Fixed Point Iterations for Finite Sum Monotone Inclusions

Morin, Martin

2022

Document Version:
Publisher's PDF, also known as Version of record

Fixed Point Iterations for Finite Sum Monotone Inclusions

Martin Morin

LUND
UNIVERSITY

Department of Automatic Control

PhD Thesis TFRT-1138
ISBN 978-91-8039-409-3 (print)
ISBN 978-91-8039-410-9 (web)
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Media-Tryck.
Lund 2022

Abstract

This thesis studies two families of methods for finding zeros of finite sums of monotone operators, the first being variance-reduced stochastic gradient (VRSG) methods. This is a large family of algorithms that use random sampling to improve the convergence rate compared to more traditional approaches. We examine the optimal sampling distributions and their interaction with the epoch length. Specifically, we show that in methods like SAGA, where the epoch length is directly tied to the random sampling, the optimal sampling becomes more complex compared to for instance L-SVRG, where the epoch length can be chosen independently. We also show that biased VRSG estimates in the style of SAG are sensitive to the problem setting. More precisely, a significantly larger step-size can be used when the monotone operators are cocoercive gradients compared to when they just are cocoercive. This is noteworthy since the standard gradient descent is not affected by this change and the fact that the sensitivity to the problem assumption vanishes when the estimates are unbiased.

The second set of methods we examine are deterministic operator splitting methods and we focus on frameworks for constructing and analyzing such splitting methods. One such framework is based on what we call nonlinear resolvents and we present a novel way of ensuring convergence of iterations of nonlinear resolvents by the means of a momentum term. This approach leads in many cases to cheaper per-iteration cost compared to a previously established projection approach. The framework covers many existing methods and we provide a new primal-dual method that uses an extra resolvent step as well as a general approach for adding momentum to any special case of our nonlinear resolvent method. We use a similar concept to the nonlinear resolvent to derive a representation of the entire class of frugal splitting operators, which are splitting operators that use exactly one direct or resolvent evaluation of each operator of the monotone inclusion problem. The representation reveals several new results regarding lifting numbers, existence of solution maps, and parallelizability of the forward/backward evaluations. We show that the minimal lifting is $n-1-f$ where n is the number of monotone operators and f is the number of direct evaluations in the splitting. A new convergent and parallelizable frugal splitting operator with minimal lifting is also presented.

Acknowledgements

Thanks should of course go out to my supervisors: Pontus Giselsson, Sebastian Banert, and Bo Bernhardsson. Without you, these last five years would have been very different and this doctoral thesis would not have existed. To the rest of the people that make the department go round, thanks. Without the administrative staff, things would not run half as smooth and I know I always can count on you to help with what is in your power. The rest of my colleagues, both new and old, have all provided good company, stimulating talks, and enjoyable coffee breaks. Special thanks should go to my family and friends for supporting me and sharing both good and bad times during the entirety of my life.

Perhaps to the dismay of my office mates, I would like to thank the band and artists that have provided the soundtrack for the last five years. Most of those bands and artists I found via the writers at `angrymetalguy.com`, all of which deserve thanks for broadening and stimulating my musical interest. Further thanks (and perhaps some guilt) are given to the team behind `typeracer.com`. I'll never become a fast enough typist to regain the time I've spent on your site.

Contents

1. Introduction **9**
1.1 Outline 10

2. Background **11**
2.1 Notation and Preliminaries 11
2.2 Monotone Inclusion Problems 13
2.3 Convex Optimization as Monotone Inclusion 15
2.4 Fixed Point Iterations 17
2.5 Operator Splitting Methods 17
2.6 Variance-Reduced Stochastic Gradient Methods 20

3. Contributions **23**

Bibliography **26**

Paper I. Sampling and Update Frequencies in Proximal Variance-Reduced Stochastic Gradient Methods **31**
1 Introduction 32
2 Preliminaries 33
3 Problem and Algorithm 34
4 Convergence Analysis 36
5 Special Cases 39
6 Sampling Design 40
7 Numerical Experiments 42
8 Conclusion 45
A Proofs of Proposition and Lemmas 45
B Proofs of Theorems 48
C Proof of Corollaries 50
References 52

Paper II. Cocoercivity, Smoothness and Bias in Variance-Reduced Stochastic Gradient Methods **55**
1 Introduction 56
2 Preliminaries and Notation 60

3 Convergence 62
4 Numerical Experiments 73
5 Conclusion 76
References 77

Paper III. Nonlinear Forward-Backward Splitting with Momentum Correction **83**
1 Introduction 84
2 Problem and Algorithm 87
3 Convergence 89
4 Additional Momentum 92
5 Forward-Half-Reflected-Backward Splitting 94
6 Primal-Dual Methods 97
7 Conclusion 108
References 108

Paper IV. Frugal Splitting Operators: Representation, Minimal Lifting and Convergence **113**
1 Introduction 114
2 Preliminaries 116
3 Frugal Splitting Operators 120
4 Generalized Primal-Dual Resolvents 121
5 Representation of Frugal Splitting Operators 128
6 Minimal Lifting 134
7 Convergence 138
8 A New Frugal Splitting Operator With Minimal Lifting 145
9 Conclusion 149
References 150
Supplementary Material 153

Popular Science Summary (in Swedish) **179**

1

Introduction

Understanding the methods we use to solve problems is crucial. The methods' strengths and weaknesses, their applicability and limitations, and their complexity and ease of use are all important to know so the right method for a particular situation can be chosen. Having a thorough understanding is also vital, both for the improvement of existing methods and for the development of new ones. These things hold not only for the methods we use to solve mathematical problems but for the tools and approaches we use to design those mathematical methods.

This thesis will cover the design and analysis of several methods for solving a class of problems known as monotone inclusion problems. Monotone inclusion problems cover a range of commonly occurring problems in engineering and various scientific fields, one of the prime examples being convex optimization. While solving monotone inclusion problems fast and efficiently is the main motivation behind the work in this thesis, concrete problem instances will not be the focus. Some of the included works will provide illustrative or exploratory examples—mainly within the field of optimization—but the focus will first and foremost be on the solution methods themselves.

The methods covered will be from one of two families: variance-reduced stochastic gradient (VRSG) methods and operator splitting methods. We leave a more thorough presentation of these two method families to the next chapter and the individual papers but note that the distinction between them is not necessarily sharp. For our purposes, the main difference is that splitting methods are deterministic, but, if one were to allow for randomness in operator splitting methods, one could easily argue that VRSG methods are a subset of operator splitting methods. However, we will not comment more on this connection and the two groups will be treated separately.

VRSG methods, see for instance [1, 4, 7, 11, 13, 14, 17, 18, 19, 20, 21, 22, 24, 33, 34, 36, 37, 40], were first derived to solve optimization problems formed from very large datasets. Due to the size of the dataset, the limiting factor for these kinds of problems is processing the data so the basic idea behind VRSG methods is to avoid processing the entire dataset at once. Instead, smaller subsets of the data are sampled at random and used in such a way that the expected value of the sampling

is the same as it would be if the entire dataset was used. If this is done carefully, not only can the problem be solved with probability one but the expected computational time can actually be faster than for conventional deterministic methods [24, 36]. Our work on VRSG methods consists of us trying to better understand and improve the speed of VRSG methods but we also expand them to the more general monotone inclusion setting. In particular, we look at the effect of the random sampling of data on the computational cost and derive optimal sampling distributions for a subclass of VRSG methods. In the monotone inclusion setting, we examine the effects of different problem assumptions, uncovering important results to consider when designing VRSG methods.

The term "operator splitting method" usually refers to a type of divide and conquer approach for solving monotone inclusion problems. In theory, monotone inclusion problems are actually quite simple to solve with the evaluation of a map known as a resolvent. However, computing the resolvent is in many cases computationally infeasible. Because of this, several splitting approaches have been developed that split up the monotone inclusion problem into smaller pieces, each of which consists of a computationally feasible resolvent or some other cheaply computed mapping, see for instance [2, 5, 6, 8, 9, 10, 12, 15, 16, 23, 25, 26, 27, 32, 35, 38, 39]. Our contributions have mainly been in providing compact representations and convergence criteria for large sub-classes of splitting methods. Finding new ways of representing several different methods opens up new avenues for comparing and understanding their behavior. It also makes it easier to create and explore new methods, enabling the discovery of performance improvements. Furthermore, a more expressive representation can be very beneficial in the modern era of computer and data-driven design where the exploration of large design spaces can be automated.

1.1 Outline

This thesis is structured as follows. Chapter 2 introduces the basic notation and concepts covered in the thesis such as monotone inclusions, fixed point iterations, operator splitting methods, and variance-reduced stochastic gradient methods. In Chapter 3, the papers that make up this thesis are introduced together with a declaration of each author's contribution. The remainder of the thesis contains said papers.

2

Background

The following chapter will cover the basic mathematical concepts covered by the papers. It is intended as an overview for the uninitiated reader and will focus on presenting the main mathematical objects and their relation to the questions of the papers that make up the thesis. No attempts will be made to exhaustively cover all fundamental results our papers build on; the papers are complete and self-contained with all relevant preliminary results either presented or referenced. A good reference textbook on the subjects covered in this chapter, except for the variance-reduced stochastic gradient methods, is [3]. For more information on variance-reduced stochastic gradient methods we will refer to the different papers that introduced the concepts, for instance [13, 20, 24, 36, 40].

2.1 Notation and Preliminaries

The set of real numbers will be denoted by $\mathbb{R}$, the natural numbers by $\mathbb{N} = \{0, 1, \dots\}$ and the positive natural numbers by $\mathbb{N}_+ = \{1, 2, \dots\}$. Let $\mathcal{H}$ be a real Hilbert space with inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ respectively. The notation $2^{\mathcal{H}}$ denotes the power set of $\mathcal{H}$, i.e., the set of all subsets of $\mathcal{H}$. The fundamental object of this thesis is the operator.

DEFINITION 2.1—OPERATOR
An operator *A on $\mathcal{H}$ is a map from $\mathcal{H}$ to any subset of $\mathcal{H}$, i.e., $A \colon \mathcal{H} \to 2^{\mathcal{H}}$.*

Instead of mapping a point in $\mathcal{H}$ to a just single point in $\mathcal{H}$, an operator allows us to map a point in $\mathcal{H}$ to several points in $\mathcal{H}$, making operators *set-valued* or *multi-valued*. We denote the set of points $x \in \mathcal{H}$ maps to as $Ax \in 2^{\mathcal{H}}$. Since Ax can be any subset of $\mathcal{H}$, it is possible to map a point x to no points by mapping to the empty set, i.e., $Ax = \emptyset$. The *domain* of an operator is defined as the set of points that maps to non-empty sets, i.e., $\operatorname{dom} A = \{x \in \mathcal{H} \mid Ax \neq \emptyset\}$. It is also worth stressing that Ax is a subset of $\mathcal{H}$ and hence, if we wish to state that $x \in \mathcal{H}$ is mapped to $y \in \mathcal{H}$, we write $y \in Ax$ and *not* $y = Ax$. However, for *single-valued operators* we will abuse this notation slightly.

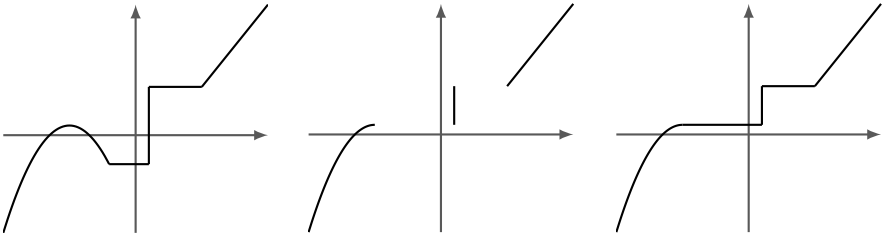


Figure 2.1 Examples of graphs of operators $\mathbb{R} \to 2^{\mathbb{R}}$ and their properties.

A single-valued operator refers to either an ordinary map $B\colon \mathcal{H} \to \mathcal{H}$ or an operator $C\colon \mathcal{H} \to 2^{\mathcal{H}}$ where Cx is a singleton for all $x \in \mathcal{H}$. For any such B and C we can always find $B'\colon \mathcal{H} \to 2^{\mathcal{H}}$ and $C'\colon \mathcal{H} \to \mathcal{H}$ such that $\{Bx\} = B'x$ and $\{C'x\} = Cx$ for all $x \in \mathcal{H}$. Because of these equivalences between B and B', and C and C' we will not make any distinction between them and let Bx and Cx denote both sets and points depending on context.

Since operators are multi-valued, we can always define the *inverse* $A^{-1}\colon \mathcal{H} \to 2^{\mathcal{H}}$ of an operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ without requiring some form of injectivity. The inverse is simply defined as $x \in A^{-1}u$ if and only if $u \in Ax$ for $x,u \in \mathcal{H}$. The *graph* of an operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is $\operatorname{gra} A = \{(x,u) \in \mathcal{H} \times \mathcal{H} \mid u \in Ax\}$, allowing the inverse to be characterized as $(u,x) \in \operatorname{gra} A^{-1}$ if and only if $(x,u) \in \operatorname{gra} A$.

For most results, we cannot allow for the full generality of arbitrary operators so the following properties will be assumed in a majority of cases.

DEFINITION 2.2—MONOTONICITY
An operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ *is* monotone *if*

$$\langle u - v, x - y \rangle \geq 0$$

for all $x,y \in \mathcal{H}$ *and all* $u \in Ax$ *and* $v \in Ay$.

DEFINITION 2.3—MAXIMALITY
A monotone operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ *is* maximal *if there exists no other monotone operator* $A'\colon \mathcal{H} \to 2^{\mathcal{H}}$ *such that* $\operatorname{gra} A \subseteq \operatorname{gra} A'$.

In the one-dimensional case, i.e., $\mathbb{R} \to 2^{\mathbb{R}}$, monotonicity is equivalent to the operator being non-decreasing, i.e., the slope of the graph is never negative. Maximality is a continuity-like assumption and states that the graph has no "holes". A few examples in the one-dimensional case are provided in Fig. 2.1. Both monotonicity and maximality are fundamental for the work in this paper but the following stronger properties will be useful in certain cases.

DEFINITION 2.4—STRONG MONOTONICITY
An operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is μ-strongly monotone for $\mu > 0$ if

$$\langle u - v, x - y \rangle \geq \mu \|x - y\|^2$$

for all $x, y \in \mathcal{H}$ and all $u \in Ax$ and $v \in Ay$.

DEFINITION 2.5—LIPSCHITZ CONTINUOUS
A single valued operator $A\colon \mathcal{H} \to \mathcal{H}$ is ℓ-Lipschitz continuous for $\ell > 0$ if

$$\ell \|x - y\| \geq \|Ax - Ay\|$$

for all $x, y \in \mathcal{H}$. A 1-Lipschitz continuous operator is said to be non-expansive.

DEFINITION 2.6—COCOERCIVITY
A single valued operator $A\colon \mathcal{H} \to \mathcal{H}$ is β-cocoercive for $\beta > 0$ if

$$\langle Ax - Ay, x - y \rangle \geq \beta \|Ax - Ay\|^2$$

for all $x, y \in \mathcal{H}$.

In the one-dimensional case, μ-strong monotonicity states that the slope of the graph is always greater than μ, ℓ-Lipschitz continuity states that the slope is between $-\ell$ and ℓ, and β-cocoercivity states that the slope is between 0 and β^{-1}. Therefore, β-cocoercivity is equivalent to monotonicity and β^{-1}-Lipschitz continuity in the one-dimensional case. In higher dimensions, this equivalence does no longer hold but β-cocoercivity still implies β^{-1}-Lipschitz continuity and monotonicity. Cocoercivity is equivalent to Lipschitz continuity and monotonicity only when the operators are so-called *subdifferentials*[3, Corollary 18.17]. Subdifferentials will be defined and explained further when discussing optimization problems in a following section.

2.2 Monotone Inclusion Problems

A monotone inclusion problem is

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in Ax \tag{2.1}$$

where $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is a maximally monotone operator. The notation $\operatorname{zer} A = \{x \in \mathcal{H} \mid 0 \in Ax\}$ will be used for denoting the solutions to this problem. It should be noted that this problem does not necessarily have a solution, i.e., it is possible that $\operatorname{zer} A = \emptyset$. However, establishing sufficient conditions for when $\operatorname{zer} A \neq \emptyset$ is beyond the scope and needs of the papers in this thesis and we simply assume that $\operatorname{zer} A$ is non-empty. This pragmatic approach is motivated by the fact that it does not makes any sense to solve a problem without a solution and, since this thesis focus

on solutions methods for (2.1), The counterargument to this is that it is useful if a solution method can detect and report that no solution exists since it might be hard to know beforehand. Although we agree with this, we see the question of certifying a lack of solution as orthogonal to the questions examined in this thesis.

Problem (2.1) is comparable to the classic matrix inversion problem of finding $x \in \mathbb{R}^n$ such that $b = Mx$ where $b \in \mathbb{R}^n$ and $M \in \mathbb{R}^{n\times n}$. Solving this kind of linear equations is the backbone of numerical linear algebra and a key component of many engineering and scientific fields. By defining the operator $x \mapsto Mx - b$, the matrix inversion can equivalently be formulated as finding a zero of this operator, just as in problem (2.1). It is therefore possible to view monotone inclusion problems as analogue to these kinds of very useful inversion problems but with linearity/affinity replaced by maximal monotonicity. If $M + M^T$ is positive semi-definite, the map $x \mapsto Mx - b$ is maximally monotone and the matrix inversion problem is then an instance of the monotone inclusion problem [3, Corollary 20.28]. Although specific instances of monotone inclusion are not the focus of this thesis, we will in Section 2.3 give an example of how a convex optimization problem can be reformulated into a monotone inclusion problem.

In certain aspects, maximal monotonicity is a lot weaker than affinity but it still provides useful and convenient properties. For instance, the solutions to (2.1) form closed convex sets. This follows from two simple facts, see [3, Proposition 20.22 and 20.36] for detailed proofs. First, Ax is always closed and convex for all maximally monotone $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ and $x \in \mathcal{H}$. Second, if $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone, then A^{-1} is also maximally monotone. Noticing that $\operatorname{zer} A = A^{-1}0$ then yields the result. Maximal monotonicity also has useful implications when designing algorithms for solving inclusion problems; we explore these aspects further when discussing operator splitting methods in Section 2.5.

The papers included in the thesis actually consider the slightly more general scenario of finite sum monotone inclusions, i.e.,

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{n} A_i x \tag{2.2}$$

where $A_i : \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone for all $i \in \{1,\dots,n\}$. Perhaps the most important thing to note regarding sums of operators is that the sum of maximally monotone operators is not necessarily a maximally monotone operator. Sufficient conditions for when the sum is maximally monotone exist [3, Theorem 25.3] but attempting to list these is again beyond the scope or needs of this thesis.

In most cases, (2.2) will be complemented by some further assumptions on one or several of the terms, e.g., cocoercivity, Lipschitz continuity, or strong monotonicity. Sometimes these assumptions carry further implications regarding the problem itself. For instance, the sum of a maximally monotone operator and cocoercive operator is guaranteed to be maximally monotone [3, Corollary 25.5] and a solution exists and is unique if in addition one of the operators is strongly monotone [3, Corollary 23.37]. However, although such problem properties will be used and

commented on in the included papers, the main purpose of these assumptions is to facilitate the design of convergent algorithms for solving (2.2).

2.3 Convex Optimization as Monotone Inclusion

A constrained convex optimization problem takes the form

$$\begin{aligned} &\underset{x \in D}{\text{minimize}} \; f(x) \\ &\text{subject to } x \in C \end{aligned} \tag{2.3}$$

where $f \colon D \to \mathbb{R}$ is convex, $D \subseteq \mathcal{H}$ and $C \subseteq \mathcal{H}$ are convex set. We will assume that the problem is feasible, i.e., $C \cap D \neq \emptyset$, that C and D are closed, and f is lower semi-continuous.

Our first goal is to write (2.3) as an unconstrained convex problem and for that two things need to be dealt with, the constraint set C and the fact that the objective function f is not necessarily defined everywhere, i.e., it is possible that $D \neq \mathcal{H}$. Both of these problems can be dealt with by introducing what is known as the *extended real line* $\mathbb{R} \cup \{-\infty, \infty\}$ where plus and minus infinity are defined as larger and smaller than any real number, respectively, $-\infty < x < \infty$ for all $x \in \mathbb{R}$. A complete arithmetic is defined as $\infty + x = \infty$, $x_+ \cdot \infty = \infty$, $x_- \cdot \infty = -\infty$, $\frac{x}{\pm\infty} = 0$ for all $x \in \mathbb{R}$, $x_+ > 0$ and $x_- < 0$ with the following being undefined: $0 \cdot \infty$, $\infty + (-\infty)$ and $\frac{\infty}{\infty}$.

We extend f by defining $\hat{f} \colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ to be equal to f on D and otherwise $\hat{f}(x) = \infty$. The constraint set C we encode by defining the indicator function $\iota_C : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ such that $\iota_C(x) = 0$ if $x \in C$ and $\iota_C(x) = \infty$ if $x \notin C$. An equivalent unconstrained optimization problem can then be written as

$$\underset{x \in \mathcal{H}}{\text{minimize}} \; \hat{f}(x) + \iota_C(x). \tag{2.4}$$

It is equivalent to (2.3) since $\hat{f}(x) + \iota_C(x) = f(x) \in \mathbb{R}$ on all feasible points $x \in C \cap D$ and otherwise the objective is equal to ∞, which is larger than all real numbers. It is also straightforward to verify that $\hat{f}$, ι_C and their sum are convex and lower semi-continuous [3, Lemma 1.27]. Although both terms of the objective of (2.4) are defined for all $x \in \mathcal{H}$, the convention for extended-real-valued functions is to define the *domain* of a function $g \colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ as $\operatorname{dom} g = \{x \in \mathcal{H} \mid g(x) < \infty\}$ which yields $\operatorname{dom} \hat{f} + \iota_C = \operatorname{dom} \hat{f} \cap \operatorname{dom} \iota_C = D \cap C$. Hence, the domain of the objective of the unconstrained problem is the same as the set of feasible points of the constrained problem. Feasibility of (2.3) is therefore equivalent to $\hat{f} + \iota_C$ being *proper*, a proper function $g \colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ satisfies $\operatorname{dom} g \neq \emptyset$.

It is well known that finding the unconstrained minimum of a differentiable convex function $g \colon \mathcal{H} \to \mathbb{R}$ is equivalent to finding a zero of the gradient, $0 = \nabla g(x)$. This is known as Fermat's rule. However, the objective of our unconstrained optimization problem is not necessarily differentiable but it is still possible to state an analogue statement using *subdifferentials*.

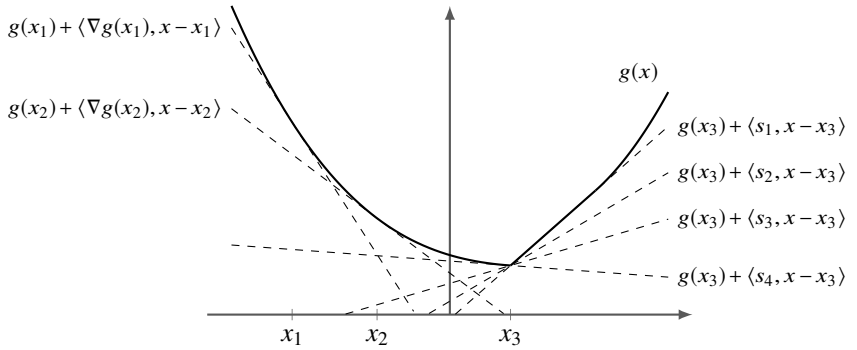


Figure 2.2 Illustration of subdifferentials. The function $g\colon \mathbb{R} \to \mathbb{R}$ is differentiable at x_1 and x_2 and hence are $\{\nabla g(x_1)\} = \partial g(x_1)$ and $\{\nabla g(x_2)\} = \partial g(x_2)$. At x_3 is g not differentiable and hence is $\partial g(x_3)$ not single-valued and $s_1, s_2, s_3, s_4 \in \partial g(x_3)$.

DEFINITION 2.7—SUBDIFFERENTIAL
Let $g\colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be a proper function. The subdifferential *of g is the operator $\partial g\colon \mathcal{H} \to 2^{\mathcal{H}}$ such that*

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

for all $x \in \mathcal{H}$. Elements in $\partial g(x)$ are known as subgradients *of g at $x \in \mathcal{H}$.*

The subdifferential of any proper function $g\colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is monotone and ∂g is maximally monotone if g is convex and lower semi-continuous [3, Example 20.3, Theorem 20.25]. Although the subdifferential is not defined via limits as ordinary derivatives are, the two notions are closely related for convex functions. Let g be convex, if g is differentiable at x, then $\{\nabla g(x)\} = \partial g(x)$. Similarly, if $\partial g(y) = \{s\}$ and g is continuous at $y \in \mathcal{H}$, then g is differentiable at y and $s = \nabla g(y)$ [3, Proposition 17.31]. For a visual example of the subdifferential, see Fig. 2.2.

Fermat's rule for subdifferentials is then that $x \in \mathcal{H}$ is a minimum of a proper function $g\colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ if and only if $0 \in \partial g(x)$ [3, Proposition 27.1]. The unconstrained problem (2.4) can then equivalently be written as the monotone inclusion problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \partial(\hat{f} + \iota_C)(x). \tag{2.5}$$

Different from the gradient, the subdifferential is not additive, i.e., $\partial(g + h) \neq \partial g + \partial h$ for some proper convex lower semi-continuous functions $g\colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ and $h\colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. However, it always holds that $\partial(g + h)(x) \supseteq \partial g(x) + \partial h(x)$ and, hence, if we find a solution to the following problem, we find a solution to (2.5),

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \partial \hat{f}(x) + \partial \iota_C(x). \tag{2.6}$$

Since both $\hat{f}$ and ι_C are proper, convex, and lower semi-continuous, both $\partial \hat{f}$ and $\partial \iota_C$ are maximally monotone operators. Problem (2.6) is therefore a finite sum

monotone inclusion problem of the form (2.2) that this thesis covers. The two problems, (2.5) and (2.6), are equivalent when $\partial(f + \iota_C) = \partial \hat{f} + \partial \iota_C$ which holds if, for instance, $\operatorname{dom} \hat{f} = \mathcal{H}$ or $\operatorname{dom} \hat{f} \cap \operatorname{int} C \neq \emptyset$ [3, Corollary 16.48]. Even weaker forms of this kind of constraint qualification exist and it should be noted that the equivalence between (2.5) and (2.6) is in practice not considered a problem for convex optimization problems.

2.4 Fixed Point Iterations

A problem that is very useful when solving monotone inclusion problems is the fixed point problem,

$$\text{find } x \in \mathcal{H} \text{ such that } x = Tx \tag{2.7}$$

where $T \colon \mathcal{H} \to \mathcal{H}$. The set of all such fixed points will be denoted $\operatorname{fix} T = \{x \in \mathcal{H} \mid x = Tx\}$. The reason for its appeal is that there exists a natural method for solving it, namely fixed point iterations. In a fixed point iteration, starting at some $x_0 \in \mathcal{H}$, one iteratively performs

$$x_{k+1} = Tx_k \tag{2.8}$$

for $k \in \mathbb{N}$. If the operator T is sufficiently "nice", the sequence $\{x_k\}_{k \in \mathbb{N}}$ will converge—weakly or strongly depending on the setting—to a fixed point of T. There is no one set of conditions that constitutes sufficiently "nice" and fixed point iterations can converge under a varied set of assumptions on T.

One such classic assumption is ℓ-Lipschitz continuity of T with $\ell < 1$, such operators are also known as contractions. Not only are contractions guaranteed to have a single fixed point, but the distance to this fixed point is decreased by a constant factor each iteration of the fixed point iteration,

$$\|x_k - x\| = \|Tx_{k-1} - Tx\| \leq \ell \|x_{k-1} - x\|.$$

where $\{x\} = \operatorname{fix} T$. Repeatedly using this inequality yields $\|x_k - x\| \leq \ell^k \|x_0 - x\|$ and, since $\ell < 1$ and hence $\ell^k \to 0$, we see that $\|x_k - x\| \to 0$ and $x_k \to x$ as $k \to \infty$. However, requiring T to be a contraction is in many cases too strong of an assumption. One of the key components of all papers in this thesis is therefore establishing and analyzing the convergence of fixed point iterations under varying sets of assumptions on T.

2.5 Operator Splitting Methods

Consider the finite sum monotone inclusion problem in (2.2). An equivalent fixed point problem can be stated as

$$\text{find } x \in \mathcal{H} \text{ such that } x \in x - \gamma \sum_{i=1}^{n} A_i x = (\mathrm{Id} - \gamma \sum_{i=1}^{n} A_i)x \tag{2.9}$$

where $\gamma > 0$ and $\mathrm{Id}\colon \mathcal{H} \to \mathcal{H}$ is the identity operator, i.e., $\mathrm{Id}\colon x \mapsto x$. The equivalence between (2.9) and (2.2) comes from

$$x \in x - \gamma \sum_{i=1}^{n} A_i x \iff 0 \in -\gamma \sum_{i=1}^{n} A_i x \iff 0 \in \sum_{i=1}^{n} A_i x.$$

As outlined in Section 2.4, this fixed point problem provides a potential solution method for the monotone inclusion method. However, in order for this fixed point problem to be well-posed, $\sum_{i=1}^{n} A_i$ must be single valued at the very least. Even further assumptions are needed to guarantee the convergence of the associated fixed point iteration. For instance, if $\sum_{i=1}^{n} A_i$ is cocoercive and strongly monotone and γ is chosen small enough, $\mathrm{Id} - \gamma \sum_{i=1}^{n} A_i$ is a contraction and the fixed point iteration converges strongly [3, Proposition 26.16]. If $\sum_{i=1}^{n} A_i$ is only cocoercive, it is possible to show that the fixed point iteration converges weakly if γ is small enough [3, Definition 4.10 and Theorem 5.14]. However, this method is difficult to use in practice since it is enough for one term to not be cocoercive or single-valued for the sum $\sum_{i=1}^{n} A_i$ to not be cocoercive or single-valued, respectively.

Another fixed point problem equivalent to (2.2) is

$$\text{find } x \in \mathcal{H} \text{ such that } x \in (\mathrm{Id} + \gamma \sum_{i=1}^{n} A_i)^{-1} x = \mathrm{J}_{\gamma \sum_{i=1}^{n} A_i}\, x \tag{2.10}$$

where $\gamma > 0$ and $\mathrm{J}_B = (\mathrm{Id} + B)^{-1}$ is known as the *resolvent* of the operator $B\colon \mathcal{H} \to 2^{\mathcal{H}}$. The equivalence between (2.10) and (2.2) comes from

$$x \in (\mathrm{Id} + \gamma \sum_{i=1}^{n} A_i)^{-1} x \iff (\mathrm{Id} + \gamma \sum_{i=1}^{n} A_i) x \ni x \iff 0 \in \sum_{i=1}^{n} A_i x.$$

The are several benefits of this formulation compared to (2.9). First of all, the resolvent $\mathrm{J}_{\gamma \sum_{i=1}^{n} A_i}$ is single valued even if $\sum_{i=1}^{n} A_i$ only is maximally monotone. Secondly, fixed point iterations of the resolvent of $\sum_{i=1}^{n} A_i$ are guaranteed to converge if $\sum_{i=1}^{n} A_i$ is maximally monotone, regardless of the choice of γ [3, Example 23.40]. This makes fixed point iterations of resolvents attractive but they can still be problematic. As noted in Section 2.2, $\sum_{i=1}^{n} A_i$ is not necessary maximally monotone even if all terms are maximally monotone. Similarly, the evaluation of $\mathrm{J}_{\gamma \sum_{i=1}^{n} A_i}$ is not necessarily tractable even if $\mathrm{J}_{\gamma A_i}$ is easily computable for all $i \in \{1, \ldots, n\}$.

Operator splitting is a way of getting around the problems of both of the previously presented approaches. Although there is no real formal definition, the term usually refers to methods that solve finite sum monotone inclusion problems (2.2) via some equivalent fixed point problem. What makes them "splitting" methods is that they split up the sum $\sum_{i=1}^{n} A_i$ and use each term separately, either in the form of the resolvent $\mathrm{J}_{\gamma A_i}$ or the direct evaluation of A_i. This way, the most can be made from the available information; if some of the terms are single-valued and cocoercive, direct evaluations of them can be performed and resolvent evaluations can be left to the terms that require it, i.e., are set-valued.

A classic example of an operator splitting method is the *forward-backward* method [16, 25]. It considers the two term monotone inclusion case, i.e., $n = 2$ in

(2.2), and assumes that the first operator A_1 is maximally monotone and the second A_2 is β-cocoercive. The reformulation of the inclusion problem to an equivalent fixed point problem is

$$\text{find } x \in \mathcal{H} \text{ such that } x = \mathrm{J}_{\gamma A_1}(\mathrm{Id} - \gamma A_2)x \tag{2.11}$$

where $\gamma > 0$. This problem can be seen as a combination of (2.9) and (2.10), taking a so called forward step on A_2 as in (2.9) and a backward step on A_1 as in (2.10). Fixed point iterations of $\mathrm{J}_{\gamma A_1} \circ (\mathrm{Id} - \gamma A_2)$ converge to a solution of (2.11) as long as $\gamma < 2\beta$ [3, Theorem 26.14].

Another foundational operator splitting method is the *Douglas–Rachford* method [26]. It also considers the two-term monotone inclusion problem but only assumes that A_1 and A_2 are maximally monotone and solves the fixed point problem

$$\text{find } x \in \mathcal{H} \text{ such that } x = \tfrac{1}{2}x + \tfrac{1}{2}(2\,\mathrm{J}_{\gamma A_1} - \mathrm{Id})(2\,\mathrm{J}_{\gamma A_2} - \mathrm{Id})x \tag{2.12}$$

where $\gamma > 0$. The fixed point iteration associated with this problem is guaranteed to converge to a fixed point, regardless of the choice of γ [3, Theorem 26.11]. However, the Douglas–Rachford method differs from the previous examples in that a solution x to (2.12) does not directly solve the inclusion problem, i.e., $x \notin \operatorname{zer} A_1 + A_2$. Instead, a solution to the inclusion problem can be recovered from x as $\mathrm{J}_{\gamma A_2} x \in \operatorname{zer} A_1 + A_2$. In practice, this makes little difference since $\mathrm{J}_{\gamma A_2}$ is already evaluated each iteration of a fixed point iteration and the recovery of a solution does not amount to any significant additional cost.

If one looks at the forward-backward fixed point problem (2.11), it might be tempting to try a backward-backward or forward-forward method, i.e., finding fixed points of either $\mathrm{J}_{\gamma A_1} \circ \mathrm{J}_{\gamma A_2}$ or $(\mathrm{Id} - \gamma A_1) \circ (\mathrm{Id} - \gamma A_2)$. However, without stricter assumptions on A_1 and A_2, the fixed points of these operators are not solutions to their associated monotone inclusion problems, nor can they easily be mapped to solutions as in the Douglas–Rachford method. This exemplifies the care needed to construct operator splitting methods. Many more examples will be given in the included papers that, for instance, relax the cocoercivity assumption on the forward step or allow for an arbitrary number of terms in the finite sum monotone inclusion problem. There will also be several methods that include auxiliary variables in the fixed point problem, e.g.,

$$\text{find } (x, y) \in \mathcal{H}^2 \text{ such that } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \mathrm{J}_{\gamma A_1}(x - \gamma A_2 x + \theta(x - y)) \\ x \end{pmatrix}. \tag{2.13}$$

These auxiliary variables, y in the example above, are superfluous if one just considers the fixed point problem. However, they can be useful to improve the convergence speed of the algorithm. In example (2.13), the extra term $\theta(x - y)$ introduce a type of inertia or momentum to the forward step which can improve convergence. It should also be noted that even though we have discussed operator splitting methods

in terms of fixed point problems, they are not always solved with a pure fixed point iteration. The algorithm parameters can often be iteration dependent, especially for methods with a momentum term. For example, with $x_0, y_0 \in \mathcal{H}$ perform

$$\begin{aligned} x_{k+1} &= \mathrm{J}_{\gamma_k A_1}(x_k - \gamma_k A_2 x_k + \theta_k(x_k - y_k)) \\ y_{k+1} &= x_k \end{aligned} \tag{2.14}$$

for all $k \in \mathbb{N}$ where $\gamma_k, \theta_k > 0$ for all $k \in \mathbb{N}$. If A_1 is maximally monotone and A_2 is β-cocoercive, then the sequences generated by (2.13) satisfy $x_k \rightharpoonup x$ and $y_k \rightharpoonup x$ where $x \in \operatorname{zer} A_1 + A_2$ if there exists $\epsilon > 0$ such that $\epsilon \leq \gamma_k \leq 2\beta - \epsilon$ for all $k \in \mathbb{N}$, there exists $\theta \in [0,1)$ such that $0 \leq \theta_k \leq \theta$ for all $k \in \mathbb{N}$, and θ_k is chosen such that $\sum_{k\in\mathbb{N}} \theta_k \|x_k - y_k\| < \infty$, see [32]

2.6 Variance-Reduced Stochastic Gradient Methods

Traditionally, a variance-reduced stochastic gradient (VRSG) method is a type of algorithm for solving finite sum optimization problems of the form

$$\underset{x\in\mathcal{H}}{\text{minimize}} \ \sum_{i=1}^{n} f_i(x) \tag{2.15}$$

where $f_1 \colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is lower semi-continuous and convex and $f_i \colon \mathcal{H} \to \mathbb{R}$ is convex and differentiable with Lipschitz continuous gradient for all $i \in \{2,\dots,n\}$. However, these VRSG methods actually solve the equivalent finite sum monotone inclusion problem (2.2) with $A_1 = \partial f_1$ and $A_i = \nabla f_i$ for $i \in \{2,\dots,n\}$. This makes VRSG methods straightforward to generalize to more general monotone inclusion problems and in this thesis the term will refer to algorithms for solving (2.2) under the assumptions that A_1 is maximally monotone and A_i is β_i-cocoercive for all $i \in \{2,\dots,n\}$. This setting covers the above mentioned optimization problem but is more general.

Consider (2.2) under these cocoercivity assumptions. Since the sum of cocoercive operators is cocoercive, it is possible to solve this problem with the ordinary forward-backward method,

$$x_{k+1} = \mathrm{J}_{\gamma A_1}(x_k - \gamma \sum_{i=2}^{n} A_i x_k) \tag{2.16}$$

for $k \in \mathbb{N}$ where $x_0 \in \mathcal{H}$ and $\gamma > 0$. This iteration will converge as $x_k \rightharpoonup x \in \operatorname{zer} \sum_{i=1}^{n} A_i$ if γ is sufficiently small. However, for the modern machine learning and model fitting problems VRSG methods first were develop for, the number of terms n might be huge and the evaluations of all $n-1$ operators in the forward step can be very expensive. The idea behind VRSG methods is to replace the forward step with an approximation that only evaluates one operator each iteration, making the per iteration cost significantly lower. For our purposes, the approximate

forward-backward step used in VRSG methods is of the form

$$x_{k+1} = \mathrm{J}_{\gamma A_1}\left(x_k - \gamma[\theta_{i_k}(A_{i_k}x_k - y_{i_k,k}) + \sum_{j=2}^{n} y_{j,k}]\right) \tag{2.17}$$

where $\theta_i \in \mathbb{R}$ for all $i \in \{2,\dots,n\}$ and i_k is selected randomly each iteration. Except if it is initialized otherwise, the variable $y_{i,k} \in \mathcal{H}$ with $i \in \{2,\dots,n\}$ and $k \in \mathbb{N}$ contains the result of an evaluation of A_i at some point, i.e., for each $k \in \mathbb{N}$ and $i \in \{2,\dots,n\}$ there exists $x \in \mathcal{H}$ such that $y_{i,k} = A_i x$. Exactly how $y_{i,k}$ is updated and i_k is randomly selected each iteration differs between the different algorithms within the class of VRSG methods.

Although the (2.17) update is cheaper, the sequence $\{x_k\}_{k\in\mathbb{N}}$ might suffer from extremely slow convergence or might not even converge at all if the approximation is poor. Any gain from the cheaper per iteration cost can therefore still be negated by a need for more iterations. The success behind VRSG methods stems from the discovery of weight choices θ_i, random selections of i_k, and updates of $y_{i,k}$ that guarantee faster convergence to a solution than forward-backward when comparing the total number of operator evaluations [1]. For example, the first variance-reduced method that was proposed is SAG [24, 36]: let $x_0, y_{2,0},\dots,y_{n,0} \in \mathcal{H}$ and perform

$$\begin{aligned} &\text{Sample } i_k \text{ uniformly from } \{2,\dots,n\} \\ &x_{k+1} = \mathrm{J}_{\gamma A_1}\left(x_k - \gamma[A_{i_k}x_k - y_{i_k,k} + \sum_{j=2}^{n} y_{j,k}]\right) \\ &y_{i,k+1} = \begin{cases} A_i x_k & \text{if } i = i_k \\ y_{i,k} & \text{otherwise} \end{cases}, \quad \forall i \in \{2,\dots,n\} \end{aligned} \tag{2.18}$$

for all $k \in \mathbb{N}$. For SAG, only $y_{i_k,k}$ out of the variables $y_{2,k},\dots,y_{n,k}$ is updated at iteration k. Furthermore, the update of $y_{i_k,k}$ requires no additional operator evaluations since $A_{i_k,k}$ is reused, which is one of the main benefits of this update. Another example is the SVRG method [20, 40] where $x_0, y_{2,0},\dots,y_{n,0} \in \mathcal{H}$ and

$$\begin{aligned} &\text{Sample } i_k \text{ uniformly from } \{2,\dots,n\} \\ &x_{k+1} = \mathrm{J}_{\gamma A_1}\left(x_k - \gamma[n(A_{i_k}x_k - y_{i_k,k}) + \sum_{j=2}^{n} y_{j,k}]\right) \\ &y_{i,k+1} = \begin{cases} A_i x_k & \text{if } k \bmod m = 0 \\ y_{i,k} & \text{otherwise} \end{cases}, \quad \forall i \in \{2,\dots,n\}. \end{aligned} \tag{2.19}$$

for all $k \in \mathbb{N}$ where $m \in \mathbb{N}$. Here, all of $y_{2,k},\dots,y_{n,k}$ are updated every m iterations and hence $n-2$ additional operator evaluations must be performed every m iterations. The update interval m is therefore typically chosen around the same size as n, thereby keeping the average number of operator evaluations per iteration around two.

[1] Due to the random nature of VRSG methods, the convergence is almost sure and the convergence rate is measured with the expected distance to a solution.

The reason for the name variance-reduced becomes clear when looking at the approximation of the forward evaluation that is made in (2.17),

$$\sum_{i=2}^{n} A_i x_k \approx \widetilde{A}_{i_k,k} \coloneqq \theta_{i_k}(A_{i_k} x_k - y_{i_k,k}) + \sum_{j=2}^{n} y_{j,k}. \tag{2.20}$$

Let the probability $P(i_k = i) = p_i$, the variance w.r.t. the random index i_k of this approximation is then

$$\begin{aligned} &\mathbb{E}_{i_k} \|\widetilde{A}_{i_k,k} - \mathbb{E}_{i_k} A_{i_k,k}\|^2 \\ &\quad = \sum_{i=2}^{n} p_i \theta_i^2 \|A_i x_k - y_{i,k}\|^2 - \|\sum_{i=2}^{n} p_i \theta_i (A_i x_k - y_{i,k})\|^2 \\ &\quad \leq \sum_{i=2}^{n} p_i \theta_i^2 \|A_i x_k - y_{i,k}\|^2 \end{aligned} \tag{2.21}$$

and, hence, the better $y_{i,k}$ approximates $A_i x_k$, the smaller the variance of the approximation gets. It would of course be best to set $y_{i,k} = A_i x_k$ for all i and all k which recovers the ordinary forward-backward method (2.16) but, as previously mentioned, this might become expensive. VRSG methods therefore need to balance the computational cost of the update of $y_{i,k}$ with the difference $A_i x_k - y_{i,k}$. Similarly, we see that the variance becomes smaller with smaller θ_i with it being zero if $\theta_i = 0$ for all $i \in \{2,\dots,n\}$. However, looking at the expected value of the approximation,

$$\mathbb{E}_{i_k} \widetilde{A}_{i_k,k} = \sum_{i=2}^{n} p_i \theta_i A_i x_k + \sum_{i=2}^{n} (1 - p_i \theta_i) y_{i,k}, \tag{2.22}$$

we see that this choice introduces a bias to the expected value. The approximation is unbiased, i.e., $\mathbb{E}_{i_k} \widetilde{A}_{i_k,k} = \sum_{i=2}^{n} A_i x_k$, only when $\theta_i = p_i^{-1}$ for all $i \in \{2,\dots,n\}$. For all other parameter choices, the variables $y_{i,k}$ for $i \in \{2,\dots,n\}$ will not only affect the variance of the approximation, but also the expected value. This is one other aspect where SAG (2.18) and SVRG (2.19) differ; SVRG is unbiased while SAG is biased.

3

Contributions

We here briefly introduce each of the four papers included in the thesis and outline the contribution of each author. The notation and setting presented in the previous section are representative of the notation and setting used in all papers, although minor differences exist. The first two papers concern themselves with the convergence properties and conditions of variance-reduced stochastic methods while the last two aims to model larger sets of splitting methods for the design and analysis of new and existing methods.

Paper I

M. Morin and P. Giselsson. "Sampling and Update Frequencies in Proximal Variance Reduced Stochastic Gradient Methods" (2020). arXiv: 2002.05545v2 [cs, math]. URL: http://arxiv.org/abs/2002.05545v2

This first paper considers the convex optimization setting, i.e., all operators are subgradients of convex functions. It considers a class of variance-reduced stochastic proximal-gradient methods and the main result regards the sampling distribution of the gradients in each iteration. Theoretically optimal distributions are derived and the tightness of these theoretical results is supported by numerical experiments. How different algorithms and problem parameters affect the optimal distribution and convergence rate is examined and simpler to use approximations of the optimal distribution are presented for different settings.

The theoretical results and numerical experiments were derived, designed and performed by Martin Morin under the supervision of Pontus Giselsson. The manuscript was written by Martin Morin and was proofread and revised by Pontus Giselsson.

