, N\}$  that correspond to **(s, M)-sparsity**.

**Definition 34** (G-adjusted restricted isometry property in levels). Let  $G \in \mathbb{C}^{N \times N}$  be invertible. The  $(\mathbf{s}, \mathbf{M})^{\text{th}}$  **G-adjusted restricted isometry constant in levels**  $\delta_{s,M,G}$  of a matrix  $A \in \mathbb{C}^{m \times N}$  is the smallest  $\delta \geq 0$  such that

$$(1 - \delta)\|Gx\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta)\|Gx\|_2^2, \quad \forall x \in \Sigma_{s,M}. \quad (5.20)$$

If  $0 < \delta_{s,M,G} < 1$  then the matrix  $A$  is said to have the **G-adjusted restricted isometry property in levels** of order  $(\mathbf{s}, \mathbf{M})$ .

**Definition 35** (Robust Null Space Property in Levels). Let  $\{w_i\}_{i=1}^N \in \mathbb{R}^N$  be positive weights. A matrix  $A \in \mathbb{C}^{m \times N}$  has the **weighted robust null space property in levels** of order  $(\mathbf{s}, \mathbf{M})$  with constants  $0 < \rho < 1$  and  $\gamma > 0$  if

$$\|P_\Delta x\|_2 \leq \frac{\rho \|P_\Delta^\perp x\|_1}{\sqrt{|\Delta|_w}} + \gamma \|Ax\|_2, \quad (5.21)$$

for all  $x \in \mathbb{C}^N$  and  $\Delta \in D_{s,M}$ .

Similar to the robust Null Space Property (not in levels), the rNSP-L implies the NSP-L.

In this case, our matrix is given as

$$A = DFSW^{-1}, \quad (5.22)$$

where  $D$  is a variable density sampling mask,  $F$  represents the discrete Fourier transform,  $S$  applies the sensitivity maps and  $W^{-1}$  is the inverse discrete Wavelet transform. This was shown in [2, Eq. 4.6] to satisfy the RIP in levels given that the sampling mask satisfies certain properties. The number of samples required at each level to achieve the RIP in levels is

$$m_k \geq (N_k - N_{k-1}) \cdot \log(\epsilon^{-1}) \cdot \mu(A_k) \cdot s_k \cdot \log(N), \quad 1 \leq k \leq r. \quad (5.23)$$

**Theorem 5.1.** Let  $A \in \mathbb{C}^{m \times N}$  satisfy the Null Space Property of order  $s$ . Suppose a reconstruction method enforces strict data consistency, i.e.,  $Ax = b$ . Then any hallucination  $\tilde{x} = x_{\text{true}} - x_{\text{recon}} \in \text{Ker}(A)$  cannot be  $\mathbf{s}$ -sparse.

*Proof.* Since  $A(x_{\text{true}} - x_{\text{recon}}) = 0$ , the hallucination  $\tilde{x} \in \text{Ker}(A)$ . By the Null Space Property, any nonzero  $\mathbf{s}$ -sparse vector cannot lie in the null space. Hence  $\tilde{x}$  is not  $\mathbf{s}$ -sparse.  $\square$

**Theorem 5.2.** Let  $A \in \mathbb{C}^{m \times N}$  satisfy the Null Space Property in levels of order  $(\mathbf{s}, \mathbf{M})$ . Suppose a reconstruction method enforces strict data consistency, i.e.,  $Ax = b$ . Then any hallucination  $\tilde{x} = x_{\text{true}} - x_{\text{recon}} \in \text{Ker}(A)$  cannot be  $(\tilde{s}, \tilde{M})$  sparse for any  $(\tilde{s}, \tilde{M})$  such that  $\sum \tilde{s}_i \leq \sum s_i$ .

*Proof.* Suppose  $A$  satisfies the Null Space Property in levels of order  $(s, M)$ . Suppose we reconstruct an image  $\hat{x}$  enforcing that  $A\hat{x} = b$ , where  $b$  is some collected data. The hallucination  $\tilde{x}$  must live in the null space of  $A$ . If  $\tilde{x}$  is  $(\tilde{s}, \tilde{M})$ -sparse for any  $\tilde{s}$  such that  $\sum_i \tilde{s}_i < \sum_i s_i$ , choose  $\Delta_{\tilde{s}, \tilde{M}}$  such that

$$\|P_{\Delta_{\tilde{s}, \tilde{M}}} x\| > \|P_{\Delta_{\tilde{s}, \tilde{M}}}^\perp x\|_1$$

by selecting  $\Delta_{\tilde{s}, \tilde{M}}$  that corresponds to all non-zero elements. Hence the null space property in levels of order  $(\mathbf{s}, \mathbf{M})$  is violated for  $A$  and we have a contradiction. Therefore, any hallucination  $\tilde{x}$  can be no more than  $(\mathbf{s}, \mathbf{M})$ -sparse.  $\square$

## 5.6 Conclusion

In this manuscript we present empirical evidence that enforced data consistency yields a significant improvement to reconstruction quality in deep learning MRI reconstruction methods.

In this study, we introduced a novel method for enforcing data consistency in deep learning-based MRI reconstruction, addressing the critical challenge of maintaining fidelity to acquired k-space data while leveraging the power of neural networks to enhance image quality. Our approach integrates a data consistency layer within the network architecture, ensuring that reconstructed images align with the measured data while mitigating artifacts and noise amplification. Through experiments on MRI datasets, we demonstrated that our method achieves superior reconstruction quality on out-of-distribution data and mitigates the risk of hallucinated artifacts in the reconstruction.

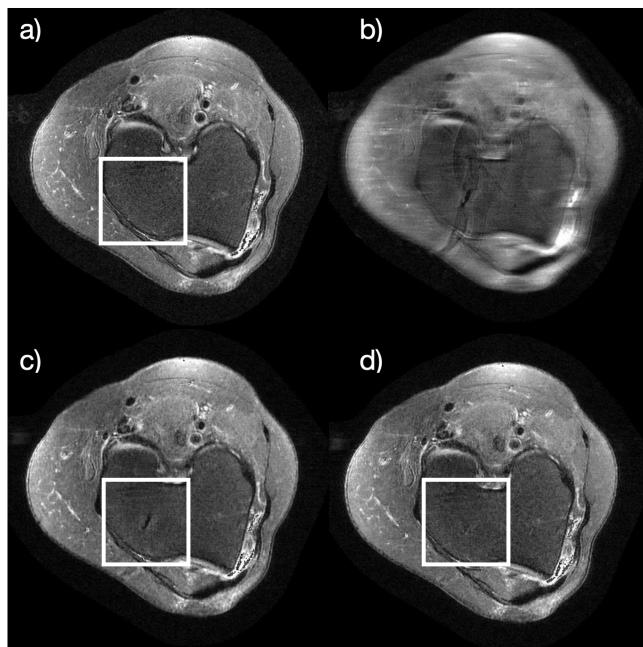


Figure 5.3: A reconstruction using the variational network of axial T2 knee data. a) shows the reconstruction with fully sampled data. b) shows the inverse Fourier transform of the zero-filled data. c) shows the reconstruction from the network trained without data consistency. d) shows the reconstruction from the network trained with data consistency. The white box highlights a medical anomaly produced in the reconstruction without data consistency that is corrected in the reconstruction with data consistency.

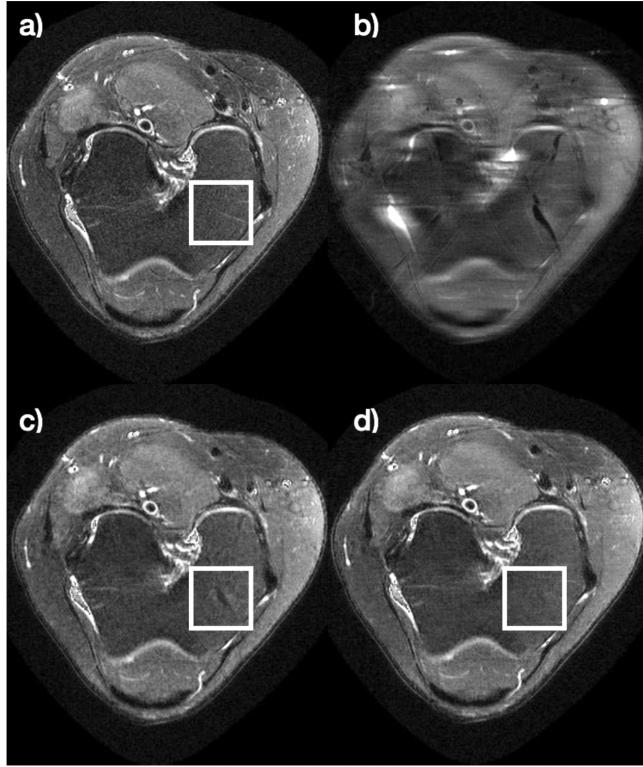


Figure 5.4: An example of a hallucination fixed by data consistency. All data is undersampled knee data using an unrolled variational network [66]. a) shows the fully sampled reconstruction with the area of the hallucination in the white box. b) shows the zero-filled reconstruction. c) shows the reconstruction from the network without data consistency, exhibiting a hallucination inside the white box. d) shows the reconstruction from the network trained with data consistency with the hallucination no longer present.

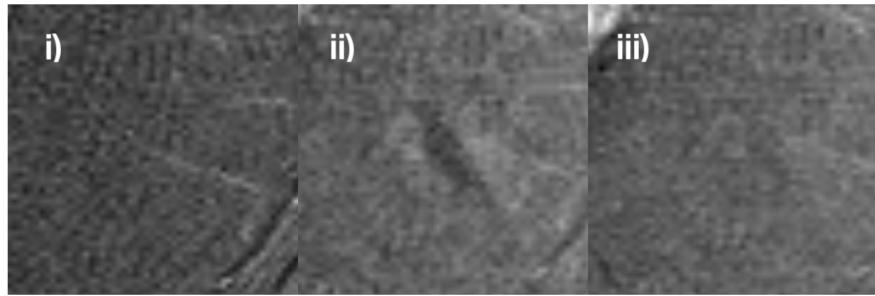


Figure 5.5: A zoomed in view of the white boxes from Fig. 3. i) shows the white box from the fully sampled reconstruction (box a from Fig. 3). ii) shows the white box from the reconstruction from the network without data consistency exhibiting the hallucination (box c from Fig. 3). iii) shows the white box from the reconstruction from the network with data consistency (box d from Fig. 3).

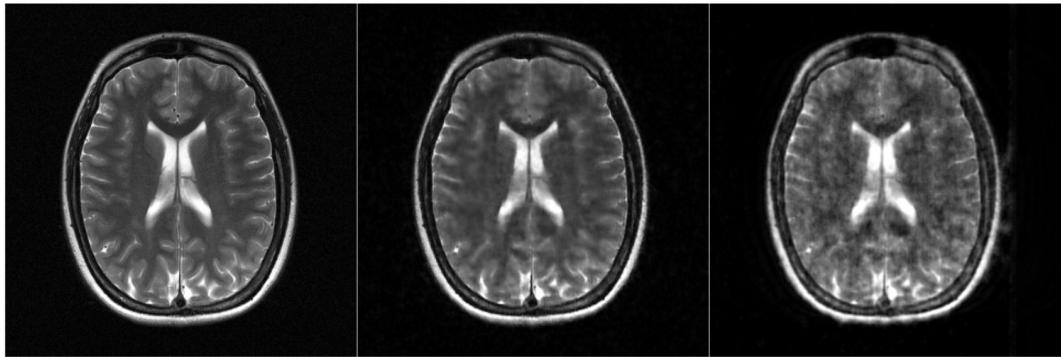


Figure 5.6: An example reconstruction from our novel unrolled network architecture, with and without data consistency. These reconstructions are made with a 10% sampling burden. On the left is the ground truth, fully sampled reconstruction. In the middle is the reconstruction from the network trained with data consistency. On the right is the reconstruction from the network trained without data consistency.

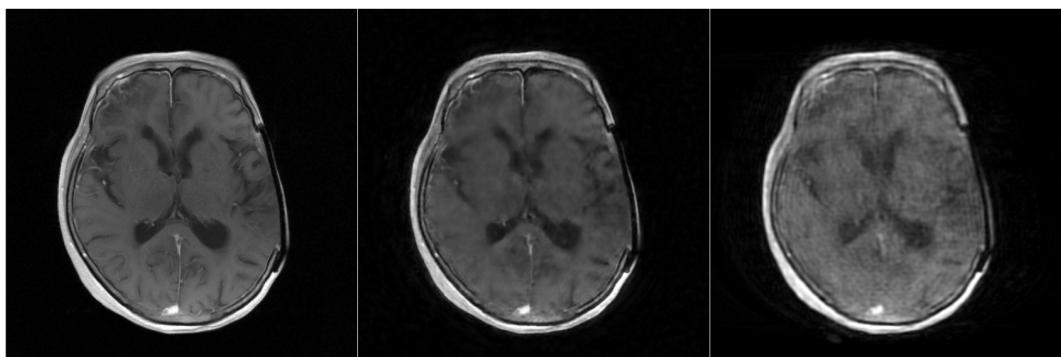


Figure 5.7: An example reconstruction from our novel unrolled network architecture, with and without data consistency. These reconstructions are made with a 7.5% sampling burden. On the left is the ground truth, fully sampled reconstruction. In the middle is the reconstruction from the network trained with data consistency. On the right is the reconstruction from the network trained without data consistency.

| Sample Rate | Metric | With DC (Median $\pm$ IQR)                                | Without DC (Median $\pm$ IQR)                             | p-value                   |
|-------------|--------|---|---|---------------------------|
| 30%         | SSIM   | 0.8964 $\pm$ 0.0621                                       | 0.8576 $\pm$ 0.1036                                       | 5.551 * 10 <sup>-14</sup> |
|             | PSNR   | 39.3468 $\pm$ 3.1115                                      | 33.6780 $\pm$ 1.8561                                      | 1.744 * 10 <sup>-16</sup> |
|             | MSE    | 5.1060 * 10 <sup>-6</sup> $\pm$ 6.9867 * 10 <sup>-6</sup> | 1.1029 * 10 <sup>-5</sup> $\pm$ 1.4153 * 10 <sup>-5</sup> | 7.779 * 10 <sup>-14</sup> |
| 25%         | SSIM   | 0.8901 $\pm$ 0.0734                                       | 0.8034 $\pm$ 0.1754                                       | 9.489 * 10 <sup>-16</sup> |
|             | PSNR   | 38.9310 $\pm$ 2.9701                                      | 32.6820 $\pm$ 1.2962                                      | 1.744 * 10 <sup>-16</sup> |
|             | MSE    | 4.6241 * 10 <sup>-6</sup> $\pm$ 6.0581 * 10 <sup>-6</sup> | 1.3974 * 10 <sup>-5</sup> $\pm$ 1.8335 * 10 <sup>-5</sup> | 1.744 * 10 <sup>-16</sup> |
| 20%         | SSIM   | 0.8603 $\pm$ 0.0896                                       | 0.8025 $\pm$ 0.1631                                       | 2.781 * 10 <sup>-8</sup>  |
|             | PSNR   | 36.3183 $\pm$ 3.6400                                      | 32.8076 $\pm$ 1.5744                                      | 6.674 * 10 <sup>-14</sup> |
|             | MSE    | 8.7200 * 10 <sup>-6</sup> $\pm$ 1.4889 * 10 <sup>-5</sup> | 1.2500 * 10 <sup>-5</sup> $\pm$ 1.6593 * 10 <sup>-5</sup> | 0.00056                   |
| 15%         | SSIM   | 0.8250 $\pm$ 0.1035                                       | 0.6870 $\pm$ 0.2225                                       | 2.781 * 10 <sup>-8</sup>  |
|             | PSNR   | 25.1529 $\pm$ 5.4100                                      | 31.6707 $\pm$ 1.2378                                      | 5.999 * 10 <sup>-10</sup> |
|             | MSE    | 1.3929 * 10 <sup>-5</sup> $\pm$ 3.1084 * 10 <sup>-5</sup> | 1.5399 * 10 <sup>-5</sup> $\pm$ 2.3989 * 10 <sup>-5</sup> | 0.8516                    |
| 10%         | SSIM   | 0.8171 $\pm$ 0.1291                                       | 0.7214 $\pm$ 0.2108                                       | 1.233 * 10 <sup>-5</sup>  |
|             | PSNR   | 33.9832 $\pm$ 4.7374                                      | 32.1588 $\pm$ 1.9665                                      | 0.07972                   |
|             | MSE    | 2.1488 * 10 <sup>-5</sup> $\pm$ 4.1942 * 10 <sup>-5</sup> | 1.2751 * 10 <sup>-5</sup> $\pm$ 1.8218 * 10 <sup>-5</sup> | 0.0019                    |
| 7.5%        | SSIM   | 0.8146 $\pm$ 0.1231                                       | 0.7543 $\pm$ 0.0.2049                                     | 5.995e $-07$              |
|             | PSNR   | 33.5464 $\pm$ 5.3073                                      | 30.7103 $\pm$ 1.4386                                      | 4.165 * 10 <sup>-5</sup>  |
|             | MSE    | 4.4233 * 10 <sup>-5</sup> $\pm$ 4.8958 * 10 <sup>-5</sup> | 2.0486 * 10 <sup>-5</sup> $\pm$ 2.9071 * 10 <sup>-5</sup> | 0.1823                    |

Table 5.4: Ablation study conducted using the unrolled network trained in a supervised learning fashion.