Paper II

M. Morin and P. Giselsson. "Cocoercivity, Smoothness and Bias in Variance-Reduced Stochastic Gradient Methods". *Numerical Algorithms* **91**:2 (2022), pp. 749–772. DOI: 10.1007/s11075-022-01280-4

This paper concerns the effect of different operator assumptions on the analysis and use of a class of variance-reduced stochastic forward methods. It compares the setting where all operators are gradients of smooth convex functions with the setting where all operators are cocoercive, the former being a special case of the latter. It is in many cases convenient to use the more general assumption and it does not necessarily yield more conservative convergence results. However, this paper shows that this is not the case when considering variance-reduced stochastic methods with the bias of the stochastic estimate having a large effect on the difference between the two settings. We show that unbiasedness is very advantageous in the cocoercive operator case while in the smooth gradient cases similar convergence rates and conditions can be achieved with both biased and unbiased estimates.

The theoretical results and numerical experiments were derived, designed and performed by Martin Morin under the supervision of Pontus Giselsson. The manuscript was written by Martin Morin and was proofread and revised by Pontus Giselsson.

Paper III

M. Morin, S. Banert, and P. Giselsson. "Nonlinear Forward-Backward Splitting with Momentum Correction" (2022). arXiv: 2112.00481v2 [math]. URL: http://arxiv.org/abs/2112.00481v2

On the surface, this paper does not cover the general finite sum monotone inclusion problem outlined in Section 2.2. Instead, it presents a forward-backward method for solving two-operator problems that use a nonlinear resolvent and provides sufficient conditions for the well-posedness and convergence of the algorithm. However, the nonlinear resolvent has a large amount of design freedom which allows the presented algorithm to capture many new or already existing algorithms as special cases, including algorithms for sums of an arbitrary number of operators. This is done by reformulating this finite sum problem to a two-operator problem, either by regrouping the terms or using some product-space or primal-dual formulation.

The original idea behind the main convergence proof came from Sebastian Banert and Pontus Giselsson. The convergence result was finalized and refined by Martin Morin along with with all other results and special cases of the paper. The manuscript was written by Martin Morin and was proofread and revised by Pontus Giselsson and Sebastian Banert.

Paper IV

M. Morin, S. Banert, and P. Giselsson. "Frugal Splitting Operators: Representation, Minimal Lifting and Convergence" (2022). arXiv: 2206.11177v1 [cs, math]. URL: http://arxiv.org/abs/2206.11177v1

In this paper, the modeling ideas of Paper III are expanded such that all frugal splitting operators can be modeled. Informally, frugal splitting operators are splitting operators that evaluate each term of the finite sum monotone inclusion problem exactly once, either directly or via a resolvent. The main result is an equivalence between the class of frugal splitting operators and a class of operators closely related to the fixed point operator used in the non-linear forward-backward method in Paper III. This equivalence leads to new results regarding the memory requirement of the fixed point iteration of any frugal splitting operator and allows us to formulate sufficient conditions for the convergence of said fixed point iteration.

Martin Morin derived all results and wrote the manuscript. Pontus Giselsson and Sebastian Banert proofread and revised the manuscript.

Bibliography

[1] Z. Allen-Zhu. "Katyusha: The First Direct Acceleration of Stochastic Gradient Methods". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2017. ACM, New York, NY, USA, 2017, pp. 1200–1205. ISBN: 978-1-4503-4528-6. DOI: 10.1145/3055399.3055448.

[2] F. Alvarez and H. Attouch. "An Inertial Proximal Method for Maximal Monotone Operators via Discretization of a Nonlinear Oscillator with Damping". *Set-Valued Analysis* **9**:1 (2001), pp. 3–11. DOI: 10.1023/A:1011253113155.

[3] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Second. CMS Books in Mathematics. Springer International Publishing, 2017. ISBN: 978-3-319-48310-8.

[4] A. Bibi, A. Sailanbayev, B. Ghanem, R. M. Gower, and P. Richtárik. *Improving SAGA via a Probabilistic Interpolation with Gradient Descent*. 2018. arXiv: 1806.05633 [math]. URL: http://arxiv.org/abs/1806.05633 (visited on 2018-08-27).

[5] L. M. Briceño-Arias and D. Davis. "Forward-Backward-Half Forward Algorithm for Solving Monotone Inclusions". *SIAM Journal on Optimization* **28**:4 (2018), pp. 2839–2871. DOI: 10.1137/17M1120099.

[6] A. Chambolle and T. Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". *Journal of Mathematical Imaging and Vision* **40**:1 (2011), pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

[7] T. Chavdarova, G. Gidel, F. Fleuret, and S. Lacoste-Julien. "Reducing Noise in GAN Training with Variance Reduced Extragradient". *Advances in Neural Information Processing Systems* **32** (2019), pp. 393–403. URL: https://proceedings.neurips.cc/paper/2019/hash/58a2fc6ed39fd083f55d4182bf88826d-Abstract.html (visited on 2020-12-17).

[8] P. L. Combettes. “Systems of Structured Monotone Inclusions: Duality, Algorithms, and Applications”. *SIAM Journal on Optimization* **23**:4 (2013), pp. 2420–2447. DOI: 10.1137/130904160.

[9] P. L. Combettes and J.-C. Pesquet. “Primal-Dual Splitting Algorithm for Solving Inclusions with Mixtures of Composite, Lipschitzian, and Parallel-Sum Type Monotone Operators”. *Set-Valued and Variational Analysis* **20**:2 (2012), pp. 307–330. DOI: 10.1007/s11228-011-0191-y.

[10] L. Condat. “A Primal–Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms”. *Journal of Optimization Theory and Applications* **158**:2 (2013), pp. 460–479. DOI: 10.1007/s10957-012-0245-9.

[11] D. Davis. *SMART: The Stochastic Monotone Aggregated Root-Finding Algorithm*. 2016. arXiv: 1601.00698 [math]. URL: http://arxiv.org/abs/1601.00698 (visited on 2018-08-27).

[12] D. Davis and W. Yin. “A Three-Operator Splitting Scheme and its Optimization Applications”. *Set-Valued and Variational Analysis* **25**:4 (2017), pp. 829–858. DOI: 10.1007/s11228-017-0421-z.

[13] A. Defazio, F. Bach, and S. Lacoste-Julien. “SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives”. In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 1646–1654. URL: http://papers.nips.cc/paper/5258-saga-a-fast-incremental-gradient-method-with-support-for-non-strongly-convex-composite-objectives.pdf (visited on 2018-08-27).

[14] A. Defazio, J. Domke, and Caetano. “Finito: A Faster, Permutable Incremental Gradient Method for Big Data Problems”. In: *International Conference on Machine Learning*. 2014, pp. 1125–1133. URL: http://proceedings.mlr.press/v32/defazio14.html (visited on 2018-08-27).

[15] P. Giselsson. “Nonlinear Forward-Backward Splitting with Projection Correction”. *SIAM Journal on Optimization* (2021), pp. 2199–2226. DOI: 10.1137/20M1345062.

[16] A. A. Goldstein. “Convex Programming in Hilbert Space”. *Bulletin of the American Mathematical Society* **70**:5 (1964), pp. 709–711. DOI: 10.1090/S0002-9904-1964-11178-2.

[17] R. M. Gower, P. Richtárik, and F. Bach. “Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching”. *Mathematical Programming* **188**:1 (2021), pp. 135–192. DOI: 10.1007/s10107-020-01506-0.

[18] F. Hanzely, K. Mishchenko, and P. Richtárik. “SEGA: Variance Reduction via Gradient Sketching”. In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 2082–2093. URL: http://

papers.nips.cc/paper/7478-sega-variance-reduction-via-gradient-sketching.pdf (visited on 2020-04-30).

[19] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams. "Variance Reduced Stochastic Gradient Descent with Neighbors". In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 2305–2313. URL: http://papers.nips.cc/paper/5919-variance-reduced-stochastic-gradient-descent-with-neighbors.pdf (visited on 2018-08-27).

[20] R. Johnson and T. Zhang. "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 315–323. URL: http://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descent-using-predictive-variance-reduction.pdf (visited on 2018-08-27).

[21] J. Konečný and P. Richtárik. "Semi-Stochastic Gradient Descent Methods". *Frontiers in Applied Mathematics and Statistics* **3** (2017). DOI: 10.3389/fams.2017.00009.

[22] D. Kovalev, S. Horváth, and P. Richtárik. "Don't Jump Through Hoops and Remove Those Loops: SVRG and Katyusha are Better Without the Outer Loop". In: *Proceedings of the 31st International Conference on Algorithmic Learning Theory*. PMLR, 2020, pp. 451–467. URL: https://proceedings.mlr.press/v117/kovalev20a.html (visited on 2021-09-27).

[23] P. Latafat and P. Patrinos. "Asymmetric Forward–Backward–Adjoint Splitting for Solving Monotone Inclusions Involving Three Operators". *Computational Optimization and Applications* **68**:1 (2017), pp. 57–93. DOI: 10.1007/s10589-017-9909-6.

[24] N. Le Roux, M. Schmidt, and F. Bach. "A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets". In: *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 2663–2671. URL: http://papers.nips.cc/paper/4633-a-stochastic-gradient-method-with-an-exponential-convergence-_rate-for-finite-training-sets.pdf (visited on 2018-12-14).

[25] E. S. Levitin and B. T. Polyak. "Constrained Minimization Methods". *USSR Computational mathematics and mathematical physics* **6**:5 (1966), pp. 1–50.

[26] P. L. Lions and B. Mercier. "Splitting Algorithms for the Sum of Two Nonlinear Operators". *SIAM Journal on Numerical Analysis* **16**:6 (1979), pp. 964–979. DOI: 10.1137/0716071.

[27] Y. Malitsky and M. K. Tam. "A Forward-Backward Splitting Method for Monotone Inclusions Without Cocoercivity". *SIAM Journal on Optimization* **30**:2 (2020), pp. 1451–1472. DOI: 10.1137/18M1207260.

[28] M. Morin, S. Banert, and P. Giselsson. "Frugal Splitting Operators: Representation, Minimal Lifting and Convergence" (2022). arXiv: 2206.11177v1 [cs, math]. URL: http://arxiv.org/abs/2206.11177v1.

[29] M. Morin, S. Banert, and P. Giselsson. "Nonlinear Forward-Backward Splitting with Momentum Correction" (2022). arXiv: 2112.00481v2 [math]. URL: http://arxiv.org/abs/2112.00481v2.

[30] M. Morin and P. Giselsson. "Cocoercivity, Smoothness and Bias in Variance-Reduced Stochastic Gradient Methods". *Numerical Algorithms* **91**:2 (2022), pp. 749–772. DOI: 10.1007/s11075-022-01280-4.

[31] M. Morin and P. Giselsson. "Sampling and Update Frequencies in Proximal Variance Reduced Stochastic Gradient Methods" (2020). arXiv: 2002.05545v2 [cs, math]. URL: http://arxiv.org/abs/2002.05545v2.

[32] A. Moudafi and M. Oliny. "Convergence of a Splitting Inertial Proximal Method for Monotone Operators". *Journal of Computational and Applied Mathematics* **155**:2 (2003), pp. 447–454. DOI: 10.1016/S0377-0427(02)00906-8.

[33] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. "SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. JMLR.org, Sydney, NSW, Australia, 2017, pp. 2613–2621. URL: http://proceedings.mlr.press/v70/nguyen17b.html (visited on 2020-04-30).

[34] X. Qian, Z. Qu, and P. Richtárik. "SAGA with Arbitrary Sampling". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 5190–5199. URL: https://proceedings.mlr.press/v97/qian19a.html (visited on 2021-09-27).

[35] E. K. Ryu and B. C. Vũ. "Finding the Forward-Douglas–Rachford-Forward Method". *Journal of Optimization Theory and Applications* **184**:3 (2020), pp. 858–876. DOI: 10.1007/s10957-019-01601-z.

[36] M. Schmidt, N. Le Roux, and F. Bach. "Minimizing Finite Sums with the Stochastic Average Gradient". *Mathematical Programming* **162**:1 (2017), pp. 83–112. DOI: 10.1007/s10107-016-1030-6.

[37] Z. Shi, X. Zhang, and Y. Yu. "Bregman Divergence for Stochastic Variance Reduction: Saddle-Point and Adversarial Prediction". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6033–6043. ISBN: 978-1-5108-6096-4.

[38] P. Tseng. “A Modified Forward-Backward Splitting Method for Maximal Monotone Mappings”. *SIAM Journal on Control and Optimization* **38**:2 (2000), pp. 431–446. DOI: 10.1137/S0363012998338806.

[39] B. C. Vũ. “A Splitting Algorithm for Dual Monotone Inclusions Involving Cocoercive Operators”. *Advances in Computational Mathematics* **38**:3 (2013), pp. 667–681. DOI: 10.1007/s10444-011-9254-8.

[40] L. Xiao and T. Zhang. “A Proximal Stochastic Gradient Method with Progressive Variance Reduction”. *SIAM Journal on Optimization* **24**:4 (2014), pp. 2057–2075. DOI: 10.1137/140961791.

Paper I

Sampling and Update Frequencies in Proximal Variance-Reduced Stochastic Gradient Methods

Martin Morin **Pontus Giselsson**

Abstract

Variance-reduced stochastic gradient methods have gained popularity in recent times. Several variants exist with different strategies for the storing and sampling of gradients and this work concerns the interactions between these two aspects. We present a general proximal variance-reduced gradient method and analyze it under strong convexity assumptions. Special cases of the algorithm include SAGA, L-SVRG and their proximal variants. Our analysis sheds light on epoch-length selection and the need to balance the convergence of the iterates with how often gradients are stored. The analysis improves on other convergence rates found in the literature and produces a new and faster converging sampling strategy for SAGA. Problem instances for which the predicted rates are the same as the practical rates are presented together with problems based on real world data.

Submitted and under review.

1. Introduction

The problem of finding a minimum of a finite sum of functions is common in classification, regression, and general empirical risk minimization. Each term of the objective is in these cases associated with some error or loss corresponding to a particular data point. In contemporary problems, the datasets are typically very large and hence the number of terms in the objective function is large. Traditional iterative minimization algorithms that evaluate the full objective or its gradient each iteration can then become computationally expensive. Stochastic gradient (SG) methods [22] have therefore become the methods of choice in this setting [4], since in each iteration they only evaluate the gradients of a random subset of the terms.

A family of SG methods that have gathered much attention due to their improved convergence properties over ordinary the ordinary SG method are *variance-reduced* SG methods, see [7, 10, 11, 12, 13, 16, 24, 28]. All variance-reduced methods have a memory over previously evaluated gradients and use them to improve the stochastic estimate of the full gradient. Although other differences exists, the main separating property between different variance-reduced stochastic gradient method is how the gradient memory is updated. This work will focus on the effects of how often the memory is updated and of how the stochastic gradient is sampled.

The majority of research into sampling strategies for randomized gradient methods has been on coordinate gradient methods. Instead of randomly selecting one function from a finite sum, coordinate gradient methods select a random set of coordinates of the gradient and update only those. One of the first proposed distributions on how these coordinates should be sampled is to sample proportional to a power of the coordinate-wise gradient Lipschitz constant [15]. An arbitrary distribution is allowed in [20] and [30] argue that the optimal distribution should be proportional the norm of the coordinate-wise gradient at the current iterate. Beyond that, [6, 18, 19, 21, 27] present approaches that allow for a combination of randomized mini-batching and arbitrary sampling.

For stochastic gradient and its variance-reduced variants, importance sampling is not as developed. Variants of importance sampling for the Kaczmarz algorithm and ordinary stochastic gradient are treated in [14, 26]. For variance-reduced methods, [28] allows for importance sampling in the SVRG setting, while [23] analyzes SAGA under importance sampling. The results for SAGA are further improved and generalized in [8, 17] to include arbitrary randomized mini-batching with importance sampling. In this paper, we introduce a general variance-reduced algorithm and prove its linear convergence in the smooth strongly convex regime. The algorithm allows for importance sampling and have, among others, SAGA [7] and L-SVRG [12] as special cases.

The analysis reveals a trade-off between the convergence of the primal iterate (approximate solution) and the dual iterates (stored gradients). For SAGA, where primal and dual updates are coupled, it is crucial to consider this trade-off when designing samplings and we provide a new sampling strategy that improves on the

known convergence rates for SAGA. For algorithms like L-SVRG, where the memory update is independent of the sampling, it is always beneficial in terms of convergence rate to update more often. However, this incurs a higher computational cost so we present an update strategy that balances the computational cost against the convergence rate. Our new rates and computational complexity improve on the previously known results for L-SVRG.

The algorithm in this paper has similarities to the algorithms analyzed in [9] and [29]. Compared to the memorization algorithm in [9], our algorithm allows for a proximal term and has a less restrictive gradient memory update. Our algorithm also allows for importance sampling in SAGA, something that is not supported by the analysis in [29]. Furthermore, the algorithm of [29] is applied to a larger class of monotone inclusion problems, potentially making the analysis more conservative.

2. Preliminaries

Let $\mathbb{R}$ be the set of real numbers. We will work in finite dimensional real spaces $\mathbb{R}^N$. Let $\langle\cdot,\cdot\rangle$ denote the standard Euclidean inner product and let $\|\cdot\|$ be the norm induced by the inner product. The expected value conditioned on the filtration $\mathcal{F}$ is $\mathbb{E}[\cdot|\mathcal{F}]$. The probability of a discrete random variable taking value i is $P(\cdot = i)$. We define $\mathbf{1}_X = 1$ if the predicate X is true, otherwise $\mathbf{1}_X = 0$.

A convex function $f : \mathbb{R}^d \to \mathbb{R}$ is *L-smooth* with $L > 0$ if it is differentiable and its gradient is $\frac{1}{L}$-cocoercive, i.e.,

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq \tfrac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2, \quad \forall x, y \in \mathbb{R}^d.$$

Note that the definition of smoothness implies *L-Lipschitz continuity* of the gradient ∇f. In fact, for convex f, Lipschitz continuity and cocoercivity of ∇f are equivalent [2, Corollary 18.17]. A proper function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is *μ-strongly convex* with $\mu > 0$ if $f - \frac{\mu}{2}\|\cdot\|^2$ is convex.

The subdifferential of a μ-strongly convex function is μ-strongly monotone [2, Example 22.4], i.e.,

$$\langle u - v, x - y\rangle \geq \mu\|x - y\|^2$$

holds $\forall x, y \in \operatorname{dom} \partial f$ and $\forall u \in \partial f(x), \forall v \in \partial f(y)$. A closed, proper and strongly convex function has a unique minimum [2, Corollary 11.17].

The *proximal operator* of a closed, convex and proper function $g : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is defined as

$$\operatorname{prox}_g(z) = \arg\min_x g(x) + \tfrac{1}{2}\|x - z\|^2.$$

Due to strong convexity of $g + \frac{1}{2}\|\cdot - z\|^2$, the minimum exist and is unique. Furthermore, the operator prox_g is *non-expansive*, i.e., Lipschitz continuous with constant 1 [2, Proposition 12.28].

A *Lipschitz distribution* or *Lipschitz sampling* is a probability distribution on $i \in \{1,\dots,n\}$ proportional to the Lipschitz constants L_i of ∇f_i in (1).

3. Problem and Algorithm

We consider the regularized finite sum problem

$$\min_{x\in\mathbb{R}^N} \quad g(x) + F(x), \tag{1}$$

where $g : \mathbb{R}^N \to \mathbb{R} \cup \{\infty\}$ and F is of finite sum form $F(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ with $f_i : \mathbb{R}^N \to \mathbb{R}$ for all $i \in \{1,\dots,n\}$. We will further make the following assumption on (1).

ASSUMPTION 3.1—PROBLEM PROPERTIES
The function $g : \mathbb{R}^N \to \mathbb{R} \cup \{\infty\}$ is closed, convex and proper. For all $i \in \{1,\dots,n\}$, the function $f_i : \mathbb{R}^N \to \mathbb{R}$ is convex, differentiable and L_i-smooth. The function $F : \mathbb{R}^N \to \mathbb{R}$ is μ-strongly convex, differentiable and L-smooth with $L \le \frac{1}{n}\sum_{i=1}^n L_i$.

As a consequence of Assumption 3.1, $g + F$ is closed, proper and μ-strongly convex and hence there exists a unique solution to (1), which we denote $x^\star$. We propose the following proximal variance-reduced stochastic gradient (PVRSG) method for solving (1).

Algorithm 3.1 PVRSG - Proximal Variance-Reduced Stochastic Gradient

Given the function g, the functions $f_1,\dots,f_n$, initial primal and dual points, x^0 and $y_1^0,\dots y_n^0$, iteratively perform the following for $k \in \{0,1,\dots\}$.

Sampling: Randomly sample $(I^k, U_1^k,\dots,U_n^k)$ from $\{1,\dots,n\}\times\{0,1\}^n$.

Primal Update:
$$z^{k+1} = x^k - \frac{\lambda}{n}\left(\frac{1}{p_{I^k}}(\nabla f_{I^k}(x^k) - y_{I^k}^k) + \sum_{i=1}^n y_i^k\right),$$
$$x^{k+1} = \operatorname{prox}_{\lambda g}(z^{k+1}).$$

Dual Update: $y_i^{k+1} = y_i^k + U_i^k(\nabla f_i(x^k) - y_i^k), \quad \forall i \in \{1,\dots,n\}.$

The sampling distributions of $(I^k, U_1^k,\dots,U_n^k)$ for all $k \in \{0,1,\dots\}$ are the same and independent. Furthermore, the distribution is such that $P(I^k = i) = p_i > 0$ and the expected update frequency $\eta_i > 0$, see Definition 3.2, for all $i \in \{1,\dots,n\}$. The step-size satisfies $\lambda > 0$.

In Algorithm 3.1, the primal variable x^k is updated with a stochastic approximation of the standard proximal gradient (PG) step. This approximation becomes better the closer the dual variables $y_1^k,\dots,y_n^k$ are to the true gradients $\nabla f_1(x^k),\dots,\nabla f_n(x^k)$.

The purpose of the dual update is to bring these dual variables closer to the true gradients by updating a selection of them with the corresponding gradients at the current iteration. The more often a dual variable is updated, the closer it will be to the true gradient on average. We quantify the frequency of the dual updates with the *expected update frequency*, or in short *update frequency*.

DEFINITION 3.2—EXPECTED UPDATE FREQUENCY
Let $U_1^k,\dots,U_n^k$ be given by the sampling in Algorithm 3.1. The expected update frequency of the ith dual variable is

$$\eta_i = \mathbb{E}[U_i^k|\mathcal{F}^k],$$

where $\mathcal{F}^k = \cup_{i=1}^k \mathcal{X}^i$ and $\mathcal{X}^k = \{x^k, y^k, I^{k-1}, U_1^{k-1},\dots,U_n^{k-1}\}$.

Note, the expected update frequency does not depend on the iteration number k since $(I^k, U_1^k,\dots,U_n^k)$ is independently sampled and its distribution does not depend on k.

By the nature of the dual update, the dual variables $y_1^k,\dots,y_n^k$ does not necessarily contain gradients evaluated at the same point, i.e., there might not exists $\hat{x}$ such that $\frac{1}{n}\sum_{i=1}^n y_i^k = \nabla F(\hat{x})$. However, it turns out that if, for all $k \in \{0,1,\dots\}$, there exists such $\hat{x}$, an improved analysis can be made. This leads to the following assumption.

ASSUMPTION 3.3—COHERENT DUAL UPDATE
For all $i \in \{1,\dots,n\}$, the initial dual variables satisfy $y_i^0 = \nabla f_i(\hat{x})$ for some $\hat{x}$ and it holds that

$$U_1^k = U_2^k = \cdots = U_n^k$$

for all $k \in \{0,1,\dots\}$.

Algorithm 3.1 contains many special cases with different samplings leading to different algorithms. The two main algorithms of relevance are SAGA [7] and L-SVRG [12].

SAGA: SAGA [7] only evaluates one gradient each iteration and always save it, i.e., the sampling is defined such that $U_i^k = \mathbf{1}_{i=I^k}$ for all $i \in \{1,\dots,n\}$ which gives the update frequency $\eta_i = p_i$.

L-SVRG: L-SVRG [12] is inspired by SVRG [10], but, instead of a deterministic update of the dual variables, the dual update is based on a weighted coin toss, i.e., $U_i^k = \mathbf{1}_{Q^k<q}$ where $0 < q \leq 1$ and Q^k is independently and uniformly sampled from $[0,1]$. The expected update frequency is $\eta_i = q$. Assumption 3.3 is satisfied if the dual variables are initialized in the same point, i.e., there exists $\hat{x}$ s.t. $y_i^0 = \nabla f_i(\hat{x})$ for all $i \in \{1,\dots,n\}$.

We introduce two more special cases to examine the effects of Assumption 3.3 and the expected update frequency η_i.

IL-SVRG (Incoherent Loopless-SVRG): IL-SVRG purposefully break the coherent dual assumption, Assumption 3.3, in L-SVRG. Each dual variable is independently updated, $U_i^k = \mathbf{1}_{Q_i^k < q}$ where $0 < q \leq 1$ and Q_i^k is independently and uniformly sampled from $[0,1]$. The update frequency is the same as for L-SVRG, $\eta_i = q$.

q-SAGA: In q-SAGA [9], for each iteration, $q \leq n$ indices are sampled uniformly and independently from $\{1,\ldots,n\}$ and the corresponding dual variables are updated. Hence, the sampling is $U_i^k = \mathbf{1}_{i \in J_q}$ where J_q is the set of sampled indices and the update frequency becomes $\eta_i = q/n$.

4. Convergence Analysis

We here analyze Algorithm 3.1 under Assumption 3.1 and prove its linear convergence, both with and without the coherent dual update assumption, Assumption 3.3. The main results of this analysis can be found in Theorems 4.7 and 4.8 and all proofs will be deferred to Sections A and B. Before moving forward with the analysis, we introduce the following necessary quantities that will be used in our Lyapunov analysis.

DEFINITION 4.1
Let $x^\star$ be the solution to (1) and $y_i^\star = \nabla f_i(x^\star)$ for all $i \in \{1,\ldots,n\}$. With $y = (y_1,\ldots,y_n)$, where $y_i \in \mathbb{R}^n$ for $i \in \{1,\ldots,n\}$, we define

$$\mathcal{P}(x) = \|x - x^\star\|^2 - 2\lambda\langle \nabla F(x) - \nabla F(x^\star), x - x^\star\rangle + \lambda^2 \mathcal{V}(x)$$

and

$$\mathcal{D}(y) = \sum_{i=1}^n (1 - \eta_i + \tfrac{1}{\gamma_i})\widehat{\gamma}_i \|y_i - y_i^\star\|^2 - (1+\delta^{-1})\lambda^2 \|\tfrac{1}{n}\sum_{i=1}^n y_i - y_i^\star\|^2,$$

where

$$\mathcal{V}(x) = \sum_{i=1}^n \tfrac{(1+\delta)}{n^2 p_i}(\tfrac{\eta_i \gamma_i}{\delta} + 1)\|\nabla f_i(x) - \nabla f_i(x^\star)\|^2 - \delta\|\nabla F(x) - \nabla F(x^\star)\|^2$$

with $\gamma_i \geq 0$, $\delta > 0$ and $\widehat{\gamma}_i = \gamma_i \frac{(1+\delta^{-1})\lambda^2}{n^2 p_i}$. If $\gamma_i = 0$ we define $\frac{\gamma_i}{\gamma_i} := 1$.

The variables γ_i and δ are meta-parameters and which will be specified in each proof of the main convergence theorems. The base of our convergence analysis will be the following proposition.

PROPOSITION 4.2
Let the filtration $\mathcal{F}^k = \cup_{i=1}^k \mathcal{X}^i$ be given by the state $\mathcal{X}^k = \{x^k, y^k, I^{k-1}, U_1^{k-1}, \ldots, U_n^{k-1}\}$. If Assumption 3.1 holds, the iterates of Algorithm 3.1 satisfy

$$\mathbb{E}\left[\|x^{k+1} - x^\star\|^2 + \sum_{i=1}^n \widehat{\gamma}_i \|y_i^{k+1} - y_i^\star\|^2 \middle| \mathcal{F}^k\right] \leq \mathcal{P}(x^k) + \mathcal{D}(y^k), \tag{2}$$

where $x^\star$ is the unique solution of (1) and $y_i^\star = \nabla f_i(x^\star)$. See Definition 4.1 for $\mathcal{P}$, $\mathcal{D}$, and $\mathcal{V}$.

If the primal updates satisfy

$$\mathcal{P}(x^k) \leq (1-\rho_P)\|x^k - x^\star\|^2 \tag{3}$$

and the dual updates satisfy

$$\mathcal{D}(y^k) \leq (1-\rho_D)\sum_{i=1}^n \widehat{\gamma}_i \|y_i^k - y_i^\star\|^2 \tag{4}$$

with $\rho_P, \rho_D \in (0,1]$ then Algorithm 3.1 converges linearly according to

$$\mathbb{E}\left[\|x^k - x^\star\|^2 + \sum_{i=1}^n \widehat{\gamma}_i \|y_i^k - y_i^\star\|^2\right] \in O((1-\min(\rho_P,\rho_D))^k).$$

Proof. See Section A. □

If we can find algorithm parameters and meta-parameters such that Algorithm 3.1 satisfy the primal and dual contractions, (3) and (4) respectively, this proposition proves that Algorithm 3.1 convergence to a solution. The following lemma provides these necessary contraction results.

LEMMA 4.3—PRIMAL CONTRACTION
Let Assumption 3.1 hold, the primal iterates of Algorithm 3.1 satisfy the primal contraction (3) with

$$\rho_P = \mu\lambda(2-\nu\lambda)$$

where $\nu = \max_i (1+\delta^{-1})\frac{L_i\eta_i\gamma_i}{np_i} + (1+\delta)\frac{L_i}{np_i} - \delta\mu$.

Proof. See Section A. □

LEMMA 4.4—DUAL CONTRACTION
Let Assumption 3.1 and $\gamma_i > 0$ hold for all $i \in \{1,\ldots,n\}$, the dual iterates of Algorithm 3.1 satisfy the dual contraction (4) with

$$\rho_D = \min_i \eta_i - \tfrac{1}{\gamma_i}.$$

Proof. See Section A. □

LEMMA 4.5—DUAL CONTRACTION - COHERENT UPDATES
Let Assumption 3.1 and 3.3 hold and, for all $i \in \{1,\ldots,n\}$, let γ_i be such that $\gamma_i \geq 0$ and $\frac{L_i}{np_i} \leq \mu$ if $\gamma_i = 0$. The dual iterates of Algorithm 3.1 satisfy the dual contraction (4) with

$$\rho_D = \min_i \begin{cases} \eta - (1-\frac{np_i}{L_i}\mu)\frac{1}{\gamma_i} & \text{if } \gamma_i > 0 \\ 1 & \text{if } \gamma_i = 0 \end{cases}.$$

Proof. See Section A. □

The proofs of our main convergence results, Theorems 4.7 and 4.8, consist of establishing meta-parameters δ and γ_i for all $i \in \{1,\dots,n\}$ such that these contractions are sufficiently large, i.e., $\min(\rho_P,\rho_D) > 0$. However, in order to establish these meta-parameters, the following lemma regarding the relationship between the smoothness constants $L_1,\dots,L_n$ and strong convexity constant μ is needed.

LEMMA 4.6
Let L_i and μ be from Assumption 3.1 and p_i from Algorithm 3.1, then $\max_i \frac{L_i}{np_i} \geq \mu$. *Furthermore, if* $\max_i \frac{L_i}{np_i} = \mu$ *then* $\frac{L_i}{np_i} = \mu$ *for all* $i \in \{1,\dots,n\}$.

Proof. See Section A. □

The main convergence theorems can now be stated.

THEOREM 4.7—PVRSG CONVERGENCE
Given $\max_i \frac{L_i}{np_i} > \mu$ *and Assumption 3.1, if there exists* $\rho \in (0, \min_i \eta_i)$ *such that*

$$\begin{aligned}\rho &= \mu\lambda(2-\nu\lambda)\\ \nu &= \min_{\delta>0} \max_i (1+\delta^{-1})\tfrac{L_i}{np_i}\tfrac{\eta_i}{\eta_i-\rho} + (1+\delta)\tfrac{L_i}{np_i} - \delta\mu,\end{aligned}$$

then the iterates of Algorithm 3.1 converge according to

$$\mathbb{E}\left[\|x^k - x^\star\|^2 + \sum_{i=1}^n \widehat{\gamma}_i \|y_i^k - y_i^\star\|^2\right] \in O((1-\rho)^k)$$

where $\widehat{\gamma}_1,\dots,\widehat{\gamma}_n$ *are given by* $\widehat{\gamma}_i = \frac{\lambda^2}{n^2 p_i}\frac{1}{\eta_i-\rho}(1+\frac{1}{\delta^\star})$ *and* $\delta^\star$ *is the unique minimizer of the minimization problem defining* ν.

If instead $\max_i \frac{L_i}{np_i} = \mu$ *then* $\nu = \mu + \mu \max_i \frac{\eta_i}{\eta_i-\rho}$ *and the convergence is such that*

$$\mathbb{E}\left[\|x^k - x^\star\|^2 + \sum_{i=1}^n \tfrac{\lambda^2}{n^2 p_i}\tfrac{1}{\eta_i-\tilde{\rho}} \|y_i^k - y_i^\star\|^2\right] \in O((1-\tilde{\rho})^k)$$

holds for all $\tilde{\rho} \in (0,\rho)$.

Proof. See Section B. □

THEOREM 4.8—PVRSG CONVERGENCE - COHERENT DUAL UPDATES
Given Assumption 3.1 and 3.3 and $\max_i \frac{L_i}{np_i} > \mu$, *if there exists* $\rho \in (0,\eta)$ *such that*

$$\begin{aligned}\rho &= \mu\lambda(2-\nu\lambda)\\ \nu &= \mu + \left(\max_i \tfrac{L_i}{np_i} - \mu\right)\left(1+\sqrt{\tfrac{\eta}{\eta-\rho}}\right)^2,\end{aligned}$$

then the iterates of Algorithm 3.1 converge according to

$$\mathbb{E}\left[\|x^k - x^\star\|^2 + \sum_{i=1}^n \widehat{\gamma}_i \|y_i^k - y_i^\star\|^2\right] \in O((1-\rho)^k)$$

where $\widehat{\gamma}_1, \ldots, \widehat{\gamma}_n$ *are given by* $\widehat{\gamma}_i = \frac{\lambda^2}{n^2 p_i} \frac{1}{\eta - \rho} \max(0, 1 - \frac{n p_i \mu}{L_i}) \left(1 + \sqrt{\frac{\eta - \rho}{\eta}}\right)$.

If instead $\max_i \frac{L_i}{n p_i} = \mu$ *then* $\nu = \mu$ *and* $\widehat{\gamma}_i = 0$ *for all* $i \in \{1, \ldots, n\}$. *Furthermore, the rate is not restricted to* $\rho \in (0, \eta)$, *but to* $\rho \in (0, 1]$.

Proof. See Section B. □

Note that the theorems do not provide explicit expressions for the convergence rates, but instead implicitly define them. Because of this, when we reference the rates of these theorems, we will refer to a numerically computed value. This computation is done with a combination of convex optimization—for computing ν—and bisection—for finding ρ such that $0 = \rho - \mu\lambda(2 - \nu\lambda)$.

Apart from the coherent dual assumption, our convergence results depend only on the update frequency η_i and not on the specifics of the dual sampling that generated it. Comparing the two theorems, we see that coherent dual updates have greatest effect when the problem is well-conditioned, i.e., when $\frac{L_i}{\mu}$ is small for all $i \in \{1, \ldots, n\}$. This stems from the fact that the contraction factor for coherent updates in Lemma 4.5 goes towards the contraction factor without coherent updates in Lemma 4.4 when $\frac{L_i}{\mu}$ increases for all $i \in \{1, \ldots, n\}$.

In the extremely well-conditioned case when $\max_i \frac{L_i}{n p_i} = \mu$, we see that $\widehat{\gamma}_i = 0$ for all $i \in \{1, \ldots, n\}$ and the dual term of the Lyapunov function vanishes completely in Theorem 4.8. This is possible due the fact that, in this case, the primal update actually is equal to the true proximal gradient step, regardless of the dual variables. Also notice that we, as expected, recover the rate for ordinary proximal-gradient.

5. Special Cases

In order to provide easily compared rates for SAGA and L-SVRG, we present simplified corollaries of Theorems 4.7 and 4.8 that provide explicit rates. These rates are by construction conservative compared to the theorems but still improve on previously known best rates. The corollaries also provide explicit upper bounds on the step-sizes. Unlike the rates, the bounds are not conservative and match the implicit bounds in Theorems 4.7 and 4.8. The proofs of the corollaries are found in Section C.

COROLLARY 5.1—SAGA - CONSERVATIVE BOUNDS
Given Assumption 3.1, the maximal and recommended step-sizes, $\lambda_{\max}$ *and* $\lambda^\star$, *for*

SAGA are:

$$\text{If } p_i = \frac{1}{n}, \qquad \lambda_{\max} = \frac{2}{C_U L_{\max}}, \qquad \lambda^\star = \frac{2}{C_U L_{\max} + n\mu + \sqrt{(C_U L_{\max})^2 + (n\mu)^2}}.$$

$$\text{If } p_i \propto L_i, \qquad \lambda_{\max} = \frac{2}{C_L \bar{L}}, \qquad \lambda^\star = \frac{2}{C_L \bar{L} + p_{\min}^{-1}\mu + \sqrt{(C_L \bar{L})^2 + (p_{\min}^{-1}\mu)^2}}.$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$, $C_U = 2 + 2\sqrt{1 - \frac{\mu}{L_{\max}}}$, *and* $C_L = 2 + 2\sqrt{1 - \frac{\mu}{\bar{L}}}$. *The iterates converge with a rate of* $\mathbb{E}\,\|x^k - x^\star\|^2 \in O\big((1 - \mu\lambda^\star)^k\big)$ *when then step-size* $\lambda^\star$ *is used.*

Proof. See Section C. □

COROLLARY 5.2—L-SVRG - CONSERVATIVE BOUNDS
Given Assumption 3.1, the maximal and recommended step-sizes, $\lambda_{\max}$ *and* $\lambda^\star$*, for L-SVRG are:*

$$\text{If } p_i = \frac{1}{n}, \qquad \lambda_{\max} = \frac{2}{D_U L_{\max}}, \qquad \lambda^\star = \frac{2}{D_U L_{\max} + \eta^{-1}\mu + \sqrt{(D_U L_{\max})^2 + (\eta^{-1}\mu)^2}}.$$

$$\text{If } p_i \propto L_i, \qquad \lambda_{\max} = \frac{2}{D_L \bar{L}}, \qquad \lambda^\star = \frac{2}{D_L \bar{L} + \eta^{-1}\mu + \sqrt{(D_L \bar{L})^2 + (\eta^{-1}\mu)^2}}.$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$, $D_U = 4 - 3\frac{\mu}{L_{\max}}$ *and* $D_L = 4 - 3\frac{\mu}{\bar{L}}$. *Note that* $4 > D_U \geq D_L \geq 1$. *The iterates converge with a rate of* $\mathbb{E}\,\|x^k - x^\star\|^2 \in O\big((1 - \mu\lambda^\star)^k\big)$ *when then step-size* $\lambda^\star$ *is used.*

Proof. See Section C. □

The *recommended* step-sizes $\lambda^\star$ are the step-sizes we found that yield the best explicit rates. However, they are not necessarily optimal w.r.t. the implicit rates in Theorems 4.7 and 4.8.

6. Sampling Design

Before we present our suggested sampling distributions for SAGA and L-SVRG, we make a few remarks on the parameter selection in Algorithm 3.1.

A higher update frequency always yields faster convergence. However, more frequent dual updates incur a higher computational cost since this require more gradient evaluations. The update frequencies therefore needs to be based on the total computational complexity of reaching an ϵ-accurate solution in expectation, i.e., $\mathbb{E}\,\|x^k - x^\star\|^2 \leq \epsilon$.

The choice of distribution of $p_1, \ldots, p_n$ does not change the iteration cost so it can be optimized by only considering the convergence rate, not the computational

complexity. If the update frequencies are uniform, $\eta_i = \eta_j, \forall i,j \in \{1,\dots,n\}$, the meta-parameters $\gamma_1,\dots,\gamma_n$ in Theorems 4.7 and 4.8 also are uniform. In this case, it can be seen that Lipschitz sampling maximizes the convergence rate, i.e., $p_i \sim L_i$. However, this is not necessarily true in cases with non-uniform update frequencies.

For SAGA, the expected update frequencies depend on $p_1,\dots,p_n$ and we can therefore not use the optimal choice of uniform update frequencies and Lipschitz sampling of I^k. Instead, we present choice of $p_1,\dots,p_n$ that considers the dependency between the primal and dual update and blends Lipschitz and uniform sampling. The proposed distribution improves on all other samplings in terms of convergence rate and computational complexity, see Corollary 6.1.

COROLLARY 6.1—SAGA - IMPROVED SAMPLING
Let the sampling distribution and step-size be

$$p_i \propto 4L_i + n\mu + \sqrt{(4L_i)^2 + (n\mu)^2}, \quad \lambda = \tfrac{2}{S}$$

where $S = \frac{1}{n}\sum_{i=1}^n (4L_i + n\mu + \sqrt{(4L_i)^2 + (n\mu)^2})$. *SAGA converges with a rate of* $\mathbb{E}\,\|x^k - x^\star\|^2 \in O\big((1-\mu\lambda)^k\big)$ *and achieves an* ϵ*-accurate solution in expectation within*

$$O\big(\tfrac{1}{2}\big(\tfrac{1}{n}\sum_{i=1}^n \tfrac{4L_i}{\mu} + n + \sqrt{(\tfrac{4L_i}{\mu})^2 + n^2}\big)\log\tfrac{1}{\epsilon}\big)$$

iterations.

Proof. See Section C. □

Unlike in SAGA, $\eta_1,\dots,\eta_n$ are always uniform in L-SVRG and can be tuned independently of the primal update. As remarked on earlier, Lipschitz sampling is then the optimal primal sampling and is therefore used in the following complexity results. We assume one gradient evaluation is needed in the primal update [1] and that, in expectation, $n\eta$ are needed in the dual update.

COROLLARY 6.2—L-SVRG - COMPUTATIONAL COMPLEXITY
Let Lipschitz sampling—$p_i \sim L_i$ *for all* $i \in \{1,\dots,n\}$*—and the step-size from Corollary 5.2 be used. L-SVRG achieves an* ϵ*-accurate solution within*

$$O\Big((1+n\eta)\Big(D_L\tfrac{\bar{L}}{\mu} + \tfrac{1}{\eta}\Big)\log\tfrac{1}{\epsilon}\Big)$$

iterations where $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$ *and* D_L *is given by Corollary 5.2. The expected update frequency that minimizes the complexity, and the corresponding complexity,*

[1] This assumes all dual variables $y_1^k,\dots,y_n^k$ are stored. One benefit of PVRSG instances that satisfy Assumption 3.3 is that they can be implemented without storing all dual variables at the cost of one extra gradient evaluation. We use the higher memory cost variant in order to compare to SAGA under equal memory requirements.

are

$$\eta^\star = \sqrt{\frac{\mu}{nD_L\bar{L}}} \quad \textit{and} \quad O\left(\left(\sqrt{n}+\sqrt{D_L\frac{\bar{L}}{\mu}}\right)^2 \log\tfrac{1}{\epsilon}\right).$$

Proof. See Section C. □

The complexity of L-SVRG in *Corollary* 6.2 is worse than that of SAGA in Corollary 6.1 when $n > 2$. The cheaper iteration cost of SAGA clearly outweighs loss of the coherent dual update, Assumption 3.3. With the choice of update frequency for L-SVRG in Corollary 6.2, the expected time between dual updates is $\frac{1}{\eta^\star} \propto \sqrt{n\frac{\bar{L}}{\mu}}$. This is in contrast to most results for SVRG and L-SVRG that have epoch lengths proportional to either n or $\frac{L}{\mu}$ [1, 10, 12, 25, 28].

7. Numerical Experiments

All algorithms have been implemented in `Julia` [3] and can be found at `https://github.com/mvmorin/VarianceReducedSG.jl`.

Simple Least Squares The analysis predicts performance accurately for a one-dimensional least squares problem,

$$\min_x \tfrac{1}{n}\sum_{i=1}^n (a_i x - b_i)^2.$$

A comparison of theoretical and practical rates for this problem is found in Figure 1. The data a_i and b_i have been independently drawn from a unit normal distribution and the number of functions is $n = 100$.

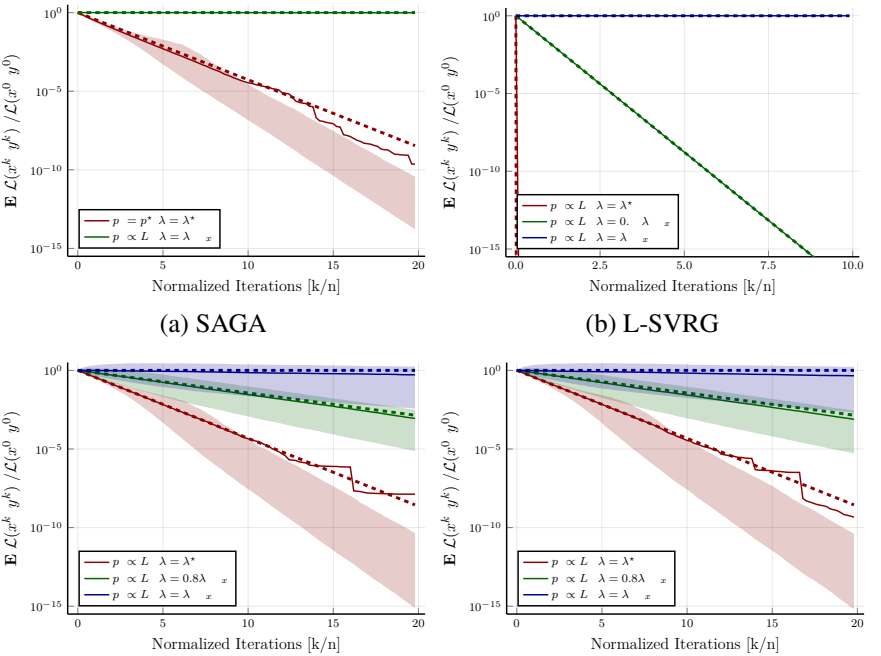
For L-SVRG, Figure 1 shows fast convergence and very narrow 5-95 percentile—it is not even visible. This is due to the $\max_i \frac{L_i}{np_i} = \mu$ condition being satisfied and then the gradient estimate is exact. Since the condition number of the problem is equal to 1, it is possible to solve the problem in one iteration.

For SAGA, we see in Figure 1 that both the maximal and optimal step-sizes are predicted well. However, note that the sampling distribution $p_1,\dots,p_n$ are not the same for the two cases.

Comparing q-SAGA and IL-SVRG in Figure 1, we see similar performance. This was predicted by Theorem 4.7 since, despite the dual updates being different, the algorithms have the same expected update frequency. Comparing to L-SVRG in Figure 1 we see the huge impact of the coherent dual assumption in this very well-conditioned case.

Lasso Problem Here we consider a Lasso regression problem of the form

$$\min \|Ax - b\|_2^2 + \xi\|x\|_1,$$



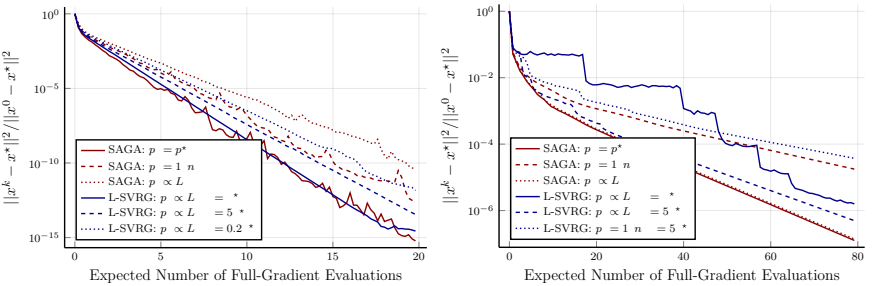
(a) SAGA (b) L-SVRG

(c) q-SAGA (d) IL-SVRG

Figure 1. One-dimensional least squares. The expected value $\mathbb{E}\,\mathcal{L}(x^k, y^k)$ is estimated with the average of 10000 runs where $\mathcal{L}(x, y)$ is taken from Theorems 4.7 and 4.8. The shaded areas represent the 5-95 percentile of the runs. The dashed lines are the predicted rates. The step-sizes $\lambda^\star$ and $\lambda^{\max}$ are the optimal and maximal step-sizes according to Theorems 4.7 and 4.8. The expected update frequency is $\eta = \frac{1}{n}$ for all algorithms except SAGA where it depends on the sampling p_i. The sampling $p_i^\star$ is from Corollary 6.1.

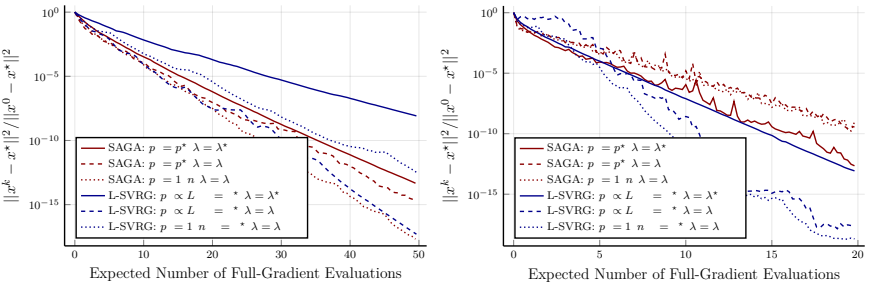
where the matrix A and vector b consist of the features and classes from different datasets from the `LibSVM` database [5]. The regularization parameter ξ is tuned for each problem such that the solution have roughly 15-20% sparsity. SAGA and L-SVRG with different sampling and update frequencies are compared in Figure 2(a)-(b). L-SVRG was tested with $\eta \in \{0.2\eta^\star, \eta^\star, 5\eta^\star\}$ and either Lipschitz or uniform sampling and the three best perform configuration are shown in Figure 2. Further comparisons between SAGA and L-SVRG with larger step-size choices can be found in Figure 2(c)-(d). In these experiments the regularization parameter was set to $\xi = 0$.

We see that it is sometime possible achieve better convergence rate by deviating from the optimal parameter choices in Corollaries 5.1, 5.2, 6.1 and 6.2. However, these are single realizations of random processes and there will be variance between



(a) Lasso: `ijcnn1`, $\xi = 0.001$, $n = 49990$, $\frac{\bar{L}}{\mu} \approx 165$

(b) Lasso: `protein`*, $\xi = 0.0001$, $n = 17766$, $\frac{\bar{L}}{\mu} \approx 3 \cdot 10^6$



(c) Least Squares: `splice`, $\xi = 0$, $n = 1000$, $\frac{\bar{L}}{\mu} \approx 806$

(d) Least Squares: `acoustic`, $\xi = 0$, $n = 78823$, $\frac{\bar{L}}{\mu} \approx 1480$

Figure 2. The *Expected Number of Full-Gradient Evaluations* is the number of full gradient evaluations the algorithms is expected to perform in k iterations, $\frac{1}{n}k$ for SAGA and $(\frac{1}{n} + \eta)k$ for L-SVRG. The step-sizes, $\lambda^\star$ or $\lambda_{\max}$ are taken from the corresponding result in Corollary 5.1-5.2, $\lambda^\star$ is used if no step-size is given. The sampling $p_i^\star$ is from Corollary 6.1 and $\eta^\star$ is the update frequency from Corollary 6.2. The average condition number $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$. The `protein` dataset has a feature consisting of only zeros. `protein`* has this feature removed in order to preserve strong convexity.

runs, especially for the larger step-sizes. The consistency of our suggested parameter choices should be noted though. Especially SAGA with the sampling from Corollary 6.1 are always among the better alternatives.

Note the slow step-like convergence of L-SVRG with $\eta = \eta^\star$ in the `protein`* example. The faster convergence of $\eta = 5\eta^\star$ suggest that $\eta^\star$ does not properly balance the primal and dual updates. Since we perform a worst case analysis, there are many reasons for why this might be the case. Our analysis also only focus on asymptotic linear rates and does not capture transient behavior. The last point is especially important when considering very ill-conditioned or maybe even non-strongly con-

vex problems. In these cases, the transient phase is the most important part since the achievable linear rates are very small or zero.

8. Conclusion

A general stochastic variance-reduced gradient method has been analyzed and problems have been presented where the predicted rates are close to real world rates. We have demonstrated the need to balance the updates of the primal and dual variables. For L-SVRG, we presented a new condition number dependent update probability for the dual variables. For SAGA, and other methods where the dual update depends on the primal update, the primal sampling needs to consider both updates. Lipschitz sampling, which appears to be optimal for methods with independent dual updates, can for SAGA lead to slow convergence. We have presented a new sampling for SAGA that balances the primal and dual update and consistently performs well.

A. Proofs of Proposition and Lemmas

Proof of Proposition 4.2. Let $x^\star$ be the unique solution to (1). With g being proper, closed and convex, the primal updates satisfy

$$
\begin{aligned}
&\mathbb{E}[\|x^{k+1} - x^\star\|^2 | \mathcal{F}^k] \\
&\quad = \mathbb{E}[\|\mathrm{prox}_{\lambda g}(z^{k+1}) - \mathrm{prox}_{\lambda g}(x^\star - \lambda \nabla F(x^\star))\|^2 | \mathcal{F}^k] \\
&\quad \leq \mathbb{E}[\|x^k - \tfrac{\lambda}{n}\big(\tfrac{1}{p_{I^k}}(\nabla f_{I^k}(x^k) - y_{I^k}^k) + \textstyle\sum_{i=1}^n y_i^k\big) - x^\star + \lambda \nabla F(x^\star)\|^2 | \mathcal{F}^k] \\
&\quad = \|(x^k - \lambda \nabla F(x^k)) - (x^\star - \lambda \nabla F(x^\star))\|^2 \\
&\qquad + \lambda^2 \mathbb{E}\left[\left\|\big(\tfrac{1}{np_{I^k}} \nabla f_{I^k}(x^k) - \nabla F(x^k)\big) - \big(\tfrac{1}{np_{I^k}} y_{I^k}^k - \tfrac{1}{n}\textstyle\sum_{i=1}^n y_i^k\big)\right\|^2 | \mathcal{F}^k\right] \\
&\quad = \|x^k - x^\star\|^2 - 2\lambda \langle \nabla F(x^k) - \nabla F(x^\star), x^k - x^\star \rangle + \lambda^2 \|\nabla F(x^k) - \nabla F(x^\star)\|^2 \\
&\qquad + \lambda^2 \mathbb{E}\left[\left\|\big(\tfrac{1}{np_{I^k}} \nabla f_{I^k}(x^k) - \nabla F(x^k)\big) - \big(\tfrac{1}{np_{I^k}} y_{I^k}^k - \tfrac{1}{n}\textstyle\sum_{i=1}^n y_i^k\big)\right\|^2 | \mathcal{F}^k\right].
\end{aligned}
\tag{5}
$$

The first equality is given by the solution being a fixed point to the proximal-gradient update $x^\star = \mathrm{prox}_{\lambda g}(x^\star - \lambda \nabla F(x^\star))$ [2, Corollary 28.9]. The first inequality is due to the non-expansiveness of $\mathrm{prox}_{\lambda g}$. The second to last equality is given by $\mathbb{E}\,\|X\|^2 = \|\mathbb{E}\,X\|^2 + \mathbb{E}\,\|X - \mathbb{E}\,X\|^2$ where X is a random variable.

The last term in (5) satisfies the following upper bound for all $\delta > 0$:

$$\begin{aligned}
&\mathbb{E}[\|(\tfrac{1}{np_{I^k}}\nabla f_{I^k}(x^k) - \nabla F(x^k)) - (\tfrac{1}{np_{I^k}}y^k_{I^k} - \tfrac{1}{n}\textstyle\sum_{i=1}^n y_i^k)\|^2|\mathcal{F}^k] \\
&\quad = \mathbb{E}[\|\tfrac{1}{np_{I^k}}(\nabla f_{I^k}(x^k) - \nabla f_{I^k}(x^\star)) - (\nabla F(x^k) - \nabla F(x^\star)) \\
&\qquad\quad - \tfrac{1}{np_{I^k}}(y^k_{I^k} - y^\star_{I^k}) + \tfrac{1}{n}\textstyle\sum_{i=1}^n (y_i^k - y_i^\star)\|^2|\mathcal{F}^k] \\
&\quad \leq (1+\delta)\mathbb{E}[\|\tfrac{1}{np_{I^k}}(\nabla f_{I^k}(x^k) - \nabla f_{I^k}(x^\star)) - (\nabla F(x^k) - \nabla F(x^\star))\|^2|\mathcal{F}^k] \\
&\qquad + (1+\delta^{-1})\mathbb{E}[\|\tfrac{1}{np_{I^k}}(y^k_{I^k} - y^\star_{I^k}) - \tfrac{1}{n}\textstyle\sum_{i=1}^n (y_i^k - y_i^\star)\|^2|\mathcal{F}^k] \qquad (6) \\
&\quad = (1+\delta)\big(\mathbb{E}[\|\tfrac{1}{np_{I^k}}(\nabla f_{I^k}(x^k) - \nabla f_{I^k}(x^\star))\|^2|\mathcal{F}^k] - \|\nabla F(x^k) - \nabla F(x^\star)\|^2\big) \\
&\qquad + (1+\delta^{-1})\big(\mathbb{E}[\|\tfrac{1}{np_{I^k}}(y^k_{I^k} - y^\star_{I^k})\|^2|\mathcal{F}^k] - \|\tfrac{1}{n}\textstyle\sum_{i=1}^n (y_i^k - y_i^\star)\|^2\big) \\
&\quad = (1+\delta)\textstyle\sum \frac{1}{n^2 p_i}\|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2 - (1+\delta)\|\nabla F(x^k) - \nabla F(x^\star)\|^2 \\
&\qquad + (1+\delta^{-1})\textstyle\sum \frac{1}{n^2 p_i}\|y_i^k - y_i^\star\|^2 - (1+\delta^{-1})\|\frac{1}{n}\textstyle\sum_{i=1}^n (y_i^k - y_i^\star)\|^2.
\end{aligned}$$

The inequality is given by Young's inequality, the second to last equality is given by $\mathbb{E}\|X - \mathbb{E}X\|^2 = \mathbb{E}\|X\|^2 - \|\mathbb{E}X\|^2$ where X is a random variable.

The dual updates satisfy

$$\begin{aligned}
\|y_i^{k+1} - y_i^\star\|^2 &= \|y_i^k + U_i^k(\nabla f_i(x^k) - y_i^k) - y_i^\star\|^2 \\
&= (1 - U_i^k)\|y_i^k - y_i^\star\|^2 + U_i^k\|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2
\end{aligned}$$

since $U_i^k \in \{0,1\}$. Summing over all terms, taking expected value and using linearity of the expected value give

$$\begin{aligned}
\mathbb{E}[\textstyle\sum_{i=1}^n \widehat{\gamma}_i\|y_i^{k+1} - y_i^\star\|^2|\mathcal{F}^k] &= \textstyle\sum_{i=1}^n (1-\eta_i)\widehat{\gamma}_i\|y^k - y^\star\|^2 \\
&\quad + \textstyle\sum_{i=1}^n \eta_i\widehat{\gamma}_i\|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2.
\end{aligned} \qquad (7)$$

Adding (7) to (5) and substituting in (6) and using the definition $\frac{\gamma_i}{\gamma_i} = 1$ when $\gamma_i = 0$ then yield (2). Applying (3) and (4), using the law of total expectation and telescoping the inequalities give the stated rate. □

Proof of Lemma 4.3. First note that μ-strong monotonicity and the Cauchy–Schwarz inequality imply

$$\|\nabla F(x^k) - \nabla F(x^\star)\| \geq \mu\|x^k - x^\star\|. \qquad (8)$$

Consider the terms of $\mathcal{V}(x^k)$. Using (8) and Cauchy–Schwarz in the last term yield

$$\begin{aligned}
\|\nabla F(x^k) - \nabla F(x^\star)\|^2 &\geq \mu\|\nabla F(x^k) - \nabla F(x^\star)\|\|x^k - x^\star\| \\
&\geq \mu\langle\nabla F(x^k) - \nabla F(x^\star), x^k - x^\star\rangle.
\end{aligned}$$

Using $\frac{1}{L_i}$-cocoercivity of ∇f_i in the first term of $\mathcal{V}(x^k)$ yields

$$\begin{aligned}
&\sum_{i=1}^n (\tfrac{\eta_i\gamma_i}{\delta}+1)\tfrac{1}{n^2p_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2 \\
&\quad\le \sum_{i=1}^n (\tfrac{\eta_i\gamma_i}{\delta}+1)\tfrac{L_i}{n^2p_i}\langle \nabla f_i(x^k)-\nabla f_i(x^\star), x^k-x^\star\rangle \\
&\quad\le \max_i((\tfrac{\eta_i\gamma_i}{\delta}+1)\tfrac{L_i}{np_i})\langle \nabla F(x^k)-\nabla F(x^\star), x^k-x^\star\rangle.
\end{aligned}$$

Adding the terms back together yields

$$\mathcal{V}(x^k)\le \lambda^2\nu\langle \nabla F(x^k)-\nabla F(x^\star), x^k-x^\star\rangle$$

where $\nu=\max_i((1+\delta^{-1})\frac{L_i\eta_i\gamma_i}{np_i}+(1+\delta)\frac{L_i}{np_i}-\delta\mu)$. This can now be summarized as

$$\begin{aligned}
\mathcal{P}(x^k) &\le \|x^k-x^\star\|^2-\lambda\big(2-\nu\lambda\big)\langle \nabla F(x^k)-\nabla F(x^\star), x^k-x^\star\rangle \\
&\le \|x^k-x^\star\|^2-\mu\lambda\big(2-\nu\lambda\big)\|x^k-x^\star\|^2 \\
&= (1-\rho_P)\|x^k-x^\star\|^2,
\end{aligned}$$

where $\rho_P=\mu\lambda(2-\nu\lambda)$ and the last inequality is given by the strong monotonicity of ∇F. □

Proof of Lemma 4.4. Since norms are non-negative, we have

$$\mathcal{D}(y^k)\le \sum_{i=1}^n (1-\eta_i+\tfrac{1}{\gamma_i})\widehat{\gamma}_i\|y_i^k-y_i^\star\|^2\le (1-\rho_D)\sum_{i=1}^n \widehat{\gamma}_i\|y_i^k-y_i^\star\|^2,$$

where $\rho_D=\min_i(\eta_i-\frac{1}{\gamma_i})$. □

Proof of Lemma 4.5. From Assumption 3.3, we know that there exists ϕ^k such that $y_i^k=\nabla f_i(\phi^k),\forall i\in\{1,\dots,n\}$. Using this and $y_i^\star=\nabla f_i(x^\star)$ yield

$$\begin{aligned}
\|\tfrac{1}{n}\sum_{i=1}^n\big[y_i^k-y_i^\star\big]\|^2 &= \|\nabla F(\phi^k)-\nabla F(x^\star)\|^2\ge \mu\|\nabla F(\phi^k)-\nabla F(x^\star)\|\|\phi^k-x^\star\| \\
&\ge \mu\langle \nabla F(\phi^k)-\nabla F(x^\star),\phi^k-x^\star\rangle = \mu\frac{1}{n}\sum_{i=1}^n\langle \nabla f_i(\phi^k)-\nabla f_i(x^\star),\phi^k-x^\star\rangle \\
&\ge \mu\frac{1}{n}\sum_{i=1}^n\frac{1}{L_i}\|\nabla f_i(\phi^k)-\nabla f_i(x^\star)\|^2=\sum_{1=1}^n\frac{\mu}{nL_i}\|y_i^k-y_i^\star\|^2,
\end{aligned}$$

where the inequalities are given by μ-strong monotonicity of ∇F, Cauchy–Schwarz, and $\frac{1}{L_i}$-cocoercivity of ∇f_i. Inserting this into $\mathcal{D}(y^k)$ and using that $\frac{\widehat{\gamma}_i}{\gamma_i}=\frac{(1+\delta^{-1})\lambda^2}{n^2p_i}$ for all $i\in\{1,\dots,n\}$—note that we defined $\frac{\gamma_i}{\gamma_i}=1$ if $\gamma_i=0$— give

$$\mathcal{D}(y^k)\le\sum_{i=1}^n(1-\eta+(1-np_i\tfrac{\mu}{L_i})\tfrac{1}{\gamma_i})\widehat{\gamma}_i\|y_i^k-y_i^\star\|^2.$$

For each term we see that if $\gamma_i > 0$ then

$$(1-\eta+(1-np_i\tfrac{\mu}{L_i})\tfrac{1}{\gamma_i})\widehat{\gamma}_i\|y_i^k-y_i^\star\|^2 \le (1-\rho_i^+)\widehat{\gamma}_i\|y_i^k-y_i^\star\|^2$$

with $\rho_i^+ = \eta-(1-np_i\frac{\mu}{L_i})\frac{1}{\gamma_i}$. If $\gamma_i = 0$, then $\frac{L_i}{np_i} \le \mu$ and

$$\begin{aligned}&(1-\eta+(1-np_i\tfrac{\mu}{L_i})\tfrac{1}{\gamma_i})\widehat{\gamma}_i\|y_i^k-y_i^\star\|^2\\ &\quad= (1-np_i\tfrac{\mu}{L_i})\tfrac{(1+\delta^{-1})\lambda^2}{n^2p_i}\|y_i^k-y_i^\star\|^2 \le 0 \le (1-1)\widehat{\gamma}_i\|y_i^k-y_i^\star\|^2.\end{aligned}$$

This gives $\mathcal{D}(y^k) \le (1-\rho_D)\sum_{i=1}^n \widehat{\gamma}_i\|y_i^k-y_i^\star\|^2$ where

$$\rho_D = \begin{cases}\eta-(1-\frac{np_i}{L_i}\mu)\frac{1}{\gamma_i} & \text{if } \gamma_i > 0\\ 1 & \text{if } \gamma_i = 0\end{cases}. \qquad \square$$

Proof of Lemma 4.6. An L-smooth and μ-strongly convex function must satisfy $L \ge \mu$. Assuming $\max_i(\frac{L_i}{np_i}) < \mu$ yields the following contradiction

$$\mu > \max_i \tfrac{L_i}{np_i} = \sum_{j=1}^n p_j \max_i \tfrac{L_i}{np_i} \ge \sum_{j=1}^n p_j \tfrac{L_j}{np_j} = \sum_{i=1}^n \tfrac{L_i}{n} \ge L.$$

If $\max_i(\frac{L_i}{np_i}) = \mu$, equality must hold everywhere and we have

$$0 = \sum_{j=1}^n p_j \max_i \tfrac{L_i}{np_i} - \sum_{j=1}^n p_j \tfrac{L_j}{np_j} = \sum_{j=1}^n p_j(\max_i \tfrac{L_i}{np_i} - \tfrac{L_j}{np_j}).$$

Since $p_j > 0$ and $\max_i \frac{L_i}{np_i} - \frac{L_j}{np_j} \ge 0$, we have $\max_i \frac{L_i}{np_i} = \frac{L_j}{np_j}$ for all $j \in \{1,\dots,n\}$.$\square$

B. Proofs of Theorems

Proof of Theorem 4.7. Application of Lemma 4.3 and 4.4 in Proposition 4.2 yields the convergence rate

$$\mathbb{E}\left[\|x^k-x^\star\|^2 + \sum_{i=1}^n \widehat{\gamma}_i\|y_i^k-y_i^\star\|^2\right] \in O((1-\min(\rho_P,\rho_D))^k)$$

with

$$\begin{aligned}\rho_P &= \mu\lambda(2-\nu\lambda)\\ \rho_D &= \min_i \eta_i - \tfrac{1}{\gamma_i}\\ \nu &= \max_i (1+\delta^{-1})\tfrac{L_i\eta_i\gamma_i}{np_i} + (1+\delta)\tfrac{L_i}{np_i} - \delta\mu,\end{aligned}$$

which hold for all choices of $\delta > 0$ and $\gamma_i > 0$ for all $i \in \{1,\dots,n\}$. If there exists δ and $\gamma_1,\dots,\gamma_n$ such that $\min(\rho_P,\rho_D) \in (0,1]$ we have convergence. We restrict ourselves to only search for δ and $\gamma_1,\dots,\gamma_n$ such that $\rho_P = \rho_D = \rho$ for some $\rho \in (0,1]$. For all $i \in \{1,\dots,n\}$, select $\gamma_i = \frac{1}{\eta_i - \rho}$, which is positive when $\rho < \eta_i$, and convergence is then proved if there exists $\rho \in (0, \min_i \eta_i)$ and $\delta > 0$ such that

$$\begin{aligned}\rho &= \mu\lambda(2 - \nu\lambda)\\ \nu &= \max_i (1+\delta^{-1})\tfrac{L_i}{np_i}\tfrac{\eta_i}{\eta_i-\rho} + (1+\delta)\tfrac{L_i}{np_i} - \delta\mu.\end{aligned}$$

The variable ν can be minimized w.r.t. δ if $\max_i \frac{L_i}{np_i} > \mu$. The minimum then exists and is unique since ν as a function of δ is continuous, strictly convex, and $\nu \to \infty$ both when $\delta \to 0^+$ and $\delta \to \infty$. Calling the minimum point $\delta^\star$, noting that $\delta^\star > 0$, and inserting it and the choice of γ_i in the expression for $\widehat{\gamma}_i$ from Definition 4.1 yield the first statement of the theorem.

When $\max_i \frac{L_i}{np_i} \not> \mu$, Lemma 4.6 gives $\frac{L_i}{np_i} = \mu$ for all $i \in \{1,\dots,n\}$ and

$$\nu = \mu + \mu(1+\delta^{-1}) \max_i \tfrac{\eta_i}{\eta_i - \rho}.$$

This can not be minimized w.r.t. δ since the inf is not attained. However, any $\delta > 0$ will yield a valid ρ and $\widehat{\gamma}_i$, giving the rate

$$\begin{aligned}&\mathbb{E}\left[\|x^k - x^\star\|^2 + \sum\nolimits_{i=1}^n \tfrac{\lambda^2}{n^2 p_i}\tfrac{1}{\eta_i-\rho}\|y_i^k - y_i^\star\|^2\right]\\ &\quad\le \mathbb{E}\left[\|x^k - x^\star\|^2 + \sum\nolimits_{i=1}^n \widehat{\gamma}_i\|y_i^k - y_i^\star\|^2\right] \in O((1-\rho)^k).\end{aligned}$$

Taking the limit as $\delta \to \infty$ results in the stated interval. □

Proof of Theorem 4.8. The proof is analogous to the proof of Theorem 4.7 but with Lemma 4.5 instead of Lemma 4.4, yielding

$$\begin{aligned}\rho_P &= \mu\lambda(2 - \nu\lambda)\\ \rho_D &= \min_i \begin{cases}\eta - (1 - \frac{np_i}{L_i}\mu)\frac{1}{\gamma_i} & \text{if } \gamma_i > 0\\ 1 & \text{if } \gamma_i = 0\end{cases}\\ \nu &= \max_i (1+\delta^{-1})\tfrac{L_i\eta\gamma_i}{np_i} + (1+\delta)\tfrac{L_i}{np_i} - \delta\mu.\end{aligned}$$

where $\delta > 0$, $\gamma_i \ge 0$ and $\gamma_i = 0$ implies $\frac{L_i}{np_i} \le \mu$ for all $i \in \{1,\dots,n\}$. Let $\gamma_i = \frac{1}{\eta-\rho_D}\max(0, 1 - \frac{np_i\mu}{L_i})$ and $\delta = \sqrt{\frac{\eta}{\eta-\rho_D}}$. Both choices are valid if $\rho_D < \eta$ since then $\delta > 0$, $\gamma_i \ge 0$ and $\gamma_i = 0$ only if $\frac{L_i}{np_i} \le \mu$.

Assuming $\max_i \frac{L_i}{np_i} > \mu$ yields

$$\begin{aligned}
\nu &= \max_i (1+\delta^{-1})\tfrac{L_i}{np_i}\tfrac{\eta}{\eta-\rho_D}\max(0, 1-\tfrac{np_i\mu}{L_i}) + (1+\delta)\tfrac{L_i}{np_i} - \delta\mu \\
&= \max_i (1+\delta^{-1})\tfrac{\eta}{\eta-\rho_D}\max(0, \tfrac{L_i}{np_i} - \mu) + (1+\delta)\tfrac{L_i}{np_i} - \delta\mu \\
&= (1+\delta^{-1})\tfrac{\eta}{\eta-\rho_D}(\max_i \tfrac{L_i}{np_i} - \mu) + (1+\delta)(\max_i \tfrac{L_i}{np_i}) - \delta\mu \\
&= \mu + \left(\max_i \tfrac{L_i}{np_i} - \mu\right)\left(1 + \sqrt{\tfrac{\eta}{\eta-\rho_D}}\right)^2.
\end{aligned}$$

Restricting the problem to $\rho = \rho_D = \rho_P$ and only considering the convergent rates $\rho \in (0,1]$ yield the problem in the theorem. The first statement of the theorem comes from Proposition 4.2 with γ_i and δ inserted in the expression for $\widehat{\gamma}_i$ from Definition 4.1.

When $\max_i \frac{L_i}{np_i} = \mu$, Lemma 4.6 gives $\frac{L_i}{np_i} = \mu$ for all $i \in \{1,\dots,n\}$, meaning $\gamma_i = 0$ is a valid choice for all $i \in \{1,\dots,n\}$. With this choice, $\nu = \mu$ regardless of δ, and ρ_D is no longer limited by η with $\rho_D = 1$. The statement of the theorem then follows. □

C. Proof of Corollaries

Proof of Corollary 5.1. The expected update frequency is $\eta_i = p_i$. Assuming $\max_i \frac{L_i}{np_i} > \mu$ and using Theorem 4.7 the convergence rate for SAGA is given by the $\rho \in (0, p_{\min})$ that satisfies

$$\begin{aligned}
&\rho = \mu\lambda(2-\nu\lambda) \\
&\nu = \mu \min_{\delta>0} \max_i (1+\delta^{-1})\tfrac{L_i}{np_i\mu}\tfrac{p_i}{p_i-\rho} + (1+\delta)\tfrac{L_i}{np_i\mu} - \delta.
\end{aligned} \tag{9}$$

If we write ν as a function of ρ, this can equivalently be written as finding $\rho \in (0, p_{\min})$ such that $\rho + \lambda^2\mu\nu(\rho) = 2\mu\lambda$. Since $\nu(\rho)$ is continuous and $\nu(\rho) \to \infty$ as $\rho \to p_{\min}$ from below, if we find a $\tilde{\rho} \in (0, p_{\min})$ such that $\tilde{\rho} + \lambda 2\mu\nu(\tilde{\rho}) \leq 2\mu\lambda$, it exists $\rho \in [\tilde{\rho}, p_{\min})$ such that (9) hold. Hence, if we replace ν in (9) with an upper bound, we can find a lower bound on the contraction ρ.

Let $\kappa_{\max} = \max_i \frac{L_i}{np_i\mu}$ and $p_{\min} = \min_i p_i$ and upper bound ν as

$$\begin{aligned}
\nu &\leq \mu \min_{\delta>0} (1+\delta^{-1})\kappa_{\max}\tfrac{p_{\min}}{p_{\min}-\rho_D} + (1+\delta)\kappa_{\max} - \delta \\
&= \mu + \mu\left[\sqrt{\kappa_{\max}\tfrac{p_{\min}}{p_{\min}-\rho}} + \sqrt{\kappa_{\max}-1}\right]^2 \\
&= \mu\kappa_{\max}\tfrac{2p_{\min}-\rho}{p_{\min}-\rho} + 2\mu\sqrt{\kappa_{\max}^2 - \kappa_{\max}}\sqrt{\tfrac{p_{\min}}{p_{\min}-\rho}} \\
&\leq \mu\kappa_{\max}\tfrac{2p_{\min}-\rho}{p_{\min}-\rho} + \mu\sqrt{\kappa_{\max}^2 - \kappa_{\max}}\tfrac{2p_{\min}-\rho}{p_{\min}-\rho} \\
&= \mu\kappa_{\max}(1+\sqrt{1-\kappa_{\max}^{-1}})\tfrac{2p_{\min}-\rho}{p_{\min}-\rho}.
\end{aligned}$$

The last inequality is given by $2\sqrt{a} \leq 1 + a$ for all $a \geq 0$. It can be verified that this upper bound also is valid when $\max_i \frac{L_i}{np_i} = \mu$. Replace ν in (9) with this upper bound gives a set of equations that define a lower bound on the contraction ρ.

Inserting the two samplings and solving for λ when the lower bound on ρ is zero gives the $\lambda_{\max}$. Maximizing the lower bound on ρ w.r.t. λ yield the optimal $\lambda^\star$ and $\rho^\star$. For both uniform and Lipschitz sampling, the upper bound on ν is tight for $\rho = 0$ so it can be used to accurately determine maximal step-size according to Theorem 4.7. □

Proof of Corollary 6.1. The proof is similar to the proof of Corollary 5.1 but instead the following upper bound is used:

$$\begin{aligned}\nu &\leq \mu \max_i 2\tfrac{L_i}{np_i\mu}\tfrac{p_i}{p_i-\rho} + 2\tfrac{L_i}{np_i\mu} - \delta \leq \mu \max_i 2\tfrac{L_i}{np_i\mu}\tfrac{p_i}{p_i-\rho} + 2\tfrac{L_i}{np_i\mu} \\ &= 2\mu \max_i \tfrac{L_i}{n\mu}\left[\tfrac{1}{p_i} + \tfrac{1}{p_i-\rho}\right].\end{aligned}$$

Replacing ν in Theorem 4.7 with this upper bound and inserting the presented p_i, λ and ρ verifies the first claim.

The rate from Theorem 4.7 is of the form $\mathbb{E}\|x^k - x^\star\|^2 \in O((1-\lambda^\star\mu)^k)$. The iteration complexity to achieve an ϵ-accurate solution in expectation is then $k \in O(\frac{1}{\lambda^\star\mu}\log\frac{1}{\epsilon})$. One gradient evaluation is done per iteration so $O(\frac{1}{\lambda^\star\mu}\log\frac{1}{\epsilon})$ is also the computational complexity. Inserting $\lambda^\star$ gives the result. □

Proof of Corollary 5.2. The proof is analogous to Corollary 5.1 but Theorem 4.8 is used instead of Theorem 4.7 and ν is upper bounded by

$$\begin{aligned}\nu &\leq \mu + 2\mu(\kappa_{\max}-1)\left[\tfrac{\eta}{\eta-\rho}+1\right] \leq \frac{\mu}{2}\left[\tfrac{\eta}{\eta-\rho}+1\right] + 2\mu(\kappa_{\max}-1)\left[\tfrac{\eta}{\eta-\rho}+1\right] \\ &\leq \mu(2\kappa_{\max} - \tfrac{3}{2})\tfrac{2\eta-\rho}{\eta-\rho}\end{aligned}$$

where $\kappa_{\max} = \max_i \frac{L_i}{np_i\mu}$. □

Proof of Corollary 6.2. From Corollary 5.2 we get the iteration complexity $k \in O(\frac{1}{\lambda^\star\mu}\log\frac{1}{\epsilon})$. One gradient evaluation is needed for the primal update and $n\eta$ evaluations are needed in expectation for the dual update, this gives the computational complexity $O((1+n\eta)\frac{1}{\lambda^\star\mu}\log\frac{1}{\epsilon})$. Inserting $\lambda^\star$ from Corollary 5.2 and using $\frac{1}{2}(a+b+\sqrt{a^2+b^2}) \leq a+b$ gives the result. □

References

[1] R. Babanezhad Harikandeh, M. O. Ahmed, A. Virani, M. Schmidt, J. Konečný, and S. Sallinen. "Stop Wasting My Gradients: Practical SVRG". In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 2251–2259. URL: http://papers.nips.cc/paper/5711-stopwasting-my-gradients-practical-svrg.pdf (visited on 2020-05-24).

[2] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Second. CMS Books in Mathematics. Springer International Publishing, 2017. ISBN: 978-3-319-48310-8. URL: //www.springer.com/gp/book/9783319483108 (visited on 2019-01-15).

[3] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A Fresh Approach to Numerical Computing". *SIAM Review* **59**:1 (2017), pp. 65–98. DOI: 10.1137/141000671.

[4] L. Bottou and O. Bousquet. "The Tradeoffs of Large Scale Learning". In: *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc., 2008, pp. 161–168. (Visited on 2019-01-03).

[5] C.-C. Chang and C.-J. Lin. "LIBSVM: A Library for Support Vector Machines". *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**:3 (2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27. DOI: 10.1145/1961189.1961199.

[6] D. Csiba and P. Richtárik. "Importance Sampling for Minibatches". *Journal of Machine Learning Research* **19**:27 (2018), pp. 1–21. URL: http://jmlr.org/papers/v19/16-241.html (visited on 2022-08-18).

[7] A. Defazio, F. Bach, and S. Lacoste-Julien. "SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives". In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 1646–1654. (Visited on 2018-08-27).

[8] R. M. Gower, P. Richtárik, and F. Bach. "Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching". *Mathematical Programming* **188**:1 (2021), pp. 135–192. DOI: 10.1007/s10107-020-01506-0.

[9] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams. "Variance Reduced Stochastic Gradient Descent with Neighbors". In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 2305–2313. (Visited on 2018-08-27).

[10] R. Johnson and T. Zhang. "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 315–323. (Visited on 2018-08-27).

[11] J. Konečný and P. Richtárik. “Semi-Stochastic Gradient Descent Methods”. *Frontiers in Applied Mathematics and Statistics* **3** (2017). DOI: 10.3389/fams.2017.00009.

[12] D. Kovalev, S. Horváth, and P. Richtárik. “Don’t Jump Through Hoops and Remove Those Loops: SVRG and Katyusha are Better Without the Outer Loop”. In: *Proceedings of the 31st International Conference on Algorithmic Learning Theory*. PMLR, 2020, pp. 451–467.

[13] N. Le Roux, M. Schmidt, and F. Bach. “A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets”. In: *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 2663–2671. (Visited on 2018-12-14).

[14] D. Needell, R. Ward, and N. Srebro. “Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm”. In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 1017–1025. (Visited on 2018-08-27).

[15] Y. Nesterov. “Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems”. *SIAM Journal on Optimization* **22**:2 (2012), pp. 341–362. DOI: 10.1137/100802001.

[16] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. “SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. JMLR.org, Sydney, NSW, Australia, 2017, pp. 2613–2621. URL: http://proceedings.mlr.press/v70/nguyen17b.html (visited on 2020-04-30).

[17] X. Qian, Z. Qu, and P. Richtárik. “SAGA with Arbitrary Sampling”. In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 5190–5199. URL: https://proceedings.mlr.press/v97/qian19a.html (visited on 2021-09-27).

[18] Z. Qu and P. Richtárik. “Coordinate Descent with Arbitrary Sampling I: Algorithms and Complexity”. *Optimization Methods and Software* **31**:5 (2016), pp. 829–857. DOI: 10.1080/10556788.2016.1190360.

[19] Z. Qu, P. Richtárik, and T. Zhang. “Quartz: Randomized Dual Coordinate Ascent with Arbitrary Sampling”. In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 865–873. (Visited on 2018-08-27).

[20] P. Richtárik and M. Takáč. “Iteration Complexity of Randomized Block-Coordinate Descent Methods for Minimizing a Composite Function”. *Mathematical Programming* **144**:1 (2014), pp. 1–38. DOI: 10.1007/s10107-012-0614-z.

[21] P. Richtárik and M. Takáč. "Parallel Coordinate Descent Methods for Big Data Optimization". *Mathematical Programming* **156**:1 (2016), pp. 433–484. DOI: 10.1007/s10107-015-0901-6.

[22] H. Robbins and S. Monro. "A Stochastic Approximation Method". *The Annals of Mathematical Statistics* **22**:3 (1951), pp. 400–407. URL: https://www.jstor.org/stable/2236626 (visited on 2019-07-30).

[23] M. Schmidt, R. Babanezhad, M. Ahmed, A. Defazio, A. Clifton, and A. Sarkar. "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Vol. 38. Proceedings of Machine Learning Research. PMLR, 2015, pp. 819–828. URL: http://proceedings.mlr.press/v38/schmidt15.html (visited on 2020-02-21).

[24] M. Schmidt, N. Le Roux, and F. Bach. "Minimizing Finite Sums with the Stochastic Average Gradient". *Mathematical Programming* **162**:1 (2017), pp. 83–112. DOI: 10.1007/s10107-016-1030-6.

[25] O. Sebbouh, N. Gazagnadou, S. Jelassi, F. Bach, and R. Gower. "Towards Closing the Gap between the Theory and Practice of SVRG". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 648–658. URL: http://papers.nips.cc/paper/8354-towards-closing-the-gap-between-the-theory-and-practice-of-svrg.pdf (visited on 2020-05-24).

[26] T. Strohmer and R. Vershynin. "A Randomized Kaczmarz Algorithm with Exponential Convergence". *Journal of Fourier Analysis and Applications* **15**:2 (2008), p. 262. DOI: 10.1007/s00041-008-9030-4.

[27] M. Takáč, P. Richtárik, and N. Srebro. *Distributed Mini-Batch SDCA*. 2015. arXiv: 1507.08322. URL: http://arxiv.org/abs/1507.08322 (visited on 2018-08-27).

[28] L. Xiao and T. Zhang. "A Proximal Stochastic Gradient Method with Progressive Variance Reduction". *SIAM Journal on Optimization* **24**:4 (2014), pp. 2057–2075. DOI: 10.1137/140961791.

[29] X. Zhang, W. B. Haskell, and Z. Ye. "A Unifying Framework for Variance-Reduced Algorithms for Findings Zeroes of Monotone operators". *Journal of Machine Learning Research* **23**:60 (2022), pp. 1–44. URL: http://jmlr.org/papers/v23/19-513.html (visited on 2022-08-18).

[30] P. Zhao and T. Zhang. "Stochastic Optimization with Importance Sampling for Regularized Loss Minimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 1–9. URL: http://proceedings.mlr.press/v37/zhaoa15.html (visited on 2018-08-27).

Paper II

Cocoercivity, Smoothness and Bias in Variance-Reduced Stochastic Gradient Methods

Martin Morin **Pontus Giselsson**

Abstract

With the purpose of examining biased updates in variance-reduced stochastic gradient methods, we introduce SVAG, a SAG/SAGA-like method with adjustable bias. SVAG is analyzed in a cocoercive root-finding setting, a setting which yields the same results as in the usual smooth convex optimization setting for the ordinary proximal-gradient method. We show that the same is not true for SVAG when biased updates are used. The step-size requirements for when the operators are gradients are significantly less restrictive compared to when they are not. This highlights the need to not rely solely on cocoercivity when analyzing variance-reduced methods meant for optimization. Our analysis either match or improve on previously known convergence conditions for SAG and SAGA. However, in the biased cases they still do not correspond well with practical experiences and we therefore examine the effect of bias numerically on a set of classification problems. The choice of bias seem to primarily affect the early stages of convergence and in most cases the differences vanish in the later stages of convergence. However, the effect of the bias choice is still significant in a couple of cases.

Originally published in Numerical Algorithms. Reprinted with permission (Creative Commons Attribution 4.0 International License). Minor corrections of typos in the published version have been made to this version.

1. Introduction

Variance-reduced stochastic gradient (VR-SG) methods is a family of iterative optimization algorithms that combine the low per-iteration computational cost of the ordinary stochastic gradient descent and the attractive convergence properties of gradient descent. Just as ordinary stochastic gradient descent, VR-SG methods solve smooth optimization problems on finite sum form,

$$\min_{x\in\mathbb{R}^N} \tfrac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{1}$$

where, for all $i \in \{1,\dots,n\}$, $f_i : \mathbb{R}^N \to \mathbb{R}$ is a convex function that is L-smooth, i.e., f_i is differentiable with L-Lipschitz continuous gradient. These types of problems are common in model fitting, supervised learning, and empirical risk minimization which, together with the nice convergence properties of VR-SG methods, has lead to a great amount of research on VR-SG methods and the development of several different variants, e.g., [1, 15, 16, 21, 22, 23, 24, 25, 27, 30, 33, 40, 41, 43, 45].

Broadly speaking, VR-SG methods form a stochastic estimate of the objective gradient by combining one or a few newly evaluated terms of the gradient with previously evaluated terms. Classic examples of this can be seen in the SAG [27, 40] and SAGA [15] algorithms. Given some initial iterates $x^0, y_1^0, \dots, y_n^0 \in \mathbb{R}^N$ and step-size $\lambda > 0$, SAGA samples i^k uniformly from $\{1,\dots,n\}$ and then updates the iterates as

$$\begin{aligned}
x^{k+1} &= x^k - \lambda\left(\nabla f_{i^k}(x^k) - y_{i^k}^k + \tfrac{1}{n}\sum_{j=1}^{n} y_j^k\right),\\
y_{i^k}^{k+1} &= \nabla f_{i^k}(x^k),\\
y_j^{k+1} &= y_j^k \quad \text{for all} \quad j \neq i^k,
\end{aligned}$$

for $k \in \{0,1,\dots\}$. The update of x^{k+1} is said to be unbiased since the expected value of x^{k+1} at iteration k is equal to an ordinary gradient descent update. This is in contrast to the biased SAG, which is identical to SAGA except that the update of x^{k+1} is

$$x^{k+1} = x^k - \lambda\left(\tfrac{1}{n}\left(\nabla f_{i^k}(x^k) - y_{i^k}^k\right) + \tfrac{1}{n}\sum_{j=1}^{n} y_j^k\right)$$

and the expected value of x^{k+1} now includes a term containing the old gradients $\frac{1}{n}\sum_{i=1}^{n} y_i^k$. Although SAG shows that unbiasedness is not essential for the convergence of VR-SG methods, the effects of this bias are unclear. The majority of VR-SG methods are unbiased but existing works have not established any clear advantage of either the biased SAG or the unbiased SAGA. This paper will examine the effect of bias and its interplay with different problem assumptions for SAG/SAGA-like methods.

1.1 Problem and Algorithm

Instead of solving (1) directly, we consider a closely related but more general root-finding problem. Throughout the paper, we consider the Euclidean space $\mathbb{R}^N$ and the problem of finding $x \in \mathbb{R}^N$ such that

$$0 = Rx \coloneqq \tfrac{1}{n}\sum_{i=1}^{n} R_i x \tag{2}$$

where $R_i : \mathbb{R}^N \to \mathbb{R}^N$ is $\frac{1}{L}$-cocoercive—see Section 2—for all $i \in \{1,\dots,n\}$. Since L-smoothness of a convex function is equivalent to $\frac{1}{L}$-cocoercivity of the gradient [2, Corollary 18.17], the smooth optimization problem in (1) can be recovered by setting $R_i = \nabla f_i$ for all $i \in \{1,\dots,n\}$ in (2). Problem (2) is also interesting in its own right with it and the closely related fixed point problem of finding $x \in \mathbb{R}^N$ such that $x = (\mathrm{Id} - \alpha R)x$ where $\alpha \in (0, 2L^{-1})$ both having applications in for instance feasibility and non-linear signal recovery problems, see [8, 10, 13] and the references therein. To solve this problem, we present the *Stochastic Variance Adjusted Gradient* (SVAG) algorithm.

Algorithm 1.1 SVAG

input operators $R_i : \mathbb{R}^N \to \mathbb{R}^N$, initial state $x^0 \in \mathbb{R}^N$ and $y_1^0,\dots,y_n^0 \in \mathbb{R}^N$, step-size $\lambda > 0$, innovation weight $\theta \in \mathbb{R}$
for $k = 0,1,\dots$ **do**
  Sample i^k uniformly from $\{1,\dots,n\}$
  $x^{k+1} = x^k - \lambda\left(\frac{\theta}{n}\left(R_{i^k}x^k - y_{i^k}^k\right) + \frac{1}{n}\sum_{j=1}^{n} y_j^k\right)$
  $y_{i^k}^{k+1} = R_{i^k}x^k$
  $y_j^{k+1} = y_j^k$ for all $j \neq i^k$
end for

SVAG is heavily inspired by SAG and SAGA with both being special cases, $\theta = 1$ and $\theta = n$ respectively. Just like SAG and SAGA, in each iteration, SVAG evaluates one operator R_{i^k} and stores the results in $y_{i^k}^{k+1}$. An estimate of the full operator is then formed as

$$Rx^k \approx \widetilde{R}^k = \tfrac{\theta}{n}(R_{i^k}x^k - y_{i^k}^k) + \tfrac{1}{n}\sum_{j=1}^{n} y_j^k.$$

The scalar θ determine how much weight should be put on the new information gained from evaluating $R_{i^k}x^k$. If the innovation, $R_{i^k}x^k - y_{i^k}^k$, is highly correlated with the total innovation, $Rx^k - \frac{1}{n}\sum_{j=1}^{n} y_j^k$, a large innovation weight θ can be chosen and vice versa. The innovation weight θ also determines the bias of SVAG. Taking the expected value $\widetilde{R}^k$ given the information at iteration k gives

$$\mathbb{E}[\widetilde{R}^k | x^k, y_1^k,\dots,y_n^k] = \tfrac{\theta}{n}Rx^k + (1-\tfrac{\theta}{n})\tfrac{1}{n}\sum_{j=1}^{n} y_j^k$$

which reveals that $\widetilde{R}^k$ is an unbiased estimate of Rx^k if $\theta = n$, i.e., in the SAGA case. Any other choice, for instance SAG where $\theta = 1$, yields a bias towards $\frac{1}{n}\sum_{j=1}^{n} y_j^k$.