## Chapter 6

### Triangular Preconditioning

#### 6.1 Introduction

##### 6.1.1 Abstract

This chapter introduces a novel approach to monotone inclusion problems, developed in collaboration with Stephen Becker. We generalize the traditional preconditioned proximal point algorithm (PPA), which typically assumes a self-adjoint preconditioner—referred to here as the symmetric-PPA. In contrast, our framework allows for **asymmetric** preconditioners, expanding the design space. To the best of our knowledge, asymmetric preconditioners have no current representation in the literature.

If we specialize to a class of **triangular** preconditioners, we have an efficient algorithm for separable operators that we call **generalized backsubstitution**. Separable operators appear as the proximal operators for functions common in optimization objective functions, such as the  $\ell_1$  norm and some mixed matrix norms.

There is a tradeoff between the cost of finding a good preconditioner and the advantage it imparts. We wonder about problems where the cost of finding the preconditioner may be amortized over several use-cases. One example is single-coil MRI reconstruction, where the same sampling pattern would give rise to the same system matrix, changing only the right-hand side.

The preconditioner framework here applies to all optimization algorithms with a proximal operator. This is common in splitting algorithms for nonsmooth optimization. We analyze the

proximal point algorithm for simplicity, but the same framework applies to Douglas-Rachford splitting, proximal-gradient descent or forward-backward splitting, primal-dual hybrid gradient, and extensions of these algorithms (e.g., [32]).

### 6.1.2 Motivation

There is a deep connection between root-finding and optimization. If  $f \in \Gamma_0(\mathcal{H})$  and differentiable, then minimizing  $f$  is equivalent to finding a zero of the gradient (there was discussion of this in the background, reference equation (2.9)). If we relax the assumption that  $f$  is differentiable (and is instead **subdifferentiable**), then we want to find  $x \in \mathcal{H}$  such that

$$0 \in \partial f(x). \quad (6.1)$$

When  $f \in \Gamma_0(\mathcal{H})$ ,  $\partial f$  is a **maximally monotone operator** (see [9, Theorem 20.25]). Solving an equation of the form (6.1) is known as a **monotone inclusion** problem, and can naturally be seen as an extension of root-finding.

Solving monotone inclusion problems is an active area of research [82, 107, 33]. There is very clearly a strong connection between them and non-smooth optimization problems, which are very common in signal and image processing. Some of these methods have been discussed previously (e.g., proximal point, Douglas-Rachford splitting, proximal gradient descent; see Sec. 2.7). The idea behind this project investigated a novel class of preconditioners that could generally be applied to proximal algorithms.

### 6.1.3 Preconditioners

The use of preconditioners for proximal methods is not novel in and of itself. There has been work on preconditioned proximal point (PPPA) for at least the past 18 years [94]. These preconditioners are usually of the same class—that is, monotone and **self-adjoint**. Monotonicity of the preconditioner intuitively seems to be a prerequisite for convergence at all. Self-adjointness, however, does not. Self-adjointness certainly makes the convergence proofs **easier**, as it becomes

simple to show that the composition of a self-adjoint bounded linear operator and a maximally monotone operator is also maximally monotone in the Hilbert space endowed with the inner product induced by the linear operator. Therefore convergence of preconditioned proximal point in this case is no different than convergence of the proximal point algorithm broadly, with the caveat that the inner product changes.

The preconditioners that we propose and study here are novel in the sense that we will relax the assumption of self-adjointness; we study **lower-triangular** matrices. We will still enforce that they are monotone—for bounded linear operators, monotonicity is equivalent to being positive semi-definite. Here another note is warranted. The definition of positive definiteness often times **implicitly** contains the assumption that the underlying matrix is symmetric. We will not do that, and stick to the definition that  $A \in \mathcal{B}(\mathcal{H})$  is positive semi-definite if

$$x^*Ax \geq 0, \quad \forall x \in \mathcal{H}. \quad (6.2)$$

## 6.2 Theoretical Foundation

### 6.2.1 Classical Linear Iteration Analogy

We are motivated by the use of splitting in classical linear algebra. To begin, we want to solve the system

$$Ax = b, \quad (6.3)$$

for  $A \in \mathbb{R}^{n \times n}$ ,  $x \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^n$ . If we split  $A = M - N$ , this becomes

$$\begin{aligned} (M - N)x &= b \\ Mx &= Nx + b, \end{aligned} \quad (6.4)$$

which suggests the following fixed-point iteration

$$x_{k+1} = M^{-1}(Nx_k + b). \quad (6.5)$$

### 6.2.1.1 Jacobi and Gauss-Seidel

For a matrix  $A \in \mathbb{R}^{n \times n}$ , let  $D$  be the isolated diagonal part, and  $-L$  and  $-U$  the lower and upper triangular parts of  $A$ , respectively, so that  $A = D - L - U$ . The system of interest (6.3) is

$$(D - L - U)x = b. \quad (6.6)$$

Jacobi splitting turns this into

$$Dx = (L + U)x + b, \quad (6.7)$$

and (assuming  $D^{-1}$  exists),

$$x = D^{-1}(L + U)x + D^{-1}b. \quad (6.8)$$

This results in the **Jacobi iterative technique** [21]

$$x_{k+1} = D^{-1}(L + U)x_k + D^{-1}b. \quad (6.9)$$

The Gauss-Seidel method instead splits as

$$(D - L)x_{k+1} = Ux_k + b, \quad (6.10)$$

or

$$x_{k+1} = (D - L)^{-1}Ux_k + (D - L)^{-1}b, \quad (6.11)$$

where  $D - L$  is lower triangular so the solve is made simple by forward substitution.

### 6.2.1.2 Convergence Criterion

Let  $G = M^{-1}N$  from (6.5) be the iteration matrix. It is well known that  $\lim_{k \rightarrow \infty} G^k = 0$  if and only if the spectral radius of  $G$  is strictly less than 1, hence a sufficient condition for convergence of Eq. (6.5) is that

$$\|G\| < 1.$$

In particular, for the Jacobi iteration we have  $G = D^{-1}(L + U)$  so one way to guarantee  $\|G\| < 1$  in this case is for  $A$  to be **strictly diagonally dominant** [21]. This assumption is sufficient but

not necessary. Similarly, Gauss-Seidel is guaranteed to converge under suitable conditions, such as  $A$  being symmetric and positive-definite.

This mirrors our problem where we want to solve  $0 \in Tx$  via the preconditioned proximal point iteration

$$x_{k+1} = (V + T)^{-1} V x_k. \quad (6.12)$$

When  $V$  is lower triangular, we can solve  $(V + T)x_{k+1} = Vx_k$  efficiently via **generalized** forward substitution, assuming that  $T$  is separable.

### 6.2.2 Asymmetric-PPA

We develop the theory for the **non-self-adjoint** preconditioned proximal point algorithm here (or **asymmetric-PPA** for short), but will be able to re-use the key components later when we turn our attention to preconditioning proximal gradient descent and the Douglas-Rachford methods.

Let  $\mathcal{H}$  be a Hilbert space and  $T : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  be a maximally monotone set-valued operator. The problem we wish to solve is

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in Tx. \quad (6.13)$$

Let  $V \in \mathcal{B}(\mathcal{H})$  and add  $Vx$  to both sides of (6.13) to yield

$$Vx \in Vx + Tx \implies Vx \in (V + T)x. \quad (6.14)$$

If  $V$  is invertible:

$$0 \in Tx \iff x \in (\mathcal{I} + V^{-1}T)x. \quad (6.15)$$

However, the “naive” fixed-point iteration suggested by the above equation,  $x_{k+1} \in (\mathcal{I} + V^{-1}T)x_k$ , is not suitable, one reason being that  $T$  is set-valued so the iteration is not well-defined. Instead, we will find conditions that ensure  $V^{-1}T$  is maximally monotone, in which case the resolvent  $(\mathcal{I} + V^{-1}T)^{-1}$  is well-defined (i.e., single valued) and firmly non-expansive, hence we instead look at the problem

$$x = (\mathcal{I} + V^{-1}T)^{-1}x$$

leading to the iteration

$$x_{k+1} = (\mathcal{I} + V^{-1}T)^{-1} x_k. \quad (6.16)$$

Similarly, if  $(V + T)^{-1}$  exists everywhere and is single-valued, then converting Eq. (6.14) to  $x = (V + T)^{-1} Vx$  leads to the iteration

$$x_{k+1} = (V + T)^{-1} Vx_k. \quad (6.17)$$

For computational efficiency reasons, we will make the assumption that  $T$  is **separable**. We say an operator  $T$  is separable if its action on a vector  $x \in \mathcal{H}$  can be represented as the action of **subcomponents** of  $T$  on the coordinates of  $x$ :

$$Tx = \begin{bmatrix} T_1 x_1 \\ T_2 x_2 \\ \vdots \\ T_n x_n \end{bmatrix} \rightarrow T = \begin{bmatrix} T_1 & 0 & 0 & \dots \\ 0 & T_2 & 0 & \dots \\ & & \ddots & \\ & & 0 & T_n \end{bmatrix}. \quad (6.18)$$

### 6.2.3 Triangular matrices

We collect a few facts. For simplicity we work over the real field, and use the terms “symmetric” and “self-adjoint” synonymously. If a matrix  $V$  is lower triangular,

- (1) its inverse is also lower triangular;
- (2) its determinant is the product of the diagonal entries;
- (3) the eigenvalues are just the diagonal entries;

and obviously analogous statements hold for upper triangular matrices. When  $V$  is triangular and symmetric, it must be diagonal; since diagonal preconditioning has already been extensively studied, we do not focus on this case, so we generally have in mind a non-diagonal matrix.