1.2 Contribution

The theory behind finding roots of monotone operators in general, and cocoercive operators in particular, has been put to good use when analyzing first order optimization methods, examples include [2, 4, 14, 26, 38, 44]. For instance can both the proximal-gradient and ADMM methods be seen as instances of classic root-finding fixed-point iterations and analyzed as such, namely forward-backward and Douglas–Rachford respectively. The resulting analyses can often be simple and intuitive and even though the root-finding formulation is more general—not all cocoercive operators are gradients of convex functions—the analyses are not necessarily more conservative. For example, analyzing proximal-gradient as forward-backward splitting yields the same rates and step-size conditions as analyzing it as a minimization method in the smooth/cocoercive setting, see for instance [32, Theorem 2.1.14] and [2, Example 5.18 and Proposition 4.39]. However, the main contribution of this paper is to show that the same is not true for VR-SG methods, in particular it is not true for SVAG when it is biased.

The results consist of two main convergence theorems for SVAG: one in the cocoercive operator case and one in the cocoercive gradient case, the later being equivalent to the minimization of a smooth and convex finite sum. Both of these theorems match or improve upon previously known results for the SAG and SAGA special cases. Comparing the two settings reveal that SVAG can use significantly larger step-sizes, with faster convergence as a result, in the cocoercive gradient case compared to the general cocoercive operator case. In the operator case, an upper bound on the step-size that scales as $O(n^{-1})$ is found where n is the number of terms in (2). However, the restrictions on the step-size loosen with reduced bias and the unfavorable $O(n^{-1})$ scaling disappears completely when SVAG is unbiased. In the gradient case, this bad scaling never occurs, regardless of bias. We provide examples in which SVAG diverges with step-sizes larger than the theoretical upper bounds in the operator case. Since the gradient case is proven to converge with much larger step-sizes, this verifies the difference between the convergence behavior of cocoercive operators and gradients.

These results indicate that it is inadvisable to only rely on the more general monotone operator theory and not explicitly use the gradient property when analyzing VR-SG methods meant for optimization. However, the large impact of bias in the cocoercive operator setting also raises the question regarding its importance in other non-gradient settings as well. One such setting of interest, where the operators are not gradients of convex functions, is the case of saddle-point problems. These problems are of importance in optimization due to their use in primal-dual methods but recently they have also gained a lot of attention due to their applications in the training of GANs in machine learning. Because of this, and due to the attractive

properties of VR-SG methods in the convex optimization setting, efforts have gone into applying VR-SG methods to saddle-point problems as well [5, 7, 34, 42, 46]. Most of these efforts have been unbiased, something our analysis suggests is wise. With that said, it is important to note that our analysis is often not directly applicable due the fact that saddle-point problems rarely are cocoercive.

The main reason for the recent rise in popularity of variance-reduced stochastic methods is their use in the optimization setting, but, although bias plays a big role in the cocoercive operator case, our results are not as clear in this setting. For instance, the theoretical results for the SAG and SAGA special cases yield identical rates and step-size conditions with no clear advantage to either special case. Further experiments are therefore performed where several different choices of bias in SVAG are examined on a set of logistic regression and SVM optimization problems. However, the results of these experiments are in line with existing works with no significant advantage of any particular bias choice in SVAG. Although the performance difference is significant in some cases, no single choice of bias performs best for all problems and all bias choices eventually converge with the same rate in the majority of the cases. Furthermore, the theoretical maximal step-size can routinely be exceeded in these experiments, indicating that there is room for further theoretical improvements.

1.3 Related Work

There is a large array of options for solving (2). For $n \in \{1,2,3,4\}$, several operator splitting methods exist with varying assumptions on the operator properties, see for instance [4, 18, 19, 28, 29, 44] and the references therein. However, while these methods also can be applied for larger n by simply regrouping the terms, they do not utilize the finite sum structure of the problem. Algorithms have therefore been designed to utilize this structure for arbitrary large n with the hopes of reducing the total computational costs, e.g., [9, 10, 11, 36]. In particular the problem and method in [10] is closely related to the root-finding problem and algorithm considered in this paper.

Using the notation of [10], when $T_0 = \mathrm{Id}$, the fixed point problem of [10] can be mapped to (2) via $R_i = \omega_i(\mathrm{Id} - T_i)$ and vice verse. [1] Many applications considered in [10] can therefore, at least in part, be tackled with our algorithm as well. In particular, the problem of finding common fixed points of firmly nonexpansive operators can directly be solved by our algorithm. However, [10] is more general in that it allows for $T_0 \neq \mathrm{Id}$ and works in general real Hilbert spaces. Looking at the algorithm of [10] we see that, just as our algorithm is a generalization of SAG/SAGA, it can be seen as a generalization of Finito [16], another classic VR-SG method. It generalize Finito in several way, for instance it allows for an additional proximal/backward step and it replaces the stochastic selection with a different selection criteria. However, in the optimization setting it still suffers from the same drawback

[1] If T_i is α_i-averaged, as assumed in [10], R_i is $(2\alpha_i\omega_i)^{-1}$-cocoercive.

as Finito when compared to SAG/SAGA-like algorithms. It still needs to store a full copy of the iterate for each term in objective. Since SAG, SAGA, and SVAG only need to store the gradient of each term, they can utilize any potential structure of the gradients to reduce the storage requirements [27]. Although the differences above are interesting in their own right, the notion of bias we examine in this paper is not applicable to Finito-like algorithms.

SAG and SAGA were compared in [15] but with no direct focus on the effects of bias. Other examples of research on SAG and SAGA include acceleration, sampling strategy selection, and ways to reduce the memory requirement [20, 22, 31, 35, 39, 47]. However, none of these works, including [31] that was written by the authors, analyze the biased case we consider in this paper. Even the works considering non-uniform sampling of gradients [20, 31, 35, 39] perform some sort of bias correction in order to remain unbiased. Furthermore, in order to keep the focus on the effects of the bias we have refrained from bringing in such generalizations into this work, making it distinct from the above research. To the authors' knowledge, the only theoretical convergence result for biased VR-SG methods are the ones for SAG [27, 40]. But, since they only consider SAG, they fail to capture the breadth of SVAG and our proof is the first to simultaneously capture SAG, SAGA, and more.

Since the release of the first preprint of this paper, [17] has also provided a proof covering the gradient case of both SAG and SAGA, and some choices of bias in SVAG. All though [17] does not consider cocoercive operators, it is some sense more general with them considering a general biased stochastic estimator of the gradient. This generality comes at the cost of a more conservative analysis with their step-size scaling with $O(n^{-1})$ in all cases.

2. Preliminaries and Notation

Let $\mathbb{R}$ denote the real numbers and let the natural numbers be denoted $\mathbb{N} = \{0,1,2,\dots\}$. Let $\langle\cdot,\cdot\rangle$ denote the standard Euclidean inner product and $\|\cdot\| = \sqrt{\langle\cdot,\cdot\rangle}$ the standard 2-norm. The scaled inner product and norm we denote as $\langle\cdot,\cdot\rangle_\Sigma = \langle\Sigma(\cdot),\cdot\rangle$ and $\|\cdot\|_\Sigma = \sqrt{\langle\cdot,\cdot\rangle_\Sigma}$ where Σ is a positive definite matrix. If Σ is not positive definite, $\|\cdot\|_\Sigma$ is not a norm but we keep the notation for convenience.

Let n be the number of operators in (2). The vector $\mathbf{1}$ is the vector of all ones in $\mathbb{R}^n$ and e_i is the vector in $\mathbb{R}^n$ of all zeros except the i:th element which contains a 1. The matrix I is an identity matrix with the size derived from context and $E_i = e_i e_i^T$.

The symbol $\otimes$ denotes the Kronecker product of two matrices. The Kronecker product is linear in both arguments and the following properties hold

$$(A\otimes B)^T = A^T \otimes B^T, \qquad (A\otimes B)(C\otimes D) = (AC)\otimes(BD).$$

In the last property it is assumed that the dimensions are such that the matrix multiplications are well defined. The eigenvalues of $A\otimes B$ are given by

$$\tau_i\mu_j \text{ for all } i\in\{1,\dots,m\}, j\in\{1,\dots,l\} \tag{3}$$

where τ_i and μ_j are the eigenvalues of A and B respectively.

The Cartesian product of two sets C_1 and C_2 is defined as

$$C_1 \times C_2 = \{(c_1, c_2) \mid c_1 \in C_1, c_2 \in C_2\}.$$

From this definition we see that if C_1 and C_2 are closed and convex, so is $C_1 \times C_2$.

Let $X^\star$ be the set of all solutions of (2),

$$X^\star = \{x \mid 0 = \tfrac{1}{n}\textstyle\sum_{i=1}^n R_i x\}$$

and define $Z^\star$ as the set of primal-dual solutions

$$Z^\star = \{(x, R_1 x, \ldots, R_n x) \mid 0 = \tfrac{1}{n}\textstyle\sum_{i=1}^n R_i x\}.$$

Assuming they exists, $x^\star$ denotes a solution to (2) and $z^\star$ denotes a primal-dual solution, i.e., $x^\star \in X^\star$ and $z^\star \in Z^\star$.

A single valued operator $R : \mathbb{R}^N \to \mathbb{R}^N$ is $\frac{1}{L}$-cocoercive if

$$\langle Rx - Ry, x - y\rangle \geq \tfrac{1}{L}\|Rx - Ry\|^2 \tag{4}$$

holds for all $x, y \in \mathbb{R}^N$. An operator that is $\frac{1}{L}$-cocoercive is L-Lipschitz continuous. The set of zeros of a cocoercive operator R is closed and convex.

A differentiable convex function $f : \mathbb{R}^N \to \mathbb{R}$ is called L-smooth if the gradient is $\frac{1}{L}$-cocoercive. Equivalently, a differentiable convex function is L-smooth if

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \tfrac{L}{2}\|y - x\|^2 \tag{5}$$

holds for all $x, y \in \mathbb{R}^N$.

If $f_i : \mathbb{R}^N \to \mathbb{R}$ is a differentiable convex function for each $i \in \{1, \ldots, n\}$, the minimization of $\sum_{i=1}^n f_i(x)$ is equivalent to (2) with $R_i = \nabla f_i$.

For more details regarding monotone operators and convex functions see [2, 32].

To establish almost sure sequence convergence of the stochastic algorithm, the following propositions will be used. The first is from [37] and establishes convergence of non-negative almost super-martingales. The second is based on [12] and provides the tool to show almost sure sequence convergence.

PROPOSITION 2.1

Let $(\Omega, \mathcal{F}, P)$ be a probability space and $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots$ be a sequence of sub-σ-algebras of $\mathcal{F}$. For all $k \in \mathbb{N}$, let z^k, β^k, ξ^k and ζ^k be non-negative $\mathcal{F}_k$-measurable random variables. If $\sum_{i=0}^\infty \beta^i < \infty$, $\sum_{i=0}^\infty \xi^i < \infty$ and

$$\mathbb{E}[z^{k+1} | \mathcal{F}_k] \leq (1 + \beta^k) z^k + \xi^k - \zeta^k$$

hold almost surely for all $k \in \mathbb{N}$, then z^k converges a.s. to a finite valued random variable and $\sum_{i=0}^\infty \zeta^i < \infty$ almost surely.

Proof. See [37, Theorem 1]. □

PROPOSITION 2.2
Let Z be a non-empty closed subset of a finite dimensional Hilbert space H, let $\phi : [0,\infty) \to [0,\infty)$ be a strictly increasing function such that $\phi(t) \to \infty$ as $t \to \infty$, and let $(x^k)_{k\in\mathbb{N}}$ be a sequence of H-valued random variables. If $\phi(\|x^k - z\|)$ converges a.s. to a finite valued non-negative random variable for all $z \in Z$, then the following hold:

(i) *$(x^k)_{k\in\mathbb{N}}$ is bounded almost surely.*

(ii) *Suppose the cluster points of $(x^k)_{k\in\mathbb{N}}$ are a.s. in Z, then $(x^k)_{k\in\mathbb{N}}$ converge a.s. to a Z-valued random variable.*

Proof. In finite dimensional Hilbert spaces, these two statements are the same as statements (ii) and (iv) of [12, Proposition 2.3]. Hence, consider the proof of [12, Proposition 2.3] restricted to finite dimensional Hilbert spaces. The proof of (ii) in [12, Proposition 2.3] only relies on the a.s. convergence of $\phi(\|x^k - z\|)$ and hence is implied by the assumptions of this proposition. This proves our first statement. The proof of (iv) in [12, Proposition 2.3] only relies on (iii) of [12, Proposition 2.3] which in turn is implied by (ii) of [12, Proposition 2.3], i.e., our first statement. This proves our second statement. □

3. Convergence

Throughout the analysis we will use the following two assumptions on the operators in (2).

ASSUMPTION 3.1
For each $i \in \{1,\dots,n\}$, let R_i be $\frac{1}{L}$-cocoercive and $X^\star \neq \emptyset$, i.e., (2) *has at least one solution.*

ASSUMPTION 3.2
For each $i \in \{1,\dots,n\}$, let $R_i = \nabla f_i$ for some differentiable function f_i and define $F = \frac{1}{n}\sum_{i=1}^n f_i$. Furthermore, let Assumption 3.1 hold, i.e., f_i is L-smooth and convex and $\arg\min F(x)$ *exists.*

3.1 Reformulation

We begin by formalizing and reformulating Algorithm 1.1 into a more convenient form. Let $(\Omega,\mathcal{F},P)$ be the underlying probability space of Algorithm 1.1. The index selected at iteration k is then a uniformly distributed random variable $i^k : \Omega \to \{1,\dots,n\}$. For each $k \in \mathbb{N}$, define the random variable $z^k : \Omega \to \mathbb{R}^{N(n+1)}$ as $z^k = (x^k, y_1^k, \dots, y_n^k)$ where x^k and y_i^k for $i \in \{1,\dots,n\}$ are the iterates of Algorithm 1.1. Let $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \dots$ be a sequence of sub-σ-algebras of $\mathcal{F}$ such that z^k are $\mathcal{F}_k$-measurable and i^k is independent of $\mathcal{F}_k$. With the operator $\mathbf{B} : \mathbb{R}^{N(n+1)} \to \mathbb{R}^{2Nn}$

defined as $\mathbf{B}(x, y_1, \ldots, y_n) = (R_1 x, \ldots, R_n x, y_1, \ldots, y_n)$, one iteration of Algorithm 1.1 can be written as

$$z^{k+1} = z^k - (U_{i^k} \otimes I)\mathbf{B} z^k \tag{6}$$

where $z^0 \in \mathbb{R}^{N(n+1)}$ is given and

$$U_i = \begin{bmatrix} \frac{\lambda}{n}\theta e_i^T & -\frac{\lambda}{n}\theta e_i^T + \frac{\lambda}{n}\mathbf{1}^T \\ -E_i & E_i \end{bmatrix}$$

for all $i \in \{1, \ldots, n\}$. The vector e_i and the matrix E_i are defined in Section 2.

The following lemma characterizes the zeros of $(U_i \otimes I)\mathbf{B}$ and hence the fixed points of (6) and Algorithm 1.1.

LEMMA 3.3
Let Assumption 3.1 hold, each $z^\star$ in $Z^\star$ is then a zero of $(U_i \otimes I)\mathbf{B}$ for all $i \in \{1, \ldots, n\}$, i.e.

$$\forall z^\star \in Z^\star, \forall i \in \{1, \ldots, n\} : \quad 0 = (U_i \otimes I)\mathbf{B} z^\star.$$

Furthermore, the set $Z^\star$ is closed and convex and $R_i x^\star = R_i \bar{x}^\star$ for all $x^\star, \bar{x}^\star \in X^\star$ and for all $i \in \{1, \ldots, n\}$.

Proof of Lemma 3.3. The zero statement, $0 = (U_i \otimes I)\mathbf{B} z^\star$, follows from definition of $z^\star$. For closedness and convexity of $Z^\star$, we first prove that $R_i x^\star$ is unique for each $i \in \{1, \ldots, n\}$. Taking $x, y \in X^\star$, which implies $\sum_{i=1}^n R_i x = \sum_{i=1}^n R_i y = 0$, and using cocoercivity (4) of each R_i gives

$$\begin{aligned} 0 = \langle \textstyle\sum_{i=1}^n R_i x - \sum_{i=1}^n R_i y, x - y \rangle &= \textstyle\sum_{i=1}^n \langle R_i x - R_i y, x - y \rangle \\ &\geq \textstyle\sum_{i=1}^n \frac{1}{L} \|R_i x - R_i y\|^2 \geq 0, \end{aligned}$$

hence must $R_i x = R_i y$ for all $i \in \{1, \ldots, n\}$. The set $Z^\star$ is a Cartesian product of $X^\star$ and the points $r_i = R_i x^\star$ for $i \in \{1, \ldots, n\}$ for any $x^\star \in X^\star$. A set consisting of only one point is closed and convex and $X^\star$ is closed and convex since $\frac{1}{n}\sum_{i=1}^n R_i$ is cocoercive [2, Proposition 23.39], hence is $Z^\star$ closed and convex. □

The operator $\mathbf{B}$ in the reformulated algorithm can be used to enforce the following property on the sequence $(z^k)_{k \in \mathbb{N}}$.

LEMMA 3.4
Let $(\Omega, \mathcal{F}, P)$ be a probability space and $(z^k)_{k \in \mathbb{N}}$ be a sequence of random variables $z^k : \Omega \to \mathbb{R}^{N(n+1)}$. If $\mathbf{B}z^k \to \mathbf{B}z^\star$ a.s. where $z^\star \in Z^\star$, then any cluster point of $(z^k)_{k \in \mathbb{N}}$ will almost surely be in $Z^\star$.

Proof of Lemma 3.4. Let z be a cluster point of $(z^k)_{k\in\mathbb{N}}$. Take an $\omega \in \Omega$ such that $\mathbf{B}z^k(\omega) \to \mathbf{B}z^\star$. For this ω and for all $k \in \mathbb{N}$, we define the realizations of z and z^k as

$$z(\omega) = (\bar{x}, \bar{y}_1, \ldots, \bar{y}_n), \quad z^k(\omega) = (\bar{x}^k, \bar{y}_1^k, \ldots, \bar{y}_n^k)$$

where $\bar{x}, \bar{y}_1, \ldots, \bar{y}_n \in \mathbb{R}^N$ and $\bar{x}^k, \bar{y}_1^k, \ldots, \bar{y}_n^k \in \mathbb{R}^N$ for all $k \in \mathbb{N}$.

Since $\mathbf{B}\bar{z}^k \to \mathbf{B}z^\star$ we directly have $\bar{y}_i^k \to R_i x^\star$ for $x^\star \in X^\star$ and hence must $\bar{y}_i = R_i x^\star$ for all $i \in \{1,\ldots,n\}$. Note, $R_i x^\star$ is independent of which $x^\star \in X^\star$ was chosen, see Lemma 3.3. Furthermore, $\mathbf{B}\bar{z}^k \to \mathbf{B}\bar{z}^\star$ implies that $R_i \bar{x}^k \to R_i x^\star$ for all $i \in \{1,\ldots,n\}$. Let $(\bar{x}^{k(l)})_{l\in\mathbb{N}}$ be a subsequence converging to $\bar{x}$, then

$$\begin{aligned}\|\tfrac{1}{n}\textstyle\sum_{i=1}^n R_i\bar{x}\| &\le \|\tfrac{1}{n}\textstyle\sum_{i=1}^n R_i\bar{x}^{k(l)} - \tfrac{1}{n}\textstyle\sum_{i=1}^n R_i\bar{x}\| + \|\tfrac{1}{n}\textstyle\sum_{i=1}^n R_i\bar{x}^{k(l)}\| \\ &\le L\|\bar{x}^{k(l)} - \bar{x}\| + \|\tfrac{1}{n}\textstyle\sum_{i=1}^n R_i\bar{x}^{k(l)}\| \to \|\tfrac{1}{n}\textstyle\sum_{i=1}^n R_i x^\star\| = 0\end{aligned}$$

as $l \to \infty$ where L-Lipschitz continuity of $\frac{1}{n}\sum_{i=1}^n R_i$ was used. This concludes that $\bar{x} \in X^\star$ and since $\bar{y}_i = R_i x^\star = R_i \bar{x}$ for all $i \in \{1,\ldots,n\}$ by Lemma 3.3, we have that $z(\omega) \in Z^\star$. Since this hold for any ω such that $\mathbf{B}z^k(\omega) \to \mathbf{B}z^\star$ and the set in $\mathcal{F}$ of all such ω have probability one due to the almost sure convergence of $\mathbf{B}z^k \to \mathbf{B}z^\star$, we have $z \in Z^\star$ almost surely. □

The reformulation (6) further allows us to concisely formulate two Lyapunov inequalities.

LEMMA 3.5
Let Assumption 3.1 hold, the update (6) *then satisfies*

$$\begin{aligned}&\mathbb{E}[\|z^{k+1} - z^\star\|_{H\otimes I}^2 | \mathcal{F}_k] \\ &\quad\le \|z^k - z^\star\|_{H\otimes I}^2 - \|\mathbf{B}z^k - \mathbf{B}z^\star\|_{(2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] - \xi I)\otimes I}^2 \\ &\qquad - \xi n L \langle Rx^k, x^k - x^\star\rangle\end{aligned}$$

for all $k \in \mathbb{N}$ *and* $\xi \in [0, \frac{2\lambda}{nL}]$*, where the matrices* H *and* M *are given by*

$$H = \begin{bmatrix} 1 & -\frac{\lambda}{n}(n-\theta)\mathbf{1}^T \\ -\frac{\lambda}{n}(n-\theta)\mathbf{1} & \frac{\lambda}{L}I + \frac{\lambda^2}{n^2}(n-\theta)^2\mathbf{1}\mathbf{1}^T \end{bmatrix}$$

and

$$M = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \otimes \frac{1}{2n}\frac{\lambda}{L} I - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \frac{\lambda^2}{2n^2}(n-\theta)\mathbf{1}\mathbf{1}^T.$$

LEMMA 3.6
Let Assumption 3.2 hold, the update (6) *then satisfies*

$$\mathbb{E}[F((K\otimes I)z^{k+1})|\mathcal{F}_k] \leq F((K\otimes I)z^k) - \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{\frac{1}{2}S\otimes I}$$

for all $k\in\mathbb{N}$, *where* $K = \begin{bmatrix} 1 & \frac{\lambda}{n}\mathbf{1}^T \end{bmatrix}$ *and*

$$S = \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix} \otimes (\theta-1)\frac{\lambda}{n^3}\mathbf{1}\mathbf{1}^T - \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \otimes (\theta-1)^2\frac{L\lambda^2}{n^3} I + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \frac{\lambda}{n^2}\mathbf{1}\mathbf{1}^T.$$

Proof of Lemma 3.5. Take $k\in\mathbb{N}$, note that since U_{i^k} is independent of $\mathcal{F}_k$ and z_k is $\mathcal{F}_k$-measurable we have

$$\begin{aligned}\mathbb{E}[\langle (U_{i^k}\otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star\rangle_{H\otimes I}|\mathcal{F}_k] \\ = \langle (H\mathbb{E}[U_{i^k}]\otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star\rangle.\end{aligned}$$

The matrix $H\mathbb{E}[U_{i^k}]$ is given by

$$H\mathbb{E}[U_{i^k}] = \begin{bmatrix} \frac{\lambda}{n}\mathbf{1}^T & 0 \\ -\frac{\lambda^2}{n^2}(n-\theta)\mathbf{1}\mathbf{1}^T - \frac{\lambda}{nL}I & \frac{\lambda}{nL}I \end{bmatrix},$$

see the supplementary material for verification of this and other matrix identities. We also note that

$$\langle Rx^k - Rx^\star, x^k - x^\star\rangle = \langle (\begin{bmatrix} \frac{1}{n}\mathbf{1}^T & 0 \\ 0 & 0 \end{bmatrix}\otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star\rangle.$$

Taking $\xi\in[0,\frac{2\lambda}{nL}]$ and putting these two expression together yield

$$\begin{aligned}\mathbb{E}[\langle (U_{i^k}\otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star\rangle_{H\otimes I}|\mathcal{F}_k] - \tfrac{\xi nL}{2}\langle Rx^k - Rx^\star, x^k - x^\star\rangle \\ = \langle (\begin{bmatrix} (\frac{\lambda}{n} - \frac{\xi L}{2})\mathbf{1}^T & 0 \\ -\frac{\lambda^2}{n^2}(n-\theta)\mathbf{1}\mathbf{1}^T - \frac{\lambda}{nL}I & \frac{\lambda}{nL}I \end{bmatrix}\otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star\rangle.\end{aligned}$$

Using $\frac{1}{L}$-cocoercivity of R_i for each $i\in\{1,\ldots,n\}$ gives

$$\begin{aligned}\mathbb{E}[\langle (U_{i^k}\otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star\rangle_{H\otimes I}|\mathcal{F}_k] - \tfrac{\xi nL}{2}\langle Rx^k - Rx^\star, x^k - x^\star\rangle \\ \geq \langle (\begin{bmatrix} (\frac{\lambda}{nL} - \frac{\xi}{2})I & 0 \\ -\frac{\lambda^2}{n^2}(n-\theta)\mathbf{1}\mathbf{1}^T - \frac{\lambda}{nL}I & \frac{\lambda}{nL}I \end{bmatrix}\otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), \mathbf{B}z^k - \mathbf{B}z^\star\rangle\end{aligned}$$

Setting

$$\bar{M} = \begin{bmatrix} \frac{\lambda}{nL} I & 0 \\ -\frac{\lambda^2}{n^2}(n-\theta)\mathbf{1}\mathbf{1}^T - \frac{\lambda}{nL} I & \frac{\lambda}{nL} I \end{bmatrix}$$

gives

$$\begin{aligned}
&\mathbb{E}[\langle (U_{i^k} \otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star \rangle_{H\otimes I} | \mathcal{F}_k] - \tfrac{\xi n L}{2} \langle Rx^k - Rx^\star, x^k - x^\star \rangle \\
&\quad \geq \langle (\bar{M} \otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), \mathbf{B}z^k - \mathbf{B}z^\star \rangle \\
&\qquad - \langle (\begin{bmatrix} \frac{\xi}{2} I & 0 \\ 0 & 0 \end{bmatrix} \otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), \mathbf{B}z^k - \mathbf{B}z^\star \rangle \\
&\quad \geq \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{\frac{1}{2}(\bar{M}+\bar{M}^T)\otimes I} - \tfrac{\xi}{2}\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2 \\
&\quad = \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(M - \frac{\xi}{2} I)\otimes I}
\end{aligned}$$

where $M = \frac{1}{2}(\bar{M} + \bar{M}^T)$ is the matrix in the statement of the lemma. Finally, using this inequality and $0 = (U_{i^k} \otimes I)\mathbf{B}z^\star$ from Lemma 3.3 gives

$$\begin{aligned}
&\mathbb{E}[\|z^{k+1} - z^\star\|^2_{H\otimes I} | \mathcal{F}_k] \\
&\quad = \mathbb{E}[\| \left(z^k - (U_{i^k} \otimes I)\mathbf{B}z^k\right) - \left(z^\star - (U_{i^k} \otimes I)\mathbf{B}z^\star\right) \|^2_{H\otimes I} | \mathcal{F}_k] \\
&\quad = \|z^k - z^\star\|^2_{H\otimes I} + \mathbb{E}[\|(U_{i^k} \otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star)\|^2_{H\otimes I} | \mathcal{F}_k] \\
&\qquad - 2\mathbb{E}[\langle (U_{i^k} \otimes I)(\mathbf{B}z^k - \mathbf{B}z^\star), z^k - z^\star \rangle_{H\otimes I} | \mathcal{F}_k] \\
&\quad \leq \|z^k - z^\star\|^2_{H\otimes I} + \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{\mathbb{E}[U^T_{i^k} H U_{i^k}]\otimes I} \\
&\qquad - \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(2M - \xi I)\otimes I} - \xi n L \langle Rx^k - Rx^\star, x^k - x^\star \rangle \\
&\quad = \|z^k - z^\star\|^2_{H\otimes I} - \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(2M - \mathbb{E}[U^T_{i^k} H U_{i^k}] - \xi I)\otimes I} \\
&\qquad - \xi n L \langle Rx^k, x^k - x^\star \rangle.
\end{aligned}$$

□

Proof of Lemma 3.6. Take $k \in \mathbb{N}$ and note that

$$(K \otimes I)z^{k+1} = (K \otimes I)(z^k - (U_{i^k} \otimes I)\mathbf{B}z^k) = x^k - (Q_{i^k} \otimes I)\mathbf{B}z^k$$

where $Q_{i^k} = \frac{\lambda}{n}\begin{bmatrix} (\theta-1)e^T_{i^k} & -(\theta-1)e^T_{i^k} \end{bmatrix}$. Furthermore, with $G = \frac{1}{n}\begin{bmatrix} \mathbf{1}^T & 0 \end{bmatrix}$, we have $\nabla F(x^k) = (G \otimes I)\mathbf{B}z^k$. From the definition of $z^\star$ we have $0 = (G \otimes I)\mathbf{B}z^\star =$

$(Q_{i^k} \otimes I)\mathbf{B}z^\star$. Using L-smoothness, (5), of F yields

$$\begin{aligned}
&\mathbb{E}[F((K\otimes I)z^{k+1})|\mathcal{F}_k] \\
&\quad = \mathbb{E}[F(x^k - (Q_{i^k}\otimes I)\mathbf{B}z^k)|\mathcal{F}_k] \\
&\quad \le F(x^k) - \langle \nabla F(x^k), (\mathbb{E}[Q_{i^k}]\otimes I)\mathbf{B}z^k\rangle + \tfrac{L}{2}\,\mathbb{E}[\|(Q_{i^k}\otimes I)\mathbf{B}z^k\|^2|\mathcal{F}_k] \\
&\quad = F(x^k) - \langle (G\otimes I)\mathbf{B}z^k, (\mathbb{E}[Q_{i^k}]\otimes I)\mathbf{B}z^k\rangle + \|\mathbf{B}z^k\|^2_{\frac{L}{2}\mathbb{E}[Q_{i^k}^T Q_{i^k}]\otimes I} \\
&\quad = F(x^k) - \|\mathbf{B}z^k\|^2_{\frac{1}{2}\mathbb{E}[Q_{i^k}^T G + G^T Q_{i^k}]\otimes I} + \|\mathbf{B}z^k\|^2_{\frac{L}{2}\mathbb{E}[Q_{i^k}^T Q_{i^k}]\otimes I} \\
&\quad = F(x^k) - \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{\frac{1}{2}S_L\otimes I}
\end{aligned}$$

where $S_L = \mathbb{E}[Q_{i^k}^T G + G^T Q_{i^k} - L Q_{i^k}^T Q_{i^k}]$.

With $D = \begin{bmatrix} 0 & \mathbf{1}^T \end{bmatrix}$ we have $(K\otimes I)z^k = x^k + \frac{\lambda}{n}(D\otimes I)\mathbf{B}z^k$. Using the first order convexity condition on F and $0 = (D\otimes I)\mathbf{B}z^\star = (G\otimes I)\mathbf{B}z^\star$ yields

$$\begin{aligned}
F((K\otimes I)z^k) &= F(x^k + \tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k) \\
&\ge F(x^k) + \langle \nabla F(x^k), \tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k\rangle \\
&= F(x^k) + \langle (G\otimes I)\mathbf{B}z^k, \tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k\rangle \\
&= F(x^k) + \|\mathbf{B}z^k\|^2_{\frac{1}{2}\frac{\lambda}{n}(D^T G + G^T D)\otimes I} \\
&= F(x^k) + \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{\frac{1}{2}S_C\otimes I}
\end{aligned} \tag{7}$$

where $S_C = \frac{\lambda}{n}(D^T G + G^T D)$. Combining these two inequalities gives

$$\mathbb{E}[F((K\otimes I)z^{k+1})|\mathcal{F}_k] \le F((K\otimes I)z^k) - \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{\frac{1}{2}S\otimes I}$$

where $S = S_L + S_C$. □

3.2 Convergence Theorems

We are now ready to state the main convergence theorems for SVAG. They are stated with the notation from Algorithm 1.1 but are proved at the end of this section with the help of the reformulation in (6) and the lemmas above.

THEOREM 3.7
For all $i \in \{1,\dots,n\}$, let $(x^k)_{k\in\mathbb{N}}$ and $(y_i^k)_{k\in\mathbb{N}}$ be the sequences generated by Algorithm 1.1. If Assumption 3.1 hold and the step-size, $\lambda > 0$, and innovation weight, $\theta \in \mathbb{R}$, satisfy

$$\frac{1}{L(2+|n-\theta|)} > \lambda,$$

then $x^k \to x^\star$ and $y_i^k \to R_i x^\star$ almost surely for all $i \in \{1,\dots,n\}$, where $x^\star$ is a solution to (2)*. For all $i \in \{1,\dots,n\}$, the residuals converge a.s. as*

$$\min_{k\in\{0,\dots,t\}} \mathbb{E}[\|R_i x^k - R_i x^\star\|^2] \leq \tfrac{n}{\lambda(L^{-1}-\lambda c)}\tfrac{1}{t+1} C_R,$$
$$\min_{k\in\{0,\dots,t\}} \mathbb{E}[\|y_i^k - R_i x^\star\|^2] \leq \tfrac{n}{\lambda(L^{-1}-\lambda c)}\tfrac{1}{t+1} C_R$$

where $c = 2 + |n-\theta|$ and

$$C_R = \min_{x\in X^\star} \|x^0 - x\|^2 + \tfrac{\lambda}{L}\textstyle\sum_{i=1}^n \|y_i^0 - R_i x^\star\|^2 + \lambda^2(n-\theta)^2 \|\frac{1}{n}\sum_{i=1}^n y_i^0\|^2 - 2\lambda(n-\theta)\langle x^0 - x, \frac{1}{n}\textstyle\sum_{i=1}^n y_i^0\rangle$$

for any $x^\star \in X^\star$.

THEOREM 3.8
For all $i \in \{1,\dots,n\}$, let $(x^k)_{k\in\mathbb{N}}$ and $(y_i^k)_{k\in\mathbb{N}}$ be the sequences generated by Algorithm 1.1. If Assumption 3.2 hold and the step-size, $\lambda > 0$, and innovation weight, $\theta \in [0,n]$, satisfy

$$\frac{1}{L}\frac{1}{2 + (n-\theta)\frac{\theta-1}{n}\left(\frac{\theta-1}{n} - 1 + \frac{\theta-1}{|\theta-1|}\sqrt{2}\right)} > \lambda,$$

then $x^k \to x^\star$ and $y_i^k \to \nabla f_i(x^\star)$ almost surely, where $x^\star$ is a solution to (2)*. For all $i \in \{1,\dots,n\}$, the residuals converge a.s. as*

$$\min_{k\in\{0,\dots,t\}} \mathbb{E}[\|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2] \leq \tfrac{n}{\lambda(L^{-1}-\lambda c)}\tfrac{1}{t+1}(C_R + C_F),$$
$$\min_{k\in\{0,\dots,t\}} \mathbb{E}[\|y_i^k - \nabla f_i(x^\star)\|^2] \leq \tfrac{n}{\lambda(L^{-1}-\lambda c)}\tfrac{1}{t+1}(C_R + C_F),$$
$$\min_{k\in\{0,\dots,t\}} \mathbb{E}[F(x^k) - F(x^\star)] \leq \tfrac{1}{\lambda(1-L\lambda c)}\tfrac{1}{t+1}(C_R + C_F)$$

where

$$c = 2 + (n-\theta)\tfrac{\theta-1}{n}\left(\tfrac{\theta-1}{n} - 1 + \tfrac{\theta-1}{|\theta-1|}\sqrt{2}\right),$$
$$C_R = \min_{x\in X^\star} \|x^0 - x\|^2 + \tfrac{\lambda}{L}\textstyle\sum_{i=1}^n \|y_i^0 - R_i x^\star\|^2 + \lambda^2(n-\theta)^2 \|\frac{1}{n}\sum_{i=1}^n y_i^0\|^2 - 2\lambda(n-\theta)\langle x^0 - x, \frac{1}{n}\textstyle\sum_{i=1}^n y_i^0\rangle,$$
$$C_F = 2\lambda(n-\theta)\big(F(x^0 + \tfrac{\lambda}{n}\textstyle\sum_{i=1}^n y_i^0) - F(x^\star)\big)$$

for any $x^\star \in X^\star$.

Both Theorem 3.7 and 3.8 give the step-size condition $\lambda \in (0, \frac{1}{2L})$ for the SAGA special case, i.e., $\theta = n$. This is the same as the largest upper bound found in the

literature [15] and appears to be tight [31]. Theorem 3.8 also give this step-size condition when $\theta = 1$, i.e., SAG in the optimization case. This bound improves on upper bound of $\frac{1}{16L} \leq \lambda$ presented in [40].

In the cocoercive operator setting with $\theta \neq n$, Theorem 3.7 gives a step-size condition that scales with n^{-1}. This step-size scaling is significantly worse compared to the gradient case in Theorem 3.8 in which the step-size's dependence on n is $O(1)$ for all θ. This difference is indeed real and not an artifact of the analysis since we in Section 4 present a problem for which the cocoercivity result appears to be tight. A consequence of this unfavorable step-size scaling in the operator setting is slow convergence. There is therefore little reason to use anything else than $\theta = n$ in SVAG when R_i is not a gradient of a smooth function for all $i \in \{1,\dots,n\}$.

The rates of Theorem 3.7 and 3.8 are of $O(\frac{1}{t+1})$ type with two sets of multiplicative factors. One factor which only depend on the algorithm parameters, $\frac{n}{\lambda(L^{-1}-\lambda c)}$, and one set which depend on how the algorithm initialization relates to the solution set, C_R and $C_R + C_F$. The initialization dependent factors also depend on the algorithm parameters, but, since knowing the exact dependency requires knowing the solution set, we will not attempt to tune the parameters to decrease this factor. Only considering the first factor, the rate becomes better if c is decreased and, since c is independent of λ, the best choice of step-size is $\lambda = (2Lc)^{-1}$. This means that $\lambda = (4L)^{-1}$ and $\theta = n$ are the best parameter choices in the cocoercive operator setting. In the optimization case the best step-size is also $\lambda = (4L)^{-1}$ but the innovation weight can be selected as either $\theta = n$ or $\theta = 1$.

However, in the optimization case we do not believe that these theoretical rates reflects real world performance and parameter choices based on them might therefore not perform particularly well. We base this belief on our experience with numerical experiments. For $\theta \neq n$ and $\theta \neq 1$, we have not found any optimization problem where the step-size condition in Theorem 3.8 appears to be tight. Also, using $\lambda = (2Lc)^{-1}$ as suggested by Theorem 3.8 can in some cases lead to impractically small step-sizes. For instance, if $\lambda = (2Lc)^{-1}$ was used in the experiments in Section 4, a couple of the experiments would have step-sizes over 1000 times smaller than the ones used now. One can of course not prove that a worst case analysis can be improved with experiments but we still feel they indicate a conservative analysis, even though the analysis improves on the previous best results.

Proof of Theorem 3.7. Apply Lemma 3.5 with $\xi = 0$, the iterates given by (6) then satisfy the following for all $z^\star \in Z^\star$,

$$\begin{aligned}\mathbb{E}[\|z^{k+1} - z^\star\|^2_{H\otimes I}|\mathcal{F}_k] \\ \leq \|z^k - z^\star\|^2_{H\otimes I} - \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(2M-\mathbb{E}[U_{i^k}^T H U_{i^k}])\otimes I}.\end{aligned} \tag{8}$$

Assuming $H \succ 0$ and $2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] \succ 0$, Proposition 2.1 can be applied. We will later prove that this assumption indeed does hold. Proposition 2.1 gives a.s. summability of $\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(2M-\mathbb{E}[U_{i^k}^T H U_{i^k}])\otimes I}$ and hence will $\mathbf{B}z^k \to \mathbf{B}z^\star$ almost

surely. Lemma 3.4 then gives that all cluster points of $(z^k)_{k\in\mathbb{N}}$ are in $Z^\star$ almost surely. Finally, since Proposition 2.1 ensures the a.s. convergence of $\|z^k - z^\star\|^2_{H\otimes I}$ and since $\mathbb{R}^{N(n+1)}$ with the inner product $\langle (H\otimes I)\cdot,\cdot\rangle$ is a finite dimensional Hilbert space, Proposition 2.2 gives the almost sure convergence of $z^k \to z^\star \in Z^\star$.

There always exists a λ such that $2M - \mathbb{E}[U_{i^k}^T H U_{i^k}]$ and H are positive definite. First we show that $H \succ 0$ always holds for $\lambda > 0$. Taking the Schur complement of 1 in H gives

$$\frac{\lambda}{L} I + \frac{\lambda^2}{n^2}(n-\theta)^2 \mathbf{1}\mathbf{1}^T - \frac{\lambda^2}{n^2}(n-\theta)^2 \mathbf{1}\mathbf{1}^T = \frac{\lambda}{L} I \succ 0.$$

Hence is $H \succ 0$ since the Schur complement is positive definite.

We now show $2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] \succ 0$. Straightforward algebra, see the supplementary material, yields

$$\begin{aligned} 2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] = &\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{\lambda}{nL} I - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{\lambda^2}{n} I + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \frac{\lambda^2}{n}(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) \\ &- \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes (n-\theta)\frac{\lambda^2}{n^2}\mathbf{1}\mathbf{1}^T + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{\lambda^2}{n^2}\mathbf{1}\mathbf{1}^T. \end{aligned}$$

Positive definiteness of this matrix is established by ensuring positivity of the smallest eigenvalue $\sigma_{\min}$. The smallest eigenvalue $\sigma_{\min}$ is greater than the sum of the smallest eigenvalue of each term. For the eigenvalues of the Kronecker products, see (3). This gives that

$$\sigma_{\min} \geq \frac{\lambda}{nL} - \frac{\lambda^2}{n} - \frac{\lambda^2}{n} - \frac{\lambda^2}{n}|n-\theta| + 0 = \frac{\lambda}{n}\big(L^{-1} - \lambda(2 + |n-\theta|)\big).$$

Since $\lambda > 0$ by assumption, if

$$\frac{1}{L(2+|n-\theta|)} > \lambda.$$

we have $\sigma_{\min} > 0$ and $2M - \mathbb{E}[U_{i^k}^T H U_{i^k}]$ is positiv definite.

Rates are gotten by taking the total expectation of (8) and adding together the inequalities from $k = 0$ to $k = t$, yielding

$$\begin{aligned} \|z^0 - z^\star\|^2_{H\otimes I} &= \mathbb{E}[\|z^0 - z^\star\|^2_{H\otimes I}] - \mathbb{E}[\|z^{t+1} - z^\star\|^2_{H\otimes I}] \\ &\geq \textstyle\sum_{k=0}^t \mathbb{E}\left[\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(2M - \mathbb{E}[U_{i^k}^T H U_{i^k}])\otimes I}\right] \\ &\geq \textstyle\sum_{k=0}^t \sigma_{\min}\, \mathbb{E}[\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2] \\ &\geq \sigma_{\min}(t+1) \min_{k\in\{0,\dots,t\}} \mathbb{E}[\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2]. \end{aligned}$$

Putting in the lower bound on $\sigma_{\min}$ and rearranging yield

$$\min_{k\in\{0,\dots,t\}} \mathbb{E}[\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2] \leq \tfrac{n}{\lambda(L^{-1} - \lambda(2+|n-\theta|))(t+1)} \|z^0 - z^\star\|^2_{H\otimes I}.$$

From the definition of H in Lemma 3.5 we have

$$\begin{aligned}\|z^0 - z^\star\|^2_{H\otimes I} = \|x^0 - x^\star\|^2 + \tfrac{\lambda}{L}\textstyle\sum_{i=1}^n \|y_i^0 - R_i x^\star\|^2 + \lambda^2(n-\theta)^2\|\tfrac{1}{n}\textstyle\sum_{i=1}^n y_i^0\|^2 \\ - 2\lambda(n-\theta)\langle x^0 - x^\star, \tfrac{1}{n}\textstyle\sum_{i=1}^n y_i^0\rangle\end{aligned}$$

where $z^\star = (x^\star, R_1 x^\star, \ldots, R_n x^\star)$. Since this hold for any $z^\star \in Z^\star$ and hence any $x^\star \in X^\star$, the results of theorems follows by minimizing the RHS over $x^\star \in X^\star$. Note, since $R_i x^\star$ constant for all $x^\star \in X^\star$, the objective is convex and, since $X^\star$ is closed and convex, the minimum is then attained. □

Proof of Theorem 3.8. Combining Lemma 3.5 and 3.6 yield

$$\begin{aligned}&\mathbb{E}[\|z^{k+1} - z^\star\|^2_{H\otimes I} + 2\lambda(n-\theta)(F((K\otimes I)z^{k+1}) - F(x^\star))|\mathcal{F}_k] \\ &\quad\le \|z^k - z^\star\|^2_{H\otimes I} + 2\lambda(n-\theta)(F((K\otimes I)z^k) - F(x^\star)) \\ &\qquad - \|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] + \lambda(n-\theta)S - \xi I)\otimes I} - \xi nL\langle \nabla F(x^k), x^k - x^\star\rangle\end{aligned}$$

which holds for all $k \in \mathbb{N}$, $\xi \in [0, \frac{2\lambda}{nL}]$, and $z^\star \in Z^\star$. Since $H \succ 0$ for $\lambda > 0$, see the proof of Theorem 3.7, the first term is non-negative while the second term is non-negative if $\theta \le n$. From cocoercivity of ∇F, the last term is non-positive and we assume, for now, that there exists $\lambda > 0$ and $\frac{2\lambda}{nL} \ge \xi > 0$ such that $2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] + \lambda(n-\theta)S - \xi I \succ 0$, making the third term non-positive.

Applying Proposition 2.1 gives the a.s. summability of

$$\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2_{(2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] + \lambda(n-\theta)S - \xi I)\otimes I} + \xi nL\langle \nabla F(x^k) - \nabla F(x^\star), x^k - x^\star\rangle.$$

Since both terms are non-negative, both terms are a.s. summable. From the first term we have the a.s. convergence of $\mathbf{B}z^k \to \mathbf{B}z^\star$ and Lemma 3.4 then gives that all cluster points of $(z^k)_{k\in\mathbb{N}}$ are almost surely in $Z^\star$. For the second term we note that by convexity we have

$$\langle \nabla F(x^k) - \nabla F(x^\star), x^k - x^\star\rangle \ge F(x^k) - F(x^\star) \ge 0$$

and $F(x^k) - F(x^\star)$ then is summable a.s. since $\xi nL > 0$. Using smoothness of F, (5) and the notation from (7) gives

$$\begin{aligned}F(x^\star) &\le F((K\otimes I)z^k) \\ &= F(x^k + \tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k) \\ &\le F(x^k) + \langle (G\otimes I)\mathbf{B}z^k, \tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k\rangle + \tfrac{L}{2}\|\tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k\|^2 \\ &\le F(x^k) + \|(G\otimes I)\mathbf{B}z^k\|\|\tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k\| + \tfrac{L}{2}\|\tfrac{\lambda}{n}(D\otimes I)\mathbf{B}z^k\|^2 \\ &\to F(x^\star) \text{ a.s.}\end{aligned}$$

since $(G\otimes I)\mathbf{B}z^k \to (G\otimes I)\mathbf{B}z^\star = 0$ and $(D\otimes I)\mathbf{B}z^k \to (D\otimes I)\mathbf{B}z^\star = 0$ almost surely. Therefore we have the a.s. convergence of $F((K\otimes I)z^k) - F(x^\star) \to 0$.

From Proposition 2.1 we can also conclude that $\|z^k - z^\star\|^2_{H\otimes I} + 2\lambda(n-\theta)(F((K\otimes I)z^k) - F(x^\star))$ a.s. converge to a non-negative random variable. Since $F((K\otimes I)z^k) - F(x^\star) \to 0$ a.s. we have that $\|z^k - z^\star\|^2_{H\otimes I}$ also must a.s. converge to a non-negative random variable. Proposition 2.2 then give the almost sure convergence of $(z^k)_{k\in\mathbb{N}}$ to $Z^\star$.

We now show that there exists $\lambda > 0$ and $\xi > 0$ such that

$$
\begin{aligned}
&2M - \mathbb{E}[U_{i^k}^T H U_{i^k}] + \lambda(n-\theta)S - \xi I \\
&\quad= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\otimes\frac{\lambda}{nL}I - \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\otimes\frac{\lambda^2}{n}I + \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\otimes\frac{\lambda^2}{n}(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) + \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\otimes\frac{\lambda^2}{n^2}\mathbf{1}\mathbf{1}^T \\
&\qquad+ \begin{bmatrix}2 & -1\\ -1 & 0\end{bmatrix}\otimes(n-\theta)(\theta-1)\frac{\lambda^2}{n^3}\mathbf{1}\mathbf{1}^T - \begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\otimes(n-\theta)(\theta-1)^2\frac{L\lambda^3}{n^3}I \\
&\qquad- \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\otimes\xi I > 0.
\end{aligned}
$$

We show positive definiteness by ensuring that the smallest eigenvalue is positive. The smallest eigenvalue $\sigma_{\min}$ is greater than the sum of the smallest eigenvalues of each term,

$$
\begin{aligned}
\sigma_{\min} \geq \frac{\lambda}{nL} - \frac{\lambda^2}{n} - \frac{\lambda^2}{n} + 0 + (1 - \tfrac{\theta-1}{|\theta-1|}\sqrt{2})(n-\theta)(\theta-1)\frac{\lambda^2}{n^2} \\
- 2(n-\theta)(\theta-1)^2\frac{L\lambda^3}{n^3} - \xi.
\end{aligned}
$$

Assuming $\lambda \leq \frac{1}{2L}$ yields the following lower bound on the smallest eigenvalue

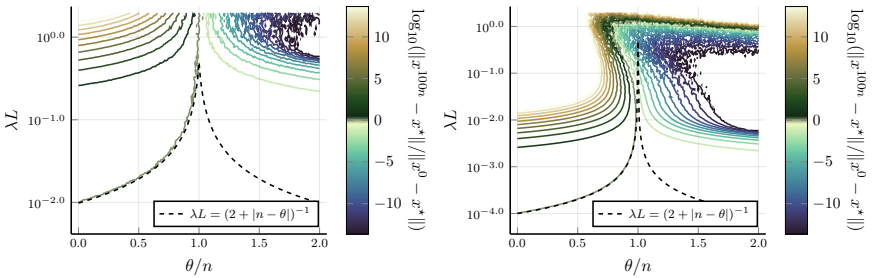
$$
\begin{aligned}
\sigma_{\min} &\geq \frac{\lambda}{nL} - \frac{2\lambda^2}{n} + (1 - \tfrac{\theta-1}{|\theta-1|}\sqrt{2})(n-\theta)(\theta-1)\frac{\lambda^2}{n^2} - (n-\theta)(\theta-1)^2\frac{\lambda^2}{n^3} - \xi \\
&= \frac{\lambda}{n}\big(L^{-1} - \lambda\big(2 + (n-\theta)\tfrac{\theta-1}{n}\big(\tfrac{\theta-1}{n} - 1 + \tfrac{\theta-1}{|\theta-1|}\sqrt{2}\big)\big)\big) - \xi.
\end{aligned}
$$

Selecting

$$
\xi = \frac{\lambda}{2n}\big(L^{-1} - \lambda\big(2 + (n-\theta)\tfrac{\theta-1}{n}\big(\tfrac{\theta-1}{n} - 1 + \tfrac{\theta-1}{|\theta-1|}\sqrt{2}\big)\big)\big),
$$

which satisfy the assumption $\frac{2\lambda}{nL} \geq \xi > 0$, yields $\sigma_{\min} \geq \xi$. Since $\lambda > 0$ by assumption, if

$$
\frac{1}{L}\frac{1}{2 + (n-\theta)\frac{\theta-1}{n}(\frac{\theta-1}{n} - 1 + \frac{\theta-1}{|\theta-1|}\sqrt{2})} > \lambda
$$



(a) $n = 100$ (b) $n = 10000$

Figure 1. Root-finding of Averaged Rotations: Relative distance to the solution after $100n$ iterations of SVAG together the step-size upper bound, $\lambda L < (2 + |n - \theta|)^{-1}$. Note how well the 0th level, i.e., the boundary between convergence and divergence, follow the upper bound on the step-size for $\theta \leq n$.

we have that $\sigma_{\min} \geq \xi > 0$ and hence that the examined matrix is positive definite. Furthermore, if λ satisfies the above inequality it also satisfies the assumption $\lambda \leq \frac{1}{2L}$.

Rates are gotten in the same way as for Theorem 3.7, the total expectation is taken of the Lyapunov inequality at the beginning of the proof and the inequalities are summed from $k = 0$ to $k = t$.

$$\begin{aligned}
&\|z^0 - z^\star\|^2_{H\otimes I} + 2\lambda(n-\theta)(F((K\otimes I)z^0) - F(x^\star)) \\
&\quad \geq \textstyle\sum_{k=0}^{t} \big(\sigma_{\min}\mathbb{E}[\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2] + \mathbb{E}[\sigma_{\min} nL\langle \nabla F(x^k), x^k - x^\star\rangle]\big) \\
&\quad \geq \sigma_{\min}(t+1) \min_{k\in\{1,\dots,t\}} \big(\mathbb{E}[\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2] + \mathbb{E}[nL\langle \nabla F(x^k), x^k - x^\star\rangle]\big) \\
&\quad \geq \sigma_{\min}(t+1) \min_{k\in\{1,\dots,t\}} \big(\mathbb{E}[\|\mathbf{B}z^k - \mathbf{B}z^\star\|^2] + nL\,\mathbb{E}[F(x^k) - F(x^\star)]\big).
\end{aligned}$$

Inserting the lower bound on $\sigma_{\min}$, rearranging and minimizing over $x^\star \in X^\star$ yield the results of the theorem. □

4. Numerical Experiments

A number of experiments, outlined below, were performed to verify the tightness of the theory in the cocoercive operator case and examine the effect of bias in the cocoercive gradient case. The experiments were implemented in `Julia` [3] and, together with several other VR-SG methods, can be found at `https://github.com/mvmorin/VarianceReducedSG.jl`.

4.1 Cocoercive Operators Case

In order for the difference between cocoercive operators and cocoercive gradients to not be an artifact of our analysis, the results in the operator case can not be overly

conservative. We therefore construct a cocoercive operator problem for which the results appear to be tight, thereby verifying the difference. Consider problem (2) where the operator $R_i : \mathbb{R}^2 \to \mathbb{R}^2$ is an averaged rotation

$$R_i = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix}$$

for all $i \in \{1,\dots,n\}$ and some $\tau \in [0,2\pi)$. The operators are 1-cocoercive and the zero vector is the only solution to (2) if $\tau \neq \pi$. The step-size condition from Theorem 3.7 appears to be tight for $\theta \in [0,n]$ when the angle of rotation τ approaches π. We therefore let $\tau = \frac{179}{180}\pi$ and solve the problem with different configurations of step-size λ and innovation weight θ.

Figure 1 displays the relative distance to the solution after $100n$ iterations of SVAG together with the upper bound on the step-size. When $\theta \in [0,n]$ and λ exceeds the upper bound, the distance to the solution increases for both $n = 100$ and $n = 10000$, i.e., the method does not converge. Hence, for $\theta \in [0,n]$, the step-size bound in Theorem 3.7 appears to be tight. However, it is noteworthy that for this particular problem it seems beneficial to exceed the step-size bound when $\theta > n$.

4.2 Cocoercive Gradients Case

Since, as we stated in Section 3.2, we do not believe that the theoretical rates are particularly tight in the optimization case, we examine the effects of the bias numerically. These experiments can of course not be exhaustive and we choose to focus on only the bias parameter θ and therefore perform all experiments with the same step-size. This also demonstrate why we believe the analysis to be conservative since the chosen step-size is in some cases a 1000 times larger than the upper bound from Theorem 3.8. Convergence with this large of a step-size have also been seen elsewhere with both [40] and [17] disregarding their own the theoretical step-size conditions.

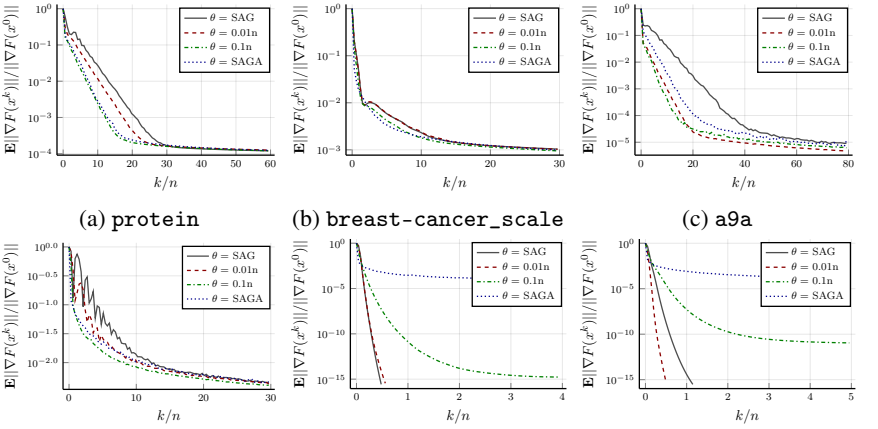
The experiments are done by performing a rough parameter sweep over the innovation weight θ on two different binary classification problems and we will look for patterns in how the convergence is affected. The first problem is logistic regression,

$$\min_{x\in\mathbb{R}^N} \tfrac{1}{n}\sum\nolimits_{i=1}^{n} \log(1 + e^{-y_i a_i^T x}).$$

The second is SVM with a square hinge loss,

$$\min_{x\in\mathbb{R}^N} \tfrac{1}{n}\sum\nolimits_{i=1}^{n} \left(\max(0, 1 - y_i a_i^T x)^2 + \tfrac{\gamma}{2}\|x\|^2\right)$$

where $\gamma > 0$ is a regularization parameter. In both problems are $y_i \in \{-1, 1\}$ the label and $a_i \in \mathbb{R}^N$ the features of the ith training data point. Note, although not initially obvious, $\max(0,\cdot)^2$ is convex and differentiable with Lipschitz continuous derivative



(a) `protein` (b) `breast-cancer_scale` (c) `a9a`

(d) `gisette_scale` (e) `mushrooms` (f) `mnist.scale`

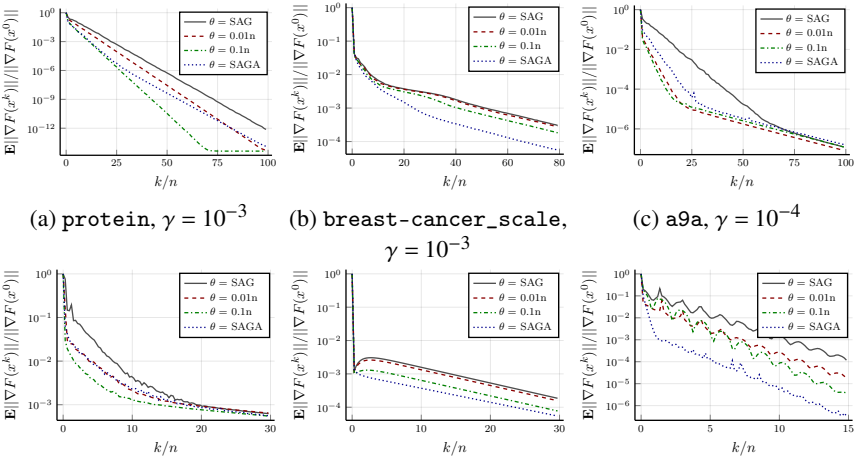
Figure 2. Logistic Regression: Expected gradient norm for each iteration. The expected value is estimated with the sample average of 100 runs. A step-size of $\lambda = \frac{1}{2L}$ was used in all cases.

and the second problem is therefore indeed smooth. The logistic regression problem does not necessarily have a unique solution and the distance to the solution set is therefore hard to estimate. For this reason, we examine the convergence of $\|\nabla F(x^k)\| \to 0$ instead of the distance to the solution set.

The datasets for both these classification problems are taken from the `LibSVM`[6] collection of datasets. The number of examples in the datasets varies between $n = 683$ and $n = 60{,}000$ while the number of features is between $N = 10$ and $N = 5{,}000$. Two of the datasets, `mnist.scale` and `protein`, consist of more than 2 classes. These are converted to binary classification problems by grouping the different classes into two groups. For the digit classification dataset `mnist.scale`, the digits are divided into the groups 0-4 and 5-9. For the `protein` dataset, the classes are grouped as 0 and 1-2. The results of solving the classification problems above can be found in Figures 2 and 3.

From Figures 2 and 3 it appears like the biggest difference between the innovation weights are in the early stages of the convergence. Most innovation weight choices appear to eventually converge with the same rate. In the cases where this does not happen, the fastest converging choice of innovation weight actually reaches machine precision. It is therefore not possible to say whether these cases would eventually reach the same rate as well. Since none of the choices of θ appears to consistently be at a significant disadvantage, even though the step-size used exceeds the upper bound in Theorem 3.8 when $\theta = 0.1n$ and $\theta = 0.01n$, we conjecture that the asymptotic rates for a given step-size is independent of θ.

The initial phase can clearly have a large impact on the convergence and it can



(a) `protein`, $\gamma = 10^{-3}$ (b) `breast-cancer_scale`, $\gamma = 10^{-3}$ (c) `a9a`, $\gamma = 10^{-4}$

(d) `gisette_scale`, $\gamma = 10^{-1}$ (e) `mushrooms`, $\gamma = 10^{-3}$ (f) `mnist.scale`, $\gamma = 10^{-1}$

Figure 3. Square Hinge Loss SVM: Expected gradient norm for each iteration. The expected value is estimated with the sample average of 100 runs. A step-size of $\lambda = \frac{1}{2L}$ was used in all cases.

therefore still be beneficial to tuning the bias. However, comparing the different choices of innovation weight yields no clear conclusion since no single choice of innovation weight consistently outperforms another. In most cases do the lower bias choices—$\theta = n$ (SAGA) or $\theta = 0.1n$—seem perform best but, when they do not, the high bias choices—$\theta = 1$ (SAG) and $\theta = 0.01n$—perform significantly better. Another observation is that lowering θ increases any oscillations. We speculate that it is due to the increased inertia and we also believe that this inertia is what allows the lower innovation weights to sometimes perform better.

5. Conclusion

We presented SVAG, a variance-reduced stochastic gradient method with adjustable bias and with SAG and SAGA as special cases. It was analyzed in two scenarios, one being the minimization of a finite sum of functions with cocoercive gradients and the other being finding a root of a finite sum of cocoercive operators. The analysis improves on the previously best known analyses in both settings and, more significantly, the two different scenarios gave different convergence conditions for the step-size. In the cocoercive operator setting a much more restrictive condition was found and it was verified numerically. This difference is not present in ordinary gradient descent and can therefore easily be overlooked, however, these results suggest that is inadvisable in the variance-reduced stochastic gradient setting.

The theoretical results in the minimization case was further examined with numerical experiments. Several choices of bias were examined but we did not find the same dependence on the bias that the theory suggests. In fact, the asymptotic convergence behavior was similar for the different choices of bias, indicating that further improvements of the theory is still needed. The bias mainly impacted the early stages of the convergence and in a couple of cases this impact was significant. There might therefore still be benefits to tuning the bias to the particular problem but further work is needed to efficiently do so.

References

[1] Z. Allen-Zhu. "Katyusha: The First Direct Acceleration of Stochastic Gradient Methods". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2017. ACM, New York, NY, USA, 2017, pp. 1200–1205. ISBN: 978-1-4503-4528-6. DOI: 10.1145/3055399.3055448.

[2] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Second. CMS Books in Mathematics. Springer International Publishing, 2017. ISBN: 978-3-319-48310-8. URL: //www.springer.com/gp/book/9783319483108 (visited on 2019-01-15).

[3] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A Fresh Approach to Numerical Computing". *SIAM Review* **59**:1 (2017), pp. 65–98. DOI: 10.1137/141000671.

[4] L. M. Briceño-Arias and D. Davis. "Forward-Backward-Half Forward Algorithm for Solving Monotone Inclusions". *SIAM Journal on Optimization* **28**:4 (2018), pp. 2839–2871. DOI: 10.1137/17M1120099.

[5] Y. Carmon, Y. Jin, A. Sidford, and K. Tian. "Variance Reduction for Matrix Games". *Advances in Neural Information Processing Systems* **32** (2019), pp. 11381–11392. URL: https://proceedings.neurips.cc/paper/2019/hash/6c442e0e996fa84f344a14927703a8c1-Abstract.html (visited on 2020-12-17).

[6] C.-C. Chang and C.-J. Lin. "LIBSVM: A Library for Support Vector Machines". *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**:3 (2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27. DOI: 10.1145/1961189.1961199.

[7] T. Chavdarova, G. Gidel, F. Fleuret, and S. Lacoste-Julien. "Reducing Noise in GAN Training with Variance Reduced Extragradient". *Advances in Neural Information Processing Systems* **32** (2019), pp. 393–403. URL: https://proceedings.neurips.cc/paper/2019/hash/58a2fc6ed39fd083f55d4182bf88826d-Abstract.html (visited on 2020-12-17).

[8] P. L. Combettes. "Solving Monotone Inclusions via Compositions of Nonexpansive Averaged Operators". *Optimization* **53**:5-6 (2004), pp. 475–504. DOI: 10.1080/02331930412331327157.

[9] P. L. Combettes and J. Eckstein. "Asynchronous Block-Iterative Primal-Dual Decomposition Methods for Monotone Inclusions". *Mathematical Programming* **168**:1 (2018), pp. 645–672. DOI: 10.1007/s10107-016-1044-0.

[10] P. L. Combettes and L. E. Glaudin. "Solving Composite Fixed Point Problems with Block Updates". *Advances in Nonlinear Analysis* **10**:1 (2021), pp. 1154–1177. DOI: 10.1515/anona-2020-0173.

[11] P. L. Combettes and J.-C. Pesquet. "Primal-Dual Splitting Algorithm for Solving Inclusions with Mixtures of Composite, Lipschitzian, and Parallel-Sum Type Monotone Operators". *Set-Valued and Variational Analysis* **20**:2 (2012), pp. 307–330. DOI: 10.1007/s11228-011-0191-y.

[12] P. L. Combettes and J.-C. Pesquet. "Stochastic Quasi-Fejér Block-Coordinate Fixed Point Iterations with Random Sweeping". *SIAM Journal on Optimization* **25**:2 (2015), pp. 1221–1248. DOI: 10.1137/140971233.

[13] P. L. Combettes and Z. C. Woodstock. "A Fixed Point Framework for Recovering Signals from Nonlinear Transformations". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 2120–2124. DOI: 10.23919/Eusipco47968.2020.9287736.

[14] D. Davis and W. Yin. "A Three-Operator Splitting Scheme and its Optimization Applications". *Set-Valued and Variational Analysis* **25**:4 (2017), pp. 829–858. DOI: 10.1007/s11228-017-0421-z.

[15] A. Defazio, F. Bach, and S. Lacoste-Julien. "SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives". In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 1646–1654. (Visited on 2018-08-27).

[16] A. Defazio, J. Domke, and Caetano. "Finito: A Faster, Permutable Incremental Gradient Method for Big Data Problems". In: *International Conference on Machine Learning*. 2014, pp. 1125–1133. URL: http://proceedings.mlr.press/v32/defazio14.html (visited on 2018-08-27).

[17] D. Driggs, J. Liang, and C.-B. Schönlieb. *On Biased Stochastic Gradient Estimation*. 2020. arXiv: 1906.01133v2 [math]. URL: http://arxiv.org/abs/1906.01133v2 (visited on 2020-03-28).

[18] P. Giselsson. "Nonlinear Forward-Backward Splitting with Projection Correction". *SIAM Journal on Optimization* (2021), pp. 2199–2226. DOI: 10.1137/20M1345062.

[19] A. A. Goldstein. "Convex Programming in Hilbert Space". *Bulletin of the American Mathematical Society* **70**:5 (1964), pp. 709–711. DOI: 10.1090/S0002-9904-1964-11178-2.

[20] R. M. Gower, P. Richtárik, and F. Bach. "Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching". *Mathematical Programming* **188**:1 (2021), pp. 135–192. DOI: 10.1007/s10107-020-01506-0.

[21] F. Hanzely, K. Mishchenko, and P. Richtárik. "SEGA: Variance Reduction via Gradient Sketching". In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 2082–2093. URL: http://papers.nips.cc/paper/7478-sega-variance-reduction-via-gradient-sketching.pdf (visited on 2020-04-30).

[22] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams. "Variance Reduced Stochastic Gradient Descent with Neighbors". In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 2305–2313. (Visited on 2018-08-27).

[23] R. Johnson and T. Zhang. "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 315–323. (Visited on 2018-08-27).

[24] J. Konečný and P. Richtárik. "Semi-Stochastic Gradient Descent Methods". *Frontiers in Applied Mathematics and Statistics* **3** (2017). DOI: 10.3389/fams.2017.00009.

[25] D. Kovalev, S. Horváth, and P. Richtárik. "Don't Jump Through Hoops and Remove Those Loops: SVRG and Katyusha are Better Without the Outer Loop". In: *Proceedings of the 31st International Conference on Algorithmic Learning Theory*. PMLR, 2020, pp. 451–467. URL: https://proceedings.mlr.press/v117/kovalev20a.html (visited on 2021-09-27).

[26] P. Latafat and P. Patrinos. "Primal-Dual Proximal Algorithms for Structured Convex Optimization: a Unifying Framework". In: *Large-Scale and Distributed Optimization*. Lecture Notes in Mathematics. Springer International Publishing, 2018, pp. 97–120. ISBN: 978-3-319-97478-1. DOI: 10.1007/978-3-319-97478-1_5.

[27] N. Le Roux, M. Schmidt, and F. Bach. "A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets". In: *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 2663–2671.

[28] E. S. Levitin and B. T. Polyak. "Constrained Minimization Methods". *USSR Computational mathematics and mathematical physics* **6**:5 (1966), pp. 1–50.

[29] P. L. Lions and B. Mercier. "Splitting Algorithms for the Sum of Two Nonlinear Operators". *SIAM Journal on Numerical Analysis* **16**:6 (1979), pp. 964–979. DOI: 10.1137/0716071.

[30] J. Mairal. "Optimization with First-order Surrogate Functions". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. JMLR.org, Atlanta, GA, USA, 2013, pp. III-783–III-791.

[31] M. Morin and P. Giselsson. "Sampling and Update Frequencies in Proximal Variance Reduced Stochastic Gradient Methods" (2020). arXiv: 2002.05545v2 [cs, math]. URL: http://arxiv.org/abs/2002.05545v2.

[32] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer US, 2004. ISBN: 978-1-4020-7553-7. URL: //www.springer.com/us/book/9781402075537 (visited on 2019-01-15).

[33] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. "SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. JMLR.org, Sydney, NSW, Australia, 2017, pp. 2613–2621. URL: http://proceedings.mlr.press/v70/nguyen17b.html (visited on 2020-04-30).

[34] B. Palaniappan and F. Bach. "Stochastic Variance Reduction Methods for Saddle-Point Problems". In: *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 1416–1424.

[35] X. Qian, Z. Qu, and P. Richtárik. "SAGA with Arbitrary Sampling". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 5190–5199. URL: https://proceedings.mlr.press/v97/qian19a.html (visited on 2021-09-27).

[36] H. Raguet, J. Fadili, and G. Peyré. "A Generalized Forward-Backward Splitting". *SIAM Journal on Imaging Sciences* **6**:3 (2013), pp. 1199–1226. DOI: 10.1137/120872802.

[37] H. Robbins and D. Siegmund. "A Convergence Theorem for Non Negative Almost Supermartingales and Some Applications". In: *Optimizing Methods in Statistics*. Academic Press, 1971, pp. 233–257. URL: https://doi.org/10.1016/B978-0-12-604550-5.50015-8.

[38] R. T. Rockafellar. "Monotone Operators and the Proximal Point Algorithm". *SIAM Journal on Control and Optimization* **14**:5 (1976), pp. 877–898. DOI: 10.1137/0314056.

[39] M. Schmidt, R. Babanezhad, M. Ahmed, A. Defazio, A. Clifton, and A. Sarkar. "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Vol. 38. Proceedings of Machine Learning Research. PMLR, 2015, pp. 819–828. URL: http://proceedings.mlr.press/v38/schmidt15.html (visited on 2020-02-21).

[40] M. Schmidt, N. Le Roux, and F. Bach. “Minimizing Finite Sums with the Stochastic Average Gradient”. *Mathematical Programming* **162**:1 (2017), pp. 83–112. DOI: 10.1007/s10107-016-1030-6.

[41] S. Shalev-Shwartz and T. Zhang. “Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization”. *Journal of Machine Learning Research* **14**:Feb (2013), pp. 567–599. URL: http://www.jmlr.org/papers/v14/shalev-shwartz13a.html (visited on 2018-08-27).

[42] Z. Shi, X. Zhang, and Y. Yu. “Bregman Divergence for Stochastic Variance Reduction: Saddle-Point and Adversarial Prediction”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6033–6043. ISBN: 978-1-5108-6096-4.

[43] M. Tang, L. Qiao, Z. Huang, X. Liu, Y. Peng, and X. Liu. “Accelerating SGD Using Flexible Variance Reduction on Large-Scale Datasets”. *Neural Computing and Applications* (2019). DOI: 10.1007/s00521-019-04315-5.

[44] P. Tseng. “A Modified Forward-Backward Splitting Method for Maximal Monotone Mappings”. *SIAM Journal on Control and Optimization* **38**:2 (2000), pp. 431–446. DOI: 10.1137/S0363012998338806.

[45] L. Xiao and T. Zhang. “A Proximal Stochastic Gradient Method with Progressive Variance Reduction”. *SIAM Journal on Optimization* **24**:4 (2014), pp. 2057–2075. DOI: 10.1137/140961791.

[46] X. Zhang, W. B. Haskell, and Z. Ye. “A Unifying Framework for Variance-Reduced Algorithms for Findings Zeroes of Monotone operators”. *Journal of Machine Learning Research* **23**:60 (2022), pp. 1–44. URL: http://jmlr.org/papers/v23/19-513.html (visited on 2022-08-18).

[47] K. Zhou, Q. Ding, F. Shang, J. Cheng, D. Li, and Z.-Q. Luo. “Direct Acceleration of SAGA using Sampled Negative Momentum”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 1602–1610.

Paper III

Nonlinear Forward-Backward Splitting with Momentum Correction

Martin Morin Sebastian Banert Pontus Giselsson

Abstract

The nonlinear, or warped, resolvent recently explored by Giselsson and Bùi-Combettes has been used to model a large set of existing and new monotone inclusion algorithms. To establish convergent algorithms based on these resolvents, corrective projection steps are utilized in both works. We present a different way of ensuring convergence by means of a nonlinear momentum term, which in many cases leads to cheaper per-iteration cost. The expressiveness of our method is demonstrated by deriving a wide range of special cases. These cases cover and expand on the forward-reflected-backward method of Malitsky-Tam, the primal-dual methods of Vũ-Condat and Chambolle-Pock, and the forward-reflected-Douglas-Rachford method of Ryu-Vũ. A new primal-dual method that uses an extra resolvent step is also presented as well as a general approach for adding momentum to any special case of our nonlinear forward-backward method, in particular all the algorithms listed above.

Submitted and under review.

1. Introduction

Given a real Hilbert space $\mathcal{H}$, we consider the problem of finding a zero $x \in \mathcal{H}$ of the sum of a maximally monotone operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ and a cocoercive operator $C\colon \mathcal{H} \to \mathcal{H}$, i.e.,

$$0 \in Ax + Cx. \tag{1}$$

If the resolvent $(\mathrm{Id} + A)^{-1}$ of A is easily computable, this problem can be solved with the forward-backward splitting method [28, 33]. Since this might not be the case, great effort has been devoted to constructing other splitting methods that can exploit any additional structure of A, sometimes further assuming $C = 0$ [10, 18, 19, 20, 22, 34, 40]. This work presents an alternative approach for analyzing and constructing such splitting methods by formulating them as different instances of a forward-backward method with a nonlinear resolvent $(M + A)^{-1} \circ M$ where $M\colon \mathcal{H} \to \mathcal{H}$ is a (potentially) nonlinear kernel.

Nonlinear resolvents–or warped resolvents in the terminology of [15]–were recently explored in [15, 26] with precursors available in [30, 31]. These works are preceded by, or developed in parallel with, several other generalizations to the concept of a resolvent. Using a resolvent with a strongly positive self-adjoint bounded linear kernel P in the standard forward-backward method has long been known to converge. In fact, it is simply forward-backward splitting applied to the scaled problem $0 \in P^{-1}Ax + P^{-1}Cx$, which is a monotone inclusion problem in the Hilbert space given by the inner product $\langle P(\cdot), \cdot \rangle$. The conditions on the kernel have been further relaxed in [32], which allows for non-self-adjoint linear kernels. In the multiple works on Bregman-distance based resolvents, for instance [3, 5, 6, 11, 14, 16, 23], the linearity condition is dropped altogether by allowing the kernel to be the gradient of some differentiable convex function. These relaxations allow the resolvent to be adapted to a particular problem, either to improve the speed of convergence or to make an otherwise intractable resolvent evaluation tractable. However, this extra freedom may come at a cost. The algorithms of [15, 26, 31, 32] all need an extra corrective projection step to ensure that any nonlinearities and asymmetries of the kernel do not prevent convergence. The primary contribution of this paper is a different approach for correcting the update, removing the need to perform a potentially expensive projection. Convergence is instead ensured with a corrective momentum term that reuses information from previous iterations, making it possible to achieve lower per-iteration costs.

The strength of nonlinear resolvents lies in their substantial modeling power which allows for a unified view of a large set of algorithms. Both [15, 26] present numerous algorithms that can be interpreted as forward-backward methods with nonlinear resolvents. Our new nonlinear forward-backward method further expands on these modeling capabilities and the second half of this paper is dedicated to deriving both new and existing algorithms as special cases.

Among already existing methods, we show that the forward-(half)-reflected-

backward method in [36] is a special case of our method and highlight its connection to the similar forward-backward-(half)-forward method [12, 42] via the nonlinear resolvent. We present two new four-operator primal-dual splitting methods, the first of which has, among others, Vũ-Condat [21, 43] and Chambolle-Pock [17] as special cases. Vũ-Condat and Chambolle-Pock have been shown to be ordinary forward-backward methods [29] and to have Douglas-Rachford splitting [34] as a special case.[1] Our first primal-dual method is an expansion of this to the nonlinear resolvent setting, giving us the forward-reflected-Douglas-Rachford method of [41] and the novel forward-half-reflected-Douglas-Rachford method as special cases. Our second primal-dual method solves the same problem as the first one but utilizes three resolvent steps, two of which are of the same operator. This method is, as far as we know, completely novel.

Different kinds of momentum have long been used to accelerate the convergence of first-order methods [1, 2, 7, 8, 9, 35, 37, 39] and, due to the use of a momentum-like correction term, our nonlinear forward-backward method naturally lend itself to modeling momentum methods. Momentum can be incorporated directly into the design of a special case of our main algorithm but we also present an approach to add momentum to any special case, regardless of whether it initially was designed with momentum or not. The approach is demonstrated on the forward-half-reflected-backward method of [36], which gives a novel momentum algorithm that extends the relaxed momentum algorithm in [36] to include a cocoercive term. Our convergence conditions compare favorably to previous work with a larger range of possible choices of the momentum parameter, even in the more restrictive special case of ordinary forward-backward splitting with momentum.

1.1 Outline

We start by presenting basic notation, preliminary results, and define some operator properties. The proposed nonlinear forward-backward algorithm, along with all necessary assumptions on both the problem (1) and the different design parameters, is presented in Section 2. Section 3 contains the main convergence proof.

In the remainder of the paper, we present and discuss new or already existing special cases of our nonlinear forward-backward method. Section 4 presents a way of adding momentum to any special case of our main algorithm. Section 5 derives the forward-half-reflected-backward method of [36] as a special case and uses the previously presented approach to add momentum to it. Two new primal-dual methods are derived in Section 6. Section 6.1 contains an algorithm that expands on the methods of Vũ-Condat and Chambolle-Pock as well as the forward-reflected-Douglas-Rachford of [41]. In Section 6.2 a, to the authors' knowledge, completely

[1] In order to formulate the standard Douglas-Rachford as a forward-backward method, singular resolvent kernels needs to be allowed. The analysis of this paper will not allow for this but can be modified to do so.

new primal-dual method that uses one additional resolvent evaluation per iteration is derived. We end the paper with a brief conclusion.

1.2 Notation and Preliminaries

Let $\mathbb{R}$ be the set of real numbers, $\mathbb{N} = \{0,1,\dots\}$ be the set of natural numbers, $\mathbb{N}_+ = \{1,2,\dots\}$ be the set of non-zero natural numbers, and let $\mathcal{H}$ be a real Hilbert space. The set $\mathcal{P}(\mathcal{H})$ is the set of bounded linear operators $S\colon \mathcal{H} \to \mathcal{H}$ that are self-adjoint and strongly positive, i.e., there exists $m > 0$ such that

$$\langle Sx, x\rangle \geq m\|x\|^2, \quad \forall x \in \mathcal{H}.$$

If $S \in \mathcal{P}(\mathcal{H})$, then S is invertible and $S^{-1} \in \mathcal{P}(\mathcal{H})$.

For the remainder of this section, we let $S \in \mathcal{P}(\mathcal{H})$. The scaled inner product is defined as $\langle \cdot,\cdot\rangle_S = \langle S(\cdot),\cdot\rangle$ and the scaled norm as $\|\cdot\|_S = \sqrt{\langle \cdot,\cdot\rangle_S}$. The unscaled and scaled norms are equivalent, i.e., there exist $M,m > 0$ such that $M\|x\| \geq \|x\|_S \geq m\|x\|$ for all $x \in \mathcal{H}$. For all $a,b,c,d \in \mathcal{H}$, we have the identity

$$2\langle a-b, d-c\rangle_S = \|a-c\|_S^2 - \|b-c\|_S^2 - \|a-d\|_S^2 + \|b-d\|_S^2. \tag{2}$$

A set-valued operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is *monotone* if

$$\langle u-v, x-y\rangle \geq 0, \quad \forall (x,u),(y,v) \in \operatorname{gra} A$$

where $\operatorname{gra} A = \{(x,u) \mid u \in Ax\}$ is the graph of A. An operator A is *maximally monotone* if it is monotone and its graph is not a proper subset of the graph of another monotone operator.

For $\mu > 0$, a maximally monotone operator $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is *μ-strongly monotone w.r.t. S* if

$$\langle u-v, x-y\rangle \geq \mu\|x-y\|_S^2, \quad \forall u \in Ax, \forall v \in Ay, \forall x,y \in \mathcal{H}.$$

This definition is equivalent to ordinary μ-strong monotonicity of $S^{-1} \circ A$ in the Hilbert space given by the scaled inner product $\langle \cdot,\cdot\rangle_S$. The analogous equivalences hold for the two following definitions as well. For $L \geq 0$, an operator $B\colon \mathcal{H} \to \mathcal{H}$ is *L-Lipschitz continuous w.r.t. S* if

$$\|Bx - By\|_{S^{-1}} \leq L\|x-y\|_S, \quad \forall x,y \in \mathcal{H}.$$

For $\ell > 0$, an operator $C\colon \mathcal{H} \to \mathcal{H}$ is *ℓ^{-1}-cocoercive w.r.t. S* if

$$\langle Cx - Cy, x-y\rangle \geq \ell^{-1}\|Cx - Cy\|_{S^{-1}}^2, \quad \forall x,y \in \mathcal{H}.$$

An ℓ^{-1}-cocoercive operator w.r.t. S is ℓ-Lipschitz continuous w.r.t. S. For all operator properties, if no scaling S is explicitly stated, we mean $S = \mathrm{Id}$.

Let C be an ℓ^{-1}-cocoercive operator w.r.t. S. Then the following three-point inequality holds:

$$\langle Cx - Cy, z - y\rangle \geq -\tfrac{\ell}{4}\|z - x\|_S^2, \quad \forall x, y, z \in \mathcal{H}. \tag{3}$$

This is shown by inserting $x - x$ in the inner product on the left-hand side and using cocoercivity and Young's inequality,

$$\begin{aligned}
\langle Cx - Cy, z - y\rangle &= \langle Cx - Cy, z - x\rangle + \langle Cx - Cy, x - y\rangle \\
&\geq \langle Cx - Cy, z - x\rangle + \ell^{-1}\|Cx - Cy\|_{S^{-1}}^2 \\
&= \langle S^{-\frac{1}{2}}(Cx - Cy), S^{\frac{1}{2}}(z - x)\rangle + \ell^{-1}\|Cx - Cy\|_{S^{-1}}^2 \\
&\geq -\tfrac{\epsilon}{2}\|Cx - Cy\|_{S^{-1}}^2 - \tfrac{1}{2\epsilon}\|z - x\|_S^2 + \ell^{-1}\|Cx - Cy\|_{S^{-1}}^2
\end{aligned}$$

where $\epsilon > 0$. Selecting $\epsilon = 2\ell^{-1}$ yields the desired inequality (3).

2. Problem and Algorithm

Apart form the general problem structure of (1), we further assume that the operators satisfy the following standard assumptions.

ASSUMPTION 2.1
The operators of (1) satisfy:

(i) $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ *is maximally monotone.*

(ii) $C\colon \mathcal{H} \to \mathcal{H}$ *is* ℓ^{-1}*-cocoercive w.r.t.* S*, where* $S \in \mathcal{P}(\mathcal{H})$.

(iii) $\operatorname{zer}(A + C) \neq \emptyset$.

If $C = 0$*, we set* $\ell = \ell^{-1} = 0$.

Since $\operatorname{dom} C = \mathcal{H}$, the sum $A + C$ is maximally monotone and the problem could be reformulated as finding a zero of the single maximally monotone operator $A + C$. However, as in ordinary forward-backward splitting, separating the problem into a maximally monotone and a cocoercive term and utilizing this structure will prove beneficial. The fact that we assume cocoercivity w.r.t. S entails no real restriction on the problem since the scaled norm $\|\cdot\|_S$ is equivalent to $\|\cdot\|$. A cocoercive operator w.r.t. S is therefore also cocoercive w.r.t. all other $\hat{S} \in \mathcal{P}(\mathcal{H})$ and vice versa, but with different cocoercivity constants.

The cocoercivity scaling S is utilized directly in our algorithm. In the simplest setting, S acts as a form of preconditioning used to better adapt the algorithm to the specific geometry of the problem. It can also be used as a more general design parameter with different choices of S yielding different instances of our algorithm, see the primal-dual methods in Section 6 for examples of this. Along with the scaling S,

the algorithm has two additional iteration-dependent design parameters, a nonlinear kernel $M_k : \mathcal{H} \to \mathcal{H}$ and a positive momentum parameter $\gamma_k > 0$:

Algorithm 2.1 Nonlinear Forward-Backward with Momentum Correction

Consider problem (1) and let S be such that Assumption 2.1 is satisfied. With $x_0, u_0 \in \mathcal{H}$, for all $k \in \mathbb{N}$ iteratively perform

$$x_{k+1} = (M_k + A)^{-1}(M_k x_k - C x_k + \gamma_k^{-1} u_k),$$
$$u_{k+1} = (\gamma_k M_k - S)x_{k+1} - (\gamma_k M_k - S)x_k,$$

where $M_k : \mathcal{H} \to \mathcal{H}$ and $\gamma_k > 0$.

Compared to [15, 26], the elements of the sequence $(x_k)_{k\in\mathbb{N}}$ are given directly by a nonlinear forward-backward step and do not need an extra projection step. Convergence is instead ensured by the addition of the corrective term u_k to the forward step. The main benefit of this approach is in how the corrective term u_k is computed. Both Algorithm 2.1 and the corresponding algorithm with projection correction [26, Algorithm 3.1] will in general need to evaluate M_k at two points. For Algorithm 2.1, the two points are x_k and x_{k+1} but this means that M_k and M_{k+1} are evaluated at the same point, i.e., x_{k+1}. The cost of one of these evaluations can then be reduced if M_k and M_{k+1} are similar, for instance if $M_{k+1}x_{k+1}$ is a scalar multiplication of $M_k x_{k+1}$. In order for [26, Algorithm 3.1] to also evaluate M_k at x_k and x_{k+1}, it is required that all $M_k = \alpha_k^{-1} S$ with $S \in \mathcal{P}(\mathcal{H})$ and $\alpha_k > 0$ for all $k \in \mathbb{N}$. The only instance of [26, Algorithm 3.1] that satisfies this condition is ordinary forward-backward splitting in the scaled metric given by $\|\cdot\|_S$. This is in contrast to our work where all but one–Algorithm 6.3–of the special cases we cover have kernels that allow this reduction in computational cost.

The more similar M_k and $\gamma_k^{-1} S$ are in Algorithm 2.1, the more similar the nonlinear resolvent is to an ordinary scaled resolvent $(\gamma_k^{-1} S + A)^{-1} \circ \gamma_k^{-1} S$ and the smaller the corrective term u_{k+1} will be. No correction, i.e., $u_{k+1} = 0$, is applied when $M_k = \gamma_k^{-1} S$ and Algorithm 2.1 then reduces to ordinary forward-backward splitting. We quantify the difference between M_k and $\gamma_k^{-1} S$ in the following assumption on the design parameters of Algorithm 2.1.

ASSUMPTION 2.2
Assume that:

(i) The sequence $(\gamma_k)_{k\in\mathbb{N}}$ is positively lower bounded, i.e., $\gamma_k \geq \gamma$ for some $\gamma > 0$ for all $k \in \mathbb{N}$.

(ii) The nonlinear kernel $M_k : \mathcal{H} \to \mathcal{H}$ is such that $\gamma_k M_k - S$ is L_k-Lipschitz continuous w.r.t. S for all $k \in \mathbb{N}$.

These assumptions will form the basis of our convergence analysis. First, we will use them to infer a few useful properties of the nonlinear kernel M_k.

PROPOSITION 2.3
Let Assumption 2.2 hold with $L_k < 1$ for all $k \in \mathbb{N}$. Then M_k is $2\gamma^{-1}$-Lipschitz continuous w.r.t. S, maximally monotone, and strongly monotone w.r.t. S for all $k \in \mathbb{N}$.

Proof. The kernel M_k satisfies $M_k = \gamma_k^{-1}(\gamma_k M_k - S) + \gamma_k^{-1} S$ and therefore is it $\gamma_k^{-1}(1 + L_k)$-Lipschitz continuous w.r.t. S. Since $L_k < 1$ and $\gamma_k \geq \gamma$, the Lipschitz continuity claim is proven. L_k-Lipschitz continuity of $\gamma_k M_k - S$ gives

$$\begin{aligned} L_k^2 \|x - y\|_S^2 &\geq \|(\gamma_k M_k - S)x - (\gamma_k M_k - S)y\|_{S^{-1}}^2 \\ &= \|\gamma_k M_k x - \gamma_k M_k y\|_{S^{-1}}^2 + \|S(x - y)\|_{S^{-1}}^2 \\ &\quad - 2\gamma_k \langle M_k x - M_k y, x - y \rangle \\ &\geq \|x - y\|_S^2 - 2\gamma_k \langle M_k x - M_k y, x - y \rangle. \end{aligned}$$

Since $L_k < 1$, rearranging this expression yields the $\frac{1-L_k^2}{2\gamma_k}$-strong monotonicity w.r.t. S of M_k. Maximality of M_k follows from its continuity and monotonicity [4, Corollary 20.28]. □

3. Convergence

The convergence of Algorithm 2.1 will be established by the convergence of a quantity $\mathcal{V}_k$, defined in Lemma 3.2. The quantity $\mathcal{V}_k$ consists of the distance from the corrected iterate $x_k + S^{-1}u_k$ to an arbitrary solution (measured in the scaled norm $\|\cdot\|_S$) and a residual term. Theorem 3.3 will then establish the main convergence result. Before that, we show that the algorithm generates a well-defined infinite sequence.

PROPOSITION 3.1
Let Assumptions 2.1 and 2.2 hold with $L_k < 1$ for all $k \in \mathbb{N}$. Then Algorithm 2.1 generates infinite sequences $(x_k)_{k\in\mathbb{N}}$ and $(u_k)_{k\in\mathbb{N}}$ uniquely determined by x_0 and u_0.