When  $V$  is not symmetric, then positive definiteness is no longer equivalent to all eigenvalues

being positive. As a simple counter example, let

$$A = \begin{bmatrix} 1 & 0 \\ -10 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (6.19)$$

The eigenvalues of  $A$  are the diagonal entries, so 1 with multiplicity 2. However  $b^*Ab = -8$  and so  $A$  is not positive definite.

We can consider the symmetric and skew-symmetric portions of  $A$ :

$$A_{\text{sym}} = \frac{1}{2}(A + A^*) = A_{\text{sym}}^*, \quad A_{\text{skew}} = \frac{1}{2}(A - A^*) = -A_{\text{skew}}^*. \quad (6.20)$$

We have  $A = A_{\text{sym}} + A_{\text{skew}}$ . We can show that  $x^*A_{\text{skew}}x = 0$  for all  $x$ , so then

$$\begin{aligned} x^*Ax &= x^*(A_{\text{sym}} + A_{\text{skew}})x \\ &= x^*A_{\text{sym}}x. \end{aligned} \quad (6.21)$$

Since  $A_{\text{sym}}$  is symmetric, then its positive definiteness is related to its eigenvalues. As we just saw, its positive definiteness is exactly related to the positive definiteness of  $A$ . This becomes the easiest way to check if nonsymmetric  $A$  is positive definite—form  $A_{\text{sym}} = \frac{1}{2}(A + A^*)$  and check its eigenvalues.

### 6.3 Monotonicity and Convergence Considerations

We are focused on the iterative method

$$x_{k+1} = (\mathcal{I} + V^{-1}T)^{-1}x_k. \quad (6.22)$$

Our goal is to find the broadest possible set of conditions on  $V$  and  $T$  that will guarantee this converges, and in particular, under what conditions  $V^{-1}T$  is a maximal monotone operator, since then  $(\mathcal{I} + V^{-1}T)^{-1}$  is full-domain and firmly non-expansive, so in finite dimensions the iteration will converge to a zero of  $T$  (if one exists) via Groetsch's theorem [9, Thm. 5.15]. We do not yet have a theory that is provably as broad as possible, but we show steps in that direction, and collect some sufficient conditions. Clearly, one sufficient condition is if  $V = \mathcal{I}$  and  $T$  is maximally

monotone. Slightly more generally, this is equivalent to the standard proximal point algorithm if  $V$  is self-adjoint and positive definite since this is just changing the inner product on the Hilbert space. The main difficulty in our case is that most theory (e.g., from the standard reference on monotone operators in Hilbert space, [9], which we henceforth refer to as B&C) assumes  $V$  to be symmetric positive definite. When  $V$  is asymmetric, there's no inner product that induces it.

### 6.3.1 Useful Theorems

There are some useful theorems in B&C that relate to the maximal monotonicity of compositions of maximally monotone operators and bounded linear operators. The usual restrictions placed on the bounded linear operator are it being monotone and self-adjoint. In a finite-dimensional Hilbert space, a bounded linear operator being monotone is equivalent to it being positive semidefinite. Recall that the **resolvent** of a set-valued operator  $A : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  is the operator  $J_A = (\mathcal{I} + A)^{-1}$ .

**Theorem 6.1** (B&C Proposition 23.25 [9]). Let  $\mathcal{K}$  be a real Hilbert space, suppose that  $L \in \mathcal{B}(\mathcal{H}, \mathcal{K})$  is such that  $LL^*$  is invertible, let  $A : \mathcal{K} \rightarrow 2^{\mathcal{K}}$  be maximally monotone, and set  $B = L^*AL$ . Then the following hold:

- (1)  $B : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  is maximally monotone.
- (2)  $J_B = \mathcal{I} - L^* \circ (LL^* + A^{-1})^{-1} \circ L$ .
- (3) Suppose that  $LL^* = \mu\mathcal{I}$  for some  $\mu > 0$ . Then  $J_B = \mathcal{I} - L^* \circ^\mu A \circ L$ .

**Theorem 6.2** (B&C Proposition 20.24 [9]). Let  $A : \mathcal{H} \rightarrow 2^{\mathcal{H}}$  be maximally monotone, let  $U \in \mathcal{B}(\mathcal{H})$  be a self-adjoint and strongly monotone operator, and let  $\mathcal{K}$  be the real Hilbert space obtained by endowing  $\mathcal{H}$  with the scalar product  $\langle x, y \rangle_{\mathcal{K}} = \langle x, U^{-1}y \rangle$ . Then  $UA : \mathcal{K} \rightarrow 2^{\mathcal{K}}$  is maximally monotone.

The proof for this theorem breaks down if  $U$  is not self-adjoint, as it no longer defines a new inner product (lack of symmetry violates properties of the inner product). However this will

be a starting point for some of our methods of trying to directly prove that  $V^{-1}T$  is maximally monotone.

One might hope that if  $V^{-1}$  has only nonnegative entries, then for a separable maximal monotone operator  $T$ , each component of  $V^{-1}T$  would be a nonnegative sum of maximal monotone operators and hence maximal monotone (in analogy to the fact that the nonnegative sum of convex functions is still convex). One avenue we pursued was assuming that  $V^{-1}$  has only positive entries. Our hope was, with a separable  $T$ , the “matrix-matrix product” of  $V^{-1}T$  would only contain maximally monotone operators. While the positive combination of two monotone operators is monotone, it is not guaranteed to be maximally monotone. This leads to one major open question:

**Open Question 6.3.1.** Under what conditions is  $V^{-1}T$  maximally monotone, when  $T$  is maximally monotone and  $V$  is lower-triangular, positive definite, but not self-adjoint? What role does separability of  $T$  or entrywise nonnegativity of  $V^{-1}$  play?

In the absence of a general theoretic guarantee, we shift our focus in the following sections to special cases and numerical behavior. In particular, we analyze the separable setting and matrix classes where monotonicity may hold empirically while investigating the convergence behavior of the method applied to proximal algorithms.

## 6.4 The Proximal Point Method

We now study the convergence of the iteration

$$x_{k+1} = (\mathcal{I} + V^{-1}T)^{-1}x_k$$

for a maximally monotone operator  $T$ , where  $V \in \mathbb{R}^{n \times n}$  is lower-triangular and invertible. This is a natural generalization of the standard **proximal point method (PPM)**, which uses the resolvent  $(\mathcal{I} + T)^{-1}$ . Classical convergence theory for PPM applies in a Hilbert space  $\mathcal{H}$ , and hinges on the fact that the resolvent of a maximally monotone operator is **firmly nonexpansive** in the associated norm. In our setting, however, the use of a non-symmetric, possibly non-normal matrix  $V$  introduces significant theoretical difficulties.

### 6.4.1 Classical Setting

If  $V \succ 0$  and is self-adjoint, then we can define a new inner product via

$$\langle x, y \rangle_V := \langle Vx, y \rangle.$$

This induces a Hilbert space norm  $\|x\|_V^2 := \langle x, x \rangle_V$ , and in this space the operator  $V^{-1}T$  is still maximally monotone whenever  $T$  is. The corresponding resolvent  $(\mathcal{I} + V^{-1}T)^{-1}$  is then **firmly nonexpansive** with respect to the  $V$ -norm, and the iteration converges by standard arguments (see [9, Prop. 23.25]).

### 6.4.2 Our Setting: $V$ Triangular, Non-Symmetric

We break from the classical setting by allowing  $V$  to be lower-triangular and **not** self-adjoint. The reasons are both theoretical and practical: triangular matrices are easy to invert, allow sequential coordinate updates (akin to Gauss–Seidel), and are common in large-scale approximations like incomplete LU.

However, without symmetry,  $\langle x, y \rangle_V := \langle x, Vy \rangle$  is no longer symmetric and thus fails to define an inner product. Consequently, the usual machinery of Hilbert space monotonicity no longer applies, and it is unclear whether  $V^{-1}T$  is monotone, let alone maximally monotone.

**Open Question 6.4.1.** Let  $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$  be maximally monotone, and  $V \in \mathbb{R}^{n \times n}$  be invertible, lower-triangular, and positive definite but not self-adjoint. Under what conditions is the composition  $V^{-1}T$  also maximally monotone? Does the answer change if  $T$  is separable (e.g.,  $T = \partial f$  with  $f(x) = \sum_i f_i(x_i)$ ) or if  $V^{-1}$  is entrywise nonnegative?