Proof. Since S, C, and M_k are single-valued, it suffices to show that $(M_k + A)^{-1}$ is also single-valued and has full domain. By Proposition 2.3, the kernel M_k is maximally monotone and strongly monotone w.r.t. S, which implies maximal monotonicity and strong monotonicity w.r.t. Id as well. The kernel has full domain, $\operatorname{dom} M_k = \mathcal{H}$, so the sum $M_k + A$ is maximally monotone and strongly monotone with $\operatorname{ran}(M_k + A) = \mathcal{H}$ and hence $\operatorname{dom}(M_k + A)^{-1} = \mathcal{H}$ [4, Corollary 25.28]. Since $M_k + A$ is strongly monotone, $(M_k + A)^{-1}$ is cocoercive and hence Lipschitz continuous and single-valued [4, Example 22.7]. □

LEMMA 3.2
Let $z \in \operatorname{zer}(A + C)$ and let Assumptions 2.1 and 2.2 hold with $L_k < 1$ for all $k \in \mathbb{N}$. Then Algorithm 2.1 satisfies

$$(1 - L_{k-1} - L_k - \tfrac{\gamma_k \ell}{2})\|x_{k+1} - x_k\|_S^2 \leq \mathcal{V}_k - \mathcal{V}_{k+1} \tag{4}$$

for all $k \in \mathbb{N}_+$ where

$$\mathcal{V}_k = \|x_k + S^{-1}u_k - z\|_S^2 + (1 - L_{k-1})L_{k-1}\|x_k - x_{k-1}\|_S^2.$$

Proof. By Proposition 3.1 we have that sequences $(x_k)_{k\in\mathbb{N}}$ and $(u_k)_{k\in\mathbb{N}}$ are well-defined, which implies that all quantities of the lemma are well-defined. Let $k \in \mathbb{N}_+$ be arbitrary. From Algorithm 2.1 we know that

$$x_{k+1} = (M_k + A)^{-1}(M_k x_k - Cx_k + \gamma_k^{-1}u_k).$$

Using the definition of $(M_k + A)^{-1}$, multiplying with γ_k and rearranging yields

$$Sx_k - Sx_{k+1} + u_k - u_{k+1} - \gamma_k Cx_k \in \gamma_k Ax_{k+1}.$$

Since $z \in \operatorname{zer}(A + C)$, we have $-Cz \in Az$. Using monotonicity of $\gamma_k A$ and multiplying by 2 gives

$$\begin{aligned}
0 &\leq 2\langle \gamma_k Ax_{k+1} - \gamma_k Az, x_{k+1} - z\rangle \\
&\leq 2\langle Sx_k - Sx_{k+1} + u_k - u_{k+1} - \gamma_k Cx_k + \gamma_k Cz, x_{k+1} - z\rangle \\
&= 2\langle Sx_k + u_k - (Sx_{k+1} + u_{k+1}), x_{k+1} - z\rangle - 2\gamma_k\langle Cx_k - Cz, x_{k+1} - z\rangle.
\end{aligned}$$

Assuming $C \neq 0$ and applying (3) on the last term gives

$$0 \leq 2\langle \xi_k - \xi_{k+1}, x_{k+1} - z\rangle_S + \tfrac{\gamma_k \ell}{2}\|x_{k+1} - x_k\|_S^2$$

where we have set $\xi_k \coloneqq x_k + S^{-1}u_k$. It is clear that this also holds when $C = 0$, since then $\ell = 0$ by definition. Applying (2) to the inner product with $a = \xi_k$, $b = \xi_{k+1}$, $c = z$, $d = x_{k+1}$ yields

$$\begin{aligned}
0 &\leq \|\xi_k - z\|_S^2 - \|\xi_{k+1} - z\|_S^2 + \tfrac{\gamma_k \ell}{2}\|x_{k+1} - x_k\|_S^2 \\
&\quad - \|\xi_k - x_{k+1}\|_S^2 + \|\xi_{k+1} - x_{k+1}\|_S^2 \\
&= \|\xi_k - z\|_S^2 - \|\xi_{k+1} - z\|_S^2 + \tfrac{\gamma_k \ell}{2}\|x_{k+1} - x_k\|_S^2 \\
&\quad - \|S^{-1}u_k - (x_{k+1} - x_k)\|_S^2 + \|u_{k+1}\|_{S^{-1}}^2.
\end{aligned} \tag{5}$$

We can expand the second to last norm, assume $L_{k-1} > 0$ and use Young's inequality to get

$$\begin{aligned}
\|S^{-1}u_k - (x_{k+1} - x_k)\|_S^2 &= \|u_k\|_{S^{-1}}^2 - 2\langle u_k, x_{k+1} - x_k\rangle + \|x_{k+1} - x_k\|_S^2 \\
&\geq -(L_{k-1}^{-1} - 1)\|u_k\|_{S^{-1}}^2 + (1 - L_{k-1})\|x_{k+1} - x_k\|_S^2.
\end{aligned}$$

By definition we have $u_k = (\gamma_{k-1}M_{k-1} - S)x_k - (\gamma_{k-1}M_{k-1} - S)x_{k-1}$ which yields

$$\begin{aligned}
&\|S^{-1}u_k + (x_k - x_{k+1})\|_S^2 \\
&\quad \geq -(1 - L_{k-1})L_{k-1}\|x_k - x_{k-1}\|_S^2 + (1 - L_{k-1})\|x_{k+1} - x_k\|_S^2
\end{aligned}$$

since $\gamma_{k-1}M_{k-1} - S$ is L_{k-1}-Lipschitz continuous w.r.t. S with $L_{k-1} < 1$. We also note that this inequality holds when $L_{k-1} = 0$ since $u_k = 0$ in that case.

Inserting this back into (5) and using Lipschitz continuity of $\gamma_k M_k - S$ on the last term yield

$$\begin{aligned}
0 &\leq \|\xi_k - z\|_S^2 - \|\xi_{k+1} - z\|_S^2 + \tfrac{\gamma_k \ell}{2}\|x_{k+1} - x_k\|_S^2 \\
&\quad + (1 - L_{k-1})L_{k-1}\|x_k - x_{k-1}\|_S^2 - (1 - L_{k-1})\|x_{k+1} - x_k\|_S^2 \\
&\quad + L_k^2\|x_{k+1} - x_k\|_S^2 \\
&= \|\xi_k - z\|_S^2 + (1 - L_{k-1})L_{k-1}\|x_k - x_{k-1}\|_S^2 \\
&\quad - \|\xi_{k+1} - z\|_S^2 - (1 - L_k)L_k\|x_{k+1} - x_k\|_S^2 \\
&\quad - (1 - L_{k-1} - L_k - \tfrac{\gamma_k \ell}{2})\|x_{k+1} - x_k\|_S^2.
\end{aligned}$$

Rearranging this expression gives the inequality of the lemma. □

THEOREM 3.3
Let Assumptions 2.1 and 2.2 hold. If there exists an $\epsilon > 0$ such that

$$1 - L_{k-1} - L_k - \tfrac{\gamma_k \ell}{2} \geq \epsilon \tag{6}$$

for all $k \in \mathbb{N}_+$, then Algorithm 2.1 satisfies the following as $k \to \infty$:

(i) $x_{k+1} - x_k \to 0$,
(ii) $u_k \to 0$,
(iii) $(A + C)x_{k+1} \ni M_k x_k - M_k x_{k+1} + \gamma_k^{-1} u_k + C x_{k+1} - C x_k \to 0$,
(iv) $x_k \rightharpoonup x^\star$ *for some* $x^\star \in \operatorname{zer}(A + C)$.

Proof. Let $z \in \operatorname{zer}(A + C)$. Applying Lemma 3.2 and adding the inequality (4) for $k = 1, \ldots, n$ yields

$$\sum_{k=1}^{n} (1 - L_{k-1} - L_k - \tfrac{\gamma_k \ell}{2})\|x_{k+1} - x_k\|_S^2 \leq \mathcal{V}_1 - \mathcal{V}_{n+1} < \mathcal{V}_1 < \infty.$$

The second to last inequality holds since $0 \leq L_k < 1$ for all $k \in \mathbb{N}$ by the assumptions and the condition (6) of the theorem and therefore is $\mathcal{V}_{n+1}$ nonnegative. Item (i) follows from letting $n \to \infty$ since $(1 - L_{k-1} - L_k - \frac{\gamma_k \ell}{2}) \geq \epsilon > 0$ for all $k \in \mathbb{N}_+$ by the condition of the theorem. Item (ii) follows from (i), the definition of u_k, and from the L_k-Lipschitz continuity of $\gamma_k M_k - S$ where $L_k < 1$ for all $k \in \mathbb{N}$.

Let $k \in \mathbb{N}$. For (iii), we first note from the nonlinear forward-backward step in Algorithm 2.1 that

$$A x_{k+1} \ni M_k x_k - M_k x_{k+1} + \gamma_k^{-1} u_k - C x_k,$$

which, by adding $C x_{k+1}$ to both sides, gives

$$(A + C)x_{k+1} \ni M_k x_k - M_k x_{k+1} + \gamma_k^{-1} u_k + C x_{k+1} - C x_k.$$

The result then follows from (i) and (ii) since for all $k \in \mathbb{N}$, $\gamma_k > \gamma$ and M_k and C are Lipschitz continuous w.r.t. S with constants $2\gamma^{-1}$ and ℓ respectively, see Proposition 2.3 and Assumption 2.1.

Since $A+C$ is maximally monotone, (iii) implies that all weak sequential cluster points of $(x_k)_{k\in\mathbb{N}}$ belong to $\operatorname{zer}(A+C)$ due to weak-strong sequential closedness of graphs of maximal monotone operators [4, Proposition 20.38]. To show the weak convergence result in (iv), in view of [4, Lemma 2.47], it is enough to show that $(\|x_k - z\|_S)_{k\in\mathbb{N}}$ converges for all $z \in \operatorname{zer}(A+C)$. The proof of [4, Lemma 2.47] actually only covers the case when $(\|x_k - z\|)_{k\in\mathbb{N}}$ converges but the generalization is straightforward.

For any $z \in \operatorname{zer}(A+C)$, Lemma 3.2 and the condition $(1 - L_{k-1} - L_k - \frac{\gamma_k \ell}{2}) \geq \epsilon > 0$ give that $(\mathcal{V}_k)_{k\in\mathbb{N}_+}$ is a nonincreasing nonnegative sequence which therefore converges, say, $\mathcal{V}_k \to \nu$. This convergence implies

$$\|x_k + S^{-1}u_k - z\|_S^2 = \mathcal{V}_k - (1 - L_{k-1})L_{k-1}\|x_k - x_{k-1}\|_S^2 \to \nu$$

due to (i) and $0 \leq L_{k-1} < 1$. The sequence $\{x_k + S^{-1}u_k - z\}_{k\in\mathbb{N}}$ is then bounded, which, together with (ii), yields

$$\begin{aligned}\|x_k - z\|_S^2 &= \|(x_k + S^{-1}u_k - z) - S^{-1}u_k\|_S^2 \\ &= \|x_k + S^{-1}u_k - z\|_S^2 + \|u_k\|_{S^{-1}}^2 - 2\langle u_k, x_k + S^{-1}u_k - z\rangle \to \nu\end{aligned}$$

which concludes the proof of (iv). □

4. Additional Momentum

Consider the following variant of Algorithm 2.1 that adds an additional scaled momentum term $\gamma_k^{-1}\theta S(x_k - x_{k-1})$.

Algorithm 4.1 Nonlinear Forward-Backward with Momentum Correction and Additional Momentum

Consider problem (1) and let S be such that Assumption 2.1 is satisfied. With $x_0, x_{-1}, u_0 \in \mathcal{H}$, for all $k \in \mathbb{N}$ iteratively perform

$$\begin{aligned}x_{k+1} &= (M_k + A)^{-1}(M_k x_k - C x_k + \gamma_k^{-1}u_k + \gamma_k^{-1}\theta S(x_k - x_{k-1})), \\ u_{k+1} &= (\gamma_k M_k - S)x_{k+1} - (\gamma_k M_k - S)x_k,\end{aligned}$$

where $M_k : \mathcal{H} \to \mathcal{H}$, $\gamma_k > 0$ and $\theta < 1$.

We will show in Corollary 4.1 that there always exists a $\theta \neq 0$–possibly negative–such that if Algorithm 2.1 converges, so does Algorithm 4.1. This shows that it is always possible to add momentum to an instance of Algorithm 2.1. We will use

this in the next section to develop a new momentum variant of the Forward-Half-Reflected-Backward method. Although it might seem like Algorithm 4.1 has more degrees of freedom than Algorithm 2.1, this is not the case. In fact, Algorithm 4.1 is equivalent to Algorithm 2.1–we show and use this in the proofs below. Algorithm 4.1 is therefore first and foremost a tool for adding momentum to an already known instance of Algorithm 2.1 and the usefulness comes via the following corollary that gives an explicit convergence condition.

COROLLARY 4.1
Let Assumptions 2.1 and 2.2 hold and let $\theta < 1$. If there exists an $\varepsilon > 0$ such that

$$1 - \theta - 2|\theta| - L_{k-1} - L_k - \gamma_k \tfrac{\ell}{2} \geq \varepsilon \tag{7}$$

for all $k \in \mathbb{N}_+$, then Algorithm 4.1 satisfies the following as $k \to \infty$:

(i) $x_{k+1} - x_k \to 0$,

(ii) $u_k \to 0$,

(iii) $(A+C)x_{k+1} \ni M_k x_k - M_k x_{k+1} + \gamma_k^{-1} u_k + \gamma_k^{-1}\theta S(x_k - x_{k-1}) + C x_{k+1} - C x_k \to 0$,

(iv) $x_k \rightharpoonup x^\star$ *for some* $x^\star \in \operatorname{zer}(A+C)$.

Proof. By defining $\hat{\gamma}_k = \frac{\gamma_k}{1-\theta}$, the update of Algorithm 4.1 can equivalently be written as

$$\begin{aligned} x_{k+1} &= (M_k + A)^{-1}(M_k x_k - C x_k + \hat{\gamma}_k^{-1}\hat{u}_k), \\ \hat{u}_{k+1} &= (\hat{\gamma}_k M_k - S)x_{k+1} - (\hat{\gamma}_k M_k - S)x_k \end{aligned} \tag{8}$$

which is the same as the update of Algorithm 2.1 but with $\hat{\gamma}_k$ and $\hat{u}_k$ instead of γ_k and u_k respectively. Algorithm 4.1 is therefore equivalent to Algorithm 2.1. Since, by Assumption 2.2, $\gamma_k M_k - S$ is L_k-Lipschitz w.r.t. S and

$$\hat{\gamma}_k M_k - S = \tfrac{1}{1-\theta}(\gamma_k M_k - S) + \tfrac{\theta}{1-\theta} S$$

we conclude that $\hat{\gamma}_k M_k - S$ is $\frac{L_k + |\theta|}{1-\theta}$-Lipschitz continuous w.r.t. S. We further have that $\hat{\gamma}_k = \frac{\gamma_k}{1-\theta} \geq \frac{\gamma}{1-\theta} > 0$ and Assumption 2.2 is therefore satisfied for (8). The convergence condition (6) from Theorem 3.3 for the algorithm update (8) is then that there exists an $\epsilon > 0$ such that

$$1 - \tfrac{L_{k-1}+|\theta|}{1-\theta} - \tfrac{L_k+|\theta|}{1-\theta} - \tfrac{\gamma_k}{1-\theta}\tfrac{\ell}{2} \geq \epsilon.$$

Multiplication of both sides by $1-\theta$ and noting that $\theta < 1$ gives the equivalent condition that there exists an $\varepsilon > 0$ such that

$$1 - \theta - 2|\theta| - L_{k-1} - L_k - \gamma_k \tfrac{\ell}{2} \geq \varepsilon.$$

The convergence results for Algorithm 4.1 follow directly from Theorem 3.3 and $\hat{u}_{k+1} = \frac{1}{1-\theta} u_{k+1} + \frac{\theta}{1-\theta} S(x_{k+1} - x_k)$. □

COROLLARY 4.2
If the conditions of Theorem 3.3 hold–implying that Algorithm 2.1 converges to a solution of (1)–there exists a $\theta \neq 0$ with $\theta < 1$ such that the conditions of Corollary 4.1 also hold and the additional momentum method in Algorithm 4.1 converges to a solution of (1).

Proof. The assumptions on A, C, S, M_k, and γ_k of Theorem 3.3 and Corollary 4.1 are identical so it is enough to conclude that there exists a $\theta \neq 0$ and $\theta < 1$ such that convergence condition (7) of Corollary 4.1 is implied by the conditions of Theorem 3.3. Since Theorem 3.3 holds, we know that

$$1 - L_{k-1} - L_k - \gamma_k \tfrac{\ell}{2} \geq \epsilon > 0.$$

Since $\epsilon > 0$ there exist a θ such that $-\frac{1}{2}\epsilon < \theta < \frac{1}{6}\epsilon$, $\theta \neq 0$, and $\theta < 1$. Selecting such a θ yields $\frac{1}{2}\epsilon > \theta + 2|\theta| > 0$ and

$$1 - L_{k-1} - L_k - \gamma_k \tfrac{\ell}{2} \geq \epsilon > \tfrac{1}{2}\epsilon + \theta + 2|\theta| > 0.$$

Subtracting $\theta + 2|\theta|$ and defining $\varepsilon = \frac{1}{2}\epsilon$ yield

$$1 - \theta - 2|\theta| - L_{k-1} - L_k - \gamma_k \tfrac{\ell}{2} \geq \varepsilon > 0$$

which is the convergence condition (7) for Algorithm 4.1. □

REMARK 4.3
From Corollary 4.2, we know that we can always add momentum to an instance of Algorithm 2.1 and still get a convergent algorithm. In most cases, the per iteration computational cost of the momentum variant is similar to that of the basic method. However, it is possible for the momentum variant not to be tractable. More precisely, it might not be possible to cheaply evaluate $(M_k + A)^{-1}$ at $M_k x_k - C x_k + \gamma_k^{-1} u_k + \gamma_k^{-1}\theta S(x_k - x_{k-1})$ even though it can be cheaply evaluated at $M_k x_k - C x_k + \gamma_k^{-1} u_k$. We will show an example of this in Algorithm 6.3. For Algorithm 6.3, this problem can be handled by introducing a θ-dependent term in the nonlinear kernel.

5. Forward-Half-Reflected-Backward Splitting

Two examples of existing algorithms that can be interpreted as instances of Algorithm 2.1 are the forward-half-reflected-backward (FHRB) method and its special case, the forward-reflected-backward (FRB) method[2] [36]. FHRB is a method for finding $x \in \mathcal{H}$ such that

$$0 \in Bx + Dx + Cx \tag{9}$$

for which the following assumption holds; FRB solves the same problem but with $C = 0$.

[2] FHRB was referred to as a three-operator splitting variant of FRB in the original work.

ASSUMPTION 5.1
The operators of (9) satisfy:

(i) $B\colon \mathcal{H} \to 2^{\mathcal{H}}$ *is maximally monotone.*

(ii) $D\colon \mathcal{H} \to \mathcal{H}$ *is* δ*-Lipschitz continuous.*

(iii) $B + D$ *is maximally monotone.*

(iv) $C\colon \mathcal{H} \to \mathcal{H}$ *is* β^{-1}*-cocoercive.*

(v) $\operatorname{zer}(B + D + C) \neq \emptyset$.

If $C = 0$*, we set* $\beta = \beta^{-1} = 0$.

It should be noted that in [36] was Assumption 5.1*(ii)* replaced with a monotonicity assumption on D. This assumption implies Assumption 5.1*(ii)* since the sum $B + D$ is maximally monotone if D is maximally monotone with full domain which is the case if D is monotone and Lipschitz continuous. However, our assumptions are slightly more general since we can allow for non-monotone D as long as B can compensate for it.

By letting $A = B + D$, problem (9) can be seen as an instance of our standard problem formulation (1). If we in addition let $S = \mathrm{Id}$, Assumption 5.1 implies that Assumption 2.1 holds with $\ell = \beta$. With these choices, FHRB is obtained from Algorithm 2.1 by choosing $M_k = \alpha_k^{-1}\,\mathrm{Id} - D$ and $\gamma_k = \alpha_k$ for some step-size $\alpha_k > 0$. The backward step of the algorithm becomes

$$(M_k + A)^{-1} = (\alpha_k^{-1}\,\mathrm{Id} - D + B + D)^{-1} = (\mathrm{Id} + \alpha_k B)^{-1} \circ \alpha_k\,\mathrm{Id}\,.$$

Note, the backward step is independent of D and the algorithm will, as we will show next, only depend on D through the forward step. The operator $\gamma_k M_k - S$ used in the correction term becomes

$$\gamma_k M_k - S = \alpha_k(\alpha_k^{-1}\,\mathrm{Id} - D) - \mathrm{Id} = -\alpha_k D,$$

and the complete forward step with momentum correction is

$$\begin{aligned} &M_k x_k - C x_k + \gamma_k^{-1} u_k \\ &\quad = \alpha_k^{-1} x_k - D x_k - C x_k - \alpha_k^{-1}(\alpha_{k-1} D x_k - \alpha_{k-1} D x_{k-1}). \end{aligned}$$

Combining the backward and forward steps yields the full FHRB algorithm, see Algorithm 5.1. In this special case, we do not need to evaluate both $M_{k-1} x_k$ and $M_k x_k$ from scratch since we can reuse the potentially expensive computation of $D x_k$.

Algorithm 5.1 Forward-Half-Reflected-Backward [36]

Consider problem (9). With $x_0, x_{-1} \in \mathcal{H}$ and $\alpha_{-1} > 0$, for all $k \in \mathbb{N}$ iteratively perform

$$x_{k+1} = (\mathrm{Id} + \alpha_k B)^{-1}(x_k - \alpha_k C x_k - (\alpha_k + \alpha_{k-1})Dx_k + \alpha_{k-1} Dx_{k-1})$$

where $\alpha_k > 0$.

COROLLARY 5.2
Let Assumption 5.1 hold and consider problem (9) and Algorithm 5.1. If there exists $\epsilon > 0$ *such that*

$$\epsilon \leq \alpha_k, \quad \alpha_k \delta + \alpha_{k+1}(\delta + \tfrac{\beta}{2}) \leq 1 - \epsilon$$

for all $k \in \mathbb{N}$*, then* $x_k \rightharpoonup x^\star$ *where* $x^\star$ *is a solution to (9).*

Proof. After Assumption 5.1, we concluded that Assumption 2.1 holds for the reformulation of (9) into (1) via $A = B + D$. Assumption 2.2 also holds since $\gamma_k = \alpha_k \geq \epsilon > 0$ and $\gamma_k M_k - S = -\alpha_k D$ is $\alpha_k \delta$-Lipschitz continuous. Inserting γ_k, β, and δ into (6) of Theorem 3.3 then directly gives the step-size condition and the results follow from the theorem. □

These step-size conditions are slightly relaxed compared to the ones in the original work [36]. Our conditions match these when a constant step-size $\alpha_k = \alpha$ is chosen. However, the original work only provides convergence conditions for non-constant step-sizes in the FRB case, i.e., $C = 0$. In that case, [36] proved convergence if $\epsilon \leq 2\alpha_k \leq \delta^{-1} - \epsilon$ for some $\epsilon > 0$ and all $k \in \mathbb{N}$ which is slightly more restrictive than our condition.

REMARK 5.3
The same nonlinear kernel that in this case generates FHRB and FRB yields the forward-backward-half-forward [12] and forward-backward-forward [42] methods when used in the nonlinear forward-backward scheme with projection correction [26]. The two sets of algorithms can therefore be seen to have the same nonlinear forward-backward step but with different correction methods to guarantee convergence. Due to the momentum correction's reuse of old information, FHRB and FRB have cheaper per-iteration costs compared to the projection correction counterparts.

5.1 Forward-Half-Reflected-Backward with Momentum

Consider again problem (9) and the operator choices that generated FHRB; $A = B + D$, $M_k = \alpha_k^{-1}\,\mathrm{Id} - D$, $S = \mathrm{Id}$, and $\gamma_k = \alpha_k$. Using these parameters in Algorithm 4.1 gives the following momentum variant of FHRB.

Algorithm 5.2 Forward-Half-Reflected-Backward with Momentum

Consider problem (9). With $x_0, x_{-1} \in \mathcal{H}$ and $\alpha_{-1} > 0$, for all $k \in \mathbb{N}$ iteratively perform

$$\bar{x}_k = x_k + \theta(x_k - x_{k-1}),$$
$$x_{k+1} = (\mathrm{Id} + \alpha_k B)^{-1}(\bar{x}_k - \alpha_k C x_k - (\alpha_k + \alpha_{k-1})Dx_k + \alpha_{k-1} Dx_{k-1})$$

where $\alpha_k > 0$ and $\theta < 1$.

COROLLARY 5.4
Let Assumption 5.1 hold and consider problem (9) and Algorithm 5.2. If there exists $\epsilon > 0$ *such that*

$$\epsilon \leq \alpha_k, \quad \alpha_k \delta + \alpha_{k+1}(\delta + \tfrac{\beta}{2}) \leq (1 - \theta - 2|\theta|) - \epsilon$$

for all $k \in \mathbb{N}$*, then* $x_k \rightharpoonup x^\star$ *where* $x^\star$ *is a solution to (9).*

Proof. The results follow from Corollary 4.1 analogously to how the results of Corollary 5.2 follow from Theorem 3.3. □

When $C = 0$ this is the same method as [36, Equation 4.1] without relaxation and when $D = 0$ it is forward-backward splitting with momentum. Both of these special cases have been shown to converge under certain conditions but our results expand these conditions in both settings. In the FRB with momentum case, Corollary 5.4 allows for step-sizes that depend on the iteration index k while [36, Theorem 4.3] only allows for constant step-size, $\alpha_k = \alpha$ for all $k \in \mathbb{N}$. In the forward-backward with momentum case, Corollary 5.4 makes it possible to find a convergent step-size α_k for all $\theta \in (-1, \frac{1}{3})$, which is the only result we know of that allows for negative momentum. This is especially interesting considering that the magnitude of negative momentum is allowed to be larger than the magnitude of positive momentum. Our upper bound on the momentum matches other results in the literature for weak sequence convergence–[36] when $C = 0$, [1] when $C = D = 0$, and [37] when $C \neq 0$ and $D = 0$.[3] In the gradient-descent case, larger upper bounds on θ and α_k have been shown to work [25]. These results guarantee ergodic convergence of function values and are not applicable to general monotone inclusion problems.

6. Primal-Dual Methods

Let $\mathcal{K}$, and $\mathcal{G}$ be real Hilbert spaces. We will present two new primal-dual methods for solving the problem of finding $y \in \mathcal{K}$ such that

$$0 \in By + (V^* \circ D \circ V)y + Ey + Fy \tag{10}$$

[3] The work in [37] does not present an explicit convergence condition for a fixed choice of θ. Instead, they present a criterion for selecting an iteration dependent θ_k adaptively. However, in a remark they mention results from [1] which, when combined with their results, yield a convergence criteria for a fixed choice of θ.

where the following assumptions hold.

ASSUMPTION 6.1
The operators of (10) satisfy:

(i) $B\colon \mathcal{K} \to 2^{\mathcal{K}}$ *and* $D\colon \mathcal{G} \to 2^{\mathcal{G}}$ *are maximally monotone.*

(ii) $E\colon \mathcal{K} \to \mathcal{K}$ *is monotone and* δ*-Lipschitz continuous.*

(iii) $F\colon \mathcal{K} \to \mathcal{K}$ *is* β^{-1}*-cocoercive.*

(iv) $V\colon \mathcal{K} \to \mathcal{G}$ *is linear and bounded.*

(v) $\operatorname{zer}(B + (V^* \circ D \circ V) + E + F) \neq \emptyset$.

If $F = 0$*, we set* $\beta = \beta^{-1} = 0$.

By a primal-dual method, we mean a method that, instead of solving (10) directly, solves the equivalent primal-dual problem of finding $y \in \mathcal{K}$ and $z \in \mathcal{G}$ such that

$$0 \in \begin{cases} By + V^* z + Ey + Fy \\ D^{-1} z - Vy. \end{cases} \tag{11}$$

The two primal-dual methods are derived by reformulating this primal-dual problem into our standard form (1) and then applying Algorithm 2.1 with different sets of design parameters. There is no unique way of reformulating (11) into (1) but we set $\mathcal{H} = \mathcal{K} \times \mathcal{G}$ and define, with some abuse of block matrix notation, $A\colon \mathcal{K} \times \mathcal{G} \to 2^{\mathcal{K} \times \mathcal{G}}$ and $C\colon \mathcal{K} \times \mathcal{G} \to \mathcal{K} \times \mathcal{G}$ as

$$A = \underbrace{\begin{bmatrix} B & 0 \\ 0 & D^{-1} \end{bmatrix}}_{\widehat{A}} + \underbrace{\begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix}}_{\widehat{E}} + \underbrace{\begin{bmatrix} 0 & V^* \\ -V & 0 \end{bmatrix}}_{\widehat{V}} \quad \text{and} \quad C = \begin{bmatrix} F & 0 \\ 0 & 0 \end{bmatrix}. \tag{12}$$

Assuming $A + C$ has at least one zero, these operators satisfy Assumption 2.1 since $A = \widehat{A} + \widehat{E} + \widehat{V}$ is the sum of a maximally monotone operator $\widehat{A}$ and two maximally monotone operators $\widehat{E}$ and $\widehat{V}$ with full domains. The properties of $\widehat{A}$, $\widehat{E}$, and $\widehat{V}$ are results of the following: maximal monotonicity of B and D; monotonicity and Lipschitz continuity of E; and the skew-adjointness and linearity of $\widehat{V}$. The first assumption of Assumption 2.1 is then satisfied and the second assumption regarding the cocoercivity of C is easily verified in the standard metric of $\mathcal{K} \times \mathcal{G}$. However, the algorithms in Sections 6.1 and 6.2 will use different scaling operators S and we will therefore defer the derivation of more precise cocoercivity constants to the respective sections since the constants depend on S.

6.1 Block-Triangular Resolvent

To derive our first primal-dual algorithm, we decompose the iterates of Algorithm 2.1 as $x_k = (y_k, z_k)$ with $y_k \in \mathcal{K}$ and $z_k \in \mathcal{G}$ for all $k \in \mathbb{N}$. The algorithm is given by the following design parameters

$$S = \begin{bmatrix} \mathrm{Id} & -\tau V^* \\ -\tau V & \tau\sigma^{-1}\mathrm{Id} \end{bmatrix}, \quad M_k = \underbrace{\begin{bmatrix} \tau^{-1}\mathrm{Id} & 0 \\ -\lambda_k V & \sigma^{-1}\mathrm{Id} \end{bmatrix}}_{\widehat{M}_k} - \widehat{E} - \widehat{V} \quad \text{and} \quad \gamma_k = \tau \tag{13}$$

where $\tau, \sigma > 0$ such that $\tau\sigma\|V\|^2 < 1$ and $\lambda_k \in \mathbb{R}$ for all $k \in \mathbb{N}$. The assumption on τ and σ guarantees that $S \in \mathcal{P}(\mathcal{K} \times \mathcal{G})$. The forward step operator and the correction operator are

$$M_k - C = \begin{bmatrix} \tau^{-1}\mathrm{Id} - E - F & -V^* \\ (1-\lambda_k)V & \sigma^{-1}\mathrm{Id} \end{bmatrix}, \quad \gamma_k M_k - S = \tau \begin{bmatrix} -E & 0 \\ (2-\lambda_k)V & 0 \end{bmatrix}.$$

Inserting these operators into the complete forward step with correction,

$$\begin{aligned} (\hat{y}_k, \hat{z}_k) \coloneqq M_k(y_k, z_k) - C(y_k, z_k) + \gamma_k^{-1}(\gamma_{k-1}M_{k-1} - S)(y_k, z_k) \\ - \gamma_k^{-1}(\gamma_{k-1}M_{k-1} - S)(y_{k-1}, z_{k-1}), \end{aligned}$$

where $(\hat{y}_k, \hat{z}_k) \in \mathcal{K} \times \mathcal{G}$, yields

$$\begin{aligned} \hat{y}_k &= \tau^{-1} y_k - V^* z_k - (2Ey_k - Ey_{k-1}) - Fy_k, \\ \hat{z}_k &= \sigma^{-1} z_k + (1-\lambda_k)Vy_k + (2-\lambda_{k-1})V(y_k - y_{k-1}). \end{aligned}$$

What remains to compute is the backward step. The kernel M_k is designed to cancel out the $\widehat{E}$ and $\widehat{V}$ terms, making only the forward step depend on these operators,

$$(M_k + A)^{-1} = (\widehat{M}_k - \widehat{E} - \widehat{V} + \widehat{A} + \widehat{E} + \widehat{V})^{-1} = (\widehat{M}_k + \widehat{A})^{-1}.$$

This is the inverse of a lower block triangular operator and it can therefore be computed with back substitution according to

$$\begin{aligned} (y_{k+1}, z_{k+1}) = (\widehat{M}_k + \widehat{A})^{-1}(\hat{y}_k, \hat{z}_k) \\ \iff \quad (\hat{y}_k, \hat{z}_k) \in (\widehat{M}_k + \widehat{A})(y_{k+1}, z_{k+1}) \\ \iff \quad \begin{cases} \hat{y}_k \in (\tau^{-1}\mathrm{Id} + B)y_{k+1} \\ \hat{z}_k \in -\lambda_k V y_{k+1} + (\sigma^{-1}\mathrm{Id} + D^{-1})z_{k+1} \end{cases} \\ \iff \quad \begin{cases} y_{k+1} = (\mathrm{Id} + \tau B)^{-1}(\tau \hat{y}_k) \\ z_{k+1} = (\mathrm{Id} + \sigma D^{-1})^{-1}(\sigma \hat{z}_k + \sigma\lambda_k V y_{k+1}). \end{cases} \end{aligned}$$

Inserting the expressions for $\hat{y}_k$ and $\hat{z}_k$ results in the following algorithm.

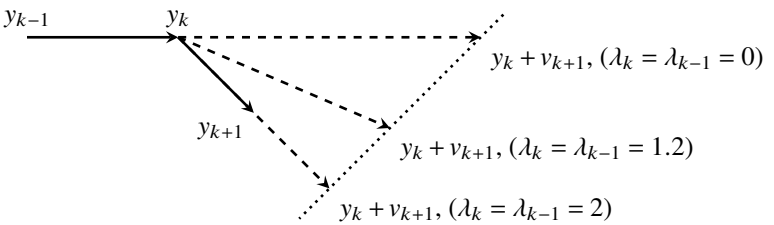


Figure 1. Update of the corrected primal iterate $y_k + v_{k+1}$ in Algorithm 6.1.

Algorithm 6.1 Primal-Dual Method with Block Triangular Resolvent

Consider problem (10). With $y_0, y_{-1} \in \mathcal{K}$, $z_0 \in \mathcal{G}$ and $\lambda_{-1} \in \mathbb{R}$, for all $k \in \mathbb{N}$ iteratively perform

$$\begin{aligned} y_{k+1} &= (\mathrm{Id} + \tau B)^{-1}(y_k - \tau V^* z_k - \tau(2Ey_k - Ey_{k-1}) - \tau F y_k), \\ v_{k+1} &= \lambda_k(y_{k+1} - y_k) + (2 - \lambda_{k-1})(y_k - y_{k-1}), \\ z_{k+1} &= (\mathrm{Id} + \sigma D^{-1})^{-1}(z_k + \sigma V(y_k + v_{k+1})), \end{aligned}$$

where $\tau, \sigma > 0$ and $\lambda_k \in \mathbb{R}$.

Due to the lower block-triangular structure of the operator in the backward step, the primal update of y_{k+1} is independent of the dual update of z_{k+1} but the opposite statement does not hold in general. This dependency is controlled by λ_k and manifests itself as a correction v_{k+1} added to the primal iterate used in the dual update. When $\lambda_k = \lambda_{k-1}$, the correction v_{k+1} is an affine combination of an extrapolation step based either on the current or previous primal update, see Fig. 1. When $\lambda_k \neq \lambda_{k-1}$, the correction can be an arbitrary linear combination of the two different extrapolations. However, the choice of the sequence $(\lambda_k)_{k\in\mathbb{N}}$ will affect the range of allowed step-sizes. The more λ_k differs from 2, the smaller the upper bound on the step-sizes is in the following convergence result.

COROLLARY 6.2
Let Assumption 6.1 hold and consider problem (10) and Algorithm 6.1. If there exists $\epsilon > 0$ such that

$$\tau\sigma\|V\|^2 + (|2 - \lambda_k| + |2 - \lambda_{k+1}|)\sqrt{\tau\sigma}\|V\| + \tau(2\delta + \tfrac{1}{2}\beta) < 1 - \epsilon$$

for all $k \in \mathbb{N}$, then $y_k \rightharpoonup y^\star$ and $z_k \rightharpoonup z^\star$ where $y^\star$ is a solution to (10) and $(y^\star, z^\star)$ is a solution to (11).

Before proceeding to the proof of Corollary 6.2, we present the following lemma on which the proof relies.

LEMMA 6.3
Let $S \in \mathcal{P}(\mathcal{K} \times \mathcal{G})$ be from (13). The inverse of S satisfies

$$S^{-1} = \begin{bmatrix} (\mathrm{Id} - \tau\sigma V^*V)^{-1} & 0 \\ 0 & (\mathrm{Id} - \tau\sigma VV^*)^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{Id} & \sigma V^* \\ \sigma V & \tau^{-1}\sigma\,\mathrm{Id} \end{bmatrix}.$$

The following inequalities hold for all $y \in \mathcal{K}$ and $z \in \mathcal{G}$:

$$\|(y,0)\|_{S^{-1}}^2 \leq \tfrac{1}{1-\tau\sigma\|V\|^2}\|y\|^2, \quad \|(0,z)\|_{S^{-1}}^2 \leq \tfrac{\tau^{-1}\sigma}{1-\tau\sigma\|V\|^2}\|z\|^2$$
$$\text{and} \quad \|y\|^2 \leq \tfrac{1}{1-\tau\sigma\|V\|^2}\|(y,z)\|_S^2.$$

Proof. The inverse is easily verified and we note that, since $\tau\sigma\|V\|^2 < 1$ by assumption, $\mathrm{Id} - \tau\sigma V^*V \in \mathcal{P}(\mathcal{K})$ and $\mathrm{Id} - \tau\sigma VV^* \in \mathcal{P}(\mathcal{G})$ and hence they are invertible. Let $y \in \mathcal{K}$, then

$$\begin{aligned} \|(y,0)\|_{S^{-1}}^2 &= \langle (\mathrm{Id} - \tau\sigma V^*V)^{-1} y, y\rangle \\ &\leq \|(\mathrm{Id} - \tau\sigma V^*V)^{-1}\|\|y\|^2 \\ &\leq \tfrac{1}{1-\tau\sigma\|V\|^2}\|y\|^2 \end{aligned}$$

which proves the first inequality of the lemma. The last step holds since $1 > \tau\sigma\|V\|^2$. Let $z \in \mathcal{G}$, then

$$\begin{aligned} \|(0,z)\|_{S^{-1}}^2 &= \tau^{-1}\sigma\langle (\mathrm{Id} - \tau\sigma VV^*)^{-1} z, z\rangle \\ &\leq \tau^{-1}\sigma\|(\mathrm{Id} - \tau\sigma VV^*)^{-1}\|\|z\|^2 \\ &\leq \tfrac{\tau^{-1}\sigma}{1-\tau\sigma\|V\|^2}\|z\|^2 \end{aligned}$$

which proves the second inequality of the lemma. Again, the last step holds since $1 > \tau\sigma\|V\|^2$. Let $y \in \mathcal{K}$ and $z \in \mathcal{G}$, then

$$\begin{aligned} \|(y,z)\|_S^2 &= \|y\|^2 + \tau\sigma^{-1}\|z\|^2 - 2\tau\langle Vy, z\rangle \\ &\geq \|y\|^2 + \tau\sigma^{-1}\|z\|^2 - \tau(\sigma\|V\|^2\|y\|^2 + \sigma^{-1}\|z\|^2) \\ &= (1 - \tau\sigma\|V\|^2)\|y\|^2 \end{aligned}$$

which proves the third inequality of the lemma. □

Proof of Corollary 6.2. As previously stated, the choice of A and C in (12) satisfies Assumption 2.1 since we assume that a solution exists. What remains to verify of Assumption 2.1 is to derive a cocoercivity constant of C. The first inequality of Lemma 6.3 directly gives

$$\begin{aligned} \|C(y,z) - C(y',z')\|_{S^{-1}}^2 &\leq \tfrac{1}{1-\tau\sigma\|V\|^2}\|Fy - Fy'\|^2 \\ &\leq \tfrac{\beta}{1-\tau\sigma\|V\|^2}\langle Fy - Fy', y - y'\rangle \\ &= \tfrac{\beta}{1-\tau\sigma\|V\|^2}\langle C(y,z) - C(y',z'), (y,z) - (y',z')\rangle \end{aligned}$$

for all $(y,z),(y',z') \in \mathcal{K}\times\mathcal{G}$. Hence, C is ℓ^{-1}-cocoercive w.r.t. S with $\ell = \frac{\beta}{1-\tau\sigma\|V\|^2}$. Note that we can set $\ell = 0$ if $F = 0$.

The assumptions placed on the design parameters, Assumption 2.2, also need to hold. For item (i) of Assumption 2.2, we directly see that $\gamma_k = \tau > 0$. We prove (ii) of Assumption 2.2, the Lipschitz continuity of

$$\gamma_k M_k - S = \tau(\widehat{M}_k - \widehat{V}) - S - \tau\widehat{E},$$

by showing Lipschitz continuity of $\tau\widehat{E}$ and of $\tau(\widehat{M}_k - \widehat{V}) - S$ separately. The Lipschitz continuity of $\gamma_k M_k - S$ then follows from the Lipschitz continuity of a sum of Lipschitz continuous operators. Starting with $\tau\widehat{E}$ and using the first and third inequalities from Lemma 6.3 and the Lipschitz continuity of E gives

$$\begin{aligned}\|\widehat{E}(y,z) - \widehat{E}(y',z')\|_{S^{-1}}^2 &\leq \tfrac{1}{1-\tau\sigma\|V\|^2}\|Ey - Ey'\|^2\\ &\leq \tfrac{\delta^2}{1-\tau\sigma\|V\|^2}\|y - y'\|^2\\ &\leq \tfrac{\delta^2}{(1-\tau\sigma\|V\|^2)^2}\|(y,z) - (y',z')\|_S^2\end{aligned}$$

for all $(y,z),(y',z') \in \mathcal{K}\times\mathcal{G}$. The term $\tau\widehat{E}$ is therefore $\frac{\tau\delta}{1-\tau\sigma\|V\|^2}$-Lipschitz continuous w.r.t. S. For $\tau(\widehat{M}_k - \widehat{V}) - S$, we first note that

$$\tau(\widehat{M}_k - \widehat{V}) - S = \begin{bmatrix} 0 & 0 \\ \tau(2-\lambda_k)V & 0 \end{bmatrix}$$

and we can use the second inequality of Lemma 6.3:

$$\begin{aligned}\|(\tau(\widehat{M}_k - \widehat{V}) - S)(y,z)\|_{S^{-1}}^2 &\leq \tfrac{\tau^{-1}\sigma}{1-\tau\sigma\|V\|^2}\|\tau(2-\lambda_k)Vy\|^2\\ &\leq (2-\lambda_k)^2\tfrac{\tau\sigma\|V\|^2}{1-\tau\sigma\|V\|^2}\|y\|^2\\ &\leq (2-\lambda_k)^2\tau\sigma\|V\|^2\tfrac{1}{(1-\tau\sigma\|V\|^2)^2}\|(y,z)\|_S^2\end{aligned}$$

for all $(y,z) \in \mathcal{K}\times\mathcal{G}$. The operator $\tau(\widehat{M}_k - \widehat{V}) - S$ is therefore Lipschitz continuous w.r.t. S with constant $|2-\lambda_k|\sqrt{\tau\sigma}\|V\|\frac{1}{1-\tau\sigma\|V\|^2}$. Adding these two Lipschitz constants yields that $\gamma_k M_k - S$ is L_k-Lipschitz continuous w.r.t. S where

$$L_k = \tfrac{1}{1-\tau\sigma\|V\|^2}(|2-\lambda_k|\sqrt{\tau\sigma}\|V\| + \tau\delta),$$

and Assumption 2.2 is satisfied. The result of the corollary now follows from Theorem 3.3 after inserting the expressions for ℓ and L_k into the convergence criterion $0 < \epsilon \leq 1 - L_k - L_{k-1} - \tau\frac{\ell}{2}$. □

Related Algorithms From Algorithm 6.1, when $E = 0$ and $\lambda_k = 2$ for all $k \in \{-1,0,\dots\}$, we obtain an instance of the Vũ-Condat algorithm [21, 43]. If $F = 0$ as

well, we get the method of Chambolle-Pock [17]. This is not surprising since both of these methods are special cases of ordinary forward-backward splitting and the kernel M_k, see (13), is linear, self-adjoint, and can be made strongly positive when $E = 0$ and $\lambda_k = 2$. Furthermore, we have that $\gamma_k M_k - S = 0$, which implies that the momentum-correction term is zero and that Algorithm 2.1 has reduced to the ordinary forward-backward method. Both when $F \neq 0$ and when $F = 0$, Corollary 6.2 regains the convergence criteria of Vũ-Condat and Chambolle-Pock respectively.

When $E = 0$, Algorithm 6.1 shares similarities with the asymmetric-kernel primal-dual method of Latafat and Patrinos [32, Algorithm 3]. They use the same resolvent kernel, but [26] showed that the Latafat-Patrinos algorithm is a special case of nonlinear forward-backward splitting with projection correction instead of momentum correction. As discussed in Section 2 when comparing momentum and projection corrections, the main benefit of Algorithm 6.1 is that the momentum correction generally yields cheaper iterations. In Algorithm 6.1, the linear composition term V and its adjoint V^* only need to be evaluated once each, while they need to be evaluated twice each for the Latafat-Patrinos method.

We can also relate Algorithm 6.1 to projective splitting methods [18, 24]. It has been shown in [13, 27] that these methods are nonlinear forward-backward method with projection correction. In fact, the synchronous projective splitting considered in [27] is using the same kernel as in Algorithm 6.1 with $E = 0$ and $\lambda_k = 0$. We can therefore think of Algorithm 6.1 with $E = F = 0$ and $\lambda_k = 0$ for all $k \in \{-1,0,\dots\}$ as a projective splitting method with momentum correction instead of a projection correction. The benefit of projective splitting methods compared to Chambolle-Pock-like primal-dual methods is that the primal and dual updates do not depend on each other and can therefore be performed in parallel. The same holds for Algorithm 6.1 since the correction v_{k+1} does not depend on y_{k+1} when $\lambda_k = 0$. The reason for this becomes evident when examining the backward step $(M_k + A)^{-1} = (\widehat{M}_k + \widehat{A})^{-1}$ since both $\widehat{M}_k$ and $\widehat{A}$ are block-diagonal when $\lambda_k = 0$, see (12) and (13).

Forward-Half-Reflected-Douglas–Rachford There is a connection between primal/dual methods and Douglas-Rachford splitting [17, 38], and this connection also exists for our first primal-dual method, Algorithm 6.1. Whenever $V = \mathrm{Id}$ and $F = 0$, choosing $\lambda_k = 2$ for all $k \in \{-1,0,\dots\}$, $\sigma = \varsigma^{-1}$ for some $\varsigma > 0$ and using Moreau's identity in the dual update of Algorithm 6.1, the forward-reflected-Douglas-Rachford (FRDR) method in [41] is obtained. Since we can allow for $F \neq 0$, we can analogously construct a forward-half-reflected-Douglas-Rachford method for solving (10).

Algorithm 6.2 Forward-Half-Reflected-Douglas–Rachford

Consider problem (10) with $V = \mathrm{Id}$. With $y_0, y_{-1} \in \mathcal{K}$ and $z_0 \in \mathcal{G}$, for all $k \in \mathbb{N}$ iteratively perform

$$\begin{aligned} y_{k+1} &= (\mathrm{Id} + \tau B)^{-1}(y_k - \tau z_k - \tau(2Ey_k - Ey_{k-1}) - \tau F y_k), \\ \hat{y}_{k+1} &= (\mathrm{Id} + \varsigma D)^{-1}(\varsigma z_k + 2y_{k+1} - y_k), \\ z_{k+1} &= z_k + \varsigma^{-1}(2y_{k+1} - y_k - \hat{y}_{k+1}), \end{aligned}$$

where $\tau, \sigma > 0$.

The convergence conditions match those of [41] when $F = 0$:

COROLLARY 6.4
Let $V = \mathrm{Id}$ and let Assumption 5.1 hold. Consider problem (10) and Algorithm 6.2. If the step-sizes satisfy

$$\tau(\varsigma^{-1} + 2\delta + \tfrac{1}{2}\beta) < 1,$$

then $y_k \rightharpoonup y^\star$ and $z_k \rightharpoonup z^\star$ where $y^\star$ is a solution to (10) and $(y^\star, z^\star)$ is a solution to (11).

Proof. Follows directly from Corollary 6.2 with $V = \mathrm{Id}$ and $\lambda_{k-1} = 2$ for all $k \in \mathbb{N}$.□

When $E = F = 0$, the standard Douglas-Rachford is retrieved from Algorithm 6.2 if the step-sizes $\tau = \varsigma$ are chosen and the variable change $z_k = y_k - \tau z_k$ is made. However, this step-size choice makes the step-size condition of Corollary 6.4 impossible to satisfy. The reason for this is that the scaling S of the underlying nonlinear forward-backward method becomes singular, which violates Assumption 2.1. Dealing with this singularity is possible if it is explicitly assumed that $E = F = 0$, but this is beyond the scope of this article, where the positive definiteness of S is assumed.

6.2 Resolvent-Compensated Kernel

Our second method for solving (10) through the primal-dual problem (11) will make further use of the nonlinearity of the kernel by including resolvent evaluations in the kernel itself. As in the previous case, we reformulate the primal-dual problem to our standard problem (1) by defining $\mathcal{H}$, A, C, $\widehat{A}$, $\widehat{E}$, and $\widehat{V}$ as in (12). The iterates of Algorithm 2.1 are decomposed as $x_k = (y_k, z_k)$ with $y_k \in \mathcal{K}$ and $z_k \in \mathcal{G}$ for all $k \in \mathbb{N}$. The second primal-dual algorithm is then given by Algorithm 2.1 with the following

design parameters:

$$M_k = \underbrace{\begin{bmatrix} \tau^{-1}\operatorname{Id} - V^* \circ (\operatorname{Id} + \sigma D^{-1})^{-1} \circ T_{-z_k} \circ \sigma V & 0 \\ 0 & \sigma^{-1}\operatorname{Id} \end{bmatrix}}_{\widehat{M}_k} - \widehat{E},$$
$$S = \begin{bmatrix} \operatorname{Id} & 0 \\ 0 & \tau\sigma^{-1}\operatorname{Id} \end{bmatrix} \quad \text{and} \quad \gamma_k = \tau \tag{14}$$

where $\tau, \sigma > 0$ and $T_a : \mathcal{G} \to \mathcal{G} : z \mapsto z - a$ is the translation by $a \in \mathcal{G}$. Note that the current iterate z_k is used in the construction of M_k and that $S \in \mathcal{P}(\mathcal{K} \times \mathcal{G})$ for all $\tau, \sigma > 0$.

With these design parameters, the correction operator becomes

$$\gamma_k M_k - S = \tau \begin{bmatrix} -E - V^* \circ (\operatorname{Id} + \sigma D^{-1})^{-1} \circ T_{-z_k} \circ \sigma V & 0 \\ 0 & 0 \end{bmatrix}. \tag{15}$$

Inserting this and the other operators into the forward step,

$$\begin{aligned}(\hat{y}_k, \hat{z}_k) \coloneqq M_k(y_k, z_k) - C(y_k, z_k) + \gamma_k^{-1}(\gamma_{k-1} M_{k-1} - S)(y_k, z_k) \\ - \gamma_k^{-1}(\gamma_{k-1} M_{k-1} - S)(y_{k-1}, z_{k-1}),\end{aligned}$$

where $(\hat{y}_k, \hat{z}_k) \in \mathcal{K} \times \mathcal{G}$, yields

$$\begin{aligned}\hat{y}_k &= \tau^{-1} y_k - (2E y_k + E y_{k-1}) - F y_k \\ &\quad - V^*(\operatorname{Id} + \sigma D^{-1})^{-1}(z_k + \sigma V y_k) \\ &\quad - V^*(\operatorname{Id} + \sigma D^{-1})^{-1}(z_{k-1} + \sigma V y_k) \\ &\quad + V^*(\operatorname{Id} + \sigma D^{-1})^{-1}(z_{k-1} + \sigma V y_{k-1}), \\ \hat{z}_k &= \sigma^{-1} z_k.\end{aligned}$$

To see that the backward step

$$(M_k + A)^{-1} = (\widehat{M}_k - \widehat{E} + \widehat{A} + \widehat{E} + \widehat{V})^{-1} = (\widehat{M}_k + \widehat{A} + \widehat{V})^{-1},$$

can be evaluated efficiently requires some extra attention. The operator $\widehat{M}_k + \widehat{A} + \widehat{V}$ does not have the lower block-triangular structure as in the algorithm in Section 6.1. We can therefore not evaluate its inverse using the same back substitution approach as before and computing it at a general point seems intractable. However, $(\widehat{M}_k + \widehat{A} + \widehat{V})^{-1}$ is only evaluated at $(\hat{y}_k, \hat{z}_k)$ and the kernel has been specifically designed such that the backward step can be efficiently evaluated in this point. First use

$$\begin{aligned}&(y_{k+1}, z_{k+1}) = (\widehat{M}_k + \widehat{A} + \widehat{V})^{-1}(\hat{y}_k, \hat{z}_k) \\ \iff\ &(\hat{y}_k, \hat{z}_k) \in (\widehat{M}_k + \widehat{A} + \widehat{V})(y_{k+1}, z_{k+1}).\end{aligned}$$

Writing out the inclusion problem explicitly yields

$$\begin{cases} \hat{y}_k \in (\tau^{-1}\,\mathrm{Id} + B)y_{k+1} - V^*(\mathrm{Id} + \sigma D^{-1})^{-1}(z_k + \sigma V y_{k+1}) + V^* z_{k+1}, \\ \hat{z}_k \in -V y_{k+1} + (\sigma^{-1}\,\mathrm{Id} + D^{-1}) z_{k+1}. \end{cases}$$

Using that $z_k = \sigma \hat{z}_k$ in the first row and solving for z_{k+1} in the second row results in

$$\begin{cases} \hat{y}_k \in (\tau^{-1}\,\mathrm{Id} + B)y_{k+1} - V^*(\mathrm{Id} + \sigma D^{-1})^{-1}(\sigma \hat{z}_k + \sigma V y_{k+1}) + V^* z_{k+1}, \\ z_{k+1} = (\mathrm{Id} + \sigma D^{-1})^{-1}(\sigma \hat{z}_k + \sigma V y_{k+1}). \end{cases}$$

Inserting the second row into the first and solving for y_{k+1} gives

$$\begin{cases} y_{k+1} = (\mathrm{Id} + \tau B)^{-1}(\tau \hat{y}_k), \\ z_{k+1} = (\mathrm{Id} + \sigma D^{-1})^{-1}(\sigma \hat{z}_k + \sigma V y_{k+1}). \end{cases}$$

Finally, inserting the expressions for $\hat{y}_k$ and $\hat{z}_k$ gives us the following algorithm.

Algorithm 6.3 Primal-Dual Method with Resolvent Corrected Kernel

Consider problem (10). With $y_0, y_{-1} \in \mathcal{K}$ and $z_0, v_0 \in \mathcal{G}$, for all $k \in \mathbb{N}$ iteratively perform

$$\begin{aligned} v_{k+1} &= (\mathrm{Id} + \sigma D^{-1})^{-1}(z_k + \sigma V y_k) \\ y_{k+1} &= (\mathrm{Id} + \tau B)^{-1}(y_k - \tau V^*(z_k + v_{k+1} - v_k) - \tau(2Ey_k - Ey_{k-1}) - \tau F y_k) \\ z_{k+1} &= (\mathrm{Id} + \sigma D^{-1})^{-1}(z_k + \sigma V y_{k+1}) \end{aligned}$$

where $\tau, \sigma > 0$.

We see that, compared to our other primal-dual method Algorithm 6.1, we require one extra evaluation of the resolvent of D^{-1} each iteration. Apart from that, Algorithm 6.3, also only requires one evaluation of $(\mathrm{Id} + \tau B)^{-1}$, V and V^*, given that $V y_{k+1}$ is stored for the next iteration. Still, the resulting per-iteration computational cost is higher compared to Algorithm 6.1 and most other primal-dual methods. Exactly how much more expensive this method is will depend on the problem being solved and in some cases it is negligible. The main reason for presenting Algorithm 6.3, apart from its novelty, is to further demonstrate the flexibility of the nonlinear kernel framework.

COROLLARY 6.5

Let Assumption 6.1 hold and consider problem (10) and Algorithm 6.3. If the stepsizes satisfy

$$2\tau\sigma\|V\|^2 + \tau(2\delta + \tfrac{\beta}{2}) < 1,$$

then $y_k \rightharpoonup y^\star$ and $z_k \rightharpoonup z^\star$ where $y^\star$ is a solution to (10) and $(y^\star, z^\star)$ is a solution to (11).

Proof. Due to the structures of S and C we can conclude that C is β^{-1}-cocoercive w.r.t. S since

$$\begin{aligned}\|C(y,z)-C(y',z')\|_{S^{-1}}^2 &= \|Fy-Fy'\|^2\\ &\leq \beta\langle Fy-Fy', y-y'\rangle\\ &= \beta\langle C(y,z)-C(y',z'),(y,z)-(y',z')\rangle\end{aligned}$$

for all $(y,z)\in\mathcal{K}\times\mathcal{G}$. We have previously established that A is maximally monotone and, since we assume a solution exists, Assumption 2.1 holds.

For Assumption 2.2, we first note that $\gamma_k = \tau > 0$ and, hence, that the first assumption is satisfied. For the Lipschitz continuity of $\gamma_k M_k - S$ we recall the definition of the operator in (15). The operator E is, by assumption, δ-Lipschitz continuous, and $(\mathrm{Id}+\sigma D^{-1})^{-1}\circ T_{-}z_k$ is 1-Lipschitz since both the resolvent and translation are 1-Lipschitz continuous. The operator $-\tau(E+V^*\circ(\mathrm{Id}+\sigma D^{-1})^{-1}\circ T_{-}z_k\circ\sigma V)$ is therefore $(\tau\delta+\tau\sigma\|V\|^2)$-Lipschitz continuous for all $k\in\mathbb{N}$. Since

$$\begin{aligned}&\|(\gamma_k M_k - S)(y,z)-(\gamma_k M_k - S)(y',z')\|_{S^{-1}}^2\\ &\quad= \|\tau(E+V^*(\mathrm{Id}+\sigma D^{-1})^{-1}(z_k+\sigma V))y\\ &\qquad -\tau(E+V^*(\mathrm{Id}+\sigma D^{-1})^{-1}(z_k+\sigma V))y'\|^2\\ &\quad\leq(\tau\delta+\tau\sigma\|V\|^2)^2\|y-y'\|^2\\ &\quad\leq(\tau\delta+\tau\sigma\|V\|^2)^2\|(y,z)-(y',z')\|_S^2\end{aligned}$$

for all $(y,z)\in\mathcal{K}\times\mathcal{G}$, $\gamma_k M_k - S$ is $(\tau\delta+\tau\sigma\|V\|^2)$-Lipschitz continuous w.r.t. S for all $k\in\mathbb{N}$. The result now follows from Theorem 3.3. □

REMARK 6.6

As stated in Remark 4.3, the approach for adding momentum presented in Section 4 and Algorithm 4.1 does not yield a tractable algorithm when applied to Algorithm 6.3. The kernel of Algorithm 6.3 was designed in such a way that the backward step is only cheaply computed at the point given by the forward step and it is therefore not straightforward to apply the latter to the forward step with momentum. However, this is easily fixed. We regain computability of the backward step if we add $\theta(z_k - z_{k-1})$ according to

$$M_k = \begin{bmatrix}\tau^{-1}\,\mathrm{Id} - V^*\circ(\mathrm{Id}+\sigma D^{-1})^{-1}\circ T_{-z_k-\theta(z_k-z_{k-1})}\circ\sigma V & 0\\ 0 & \sigma^{-1}\,\mathrm{Id}\end{bmatrix} - \widehat{E}$$

and use this kernel in Algorithm 4.1 instead. Since this operator only differs from the one in (14) by a translation, it does not modify any Lipschitz constants, and the convergence can be proved using the same approach as in Corollary 4.1.

7. Conclusion

We have presented a forward-backward method with a nonlinear resolvent and a novel momentum correction. The design freedom of the nonlinear resolvent allows us to interpret numerous methods as special cases of this forward-backward method. Existing special cases include the forward-(half)-reflected-backward method, the forward-reflected-Douglas-Rachford method and the primal-dual methods of Vũ-Condat and Chambolle-Pock. New algorithms include momentum versions of the previously mentioned algorithms and two new four-operator primal-dual splitting methods. Our convergence conditions either regain or improve on the already known conditions for the existing methods, establishing parity of our more general analysis with the more specialized approaches. We believe that this parity of analysis and the great amount of freedom in the parameter choices of our algorithm can prove useful for the understanding of existing algorithms and the development of new ones.

References

[1] F. Alvarez and H. Attouch. "An Inertial Proximal Method for Maximal Monotone Operators via Discretization of a Nonlinear Oscillator with Damping". *Set-Valued Analysis* **9**:1 (2001), pp. 3–11. DOI: 10.1023/A:1011253113155.

[2] H. Attouch and A. Cabot. "Convergence Rates of Inertial Forward-Backward Algorithms". *SIAM Journal on Optimization* **28**:1 (2018), pp. 849–874. DOI: 10.1137/17M1114739.

[3] H. H. Bauschke, J. M. Borwein, and P. L. Combettes. "Bregman Monotone Optimization Algorithms". *SIAM Journal on Control and Optimization* **42**:2 (2003), pp. 596–636. DOI: 10.1137/S0363012902407120.

[4] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Second. CMS Books in Mathematics. Springer International Publishing, 2017. ISBN: 978-3-319-48310-8.

[5] H. H. Bauschke, M. N. Dao, and S. B. Lindstrom. "Regularizing with Bregman–Moreau Envelopes". *SIAM Journal on Optimization* **28**:4 (2018), pp. 3208–3228. DOI: 10.1137/17M1130745.

[6] H. H. Bauschke, P. L. Combettes, and D. Noll. "Joint Minimization with Alternating Bregman Proximity Operators". *Pacific journal of optimization* (2006). URL: https://hal.archives-ouvertes.fr/hal-01868791 (visited on 2021-08-30).

[7] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". *SIAM Journal on Imaging Sciences* **2**:1 (2009), pp. 183–202. DOI: 10.1137/080716542.

[8] R. I. Boţ and E. R. Csetnek. "An Inertial Forward-Backward-Forward Primal-Dual Splitting Algorithm for Solving Monotone Inclusion Problems". *Numerical Algorithms* **71**:3 (2016), pp. 519–540. DOI: 10.1007/s11075-015-0007-5.

[9] R. I. Boţ, E. R. Csetnek, and C. Hendrich. "Inertial Douglas–Rachford Splitting for Monotone Inclusion Problems". *Applied Mathematics and Computation* **256** (2015), pp. 472–487. DOI: 10.1016/j.amc.2015.01.017.

[10] R. I. Boţ, E. R. Csetnek, and E. Nagy. "Solving Systems of Monotone Inclusions via Primal-Dual Splitting Techniques". *Taiwanese Journal of Mathematics* **17**:6 (2013), pp. 1983–2009. DOI: 10.11650/tjm.17.2013.3087.

[11] L. M. Bregman. "The Relaxation Method of Finding the Common Point of Convex Sets and Its Application to the Solution of Problems in Convex Programming". *USSR Computational Mathematics and Mathematical Physics* **7**:3 (1967), pp. 200–217. DOI: 10.1016/0041-5553(67)90040-7.

[12] L. M. Briceño-Arias and D. Davis. "Forward-Backward-Half Forward Algorithm for Solving Monotone Inclusions". *SIAM Journal on Optimization* **28**:4 (2018), pp. 2839–2871. DOI: 10.1137/17M1120099.

[13] M. N. Bùi. *The Warped Resolvent of a Set-Valued Operator: Theory and Applications*. PhD thesis. North Carolia State University, 2021. URL: https://repository.lib.ncsu.edu/bitstream/handle/1840.20/39099/etd.pdf (visited on 2021-10-10).

[14] M. N. Bùi and P. L. Combettes. "Bregman Forward-Backward Operator Splitting". *Set-Valued and Variational Analysis* **29**:3 (2021), pp. 583–603. DOI: 10.1007/s11228-020-00563-z.

[15] M. N. Bùi and P. L. Combettes. "Warped Proximal Iterations for Monotone Inclusions". *Journal of Mathematical Analysis and Applications* **491**:1 (2020), p. 124315. DOI: 10.1016/j.jmaa.2020.124315.

[16] R. Burachik and J. Dutta. "Inexact Proximal Point Methods for Variational Inequality Problems". *SIAM Journal on Optimization* **20**:5 (2010), pp. 2653–2678. DOI: 10.1137/080733437.

[17] A. Chambolle and T. Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". *Journal of Mathematical Imaging and Vision* **40**:1 (2011), pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

[18] P. L. Combettes and J. Eckstein. "Asynchronous Block-Iterative Primal-Dual Decomposition Methods for Monotone Inclusions". *Mathematical Programming* **168**:1 (2018), pp. 645–672. DOI: 10.1007/s10107-016-1044-0.

[19] P. L. Combettes and L. E. Glaudin. "Solving Composite Fixed Point Problems with Block Updates". *Advances in Nonlinear Analysis* **10**:1 (2021), pp. 1154–1177. DOI: 10.1515/anona-2020-0173.

[20] P. L. Combettes and J.-C. Pesquet. "Primal-Dual Splitting Algorithm for Solving Inclusions with Mixtures of Composite, Lipschitzian, and Parallel-Sum Type Monotone Operators". *Set-Valued and Variational Analysis* **20**:2 (2012), pp. 307–330. DOI: 10.1007/s11228-011-0191-y.

[21] L. Condat. "A Primal–Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms". *Journal of Optimization Theory and Applications* **158**:2 (2013), pp. 460–479. DOI: 10.1007/s10957-012-0245-9.

[22] D. Davis and W. Yin. "A Three-Operator Splitting Scheme and its Optimization Applications". *Set-Valued and Variational Analysis* **25**:4 (2017), pp. 829–858. DOI: 10.1007/s11228-017-0421-z.

[23] J. Eckstein. "Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming". *Mathematics of Operations Research* **18**:1 (1993), pp. 202–226. URL: http://www.jstor.org/stable/3690161 (visited on 2021-08-30).

[24] J. Eckstein and B. F. Svaiter. "General Projective Splitting Methods for Sums of Maximal Monotone Operators". *SIAM Journal on Control and Optimization* **48**:2 (2009), pp. 787–811. DOI: 10.1137/070698816.

[25] E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson. "Global Convergence of the Heavy-Ball Method for Convex Optimization". In: *2015 European Control Conference (ECC)*. 2015, pp. 310–315. DOI: 10.1109/ECC.2015.7330562.

[26] P. Giselsson. "Nonlinear Forward-Backward Splitting with Projection Correction". *SIAM Journal on Optimization* (2021), pp. 2199–2226. DOI: 10.1137/20M1345062.

[27] P. Giselsson. *Nonlinear Forward-Backward Splitting with Projection Correction*. 2021. arXiv: 1908.07449v3. URL: http://arxiv.org/abs/1908.07449v3.

[28] A. A. Goldstein. "Convex Programming in Hilbert Space". *Bulletin of the American Mathematical Society* **70**:5 (1964), pp. 709–711. DOI: 10.1090/S0002-9904-1964-11178-2.

[29] B. He, Y. You, and X. Yuan. "On the Convergence of Primal-Dual Hybrid Gradient Algorithm". *SIAM Journal on Imaging Sciences* **7**:4 (2014), pp. 2526–2537. DOI: 10.1137/140963467.

[30] G. Kassay. "The Proximal Points Algorithm for Reflexive Banach Spaces". *Stud. Univ. Babes-Bolyai Math* **30** (1985), pp. 9–17.

[31] I. V. Konnov. "Combined Relaxation Methods for Generalized Monotone Variational Inequalities". In: *Generalized Convexity and Related Topics*. Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Heidelberg, 2006, pp. 3–31. ISBN: 978-3-540-37007-9. DOI: 10.1007/978-3-540-37007-9_1.

[32] P. Latafat and P. Patrinos. "Asymmetric Forward–Backward–Adjoint Splitting for Solving Monotone Inclusions Involving Three Operators". *Computational Optimization and Applications* **68**:1 (2017), pp. 57–93. DOI: 10.1007/s10589-017-9909-6.

[33] E. S. Levitin and B. T. Polyak. "Constrained Minimization Methods". *USSR Computational mathematics and mathematical physics* **6**:5 (1966), pp. 1–50.

[34] P. L. Lions and B. Mercier. "Splitting Algorithms for the Sum of Two Nonlinear Operators". *SIAM Journal on Numerical Analysis* **16**:6 (1979), pp. 964–979. DOI: 10.1137/0716071.

[35] D. A. Lorenz and T. Pock. "An Inertial Forward-Backward Algorithm for Monotone Inclusions". *Journal of Mathematical Imaging and Vision* **51**:2 (2015), pp. 311–325. DOI: 10.1007/s10851-014-0523-2.

[36] Y. Malitsky and M. K. Tam. "A Forward-Backward Splitting Method for Monotone Inclusions Without Cocoercivity". *SIAM Journal on Optimization* **30**:2 (2020), pp. 1451–1472. DOI: 10.1137/18M1207260.

[37] A. Moudafi and M. Oliny. "Convergence of a Splitting Inertial Proximal Method for Monotone Operators". *Journal of Computational and Applied Mathematics* **155**:2 (2003), pp. 447–454. DOI: 10.1016/S0377-0427(02)00906-8.

[38] D. O'Connor and L. Vandenberghe. "On the Equivalence of the Primal-Dual Hybrid Gradient Method and Douglas–Rachford Splitting". *Mathematical Programming* **179**:1 (2020), pp. 85–108. DOI: 10.1007/s10107-018-1321-1.

[39] B. T. Polyak. "Some Methods of Speeding up the Convergence of Iteration Methods". *USSR Computational Mathematics and Mathematical Physics* **4**:5 (1964), pp. 1–17. DOI: 10.1016/0041-5553(64)90137-5.

[40] H. Raguet, J. Fadili, and G. Peyré. "A Generalized Forward-Backward Splitting". *SIAM Journal on Imaging Sciences* **6**:3 (2013), pp. 1199–1226. DOI: 10.1137/120872802.

[41] E. K. Ryu and B. C. Vũ. "Finding the Forward-Douglas–Rachford-Forward Method". *Journal of Optimization Theory and Applications* **184**:3 (2020), pp. 858–876. DOI: 10.1007/s10957-019-01601-z.

[42] P. Tseng. "A Modified Forward-Backward Splitting Method for Maximal Monotone Mappings". *SIAM Journal on Control and Optimization* **38**:2 (2000), pp. 431–446. DOI: 10.1137/S0363012998338806.

[43] B. C. Vũ. "A Splitting Algorithm for Dual Monotone Inclusions Involving Cocoercive Operators". *Advances in Computational Mathematics* **38**:3 (2013), pp. 667–681. DOI: 10.1007/s10444-011-9254-8.

Paper IV

Frugal Splitting Operators: Representation, Minimal Lifting and Convergence

Martin Morin **Sebastian Banert** **Pontus Giselsson**

Abstract

We consider frugal splitting operators for finite sum monotone inclusion problems, i.e., splitting operators that use exactly one direct or resolvent evaluation of each operator of the sum. A novel representation of these operators in terms of what we call a generalized primal-dual resolvent is presented. This representation reveals a number of new results regarding lifting numbers, existence of solution maps, and parallelizability of the forward and backward evaluations. We show that the minimal lifting is $n-1-f$ where n is the number of monotone operators and f is the number of direct evaluations in the splitting. Furthermore, we show that this lifting number is only achievable as long as the first and last evaluations are resolvent evaluations. In the case of frugal resolvent splitting operators, these results are the same as the results of Ryu and Malitsky–Tam. The representation also enables a unified convergence analysis and we present a generally applicable theorem for the convergence and Fejér monotonicity of fixed point iterations of frugal splitting operators with cocoercive direct evaluations. We conclude by constructing a new convergent and parallelizable frugal splitting operator with minimal lifting.

Submitted and under review.

1. Introduction

It is well known that a zero of a maximally monotone operator can be found by performing a fixed point iteration of the resolvent of the operator [28]. However, the resolvent of an operator is not always easily computable, even in the case when the operator is a finite sum of maximally monotone operators for which each resolvent is easily computable. This has led to the development of splitting methods that use each term separately to form convergent fixed point iterations that find a zero of the sum of the operators. In this work, we will consider a general class of splitting operators for finite sums of maximally monotone operators which we call *frugal splitting operators*. Informally, the class contains all operators whose fixed points encode the zeros of the sum of the monotone operators and can be computed with exactly one evaluation of each operator, either directly or via a resolvent. Apart from the operator evaluations, only predetermined linear combinations of the input and operator evaluations are allowed. The class covers the classic Douglas–Rachford [21] and forward-backward [17, 20] operators along with many others, for instance [1, 5, 8, 10, 12, 22, 23, 25, 27, 29, 31, 35].

We provide an equivalence between frugal splitting operators and a class of operators we call *generalized primal-dual resolvents*. This class is inspired by the works of [6, 15, 18, 24] that made similar generalizations of the resolvent. These works provide powerful modeling tools that are able to capture many different algorithms but our resolvent generalization is the first to provably fully cover the class of frugal splitting operators. This novel representation is a key difference to the related works of [23, 29] which examined frugal resolvent splitting operators, i.e., frugal splitting operators that only use resolvents and no direct evaluations. These works focused on the *lifting number* of frugal splitting operators and, while we also provide minimal lifting results, our representation allows us to easily design new splitting operators and analyze the convergence of their fixed point iterations in a general setting. It also allows us to relax one assumption of [23, 29] regarding the existence of a *solution map* since the existence of such a solution map for any frugal splitting operator is evident directly from our new representation. The representation also directly reveals information regarding the resolvent/direct evaluations, for instance, the resolvent step-sizes and which of the individual evaluations that can be performed in parallel.

The general idea behind lifting is to trade computational complexity for storage complexity by creating an easier to solve problem in some higher dimensional space. In the context of frugal splitting operators, this means that, while the monotone inclusion problem lies in some real Hilbert space $\mathcal{H}$, the splitting operator maps to and from $\mathcal{H}^d$ for some non-zero natural number d which we call the lifting number. For example, the Douglas–Rachford and forward-backward splitting operators have lifting number one while the splitting operator of the primal-dual method

of Chambolle–Pock [8] has lifting number two[1].

In the three operator case, Ryu [29] showed that the smallest possible lifting number for a frugal resolvent splitting operator is two. Malitsky and Tam [23] later expanded this result to sums of n maximally monotone operators and found that the minimal lifting is $n-1$. In this paper, we show that this lifting number can be reduced to $n-1-f$ where f is the number of operators that are evaluated directly and not via resolvents. This is done via a simple rank constraint on a real matrix from the generalized primal-dual resolvent. We also show that the minimal lifting number is dependent on the order of the direct and resolvent evaluations. In particular, if the first or last operator is evaluated directly, the minimal lifting under these assumptions is $n-f$ instead of $n-1-f$. An example of this can be found in the three operator splitting of Davis and Yin [12], which places its only direct evaluation second and hence can achieve a lifting number of one. This is to be compared to the primal-dual method of Vũ and Condat [10, 35], which performs a direct evaluation first and hence requires a higher lifting number of two in the three operator case.

We provide sufficient conditions for the convergence of a fixed point iteration of a frugal splitting operator with cocoercive direct evaluations and show that the generated sequence is Fejér monotone w.r.t. to the fixed points of the splitting. There are a number of different general or unified approaches for analyzing algorithm classes [6, 13, 15, 18, 19, 24, 30, 32, 33]. Many of these approaches can be applied to a frugal resolvent splitting as well but our analysis has the advantage that it is performed directly on the generalized primal-dual resolvent representation. Both the conditions for convergence and the conditions for a generalized primal-dual resolvent to be a frugal resolvent splitting are then constraints on the same set of matrices, which simplifies the design of new frugal resolvent splittings. As an example, we construct a new frugal splitting operator with resolvent and direct evaluations that is convergent and parallelizable and briefly discuss its relation to existing splitting operators with minimal lifting. Note, although Malitsky–Tam [23] were the first to explicitly present a splitting operator with minimal lifting, their work—and now also this paper—retroactively proves that a number of already established splittings have minimal lifting, for instance [2, 7, 11].

Some of the proofs require the underlying real Hilbert space $\mathcal{H}$ of the monotone inclusion problem to have dimension greater than one and we will therefore make that a blanket assumption. We do not consider this a significant restriction, partly because the $\dim\mathcal{H} \leq 1$ cases are of the limited practical interest and partly because many of our results still hold in these cases. For instance, although the proof for the necessary conditions of our representation theorem no longer holds when $\dim\mathcal{H} \leq 1$, the proof for the sufficient conditions still holds. Hence, if we

[1] The Chambolle–Pock method considers monotone inclusion problems that allow for compositions with linear operators. We will implicitly assume that the linear operators are the identity operator when discussing Chambolle–Pock or other similar methods for monotone inclusions with compositions.

find a representation that yields a frugal splitting operator when $\dim \mathcal{H} \geq 2$, then it also yields a frugal splitting operator when $\dim \mathcal{H} \leq 1$. All frugal splitting operators presented in this paper are therefore also applicable to the $\dim \mathcal{H} \leq 1$ case. It should also be noted that we have been able to reestablish the necessary conditions by relaxing other assumptions placed on the inclusion problem. However, this comes at the cost of some additional technicalities which we wish to avoid in this paper.

1.1 Outline

In Section 2, we introduce some preliminary notation and results together with the main monotone inclusion problem. We define and discuss our definition of a frugal splitting operator in Section 3. Section 4 contains the definition of a generalized primal-dual resolvent which we will use to represent frugal splitting operators. The lemmas that prove our representation results can be found in Section 4.1 and they are summarized in our main convergence theorem in Section 5 which also contains some general remarks on the representation. For instance, the relationship between fixed points of the splitting and solutions to the monotone inclusion is proven and how the parallelizable resolvent/direct evaluations can be identified is demonstrated. We also show how a representation can be derived via an example. The minimal lifting results in terms of a rank bound on a structured matrix can be found in Section 6. This result covers the setting of Malitsky and Tam [23] and shows that the minimal lifting number depends on whether the first or last operator evaluation is a forward or backward evaluation. Convergence under cocoercive forward evaluations is proven in Section 7. In Section 7.1 we apply the convergence theorem and reestablish the convergence criterion of the three operator splitting of Davis and Yin [12] as well as provide conditions for the convergence of a fixed point iteration of forward-backward splitting with Nesterov-like momentum. The last part of the paper, Section 8, contains the construction of a new frugal splitting operator with minimal lifting and parallelizable forward/backward evaluations. The paper ends with a short conclusion in Section 9. In the accompanying supplement many more examples of representations of frugal splitting operators and application of our convergence theorem can be found.

2. Preliminaries

Let $\mathbb{N} = \{0, 1, \dots\}$ be the set of natural numbers and $\mathbb{N}_+ = \{1, 2, \dots\}$ be the set of non-zero natural numbers. The cardinality of a set A is denoted by $|A|$. A subset B of A is denoted by $B \subseteq A$ while a strict subset is denoted by $B \subset A$.

Let $\mathbb{R}$ be the set of real numbers. We refer to the range and kernel of a matrix $A \in \mathbb{R}^{n \times m}$ as $\operatorname{ran} A$ and $\ker A$ respectively. These are linear subspaces of $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively and their orthogonal complement with respect to the standard Euclidean inner product are denoted by $(\operatorname{ran} A)^\perp$ and $(\ker A)^\perp$ respectively, which also are linear subspaces.

With $\mathcal{H}$ being a real Hilbert space, the set of all subsets of $\mathcal{H}$ is denoted $2^{\mathcal{H}}$. The inner product and norm on $\mathcal{H}$ are denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ respectively. Let $H\colon \mathcal{H}\to\mathcal{H}$ be a bounded linear operator, we define $\langle\cdot,\cdot\rangle_H = \langle H(\cdot),\cdot\rangle$ and $\|\cdot\|_H^2 = \langle H(\cdot),\cdot\rangle$. If H is self-adjoint and strongly positive then $\langle\cdot,\cdot\rangle_H$ is an inner product and $\|\cdot\|_H$ is a norm.

Let U, V and W be sets and define the operator $A\colon U\times W\to V$. With the notation $A_w u = v$ we mean $A(u,w) = v$ and we define $A_w = A(\cdot,w)$. Since A_w is an operator from U to V, instead of writing $A\colon U\times W\to V$ we will say that $A_{(\cdot)}\colon U\to V$ is *parameterized* by W.

Let $A\colon \mathcal{H}\to 2^{\mathcal{H}}$ be a *set-valued operator* on $\mathcal{H}$, i.e., A is an operator that maps any point in $\mathcal{H}$ to a subset of $\mathcal{H}$. The *graph* of A is $\operatorname{gra} A = \{(x,u)\in\mathcal{H}\times\mathcal{H} \mid u\in Ax\}$. The *range* of A is $\operatorname{ran} A = \{u\in\mathcal{H} \mid \exists x\in\mathcal{H} \text{ s.t. } (x,u)\in\operatorname{gra} A\}$. The *domain* of A is $\operatorname{dom} A = \{x\in\mathcal{H} \mid Ax\neq\emptyset\}$. The operator A is said to have *full domain* if $\operatorname{dom} A = \mathcal{H}$. If Ax is a singleton for all $x\in\mathcal{H}$ then it is said to be *single-valued*. A single-valued operator has full domain and we will make no distinction between single-valued operators $A\colon \mathcal{H}\to 2^{\mathcal{H}}$ and mappings $A\colon \mathcal{H}\to\mathcal{H}$.

An operator $A\colon \mathcal{H}\to 2^{\mathcal{H}}$ is *monotone* if

$$\langle u-v, x-y\rangle \geq 0$$

for all $(x,u)\in\operatorname{gra} A$ and all $(y,v)\in\operatorname{gra} A$. A monotone operator is *maximal* if its graph is not contained in the graph of any other monotone operator. The *resolvent* of a maximally monotone operator A is $\mathrm{J}_A = (\mathrm{Id}+A)^{-1}$. An operator $A\colon \mathcal{H}\to 2^{\mathcal{H}}$ is *μ-strongly monotone* where $\mu>0$ if

$$\langle u-v, x-y\rangle \geq \mu\|x-y\|^2$$

for all $(x,u)\in\operatorname{gra} A$ and all $(y,v)\in\operatorname{gra} A$. An operator $A\colon \mathcal{H}\to 2^{\mathcal{H}}$ is *β-cocoercive* if it is single-valued and

$$\langle Ax-Ay, x-y\rangle \geq \beta\|Ax-Ay\|^2$$

for all $x,y\in\mathcal{H}$. The inverse of a β-cocoercive operator is β-strongly monotone.

LEMMA 2.1
Let $A\in\mathbb{R}^{n\times m}$ and $B\in\mathbb{R}^{n\times d}$. If $\operatorname{ran} A\subseteq\operatorname{ran} B$, there exists a unique $S\in\mathbb{R}^{d\times m}$ such that $A = BS$ and $\operatorname{ran} S\subseteq(\ker B)^\perp$. If $\operatorname{ran} A = \operatorname{ran} B$, such an S satisfies $\operatorname{ran} S = (\ker B)^\perp$.

Proof. Since $\operatorname{ran} A\subseteq\operatorname{ran} B$, the columns of A lie in the span of the columns of B. This means that there exists a matrix $S'\in\mathbb{R}^{d\times m}$ such that $A = BS'$. Let $\Pi_\perp$ be the orthogonal projection onto $(\ker B)^\perp$ in the standard Euclidean inner product and define $S = \Pi_\perp S'$. It is clear that $\operatorname{ran} S\subseteq(\ker B)^\perp$ and, since $B = B\Pi_\perp$, it also holds that $A = BS' = B\Pi_\perp S' = BS$.

To show uniqueness, let S and S' be such that $A = BS = BS'$ and $\operatorname{ran} S \subseteq (\ker B)^\perp$ and $\operatorname{ran} S' \subseteq (\ker B)^\perp$ and let $x \in \mathbb{R}^m$ be such that $Sx \neq S'x$. Since $Sx - S'x \neq 0$ and $Sx - S'x \in (\ker B)^\perp$ must $Sx - S'x \notin \ker B$ and we have

$$0 \neq B(Sx - S'x) = BSx - BS'x = Ax - Ax = 0$$

which is a contradiction and Sx and $S'x$ must then be equal for all $x \in \mathbb{R}^m$ which implies $S = S'$.

To show the last statement, assume $\operatorname{ran} A = \operatorname{ran} B$ and let S be the unique matrix that satisfies $A = BS$ and $\operatorname{ran} S \subseteq (\ker B)^\perp$. Assume $\operatorname{ran} S \subset (\ker B)^\perp$, then $\operatorname{rank} S < \dim(\ker B)^\perp = \operatorname{rank} B$ and $\operatorname{rank} A \leq \min(\operatorname{rank} B, \operatorname{rank} S) < \operatorname{rank} B$. However, this contradicts $\operatorname{ran} A = \operatorname{ran} B$ and hence $\operatorname{ran} S = (\ker B)^\perp$. □

LEMMA 2.2
Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{d\times n}$. If $\ker A \supseteq \ker B$, there exists a unique $S \in \mathbb{R}^{m\times d}$ such that $A = SB$ and $\ker S \supseteq (\operatorname{ran} B)^\perp$. If $\ker A = \ker B$ such an S satisfies $\ker S = (\operatorname{ran} B)^\perp$.

Proof. This is the dual to Lemma 2.1. If $\ker A \supseteq \ker B$ then $\operatorname{ran} A^T \subseteq \operatorname{ran} B^T$ and Lemma 2.1 then implies the existence of a unique $S^T \in \mathbb{R}^{d\times m}$ with $\operatorname{ran} S^T \subseteq (\ker B^T)^\perp$ such that $A^T = B^T S^T$ or equivalently $A = SB$. Since $\operatorname{ran} S^T = (\ker S)^\perp$ and $(\ker B^T)^\perp = \operatorname{ran} B$ we have $(\ker S)^\perp \subseteq \operatorname{ran} B$ or equivalently $\ker S \supseteq (\operatorname{ran} B)^\perp$. If $\ker A = \ker B$ then $\operatorname{ran} A^T = \operatorname{ran} B^T$ and Lemma 2.1 then yields $\ker S = (\operatorname{ran} B)^\perp$. □

2.1 Problem and Notation

For the remainder of this paper, let $\mathcal{H}$ be a real Hilbert space with $\dim \mathcal{H} \geq 2$. The main concern will be finding a zero of a finite sum of operators,

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^n A_i x \tag{1}$$

where $A_i \colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone for all $i \in \{1,\ldots,n\}$ and A_i is single-valued for $i \in F$ for some $F \subseteq \{1,\ldots,n\}$. However, instead of solving (1) directly, we will work with a family of primal-dual problems. Let $p \in \{1,\ldots,n\}$, a primal-dual problem associated with (1) is then

$$\text{find } (y_1,\ldots,y_n) \in \mathcal{H}^n \text{ such that } \begin{cases} 0 \in A_i^{-1} y_i - y_p \text{ for all } i \in \{1,\ldots,n\} \setminus \{p\}, \\ 0 \in A_p y_p + \sum_{j\in\{1,\ldots,n\}\setminus\{p\}} y_j. \end{cases} \tag{2}$$

We call p the primal index since the corresponding variable, y_p, solves the primal problem (1). The equivalence between (2) and (1) is straightforward to show and holds in the sense that if $y_p \in \mathcal{H}$ is a solution to (1) then there exists $y_i \in \mathcal{H}$ for all $i \in \{1,\ldots,n\} \setminus \{p\}$ such that $(y_1,\ldots,y_n)$ solves (2). Conversely, if $(y_1,\ldots,y_n) \in \mathcal{H}^n$ is a solution to (2) then y_p solves (1).

The aim of this paper is to examine a class of iterative methods for solving problems (1) and (2). We will give the exact definition of the considered class of solution methods in Section 3 and in the remainder of this section we introduce some notation in order to simplify its description.

DEFINITION 2.3—OPERATOR TUPLES
With $F \subseteq \{1,\dots,n\}$, $\mathcal{A}_n^F$ is the set of operator tuples where $A = (A_1,\dots,A_n) \in \mathcal{A}_n^F$ if and only if
(i) $A_i : \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone for all $i \in \{1,\dots,n\}$,
(ii) A_i is single-valued for all $i \in F$.
We further define $\mathcal{A}_n = \mathcal{A}_n^{\emptyset}$ and note that $\mathcal{A}_n^F \subseteq \mathcal{A}_n$ for all $F \subseteq \{1,\dots,n\}$.

This allows us to associate a tuple $A = (A_1,\dots,A_n) \in \mathcal{A}_n^F$ with each monotone inclusion problem of the form (1), or (2), and vice versa. To simplify the notation further we will also make heavy use of the following abuse of notation.

DEFINITION 2.4—MATRIX AS OPERATOR
Let $B \in \mathbb{R}^{n\times m}$ and $z = (z_1,\dots,z_m) \in \mathcal{H}^m$. We define Bz as[2]

$$Bz = \begin{bmatrix} B_{11}z_1 + \cdots + B_{1m}z_m \\ \vdots \\ B_{n1}z_1 + \cdots + B_{nm}z_m \end{bmatrix}.$$

A *primal-dual operator* $\Phi_{A,p} : \mathcal{H}^n \to 2^{\mathcal{H}^n}$ with $A \in \mathcal{A}_n^F$ and $p \in \{1,\dots,n\}$ can be identified that allows us to write a primal-dual problem (2) associated with A as

$$\text{find } y \in \mathcal{H}^n \text{ such that } 0 \in \Phi_{A,p}\, y = \Delta_{A,p}\, y + \Gamma_p\, y \tag{3}$$

where the operator $\Delta_{A,p} : \mathcal{H}^n \to 2^{\mathcal{H}^n}$ and the skew-symmetric matrix $\Gamma_p \in \mathbb{R}^{n\times n}$ are defined as

$$\Delta_{A,p}(y_1,\dots,y_n) = \widehat{A}_1 y_1 \times \cdots \times \widehat{A}_n y_n \quad \text{and} \quad \Gamma_p = R_p - R_p^T$$

where $\widehat{A}_p = A_p$, $\widehat{A}_i = A_i^{-1}$ for all $i \in \{1,\dots,n\} \setminus \{p\}$, and $R_p \in \mathbb{R}^{n\times n}$ is the matrix with ones on the pth row and zeros in all other positions. For illustration, in the case when $p = n$, the operator $\Delta_{A,n}$ and the matrix Γ_n have the following structures

$$\Delta_{A,n}(y_1,\dots,y_n) = \begin{bmatrix} A_1^{-1}y_1 \\ \vdots \\ A_{n-1}^{-1}y_{n-1} \\ A_n y_n \end{bmatrix} \quad \text{and} \quad \Gamma_n = \begin{bmatrix} 0 & \cdots & 0 & -1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & -1 \\ 1 & \cdots & 1 & 0 \end{bmatrix}.$$

All of these operators are maximally monotone: Γ_p as an operator on $\mathcal{H}^n \to \mathcal{H}^n$ is bounded linear and skew-adjoint, $\Delta_{A,p}$ is separable w.r.t. the components with each component operator being maximally monotone, and $\Phi_{A,p}$ is the sum of two maximally monotone operators with one having full domain. The primal-dual operator $\Phi_{A,p}$ will feature extensively in the rest of the paper. In fact, we will show that the considered class of operators always can be written as taking a resolvent-like step of the primal-dual operator.

[2] This operator could more accurately be represented by $B \otimes \mathrm{Id}$ where $\otimes$ is the tensor product. However, this notation will quickly become tedious.

3. Frugal Splitting Operators

A common way of solving (1) associated with some $A \in \mathcal{A}_n^F$ is with a fixed point iteration. These are methods that, given some initial iterate $z_0 \in \mathcal{H}^d$ and operator $T_A \colon \mathcal{H}^d \to \mathcal{H}^d$, iteratively perform

$$z_{k+1} = T_A z_k \tag{4}$$

for $k \in \mathbb{N}$. The operator T_A is such that the sequence $\{z_k\}_{k\in\mathbb{N}}$ converges to a fixed point from which a solution to (1) can be recovered. In this paper, this means that $T_{(\cdot)}$ is a *frugal splitting operator* and the main focus will be on examining the representation and properties of such an operator.

DEFINITION 3.1—FRUGAL SPLITTING OPERATOR
Let $d \in \mathbb{N}_+$ and $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$ be parameterized by $\mathcal{A}_n^F$. We say that $T_{(\cdot)}$ is a frugal splitting operator *over $\mathcal{A}_n^F$ if there, for all $k \in \{1,\dots,n\}$, exists $\tau_{k,(\cdot)} \colon \mathcal{H}^{d_k} \to \mathcal{H}^{d_{k+1}}$ parameterized by $\mathcal{A}_n^F$ such that, for all $A = (A_1,\dots,A_n) \in \mathcal{A}_n^F$,*

(i) $\operatorname{fix} T_A \neq \emptyset \iff \operatorname{zer} \sum_{i=1}^n A_i \neq \emptyset$,

(ii) $T_A = \tau_{n,A} \circ \tau_{n-1,A} \circ \cdots \circ \tau_{1,A}$.