### 6.4.3 Sufficient Conditions (Partial Results)

We are currently unable to give a full proof that  $V^{-1}T$  is maximally monotone under our assumptions. However, we make the following observations:

- If  $V \succ 0$  and  $V = V^T$ , then standard results apply: the induced inner product  $\langle x, y \rangle_V$  defines a Hilbert space, and  $V^{-1}T$  is maximally monotone.

- If  $V$  is diagonal with positive entries, it is self-adjoint, and again standard results apply.
- If  $V^{-1} \geq 0$  (entrywise) and  $T$  is separable, then each component of  $V^{-1}T$  applies a **positive combination** of monotone maps. This suggests that monotonicity might be preserved—though we have not yet proven it.

We presently adopt the following assumption:

**Assumption 6.4.1** (Working Assumption). For triangular, positive definite  $V$  with entrywise nonnegative inverse  $V^{-1}$ , and separable monotone operator  $T$ , the composition  $V^{-1}T$  behaves sufficiently like a monotone operator to guarantee convergence of the proximal point iteration.

This assumption is supported empirically in §?? and aligns with intuition from coordinate descent and Gauss–Seidel methods.

#### 6.4.4 On Firm Nonexpansiveness

One possible route to proving convergence, even without monotonicity of  $V^{-1}T$ , is to show that the resolvent  $(\mathcal{I} + V^{-1}T)^{-1}$  is **firmly nonexpansive** in some norm. For instance, suppose

$$\|2(\mathcal{I} + V^{-1}T)^{-1} - \mathcal{I}\| < 1.$$

This would ensure that the iteration is a contraction mapping and therefore convergent by the Banach fixed-point theorem. However, identifying a norm where this inequality holds (and proving it without monotonicity) remains an open challenge.

#### 6.4.5 Summary

In summary, convergence of the iteration

$$x_{k+1} = (\mathcal{I} + V^{-1}T)^{-1}x_k$$

is guaranteed in the classical case when  $V = V^* \succ 0$ , but remains theoretically open when  $V$  is triangular and non-symmetric. We believe that convergence may still occur in many practical cases, particularly when  $T$  is separable and  $V^{-1} \geq 0$ , but rigorous justification is currently lacking.

## 6.5 Forward Backward

Forward-backward splitting (or proximal-gradient descent) solves problems of the form

$$\underset{x \in \mathcal{H}}{\text{minimize}} f(x) + g(x), \quad (6.23)$$

where  $f$  is smooth and  $g$  has a simple proximal operator. In this case, the iteration becomes

$$x_{k+1} = (\mathcal{I} + \lambda_k \partial g)^{-1} (\mathcal{I} - \lambda_k \nabla f) x_k. \quad (6.24)$$

Applying our preconditioner  $V^{-1}$  would yield

$$x_{k+1} = \left( \mathcal{I} + V^{-1} \partial g \right)^{-1} \left( \mathcal{I} - V^{-1} \nabla f \right) x_k. \quad (6.25)$$

Standard preconditioning for the gradient descent step would try to approximate  $V \approx \nabla^2 f(x_k)$ , similar to Newton's method. This is not generally possible since  $V$  would be fixed and unchanging with  $x_k$ , and because of the assumptions of structure on  $V$ .

This is a bit subtle: when is  $V^{-1} \nabla f$  cocoercive?

Special case: for a quadratic, the operator is  $V^{-1} H$ , so it's single-valued and just a matrix.

### 6.5.1 Forward-backward algorithm

**Theorem 6.3** (Thm. 26.14 PLC, simplified to take  $\gamma = \beta$  and  $\lambda_k = 1$ ). Let  $A$  be maximally monotone and  $B$  be  $\beta$ -cocoercive, and assume there exists  $\mathbf{x}$  with  $0 \in (A + B)\mathbf{x}$ . Initialize  $\mathbf{x}_0$  and iterate for  $k = 0, 1, 2, \dots$

$$\mathbf{y}_k = \mathbf{x}_k - \beta B \mathbf{x}_k$$

$$\mathbf{x}_{k+1} = J_{\beta A}(\mathbf{y}_k) \stackrel{\text{def}}{=} (I + \beta A)^{-1} \mathbf{y}_k.$$

Then  $\mathbf{x}_k$  converges to a zero of  $A + B$ .

## 6.6 Implementation Details

### 6.6.1 Solving Eq. (6.22) when $V$ is triangular

With our assumption that  $V$  is lower triangular and  $T$  is separable, we can write (6.22) as

$$\left( \begin{bmatrix} v_{11} & 0 & 0 & \dots \\ v_{21} & v_{22} & 0 & \dots \\ v_{31} & v_{32} & v_{33} & \dots \\ \vdots & & & \end{bmatrix} + \begin{bmatrix} T_1 & 0 & 0 & \dots \\ 0 & T_2 & 0 & \dots \\ 0 & 0 & T_3 & \dots \\ \vdots & & & \end{bmatrix} \right) \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} v_{11} & 0 & 0 & \dots \\ v_{21} & v_{22} & 0 & \dots \\ v_{31} & v_{32} & v_{33} & \dots \\ \vdots & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (6.26)$$

Under the assumption that we can easily evaluate the resolvent of any of the **individual components**  $T_i$ , solving the first equation for  $y_1$  becomes

$$\begin{aligned} (v_{11} + T_1)y_1 &= v_{11}x_1 \\ (\mathcal{I} + v_{11}^{-1}T_1)y_1 &= x_1 \\ y_1 &= (\mathcal{I} + v_{11}^{-1}T_1)^{-1}x_1 \\ &= \text{prox}_{v_{11}^{-1}f_1}x_1. \end{aligned} \quad (6.27)$$

Solving the second equation now for  $y_2$ :

$$\begin{aligned} v_{21}y_1 + (v_{22} + T_2)y_2 &= v_{21}x_1 + v_{22}x_2 \\ (\mathcal{I} + v_{22}^{-1}T_1)y_2 &= v_{22}^{-1}(v_{21}x_1 + v_{22}x_2 - v_{21}y_1) \\ y_2 &= (\mathcal{I} + v_{22}^{-1}T_1)^{-1}\left(v_{22}^{-1}(v_{21}x_1 + v_{22}x_2 - v_{21}y_1)\right) \\ &= \text{prox}_{v_{22}^{-1}f_2}\left(v_{22}^{-1}(v_{21}x_1 + v_{22}x_2 - v_{21}y_1)\right) \end{aligned} \quad (6.28)$$

This is what we call **generalized forward substitution**. We continue this process on until we have completely solved for  $y$ . There are some numerical stability concerns here that should be investigated further.

## 6.7 Choosing A Good $V$

### 6.7.1 Selecting for $V^{-1}T$ To Be Monotone

One avenue we attempted is to choose  $V^{-1}$  directly so it was monotone from the definition.

For notational simplicity let  $L = V^{-1}$ . From the definition of left multiplication for operators (see, e.g., (2.14)), we have

$$LTx = \{Ly \mid y \in Tx\}. \quad (6.29)$$

Assume  $T$  is monotone, so

$$\langle x - y, x' - y' \rangle \geq 0 \quad \forall (x, x'), (y, y') \in \text{gr } T.$$

For  $LT$  to be monotone, then, we need

$$\langle x - y, Lx' - Ly' \rangle \geq 0 \quad \forall (x, x'), (y, y') \in \text{gr } T.$$

We simplify this slightly by letting  $y, y' = 0$ , so we need

$$\langle x, Lx' \rangle \geq 0 \quad \forall (x, x') \in \text{gr } T.$$

When  $L$  is symmetric and positive definite, the usual way to go about this is to use the inner product defined by  $L^{-1}$ :

$$\langle x, y \rangle_{L^{-1}} = \langle x, L^{-1}y \rangle. \quad (6.30)$$

Letting  $(x, x') \in \text{gr } T$ , then

$$\begin{aligned} \langle x, Lx' \rangle_{L^{-1}} &= \langle x, L^{-1}Lx' \rangle \\ &= \langle x, x' \rangle \geq 0, \end{aligned}$$

by definition as  $T$  is monotone. Hence operator  $LT$  is monotone in the new Hilbert space  $\mathcal{K}$  which is  $\mathcal{H}$  endowed with this new inner product from  $L^{-1}$ . This approach will not work for us since our operator  $L$  is not self-adjoint, and therefore does **not** define a new inner product.

One option we chose to pursue here is to define the inner product in terms of the **symmetric** part of  $L$ , since we assume  $L$  is monotone (read: positive semidefinite in the way defined in (6.2)).

The symmetric part of  $L$  is defined

$$L_{\text{sym}} = \frac{1}{2} (L + L^T). \quad (6.31)$$

We drop the  $\frac{1}{2}$  and define an inner product in terms of  $L_{\text{sym}}$  as

$$\langle x, y \rangle_{L_{\text{sym}}^{-1}} = \langle x, L_{\text{sym}}^{-1} y \rangle = \langle x, (L + L^T)^{-1} y \rangle. \quad (6.32)$$

We use the **Sherman-Morrison-Woodbury** matrix inversion lemma [54] to write

$$\begin{aligned} (L + L^T)^{-1} &= L^{-1} - (\mathcal{I} + L^{-T} L)^{-1} L^{-1} \\ &= (\mathcal{I} + L^{-1} L^T)^{-1} L^{-1}. \end{aligned} \quad (6.33)$$

where

$$L^{-T} = (L^{-1})^T = (L^T)^{-1}.$$

Now using the inner product defined in (6.32) and letting  $(x, x') \in \text{gr } T$  we want to show that

$$\begin{aligned} \langle x, Lx' \rangle_{L_{\text{sym}}^{-1}} &\geq 0 \\ \langle x, (\mathcal{I} + L^{-1} L^T)^{-1} L^{-1} Lx' \rangle &\geq 0 \\ \langle x, (\mathcal{I} + L^{-1} L^T)^{-1} x' \rangle &\geq 0. \end{aligned} \quad (6.34)$$

Showing that  $LT$  is monotone in the Hilbert space endowed with the  $L_{\text{sym}}^{-1}$  inner product then boils down to showing that, if  $\langle x, y \rangle \geq 0$ , then

$$\langle x, (\mathcal{I} + L^{-1} L^T)^{-1} y \rangle \geq 0 \iff \langle x, (\mathcal{I} + VV^{-T})^{-1} y \rangle \geq 0. \quad (6.35)$$

### 6.7.2 Desiderata for selecting $V^{-1}$

If we are in the forward-backward setup, then one approach is that we want this so that  $V^{-1}\nabla^2 f(x_k) \approx \mathcal{I}$ , so that we are as close as possible to Newton's method. For now, suppose  $\nabla^2 f(x_k) = H$  is a constant (i.e.,  $f$  is quadratic), so we want  $V^{-1}H \approx \mathcal{I}$ . One relaxed criteria is to ask for a good condition number  $\kappa(V^{-1}H)$ , as close to 1 as possible.

Something similar to this was done in Eq. 19 in “Preconditioning in Fast Dual Gradient Methods” by Giselsson and Boyd 2014 [52], and they refer to section 3.1 of **Linear Matrix**

**Inequalities in System and Control Theory** by Boyd, El Ghaoui, Feron, and Balakrishnan [18] for details. One potential avenue is that if all the matrices are **positive definite**, then the condition number (ratio of largest to smallest singular value) is just the ratio of largest to smallest **eigenvalue**, and that can be solved as a linear matrix inequality (LMI) or semi-definite program (SDP). So the idea would be that if we know  $H$ , we can find a good preconditioner  $V^{-1}$  by solving an LMI. This means that we can also add on side-constraints, such as the element-wise non-negativity (and the triangular structure), since those are convex.

## 6.8 Summary and Outlook

This novel preconditioner has several open questions but exciting areas of research. The proximal point algorithm was the starting point but has limited applications. The same idea could be applied to many proximal splitting algorithms, such as Douglas-Rachford splitting, forward-backward splitting, primal-dual hybrid gradient, and more, which are all broadly applicable methods to nonsmooth optimization problems.

## Bibliography

- [1] Ben Adcock and Anders C Hansen. Compressive imaging: structure, sampling, learning. Cambridge University Press, 2021.
- [2] Ben Adcock, Anders C Hansen, Clarice Poon, and Bogdan Roman. Breaking the coherence barrier: A new theory for compressed sensing. In Forum of mathematics, sigma, volume 5, page e4. Cambridge University Press, 2017.
- [3] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. Advances in neural information processing systems, 32, 2019.
- [4] Noah Amsel, Tyler Chen, Anne Greenbaum, Cameron Musco, and Chris Musco. Nearly optimal approximation of matrix functions by the lanczos method, 2024.
- [5] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1):1–3, 1966.
- [6] Corey A Baron, Nicholas Dwork, John M Pauly, and Dwight G Nishimura. Rapid compressed sensing reconstruction of 3D non-cartesian MRI. Magnetic resonance in medicine, 79(5):2685–2692, 2018.
- [7] Alexander Bastounis and Anders C Hansen. On the absence of uniform recovery in many real-world applications of compressed sensing and the restricted isometry property and nullspace property in levels. SIAM Journal on Imaging Sciences, 10(1):335–371, 2017.
- [8] Heinz H Bauschke and Patrick L Combettes. The baillon-haddad theorem revisited. arXiv preprint arXiv:0906.0807, 2009.
- [9] Heinz H Bauschke and Patrick L Combettes. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, 2011.
- [10] Heinz H Bauschke and Walaa M Moursi. The douglas–rachford algorithm for two (not necessarily intersecting) affine subspaces. SIAM Journal on Optimization, 26(2):968–985, 2016.
- [11] PJ Beatty, AC Brau, S Chang, S Joshi, CR Michelich, E Bayram, TE Nelson, RJ Herfkens, and JH Brittain. A method for autocalibrating 2D accelerated volumetric parallel imaging with clinically practical reconstruction times. In Proceedings of the International Society for Magnetic Resonance in Medicine, volume 15, page 1749, 2007.

- [12] Amir Beck. *First-order methods in optimization*. SIAM, 2017.
- [13] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [14] Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.
- [15] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.
- [16] Martin Blaimer, Felix Breuer, Matthias Mueller, Robin M Heidemann, Mark A Griswold, and Peter M Jakob. SMASH, SENSE, PILS, GRAPPA: how to choose the optimal method. *Topics in Magnetic Resonance Imaging*, 15(4):223–236, 2004.
- [17] Jelena Bock, Johannes Töger, Sebastian Bidhult, Karin Markenroth Bloch, Per Arvidsson, Mikael Kanski, Håkan Arheden, Frederik Testud, Andreas Greiser, Einar Heiberg, et al. Validation and reproducibility of cardiovascular 4D-flow MRI from two vendors using  $2 \times 2$  parallel imaging acceleration in pulsatile flow phantom and in vivo with and without respiratory gating. *Acta Radiologica*, 60(3):327–337, 2019.
- [18] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataraman Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [19] AC Brau, Philip J Beatty, Stefan Skare, and Roland Bammer. Efficient computation of autocalibrating parallel imaging reconstruction. In *Proceedings of the Annual Meeting of ISMRM*, page 2462, 2006.
- [20] Felix A Breuer, Stephan AR Kannengiesser, Martin Blaimer, Nicole Seiberlich, Peter M Jakob, and Mark A Griswold. General formulation for quantitative G-factor calculation in GRAPPA reconstructions. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 62(3):739–746, 2009.
- [21] RL Burden and JD Faires. Numerical analysis, ninth edit. Boston,: Richard Stratton, 2011.
- [22] Mark Bydder and Matthew D Robson. Partial fourier partially parallel imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 53(6):1393–1401, 2005.
- [23] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.
- [24] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.
- [25] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20:89–97, 2004.
- [26] Antonin Chambolle, Ronald A De Vore, Nam-Yong Lee, and Bradley J Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on image processing*, 7(3):319–335, 1998.

- [27] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40:120–145, 2011.
- [28] Jing Cheng, Zhuo-Xu Cui, Wenqi Huang, Ziwen Ke, Leslie Ying, Haifeng Wang, Yanjie Zhu, and Dong Liang. Learning data consistency and its application to dynamic mr imaging. *IEEE transactions on medical imaging*, 40(11):3140–3153, 2021.
- [29] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212, 2011.
- [30] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale modeling & simulation*, 4(4):1168–1200, 2005.
- [31] Laurent Condat. A direct algorithm for 1-d total variation denoising. *IEEE Signal Processing Letters*, 20(11):1054–1057, 2013.
- [32] Laurent Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of optimization theory and applications*, 158(2):460–479, 2013.
- [33] Laurent Condat, Daichi Kitahara, Andrés Contreras, and Akira Hirabayashi. Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists. *SIAM Review*, 65(2):375–435, 2023.
- [34] Amir Dembo, Thomas M Cover, and Joy A Thomas. Information theoretic inequalities. *IEEE Transactions on Information theory*, 37(6):1501–1518, 1991.
- [35] Anagha Deshmane, Vikas Gulani, Mark A Griswold, and Nicole Seiberlich. Parallel MR imaging. *Journal of Magnetic Resonance Imaging*, 36(1):55–72, 2012.
- [36] Mariya Doneva, P Börnert, Holger Eggers, and Alfred Mertins. Partial Fourier compressed sensing. *Proceedings of the Annual Meeting of the ISMRM*, 18, 2010.
- [37] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.
- [38] David L Donoho and Philip B Stark. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics*, 49(3):906–931, 1989.
- [39] Nicholas Dwork, Corey A Baron, Ethan MI Johnson, Daniel O’Connor, John M Pauly, and Peder EZ Larson. Fast variable density poisson-disc sample generation with directional variation for compressed sensing in MRI. *Magnetic Resonance Imaging*, 77:186–193, 2021.
- [40] Nicholas Dwork and Peder EZ Larson. Utilizing the structure of a redundant dictionary comprised of wavelets and curvelets with compressed sensing. *Journal of Electronic Imaging*, 31(6):063043, 2022.
- [41] Nicholas Dwork, Daniel O’Connor, Corey A Baron, Ethan MI Johnson, Adam B Kerr, John M Pauly, and Peder EZ Larson. Utilizing the wavelet transform’s structure in compressed sensing. *Signal, image and video processing*, 15:1407–1414, 2021.

- [42] Nicholas Dwork, Daniel O'Connor, Corey A Baron, Ethan MI Johnson, Adam B Kerr, John M Pauly, and Peder EZ Larson. Utilizing the wavelet transform's structure in compressed sensing. *Signal, image and video processing*, 15:1407–1414, 2021.
- [43] Jonathan Eckstein and Dimitri P Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical programming*, 55:293–318, 1992.
- [44] Charles L Epstein. *Introduction to the mathematics of medical imaging*. SIAM, 2007.
- [45] Ernie Esser, Xiaoqun Zhang, and Tony F Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.
- [46] Brecht Evens, Pieter Pas, Puya Latafat, and Panagiotis Patrinos. Convergence of the preconditioned proximal point method and douglas—rachford splitting in the absence of monotonicity. *Mathematical Programming*, pages 1–55, 2025.
- [47] Jeffrey A Fessler. Model-based image reconstruction for MRI. *Signal processing magazine*, 27(4):81–89, 2010.
- [48] Jeffrey A Fessler. Optimization methods for magnetic resonance image reconstruction: Key models and optimization algorithms. *IEEE signal processing magazine*, 37(1):33–40, 2020.
- [49] Dominique Franson, James Ahad, Yuchi Liu, Alexander Fyrdahl, William Truesdell, Jesse Hamilton, and Nicole Seiberlich. Self-calibrated through-time spiral GRAPPA for real-time, free-breathing evaluation of left ventricular function. *Magnetic Resonance in Medicine*, 2022.
- [50] Vahid Ghodrati, Jiaxin Shao, Mark Bydder, Ziwu Zhou, Wotao Yin, Kim-Lien Nguyen, Yingli Yang, and Peng Hu. MR image reconstruction using deep learning: evaluation of network structure and loss functions. *Quantitative imaging in medicine and surgery*, 9(9):1516, 2019.
- [51] Giulio Giovannetti, Luigi Landini, Maria Filomena Santarelli, and Vincenzo Positano. A fast and accurate simulator for the design of birdcage coils in MRI. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 15(1):36–44, 2002.
- [52] Pontus Giselsson and Stephen Boyd. Preconditioning in fast dual gradient methods. In *53rd IEEE Conference on Decision and Control*, pages 5040–5045. IEEE, 2014.
- [53] Pontus Giselsson, Mattias Fält, and Stephen Boyd. Line search for averaged operator iteration. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1015–1022. IEEE, 2016.
- [54] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- [55] Mark A Griswold, Martin Blaimer, Felix Breuer, Robin M Heidemann, Matthias Mueller, and Peter M Jakob. Parallel magnetic resonance imaging using the GRAPPA operator formalism. *Magnetic resonance in medicine*, 54(6):1553–1556, 2005.
- [56] Mark A Griswold, Peter M Jakob, Robin M Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase. Generalized autocalibrating partially parallel acquisitions (grappa). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 47(6):1202–1210, 2002.

- [57] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv*, 1804.09028, 2018.
- [58] Justin P. Haldar and Kawin Setsompop. Linear predictability in magnetic resonance imaging reconstruction: Leveraging shift-invariant fourier structure for faster and better imaging. *Signal Processing Magazine*, 37(1):69–82, 2020.
- [59] Kerstin Hammernik, Jo Schlemper, Chen Qin, Jinming Duan, Ronald M Summers, and Daniel Rueckert.  $\Sigma$ -net: Systematic evaluation of iterative deep neural networks for fast parallel MR image reconstruction. *arXiv*, 1912.09278, 2019.
- [60] Kerstin Hammernik, Jo Schlemper, Chen Qin, Jinming Duan, Ronald M Summers, and Daniel Rueckert. Systematic evaluation of iterative deep neural networks for fast parallel MRI reconstruction with sensitivity-weighted coil combination. *Magnetic Resonance in Medicine*, 86(4):1859–1872, 2021.
- [61] Bingsheng He and Xiaoming Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences*, 5(1):119–149, 2012.
- [62] Reinhard Heckel, Mathews Jacob, Akshay Chaudhari, Or Perlman, and Efrat Shimron. Deep learning for accelerated and robust mri reconstruction. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 37(3):335–368, 2024.
- [63] John K Hunter and Bruno Nachtergael. *Applied analysis*. World Scientific, 2001.
- [64] Chang Min Hyun, Hwa Pyung Kim, Sung Min Lee, Sungchul Lee, and Jin Keun Seo. Deep learning for undersampled MRI reconstruction. *Physics in Medicine & Biology*, 63(13):135007, 2018.
- [65] Peter M Jakob, Mark A Grisowld, Robert R Edelman, and Daniel K Sodickson. Auto-smash: a self-calibrating technique for smash imaging. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 7:42–54, 1998.
- [66] Hammernik Kerstin, Klatzer Teresa, Kobler Erich, Recht Michael P., Sodickson Daniel K., Pock Thomas, and Knoll Florian. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic Resonance in Medicine*, 79(6):3055–3071.
- [67] Florian Knoll, Kerstin Hammernik, Chi Zhang, Steen Moeller, Thomas Pock, Daniel K Sodickson, and Mehmet Akcakaya. Deep-learning methods for parallel magnetic resonance imaging reconstruction: A survey of the current approaches, trends, and issues. *IEEE signal processing magazine*, 37(1):128–140, 2020.
- [68] Florian Knoll, Tullie Murrell, Anuroop Sriram, Nafissa Yakubova, Jure Zbontar, Michael Rabbat, Aaron Defazio, Matthew J Muckley, Daniel K Sodickson, C Lawrence Zitnick, et al. Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge. *Magnetic resonance in medicine*, 84(6):3054–3070, 2020.
- [69] Florian Knoll et al. fastMRI: An open dataset and benchmarks for accelerated MRI, 2019.

- [70] Ken Kreutz-Delgado. The complex gradient operator and the cr-calculus. *arXiv preprint arXiv:0906.4835*, 2009.
- [71] Mao Li, Yingyi Ma, and Xinhua Zhang. Proximal mapping for deep regularization. *Advances in Neural Information Processing Systems*, 33:11623–11636, 2020.
- [72] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.
- [73] Rodrigo A Lobos, Chin-Cheng Chan, and Justin P Haldar. New theory and faster computations for subspace-based sensitivity map estimation in multichannel mri. *IEEE Transactions on Medical Imaging*, 43(1):286–296, 2023.
- [74] Rodrigo A Lobos and Justin P Haldar. On the shape of convolution kernels in MRI reconstruction: Rectangles versus ellipsoids. *Magnetic Resonance in Medicine*, 87(6):2989–2996, 2022.
- [75] Tianrui Luo, Douglas C Noll, Jeffrey A Fessler, and Jon-Fredrik Nielsen. A GRAPPA algorithm for arbitrary 2D/3D non-cartesian sampling trajectories with rapid calibration. *Magnetic resonance in medicine*, 82(3):1101–1112, 2019.
- [76] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.
- [77] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72–82, 2008.
- [78] Michael Lustig and John M. Pauly. SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space. *Magnetic Resonance in Medicine*, 64(2):457–471, 2010.
- [79] Albert Macovski. Noise in MRI. *Magnetic resonance in medicine*, 36(3):494–497, 1996.
- [80] Angshul Majumdar and Rabab K Ward. On the choice of compressed sensing priors and sparsifying transforms for mr image reconstruction: an experimental study. *Signal Processing: Image Communication*, 27(9):1035–1048, 2012.
- [81] Yura Malitsky and Thomas Pock. A first-order primal-dual algorithm with linesearch. *SIAM Journal on Optimization*, 28(1):411–432, 2018.
- [82] Yura Malitsky and Matthew K Tam. A forward-backward splitting method for monotone inclusions without cocoercivity. *SIAM Journal on Optimization*, 30(2):1451–1472, 2020.
- [83] G McGibney, MR Smith, ST Nichols, and A Crawley. Quantitative evaluation of several partial fourier reconstruction algorithms used in MRI. *Magnetic resonance in medicine*, 30(1):51–59, 1993.
- [84] Alex McManus, Stephen Becker, and Nicholas Dwork. Dependence of parallel imaging with linear predictability on the undersampling direction. *Journal of Electronic Imaging*, 34(2):023031–023031, 2025.
- [85] Kazuki Naganuma and Shunsuke Ono. Variable-wise diagonal preconditioning for primal-dual splitting: Design and applications. *IEEE Transactions on Signal Processing*, 2023.

- [86] Gopal Nataraj and Ricardo Otazo. Model-free deep mri reconstruction: A robustness study. In ISMRM Workshop on Data Sampling and Image, 2020.
- [87] Gopal Nataraj and Ricardo Otazo. Model-free deep MRI reconstruction: A robustness study. In Proceedings of the Annual Meeting of ISMRM, volume 28, page 3590, 2020.
- [88] Dwight G Nishimura. Principles of magnetic resonance imaging. Standford Univ., 2010.
- [89] Dwight G. Nishimura. Principles of Magnetic Resonance Imaging. Lulu.com, 2010.
- [90] Douglas C Noll, Dwight G Nishimura, and Albert Macovski. Homodyne detection in magnetic resonance imaging. IEEE Transactions on medical imaging, 10(2):154–163, 1991.
- [91] Constant Richard Noordman, Derya Yakar, Joeran Bosma, Frank Frederikus Jacobus Simonis, and Henkjan Huisman. Complexities of deep learning-based undersampled mr image reconstruction. European radiology experimental, 7(1):58, 2023.
- [92] Daniel O'Connor and Lieven Vandenberghe. On the equivalence of the primal-dual hybrid gradient method and douglas–rachford splitting. Mathematical Programming, 179(1):85–108, 2020.
- [93] Frank Ong, Shahab Amin, Shreyas Vasanawala, and Michael Lustig. Mridata. org: An open archive for sharing mri raw data. In Proc. Intl. Soc. Mag. Reson. Med, volume 26, 2018.
- [94] Lisandro A Parente, Pablo A Lotito, and Mikhail V Solodov. A class of inexact variable metric proximal point algorithms. SIAM Journal on Optimization, 19(1):240–260, 2008.
- [95] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. Foundations and trends® in Optimization, 1(3):127–239, 2014.
- [96] John Pauly. Partial k-space reconstruction. preprint, 2005.
- [97] Thomas Pock and Antonin Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In 2011 International Conference on Computer Vision, pages 1762–1769. IEEE, 2011.
- [98] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. An algorithm for minimizing the mumford-shah functional. In 2009 IEEE 12th International Conference on Computer Vision, pages 1133–1140. IEEE, 2009.
- [99] Klaas P Pruessmann. Encoding and reconstruction in parallel mri. NMR in Biomedicine: An International Journal Devoted to the Development and Application of Magnetic Resonance In vivo, 19(3):288–299, 2006.
- [100] Klaas P Pruessmann, Markus Weiger, Peter Börnert, and Peter Boesiger. Advances in sensitivity encoding with arbitrary k-space trajectories. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, 46(4):638–651, 2001.
- [101] Klaas P Pruessmann, Markus Weiger, Markus B Scheidegger, and Peter Boesiger. Sense: sensitivity encoding for fast mri. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, 42(5):952–962, 1999.

- [102] Klaas P. Pruessmann, Markus Weiger, Markus B. Scheidegger, and Peter Boesiger. Sense: Sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine*, 42(5):952–962, Nov 1999.
- [103] Alireza Radmanesh, Matthew J Muckley, Tullie Murrell, Emma Lindsey, Anuroop Sriram, Florian Knoll, Daniel K Sodickson, and Yvonne W Lui. Exploring the acceleration limits of deep learning variational network-based two-dimensional brain MRI. *Radiology: Artificial Intelligence*, 4(6):e210313, 2022.
- [104] R Tyrrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997.
- [105] Peter B Roemer, William A Edelstein, Cecil E Hayes, Steven P Souza, and Otward M Mueller. The nmr phased array. *Magnetic resonance in medicine*, 16(2):192–225, 1990.
- [106] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.
- [107] Abdurakhmon Sadiev, Laurent Condat, and Peter Richtárik. Stochastic proximal point methods for monotone inclusions under expected similarity. *arXiv preprint arXiv:2405.14255*, 2024.
- [108] Mojtaba Safari, Zach Eidex, Chih-Wei Chang, Richard LJ Qiu, and Xiaofeng Yang. Fast mri reconstruction using deep learning-based compressed sensing: A systematic review. *ArXiv*, pages arXiv–2405, 2024.
- [109] Alexei A Samsonov, Eugene G Kholmovski, Dennis L Parker, and Chris R Johnson. Pocsense: Pocs-based reconstruction for sensitivity encoded magnetic resonance imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 52(6):1397–1406, 2004.
- [110] Christopher M Sandino, Joseph Y Cheng, Feiyu Chen, Morteza Mardani, John M Pauly, and Shreyas S Vasanawala. Compressed sensing: From research to clinical practice with deep neural networks: Shortening scan times for magnetic resonance imaging. *IEEE signal processing magazine*, 37(1):117–127, 2020.
- [111] Christopher M Sandino, Joseph Y Cheng, Feiyu Chen, Morteza Mardani, John M Pauly, and Shreyas S Vasanawala. Compressed sensing: From research to clinical practice with deep neural networks: Shortening scan times for magnetic resonance imaging. *IEEE signal processing magazine*, 37(1):117–127, 2020.
- [112] Nicole Seiberlich, Felix A Breuer, Martin Blaimer, Kestutis Barkauskas, Peter M Jakob, and Mark A Griswold. Non-cartesian data reconstruction using GRAPPA operator gridding (GROG). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1257–1265, 2007.
- [113] Nalini M Singh, Neel Dey, Malte Hoffmann, Bruce Fischl, Elfar Adalsteinsson, Robert Frost, Adrian V Dalca, and Polina Golland. Data consistent deep rigid MRI motion correction. *arXiv preprint arXiv:2301.10365*, 2023.
- [114] Daniel K Sodickson and Warren J Manning. Simultaneous acquisition of spatial harmonics (smash): fast imaging with radiofrequency coil arrays. *Magnetic resonance in medicine*, 38(4):591–603, 1997.

- [115] Daniel K. Sodickson and Warren J. Manning. Simultaneous acquisition of spatial harmonics (smash): Fast imaging with radiofrequency coil arrays. *Magnetic Resonance in Medicine*, 38(4):591–603, Oct 1997.
- [116] Martin Uecker, Peng Lai, Mark J Murphy, Patrick Virtue, Michael Elad, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. Espirit—an eigenvalue approach to autocalibrating parallel mri: where sense meets grappa. *Magnetic resonance in medicine*, 71(3):990–1001, 2014.
- [117] SS Vasanawala, MJ Murphy, Marcus T Alley, P Lai, Kurt Keutzer, John M Pauly, and Michael Lustig. Practical parallel imaging compressed sensing MRI: Summary of two years of experience in accelerating body MRI of pediatric patients. In *international symposium on biomedical imaging: From nano to macro*, pages 1039–1043. IEEE, 2011.
- [118] Yinjun Wang, Haixiang Lan, and Yinyu Ye. A tuning-free primal-dual splitting algorithm for large-scale semidefinite programming. *arXiv preprint arXiv:2402.00311*, 2024.
- [119] Paul T Weavers, Eric A Borisch, Casey P Johnson, and Stephen J Riederer. Acceleration apportionment: A method of improved 2D SENSE acceleration applied to 3D contrast-enhanced MR angiography. *Magnetic resonance in medicine*, 71(2):672–680, 2014.
- [120] Meng Wen, Jigen Peng, Yuchao Tang, Chuanxi Zhu, Shigang Yue, et al. A preconditioning technique for first-order primal-dual splitting method in convex optimization. *Mathematical Problems in Engineering*, 2017, 2017.
- [121] Katherine L Wright, Jesse I Hamilton, Mark A Griswold, Vikas Gulani, and Nicole Seiberlich. Non-Cartesian parallel imaging reconstruction. *Journal of Magnetic Resonance Imaging*, 40(5):1022–1040, 2014.
- [122] Burhaneddin Yaman, Seyed Amir Hosseini, and Mehmet Akçakaya. Zero-shot self-supervised learning for mri reconstruction. *arXiv preprint arXiv:2102.07737*, 2021.
- [123] Gushan Zeng, Yi Guo, Jiaying Zhan, Zi Wang, Zongying Lai, Xiaofeng Du, Xiaobo Qu, and Di Guo. A review on deep learning MRI reconstruction without fully sampled k-space. *BMC Medical Imaging*, 21(1):195, 2021.
- [124] Di Zhao, Feng Zhao, and Yongjin Gan. Reference-driven compressed sensing mr image reconstruction using deep convolutional neural networks without pre-training. *Sensors*, 20(1):308, 2020.