Furthermore, for each $k \in \{1,\dots,n\}$ there exist $a_{ij}, b_i, c_j \in \mathbb{R}$ and $\gamma > 0$ for $i \in \{1,\dots,d_{k+1}\}$ and $j \in \{1,\dots,d_k\}$ such that

$$\tau_{k,B}(z_1,\dots,z_{d_k}) = \left(\sum_{j=1}^{d_k} a_{ij} z_j + b_i \begin{cases} \mathrm{J}_{\gamma B_k}(\sum_{j=1}^{d_k} c_j z_j) & \text{if } k \notin F \\ B_k(\sum_{j=1}^{d_k} c_j z_j) & \text{if } k \in F \end{cases} \right)_{i \in \{1,\dots,d_{k+1}\}}$$

for all $B = (B_1,\dots,B_n) \in \mathcal{A}_n^F$ and all $(z_1,\dots,z_{d_k}) \in \mathcal{H}^{d_k}$. Note, $\tau_{k,A}$ only uses A_k and $d_1 = d_{n+1} = d$.

We call a frugal splitting operator over $\mathcal{A}_n$ a resolvent splitting operator.

In line with [29], we call the class frugal since the computational requirement Definition 3.1(ii) states that each operator in A is evaluated exactly once, either directly or via a resolvent. We will also refer to a direct evaluation as a forward evaluation while a resolvent evaluation will be referred to as a backward step or backward evaluation. Apart from forward and backward evaluations, only predetermined vector additions and scalar multiplications of the inputs and results of the operator evaluations are allowed. This ensures that the evaluation cost of a frugal splitting operator is a known quantity that is mainly determined by the cost of the operator and resolvent evaluations. We also see that (ii) specifies evaluation order, i.e., it specifies that the operators A_i in the tuple A must be used in the order they appear in A when evaluating T_A. However, since the operators of the original monotone inclusion problem can be arbitrary rearranged, this entails no loss of generality.

This definition is functionally the same as the definitions of Ryu and Malitsky–Tam [23, 29] but with the addition of forward evaluations being allowed. However, both Ryu [29] and Malitsky–Tam [23] assume the existence of a computationally

tractable *solution map* $S_A : \mathcal{H}^d \to \mathcal{H}$ that maps fixed points of T_A to solutions of (1). One of our results, Proposition 5.3, shows that this is unnecessary since (i) and (ii) together imply the existence of such a solution map.

Definition 3.1 covers many classic operators like the operators used in the forward-backward method [17, 20] and Douglas–Rachford method [21], but also operators of more recent methods such that the three operator splitting of Davis and Yin [12] and primal-dual methods in the style of Chambolle and Pock [8] among many others. However, it does not allow for multiple evaluations of an operator and can therefore not cover the forward-backward-forward method of Tseng [34] although it does cover the forward-reflected-backward method of Malitsky and Tam [22]. Since it also does not allow for second order information or online search criteria it does not cover Newton's method, backtracking gradient descent or other similar methods.

Note that Definition 3.1 only considers the solution encoding and the computational requirements of a splitting and does not make any assumptions regarding the convergence of its fixed point iteration. For instance, consider a fixed point iteration of the forward-backward splitting operator $T_{(A_1,A_2)} = \mathrm{J}_{\gamma A_2} \circ (\mathrm{Id} - \gamma A_1)$ where $\gamma > 0$. This is a frugal splitting operator over $\mathcal{A}_2^{\{1\}}$ but it is well known that its fixed point iteration can fail to converge without further assumptions on A_1 and/or A_2. The standard assumption is cocoercivity of A_1 but even then the fixed point iteration can fail to converge if the step-size γ is too large. Further examples exist with both [23, 29] providing resolvent splitting operators whose fixed point iterations fail to converge in general. Since we consider the question of convergence as separate to the definition of a frugal splitting operator, we will treat convergence separately in Section 7 where we provide sufficient convergence conditions that are applicable to any fixed point iteration of a frugal splitting operator.

4. Generalized Primal-Dual Resolvents

Although Definition 3.1 fully defines all operators we aim to consider, we do not find it conducive for analysis. In this section, we will therefore develop an equivalent representation of the class of frugal splitting operators that will be used in the subsequent analysis.

The representation will be in the form of what we call a generalized primal-dual resolvent.

DEFINITION 4.1—GENERALIZED PRIMAL-DUAL RESOLVENT
Let $d \in \mathbb{N}_+$ *and* $T_{(\cdot)} : \mathcal{H}^d \to \mathcal{H}^d$ *be parameterized by* $\mathcal{A}_n^F$. *We say that* $T_{(\cdot)}$ *is a* generalized primal-dual resolvent *if there exist* $p \in \{1,\dots,n\}$, $M \in \mathbb{R}^{n\times n}$, $N \in \mathbb{R}^{n\times d}$, $U \in \mathbb{R}^{d\times d}$ *and* $V \in \mathbb{R}^{d\times n}$ *such that, for all* $A \in \mathcal{A}_n^F$ *and all* $z \in \mathcal{H}^d$, $(M + \Phi_{A,p})^{-1} \circ N$

is single-valued at z and

$$\begin{aligned} y &= (M + \Phi_{A,p})^{-1} N z, \\ T_A z &= z - Uz + Vy, \end{aligned} \tag{5}$$

where $y \in \mathcal{H}^n$ and $\Phi_{A,p}$ is defined in (3).

We call the tuple (p, M, N, U, V) a representation *of $T_{(\cdot)}$.*

When $M = N = \gamma^{-1} I$ and $U = V = \theta I$, where $\theta \in (0,2]$, $\gamma > 0$ and $I \in \mathbb{R}^{n \times n}$ is the identity matrix, the generalized primal-dual resolvent becomes an ordinary relaxed resolvent operator of the primal-dual operator $\Phi_{A,p}$,

$$(1-\theta)\,\mathrm{Id} + \theta(\gamma^{-1}\,\mathrm{Id} + \Phi_{A,p})^{-1} \circ \gamma^{-1}\,\mathrm{Id} = (1-\theta)\,\mathrm{Id} + \theta\,\mathrm{J}_{\gamma \Phi_{A,p}}\,.$$

While the ordinary relaxed resolvent operator is always single-valued, the operator $(M + \Phi_{A,p})^{-1} \circ N$ used in (5) is not necessarily single-valued or computationally tractable for an arbitrary choice of $M \in \mathbb{R}^{n \times n}$, $N \in \mathbb{R}^{n \times d}$ and $A \in \mathcal{A}_n^F$. However, for the purposes of this paper, it is not necessary to find general conditions for when $(M + \Phi_{A,p})^{-1} \circ N$ is single-valued or easy to compute. The objective is to parameterize frugal splitting operators which are computationally tractable if the evaluations of the forward and backward steps are tractable. Furthermore, as it turns out, the *kernel M* in a representation of a frugal splitting operator is highly structured, making the single-valuedness of $(M + \Phi_{A,p})^{-1} \circ N$ easy to establish.

DEFINITION 4.2—p-KERNEL OVER $\mathcal{A}_n^F$
We call a matrix $M \in \mathbb{R}^{n \times n}$ a p-kernel over $\mathcal{A}_n^F$ *if $p \notin F$ and $M + \Gamma_p$ is lower triangular with $M_{i,i} \geq 0$ for $i \in \{1, \ldots, n\}$ and $M_{i,i} = 0$ if and only if $i \in F$ where $M_{i,i}$ denotes the ith element on the diagonal of M. The matrix Γ_p is defined in (3).*

The well-posedness of a generalized primal-dual resolvent with this kernel structure then follows from the following result.

PROPOSITION 4.3
Let $M \in \mathbb{R}^{n \times n}$ be a p-kernel over $\mathcal{A}_n^F$, then

$$(M + \Phi_{A,p})^{-1} : \mathcal{H}^n \to 2^{\mathcal{H}^n}$$

is single-valued—and hence has full domain—for all $A \in \mathcal{A}_n^F$.

Proof. Let $A \in \mathcal{A}_n^F$, $z = (z_1, \ldots, z_n) \in \mathcal{H}^n$, $y = (y_1, \ldots, y_n) \in \mathcal{H}^n$, and $L = M + \Gamma_p$. By definition we have that $y \in (M + \Phi_{A,p})^{-1} x$ is equivalent to

$$x \in (M + \Phi_{A,p}) y = (L + \Delta_{A,p}) y.$$

Since L is lower triangular, see Definition 4.2, this can be written as

$$\begin{aligned}
x_1 &\in L_{1,1}y_1 + \widehat{A}_1 y_1 \\
x_2 &\in L_{2,1}y_1 + L_{2,2}y_2 + \widehat{A}_2 y_2 \\
&\vdots \\
x_n &\in L_{n,1}y_1 + L_{n,2}y_2 + \cdots + L_{n,n}y_n + \widehat{A}_n y_n
\end{aligned}$$

where $L_{i,j}$ is the i,jth element of L and $\widehat{A}_i = A_i^{-1}$ for all $i \in \{1,\ldots,n\} \setminus \{p\}$ and $\widehat{A}_p = A_p$. Note, $\widehat{A}_i$ is maximally monotone for all $i \in \{1,\ldots,n\}$ since $A \in \mathcal{A}_n^F$. We get

$$\begin{aligned}
y_1 &\in (L_{1,1}\,\mathrm{Id} + \widehat{A}_1)^{-1} x_1 \\
y_2 &\in (L_{2,2}\,\mathrm{Id} + \widehat{A}_2)^{-1}(x_2 - L_{2,1}y_1) \\
y_3 &\in (L_{3,3}\,\mathrm{Id} + \widehat{A}_3)^{-1}(x_3 - L_{3,1}y_1 - L_{3,2}y_2) \\
&\vdots \\
y_n &\in (L_{n,n}\,\mathrm{Id} + \widehat{A}_n)^{-1}(x_n - \textstyle\sum_{j=1}^{n-1} L_{n,j}y_j).
\end{aligned} \tag{6}$$

For all $i \in F$, $L_{i,i} = 0$ and $i \neq p$ by Definition 4.2 and hence

$$(L_{i,i}\,\mathrm{Id} + \widehat{A}_i)^{-1} = (A_i^{-1})^{-1} = A_i,$$

which is single-valued since $A \in \mathcal{A}_n^F$. For all $i \in \{1,\ldots,n\} \setminus F$, $L_{i,i} > 0$ and

$$(L_{i,i}\,\mathrm{Id} + \widehat{A}_i)^{-1} = \mathrm{J}_{L_{i,i}^{-1}\widehat{A}_i} \circ (L_{i,i}^{-1}\,\mathrm{Id})$$

which also is single-valued since $\widehat{A}$ is maximally monotone. Therefore, regardless of $x_1, y_1, \ldots, x_n, y_n \in \mathcal{H}$, the right hand sides of all lines in (6) are always singletons and the inclusions can be replaced by equalities. Furthermore, in (6) we see that x_1 uniquely determines y_1 which in turn implies that x_1 and x_2 uniquely determine y_2, and so forth. Hence, for all $x = (x_1,\ldots,x_n) \in \mathcal{H}^n$ there exists a unique $y = (y_1,\ldots,y_n) \in \mathcal{H}^n$ such that (6) holds. Since (6) is equivalent to $y \in (M + \Phi_{A,p})^{-1}x$ we can conclude that $(M + \Phi_{A,p})^{-1}$ is single-valued. □

4.1 Representation Lemmas

The remainder of this section will be devoted to the equivalence between frugal splitting operators and generalized primal-dual resolvents whose representations satisfy certain conditions. These results will be summarized in the next section in our main representation theorem, Theorem 5.1. We will start by showing that any frugal splitting operator is a generalized primal-dual resolvent with a particular kernel.

LEMMA 4.4
Let $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ be a frugal splitting operator over $\mathcal{A}_n^F$ where $F \subset \{1,\dots,n\}$, then $T_{(\cdot)}$ is a generalized primal-dual resolvent. For each $p \in \{1,\dots,n\} \setminus F$, there exists a representation (p,M,N,U,V) of $T_{(\cdot)}$ where M is a p-kernel over $\mathcal{A}_n^F$.

Proof. Let $p \in \{1,\dots,n\} \setminus F$, and let $A = (A_1,\dots,A_n) \in \mathcal{A}_n^F$ and $z \in \mathcal{H}^d$ be arbitrary and consider the evaluation of $T_A z$.

From Definition 3.1(ii) we know that $T_A = \tau_{n,A} \circ \cdots \circ \tau_{1,A}$ where $\tau_{i,A}\colon \mathcal{H}^{d_i} \to \mathcal{H}^{d_{i+1}}$ and $d_1 = d_{n+1} = d$. If we introduce variables for the intermediate results, i.e., $z_1 = z$ and $z_{i+1} = \tau_{i,A} z_i$ for all $i \in \{1,\dots,n\}$, we can write $T_A z = z_{n+1}$. Note, the z_i variables of this proof should not be confused with the variables of a fixed point iteration (4) of T_A. Furthermore, from the definition of $\tau_{i,A}$ in Definition 3.1(ii) we can conclude that for all $i \in \{1,\dots,n\}$ there exist matrices $B_i \in \mathbb{R}^{d_{i+1}\times d_i}$, $C_i \in \mathbb{R}^{d_{i+1}\times 1}$ and $D_i \in \mathbb{R}^{1\times d_i}$ such that

$$z_{i+1} = \tau_{i,A} z_i = \begin{cases} B_i z_i + C_i \operatorname{J}_{\gamma_i A_i} D_i z_i & \text{if } i \notin F, \\ B_i z_i + C_i A_i D_i z_i & \text{if } i \in F. \end{cases}$$

Now, for all $i \in \{1,\dots,n\} \setminus (F \cup \{p\})$, we apply the Moreau identity, $\operatorname{J}_{\gamma_i A_i} = \operatorname{Id} - \gamma_i \operatorname{J}_{\gamma_i^{-1} A_i^{-1}} \circ \gamma_i^{-1} \operatorname{Id}$, and rewrite $z_{i+1} = \tau_{i,A} z_i$ as

$$\begin{aligned} z_{i+1} &= B_i z_i + C_i \operatorname{J}_{\gamma_i A_i} D_i z_i \\ &= (B_i + C_i D_i) z_i - (\gamma_i C_i) \operatorname{J}_{\gamma_i^{-1} A_i^{-1}} (\gamma_i^{-1} D_i) z_i \\ &= \widehat{B}_i z_i + \widehat{C}_i (l_{i,i} \operatorname{Id} + \widehat{A}_i)^{-1} \widehat{D}_i z_i \end{aligned}$$

where $\widehat{B}_i = B_i + C_i D_i$, $\widehat{C}_i = -\gamma_i C_i$, $\widehat{D}_i = D_i$, $l_{i,i} = \gamma_i$ and $\widehat{A}_i = A_i^{-1}$. We can write $\tau_{i,A} z_i$ in a similar form for all $i \in F$ as well,

$$\begin{aligned} z_{i+1} &= B_i z_i + C_i A_i D_i z_i \\ &= \widehat{B}_i z_i + \widehat{C}_i (l_{i,i} \operatorname{Id} + \widehat{A}_i)^{-1} \widehat{D}_i z_i \end{aligned}$$

where $\widehat{B}_i = B_i$, $\widehat{C}_i = C_i$, $\widehat{D}_i = D_i$, $l_{i,i} = 0$ and $\widehat{A}_i = A_i^{-1}$. Finally, since $p \notin F$ we can write $\tau_{p,A} z_p$ as

$$\begin{aligned} z_{p+1} &= B_p z_p + C_p \operatorname{J}_{\gamma_p A_p} D_p z_p \\ &= B_p z_p + C_p (\gamma_p^{-1} \operatorname{Id} + A_p)^{-1} (\gamma_p^{-1} D_p) z_p \\ &= \widehat{B}_p z_p + \widehat{C}_p (l_{p,p} \operatorname{Id} + \widehat{A}_p)^{-1} \widehat{D}_p z_p \end{aligned}$$

where $\widehat{B}_p = B_p$, $\widehat{C}_p = C_p$, $\widehat{D}_p = \gamma_p^{-1} D_p$, $l_{p,p} = \gamma_p^{-1}$ and $\widehat{A}_p = A_p$. With these notations in place, we define $y_i \in \mathcal{H}$ as

$$y_i = (l_{i,i} \operatorname{Id} + \widehat{A}_i)^{-1} \widehat{D}_i z_i, \quad \forall i \in \{1,\dots,n\} \tag{7}$$

which gives

$$z_{i+1} = \widehat{B}_i z_i + \widehat{C}_i y_i, \quad \forall i \in \{1,\dots,n\}$$

and for clarity we note that $\widehat{B}_i \in \mathbb{R}^{d_{i+1}\times d_i}$, $\widehat{C}_i \in \mathbb{R}^{d_{i+1}\times 1}$ and $\widehat{D}_i \in \mathbb{R}^{1\times d_i}$ for all $i \in \{1,\dots,n\}$. Unrolling this iteration gives

$$z_i = \widehat{B}_{(i-1)\dots 1} z + \sum\nolimits_{j=1}^{i-1} \widehat{B}_{(i-1)\dots(j+1)} \widehat{C}_j y_j, \quad \forall i \in \{1,\dots,n+1\}$$

where $\widehat{B}_{a\dots b} = \widehat{B}_a \widehat{B}_{a-1} \dots \widehat{B}_b$ and $\widehat{B}_{a\dots b} = I$ if $a < b$ where $I \in \mathbb{R}^{d_b\times d_b}$ is the identity matrix. The expression for y_i in (7) can then be rewritten as

$$l_{i,i} y_i + \widehat{A}_i y_i \ni \widehat{D}_i \widehat{B}_{(i-1)\dots 1} z + \sum\nolimits_{j=1}^{i-1} \widehat{D}_i \widehat{B}_{(i-1)\dots(j+1)} \widehat{C}_j y_j, \quad \forall i \in \{1,\dots,n\}.$$

If we define $N_i = \widehat{D}_i \widehat{B}_{(i-1)\dots 1} \in \mathbb{R}^{1\times d}$, $l_{i,j} = -\widehat{D}_i \widehat{B}_{(i-1)\dots(j+1)} \widehat{C}_j \in \mathbb{R}$ for $j < i$ and rearrange this expression we get

$$\sum\nolimits_{j=1}^{i} l_{i,j} y_j + \widehat{A}_i y_i \ni N_i z, \quad \forall i \in \{1,\dots,n\}.$$

If we define $y = (y_1,\dots,y_n) \in \mathcal{H}^n$, the lower triangular matrix $L \in \mathbb{R}^{n\times n}$ with elements $L_{i,j} = l_{i,j}$ for all $j \le i$ and the matrix $N \in \mathbb{R}^{n\times d}$ whose ith row is given by N_i, this can be written as $Ly + \Delta_{A,p} y \ni Nz$ or equivalently

$$y = (L + \Delta_{A,p})^{-1} N z = (M + \Phi_{A,p})^{-1} N z$$

where $M = L - \Gamma_p$. Since the diagonal elements of M are $M_{i,i} = l_{i,i} \ge 0$ and $l_{i,i} = 0$ if and only if $i \in F$ we can from Definition 4.2 we conclude that M is a p-kernel over $\mathcal{A}_n^F$ and the single-valuedness of $(M + \Phi_{A,p})$ then follows from Proposition 4.3.

Finally, if we define the matrices $U = I - \widehat{B}_{n\dots 1} \in \mathbb{R}^{d\times d}$ and $V \in \mathbb{R}^{d\times n}$ where the jth column of V is given by $\widehat{B}_{n\dots(j+1)} \widehat{C}_j$ we can write the expression of z_{n+1} as

$$T_A z = z_{n+1} = z - Uz + Vy$$

which shows that (p, M, N, U, V) is a representation of $T_{(\cdot)}$. □

Next we prove that a representation of a frugal splitting operator needs to satisfy certain range and kernel constraints.

LEMMA 4.5
Let (p, M, N, U, V) be a representation of a frugal splitting operator over $\mathcal{A}_n^F$ with $p \in \{1,\dots,n\} \setminus F$, then

(i) $\ker \begin{bmatrix} N & -M \end{bmatrix} \supseteq \ker \begin{bmatrix} U & -V \end{bmatrix}$,

(ii) $\operatorname{ran} U \supseteq \operatorname{ran} V$.

Proof. Let $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ be a frugal splitting operator over $\mathcal{A}_n^F$ with representation (p,M,N,U,V). Furthermore, let $K\colon \mathcal{H} \to \mathcal{H}$ be a bounded linear skew-adjoint operator, and let $v \in \mathcal{H}$ be such that $Kv \neq 0$ and hence $v \neq 0$. Such an operator K and point v exist due to the $\dim \mathcal{H} \geq 2$ assumption.

We show the necessity of (i) by providing a counterexample. In particular we will show that if $U\hat{z} - V\hat{y} = 0$ while $N\hat{z} - M\hat{y} \neq 0$ for some $\hat{z} \in \mathbb{R}^d$ and $\hat{y} \in \mathbb{R}^n$, we can always construct some operator tuple $A \in \mathcal{A}_n^F$ such that the $\operatorname{fix} T_A \neq \emptyset \iff \operatorname{zer} \sum_{i=1}^n A_i \neq \emptyset$ equivalence of Definition 3.1(i) leads to a contradiction.

Assume $\hat{z} \in \mathbb{R}^d$ and $\hat{y} \in \mathbb{R}^n$ are such that $U\hat{z} - V\hat{y} = 0$ but $N\hat{z} - M\hat{y} \neq 0$. Then $z = (\hat{z}_1 v, \ldots, \hat{z}_d v) \in \mathcal{H}^d$ and $y = (y_1, \ldots, y_n) = (\hat{y}_1 v, \ldots, \hat{y}_n v) \in \mathcal{H}^n$ are such that $Uz - Vy = 0$ and $Nz - My \neq 0$. Define $(a_1, \ldots, a_n) = Nz - My \neq 0$ and note that all of $a_1, \ldots, a_n$ are parallel to v. Let $l \in \{1, \ldots, n\} \setminus \{p\}$ and define $A = (A_1, \ldots, A_n) \in \mathcal{A}_n^F$ where

$$A_i x = \begin{cases} K(x - y_p) + a_p - \sum_{j \in \{1,\ldots,n\} \setminus \{p\}} y_j & \text{if } i = p, \\ y_l - K(x - y_p - a_l) & \text{if } i = l, \\ y_i & \text{otherwise} \end{cases}$$

for all $x \in \mathcal{H}$ and all $i \in \{1, \ldots, n\}$. Note, both K and $-K$ are maximally monotone since K is skew-adjoint and hence is A_i maximally monotone for all $i \in \{1, \ldots, n\}$. The primal-dual operator $\Phi_{A,p}$ of this tuple evaluated at the y specified above satisfies

$$Nz - My = (a_1, \ldots, a_n) \in \Phi_{A,p} y \quad \text{and hence} \quad y = (M + \Phi_{A,p})^{-1} Nz$$

since $(M + \Phi_{A,p})^{-1}$ is singe-valued by Definition 4.1. We then have $T_A z = z - Uz + Vy$ and since $Uz - Vy = 0$ by assumption we have $T_A z = z$ and $\operatorname{fix} T_A \neq \emptyset$. Definition 3.1(i) then implies $\operatorname{zer} \sum_{i=1}^n A_i \neq \emptyset$ and since

$$\sum\nolimits_{i=1}^n A_i x = a_p + K a_l$$

for all $x \in \mathcal{H}$ must $K a_l = -a_p$. Since a_p and a_l are parallel to v there exist $\lambda_p, \lambda_l \in \mathbb{R}$ such that $a_p = \lambda_p v$ and $a_l = \lambda_l v$. Furthermore, since K is skew-adjoint—hence $\langle Kx, x \rangle = 0$ for all $x \in \mathcal{H}$—must

$$0 = \langle K a_l, a_l \rangle = \langle -a_p, a_l \rangle = \langle -\lambda_p v, \lambda_l v \rangle = -\lambda_p \lambda_l \|v\|^2$$

and since $v \neq 0$ is at least one of λ_p and λ_l zero. In fact, both must be zero since

$$-\lambda_p v = -a_p = K a_l = \lambda_l K v$$

and both $v \neq 0$ and $Kv \neq 0$. This means that $a_p = a_l = 0$ but, since $l \in \{1, \ldots, n\} \setminus \{p\}$ was arbitrary, we must have $a_i = 0$ for all $i \in \{1, \ldots, n\}$ and hence $(a_1, \ldots, a_n) = 0$ which is a contradiction. This concludes the necessity of (i).

For the necessity of (ii), let $y = (y_1, \dots, y_n) \in \mathcal{H}^n$ be arbitrary and define $B = (B_1, \dots, B_n) \in \mathcal{A}_n^F$ where

$$B_i x = \begin{cases} x - y_p + y_i & \text{if } i \neq p, \\ x - \sum_{j=1}^n y_j & \text{if } i = p \end{cases}$$

for all $x \in \mathcal{H}$ and all $i \in \{1, \dots, n\}$. These operators satisfy $\{y_p\} = \operatorname{zer} \sum_{i=1}^n B_i$ and $\{y\} = \operatorname{zer} \Phi_{B,p}$. Since $T_{(\cdot)}$ is a frugal splitting operator we must then have $\emptyset \neq \operatorname{fix} T_B$. Let $z \in \operatorname{fix} T_B$, (5) then implies the existence of some $y' \in \mathcal{H}^d$ such that $Uz - Vy' = 0$ and $Nz - My' \in \Phi_{B,p} y'$. But, (i) implies that $0 = Nz - My' \in \Phi_{B,p} y'$ and hence must $y' \in \operatorname{zer} \Phi_{B,p} = \{y\}$, i.e., $y' = y$ and $Uz = Vy$. Since the choice of y is arbitrary this implies that for all $y \in \mathcal{H}^n$ there exists $z \in \mathcal{H}^d$ such that $Uz = Vy$ which then implies the existence of $\hat{z} \in \mathbb{R}^d$ for each $\hat{y} \in \mathbb{R}^n$ such that $U\hat{z} = V\hat{y}$, i.e., $\operatorname{ran} U \supseteq \operatorname{ran} V$. This concludes the proof. □

Finally we prove that any representation that satisfies the constraints of the previous two lemmas represents a frugal splitting operator.

LEMMA 4.6
Let (p, M, N, U, V) be the representation of a generalized primal-dual resolvent where

(i) M is a p-kernel over $\mathcal{A}_n^F$,
(ii) $\ker \begin{bmatrix} N & -M \end{bmatrix} \supseteq \ker \begin{bmatrix} U & -V \end{bmatrix}$,
(iii) $\operatorname{ran} U \supseteq \operatorname{ran} V$,

then the generalized primal-dual resolvent is a frugal splitting operator over $\mathcal{A}_n^F$.

Proof. Let $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$ be a generalized primal-dual resolvent over $\mathcal{A}_n^F$ with representation (p, M, N, U, V) and let $A \in \mathcal{A}_n^F$. For each $z \in \mathcal{H}^d$ there exists $y \in \mathcal{H}^n$ such that

$$\begin{cases} y = (M + \Phi_{A,p})^{-1} N z, \\ T_A z = z - Uz + Vy \end{cases} \quad \text{or equivalently} \quad \begin{cases} Nz - My \in \Phi_{A,p} y, \\ Uz - Vy = z - T_A z. \end{cases} \tag{8}$$

Consider Definition 3.1(i). If $z \in \operatorname{fix} T_A$, then $Uz - Vy = 0$ and (ii) implies $0 = Nz - My \in \Phi_{A,p} y$ and hence $y \in \operatorname{zer} \Phi_{A,p} \neq \emptyset$. This proves the right implication of Definition 3.1(i).

For the left implication of Definition 3.1(i), let $y^\star \in \operatorname{zer} \Phi_{A,p}$. Then (iii) implies that there exists $z \in \mathcal{H}^d$ such that $Uz - Vy^\star = 0$ and (ii) then implies $Nz - My^\star = 0$. Since $y^\star \in \operatorname{zer} \Phi_{A,p}$ we then have $0 = Nz - My^\star \in \Phi_{A,p} y^\star$ and z and $y^\star$ then satisfy (8) which proves that $z \in \operatorname{fix} T_A \neq \emptyset$.

To show that Definition 3.1(ii) is implied by item (i) we first introduce some notation. Let $V_i \in \mathbb{R}^{d \times 1}$ denote the ith column of V, $N_i \in \mathbb{R}^{1 \times d}$ denote the ith row of N, and $l_{i,j}$ denote the i,j-element of the matrix $L = M + \Gamma_p$. Define the matrices

$$\widehat{B}_1 = \Big[\underbrace{I}_{\mathbb{R}^{d\times d}} \mid \underbrace{I}_{\mathbb{R}^{d\times d}} \mid \underbrace{\mathbf{0}}_{\mathbb{R}^{d\times d}} \mid \underbrace{\mathbf{0}}_{\mathbb{R}^{d\times n}} \Big]^T, \quad \widehat{B}_n = \Big[\underbrace{I - U}_{\mathbb{R}^{d\times d}} \mid \underbrace{\mathbf{0}}_{\mathbb{R}^{d\times d}} \mid \underbrace{I}_{\mathbb{R}^{d\times d}} \mid \underbrace{\mathbf{0}}_{\mathbb{R}^{d\times n}} \Big],$$

and $\widehat{B}_i = I \in \mathbb{R}^{(3d+n)\times(3d+n)}$ for all $i \in \{2,\dots,n-1\}$ where I and $\mathbf{0}$ denote identity and zero matrices of appropriate sizes, respectively. Further define the matrices

$$\widehat{C}_i = \big[\underbrace{\mathbf{0}}_{\mathbb{R}^{1\times d}} \mid \underbrace{\mathbf{0}}_{\mathbb{R}^{1\times d}} \mid V_i^T \mid \underbrace{0\dots0}_{\mathbb{R}^{1\times(i-1)}}\ 1\ \underbrace{0\dots0}_{\mathbb{R}^{1\times(n-i)}}\big]^T \quad \text{for all } i \in \{1,\dots,n-1\},$$

$$\widehat{D}_i = \big[\underbrace{\mathbf{0}}_{1\times d} \mid N_i \mid \underbrace{\mathbf{0}}_{\mathbb{R}^{1\times d}} \mid -l_{i,1}\cdots-l_{i,n}\big] \quad \text{for all } i \in \{2,\dots,n\},$$

$\widehat{C}_n = V_n$ and $\widehat{D}_1 = N_1$. Now, for $z_1 \in \mathcal{H}^d$ define

$$z_{i+1} = \widehat{B}_i z_i + \widehat{C}_i(l_{i,i}\,\mathrm{Id} + \widehat{A}_i)^{-1}\widehat{D}_i z_i$$

for $i \in \{1,\dots,n\}$ where $\widehat{A}_i = A_i^{-1}$ for all $i \in \{1,\dots,n\}\setminus\{p\}$ and $\widehat{A}_p = A_p$. By following the same procedure as in the proof of Lemma 4.4, starting at (7), it can be verified that $z_{n+1} = T_A z_1$ if (i) holds. Furthermore, by reversing the arguments in the proof of Lemma 4.4 that led to (7) we can conclude that there exist real matrices B_i, C_i, D_i such that

$$z_{i+1} = \begin{cases} B_i z_i + C_i\,\mathrm{J}_{\gamma_i A_i}\,D_i z_i & \text{if } i \notin F,\\ B_i z_i + C_i A_i D_i z_i & \text{if } i \in F,\end{cases}$$

for all $i \in \{1,\dots,n\}$. This proves that Definition 3.1(ii) holds. □

5. Representation of Frugal Splitting Operators

The following representation theorem summarizes the results of Section 4.1.

THEOREM 5.1
Let $F \subset \{1,\dots,n\}$ and $p \in \{1,\dots,n\}\setminus F$. An operator $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ parameterized by $\mathcal{A}_n^F$ is a frugal splitting operator over $\mathcal{A}_n^F$ if and only if it is a generalized primal-dual resolvent with representation (p,M,N,U,V) where

(i) M is a p-kernel over $\mathcal{A}_n^F$,
(ii) $\ker\begin{bmatrix}N & -M\end{bmatrix} \supseteq \ker\begin{bmatrix}U & -V\end{bmatrix}$,
(iii) $\operatorname{ran} U \supseteq \operatorname{ran} V$,
and $M \in \mathbb{R}^{n\times n}$, $N \in \mathbb{R}^{n\times d}$, $U \in \mathbb{R}^{d\times d}$, and $V \in \mathbb{R}^{d\times n}$.

Proof. Follows directly from Lemmas 4.4 to 4.6. □

Note that we require F to be a strict subset of $\{1,\dots,n\}$ which implies there always exists $p \in \{1,\dots,n\}\setminus F$. It is also worth remembering that it is assumed that $\dim\mathcal{H} \geq 2$. However, this assumption is only ever required for the proof of Lemma 4.5 and, since this lemma only concerns the "only if" part of Theorem 5.1, the sufficient conditions are not affected. Hence, any representation (p,M,N,U,V) that satisfies Theorem 5.1 yields a frugal splitting operator in the $\dim\mathcal{H} \leq 1$ setting

as well and all examples of frugal splitting operators in this paper are frugal splitting operators regardless of the dimension of $\mathcal{H}$. We have succeeded in finding replacements for the counterexample in the proof of Lemma 4.5 that require $\dim\mathcal{H} \geq 2$, but not without relaxing some other assumption. For instance, when $\dim\mathcal{H} = 1$ we have been able to construct sufficient counterexamples if we instead of single-valued operators allow for at most single-valued operators in the tuples of $\mathcal{A}_n^F$. However, this relaxation complicates questions regarding the domain of our generalized primal-dual resolvent and we believe the $\dim\mathcal{H} = 1$ case is of too limited practical interest to warrant these complications. Similarly, the trivial $\dim\mathcal{H} = 0$ case is also not worth handling.

The fact that the primal-dual operator appears directly in the representation and one only needs to consider simple range and structure constraints of a handful of matrices makes the representation easy to work with, see for instance Sections 6 to 8 where we analyze general splittings and construct a new splitting. The representation also makes the relationship between fixed points of a splitting and solutions to (1) clearly visible, something we will illustrate in the remainder of this section. There we will also discuss and illustrate different properties of the representation, such as how the step-sizes and parallelizability of the forward and backward evaluations are encoded. We will also demonstrate an approach for deriving a representation of a frugal splitting operator and provide an alternative factorization of a representation (p,M,N,U,V).

5.1 Alternative Factorization

Theorem 5.1 gives conditions on all four matrices of a representation (p,M,N,U,V). However, it turns out that these conditions make the kernel M uniquely defined given N, U and V. This leads to the following corollary which will be useful both in the examination of minimal lifting in Section 6 and in the convergence analysis in Section 7.

COROLLARY 5.2
Let $F \subset \{1,\dots,n\}$ and $p \in \{1,\dots,n\} \setminus F$. A generalized primal-dual resolvent with representation (p,SUP,SU,U,UP) where $U \in \mathbb{R}^{d\times d}$, $S \in \mathbb{R}^{n\times d}$ and $P \in \mathbb{R}^{d\times n}$ is a frugal splitting operator if

(i) SUP is a p-kernel over $\mathcal{A}_n^F$,
(ii) $\operatorname{ran} P \subseteq (\ker U)^\perp$ and $\ker S \supseteq (\operatorname{ran} U)^\perp$.

Furthermore, for any frugal splitting operator over $\mathcal{A}_n^F$ with representation (p,M,N,U,V) that satisfies Theorem 5.1, there exist matrices $S \in \mathbb{R}^{n\times d}$ and $P \in \mathbb{R}^{d\times n}$ such that $M = SUP$, $N = SU$ and $V = UP$ and the conditions (i) and (ii) are satisfied.

Proof. It is straightforward to verify that a representation (p,SUP,SU,U,UP) that satisfies the conditions of the theorem satisfies the conditions of Theorem 5.1 and hence is a representation of a frugal splitting operator over $\mathcal{A}_n^F$.

Assume $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ is s frugal splitting operator over $\mathcal{A}_n^F$ with representation (p,M,N,U,V) that satisfies Theorem 5.1. The theorem states that $\operatorname{ran} U \supseteq \operatorname{ran} V$

and Lemma 2.1 then proves the existence of a unique $P \in \mathbb{R}^{d\times n}$ with $\operatorname{ran} P \subseteq (\ker U)^\perp$ such that

$$V = UP.$$

Theorem 5.1 also states $\ker \begin{bmatrix} U & -V \end{bmatrix} \subseteq \ker \begin{bmatrix} N & -M \end{bmatrix}$ and Lemma 2.2 yields the existence of a unique $S \in \mathbb{R}^{n\times d}$ with $\ker S \supseteq (\operatorname{ran} \begin{bmatrix} U & -V \end{bmatrix})^\perp$ such that

$$N = SU \quad \text{and} \quad M = SV = SUP.$$

However, since $\operatorname{ran} U \supseteq \operatorname{ran} V$ we have $\operatorname{ran} \begin{bmatrix} U & -V \end{bmatrix} = \operatorname{ran} U$, this concludes the proof. □

We provide a similar factorization in Proposition 6.5 that is perhaps more useful when constructing new splittings since it does not require finding matrices S, U, and P such that SUP is a p-kernel. However, unlike Corollary 5.2, it is not guaranteed that all representations of frugal splitting operators have a factorization of the form in Proposition 6.5.

5.2 Solution Map and Fixed Points

The following two propositions reveal the relationship between fixed points of the splitting operator and solutions to the monotone inclusion problem for any frugal splitting operator. They also clarify why there is no need to assume the existence of a solution map in Definition 3.1 since there always exists a map that maps fixed points of a frugal splitting operator to solutions of the monotone inclusion problem (1). Furthermore, we see that this solution map is always evaluated within the evaluation of a frugal splitting operator itself.

PROPOSITION 5.3
Let $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ be a frugal splitting operator over $\mathcal{A}_n^F$ with representation (p,M,N,U,V) that satisfies Theorem 5.1. If $z \in \operatorname{fix} T_A$ for $A \in \mathcal{A}_n^F$, then there exists $(y_1,\dots,y_n) \in \mathcal{H}^n$ such that

$$(y_1,\dots,y_n) = (M+\Phi_{A,p})^{-1}Nz \in \operatorname{zer}\Phi_{A,p} \quad \textit{and} \quad y_p \in \operatorname{zer}\sum\nolimits_{i=1}^n A_i.$$

Proof. Let $A \in \mathcal{A}_n^F$ and $z \in \operatorname{fix} T_A$, Definition 4.1 then gives the existence of $y = (y_1,\dots,y_n) \in \mathcal{H}^n$ such that

$$\begin{cases} y = (M+\Phi_{A,p})^{-1}Nz, \\ z = z - Uz + Vy \end{cases} \quad \text{or equivalently} \quad \begin{cases} Nz - My \in \Phi_{A,p}y, \\ Uz - Vy = 0 \end{cases}$$

Since Theorem 5.1(ii) holds, $Uz - Vy = 0$ implies $Nz - My = 0$ and hence $y \in \operatorname{zer}\Phi_{A,p}$. That $y_p \in \operatorname{zer}\sum_{i=1}^n A_i$ follows from the equivalence between the primal-dual problem (3) and the primal problem (1). □

PROPOSITION 5.4
Let $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ be a frugal splitting operator over $\mathcal{A}_n^F$ and let (p,SUP,SU,U,UP) be a representation of $T_{(\cdot)}$ that satisfies Corollary 5.2. With $A \in \mathcal{A}_n^F$, the set of fixed points satisfies

$$\operatorname{fix} T_A \supseteq P \operatorname{zer} \Phi_{A,p}$$

where $P\operatorname{zer}\Phi_{A,p} = \{Py \mid y \in \operatorname{zer}\Phi_{A,p}\}$. Equality in the inclusion holds if and only if U has full rank.

Proof. Let $A \in \mathcal{A}_n^F$ and $y \in \operatorname{zer}\Phi_{A,p}$. We then have $0 = SUPy - SUPy \in \Phi_{A,p}y$, or equivalently $y = (SUP + \Phi_{A,p})^{-1}SUPy$, since $(SUP + \Phi_{A,p})^{-1}$ is single-valued due to SUP being a p-kernel over $\mathcal{A}_n^F$, see Proposition 4.3. By letting $z = Py$ in the definition of the generalized primal-dual resolvent we conclude that

$$\begin{aligned} y &= (SUP + \Phi_{A,p})^{-1}SUPy, \\ T_A Py &= Py - U(Py - Py) = Py, \end{aligned}$$

which implies that $Py \in \operatorname{fix} T_A$. Since $y \in \operatorname{zer}\Phi_{A,p}$ was arbitrary we have $\operatorname{fix} T_A \supseteq P\operatorname{zer}\Phi_{A,p}$.

Assume U does not have full rank. Then, there exists $z \in \mathcal{H}^d$ such that $Uz = 0$ and $z \neq 0$. Since $\operatorname{ran} P \subseteq (\ker U)^\perp$ by Corollary 5.2, there exists no $y' \in \mathcal{H}^n$ such that $z = Py'$ and hence $Py + z \notin P\operatorname{zer}\Phi_{A,p}$ since $y \in \operatorname{zer}\Phi_{A,p}$. However, we have

$$\begin{aligned} y = (SUP + \Phi_{A,p})^{-1}SUPy &= (SUP + \Phi_{A,p})^{-1}SU(Py + z), \\ T_A(Py + z) = Py + z - U(Py + z - Py) &= Py + z, \end{aligned}$$

and hence $Py + z \in \operatorname{fix} T_A$. The equality $\operatorname{fix} T_A = P\operatorname{zer}\Phi_{A,p}$ can therefore not hold if U does not have full rank.

Assume U has full rank. Let $z \in \operatorname{fix} T_A$, then there exists $y \in \mathcal{H}^n$ such that

$$\begin{aligned} SUz - SUPy &\in \Phi_{A,p}y, \\ U(z - Py) = z - T_A z &= 0. \end{aligned}$$

However, since U has full rank, this implies that $z = Py$ and that $0 = SU(z - Py) \in \Phi_{A,p}y$ and hence that $y \in \operatorname{zer}\Phi_{A,p}$ and $z \in P\operatorname{zer}\Phi_{A,p}$. Since $z \in \operatorname{fix} T_A$ was arbitrary we have $\operatorname{fix} T_A \subseteq P\operatorname{zer}\Phi_{A,p}$ and the opposite inclusion from before then gives equality. □

5.3 Evaluation Order and Parallelizability

The order of the evaluations of the different operators is specified in a frugal splitting operator, Definition 3.1. However, the definition only states that it should be possible to compute the frugal splitting operator using this order, it does not exclude the possibility of computing the frugal splitting operator with some other evaluation order or with some of the evaluations being performed in parallel. Especially

the ability of performing forward and/or backward evaluations in parallel is of particular interest since it can allow for distributed or multithreaded implementations. Fittingly, how one operator evaluation depends on previous evaluations is directly encoded in the kernel. It is therefore straightforward to both identify and construct parallelizable kernels, something we will use in Section 8 where we construct a new parallelizable frugal splitting operator with minimal lifting.

If we define $L = M + \Gamma_p$ for a representation (p, M, N, U, V) of a frugal splitting operator over $\mathcal{A}_n^F$ we can write the inverse used in the generalized primal-dual resolvent as

$$(M + \Phi_{A,p})^{-1} = (L + \Delta_{A,p})^{-1}.$$

Since M is a p-kernel, we know from Definition 4.2 that L is a lower triangular matrix and then $(y_1, \ldots, y_n) = (L + \Delta_{A,p})^{-1}(z_1, \ldots, z_n)$ can be computed with back-substitution,

$$y_i = (L_{i,i}\,\mathrm{Id} + A_i^{-1})^{-1}(z_i - \sum_{j=1}^{i-1} L_{i,j} y_j) \tag{9}$$

for $i \neq p$ while if $i = p$ then A_i^{-1} is simply replaced by A_i, see the proof of Proposition 4.3 for more details. It is clear that the strict lower triangular part of L, $L_{i,j}$ for $j < i$, determines the dependency on the results of previous forward or backward evaluations[3]. Hence, if for $i \in \mathbb{N}$ and some $j < i$ the element $L_{i,j} = 0$, then the ith evaluation does not directly depend on the result of the ith. If the ith evaluation does not depend on any other evaluation that depends on the jth evaluation, then the ith and jth evaluation can be performed in parallel.

For example, if we take the 4-kernel over $\mathcal{A}_4^{\{2\}}$ as

$$M = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & -1 & 1 \end{bmatrix} \quad \text{which yields} \quad L = M + \Gamma_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

we see that, in any frugal splitting operator with this kernel, the second evaluation directly depends on the first; the third depends directly on only the second but depends indirectly on the first; the fourth is independent of the third, depends directly on the second and depends indirectly on the first. The second must therefore be performed after the first while the last two must be performed after the second but can be performed in parallel with each other.

[3] The strictly lower triangular matrix $\tilde{L} \in \mathbb{R}^{n \times n}$ with elements $\tilde{L}_{i,j} = 1$ if $L_{i,j} \neq 0$ and $\tilde{L}_{i,j} = 0$ if $L_{i,j} = 0$ for $j < i$ is the transpose of the adjacency matrix of the directed dependency graph of the operator evaluations, i.e., the graph of n nodes where there is an edge from the ith to jth node if the result of forward or backward evaluation of A_i is used in the argument of the forward or backward evaluation of A_j.

5.4 Step-Sizes

It is possible to identify the resolvent step-sizes directly from the kernel. The inverse in (9) for $i \neq p$ and $L_{i,i} > 0$ can be written as

$$(L_{i,i}\,\mathrm{Id} + A_i^{-1})^{-1} = L_{i,i}^{-1}(\mathrm{Id} - \mathrm{J}_{L_{i,i}A_i}).$$

From this we see that the diagonal elements of L act as step-sizes in the resolvents of the frugal splitting operator and hence so do the diagonal elements of M since $L = M + \Gamma_p$ and Γ_p is skew-symmetric. Similarly we can conclude that $L_{p,p}^{-1}$ or equivalently $M_{p,p}^{-1}$ is the step-size used in the resolvent corresponding to the primal index. From Definition 4.2 we know that a forward evaluation is performed on the ith operator if and only if $L_{i,i} = M_{i,i} = 0$. Since they are zero, the diagonal elements of the kernel corresponding to forward evaluations are therefore not step-sizes. In fact, what would count as a step-size for a forward evaluation for a general frugal splitting operator is ill-defined. The same evaluation could be used in several different forward steps and we can therefore not point to a single scalar step-size.

5.5 Example of a Representation

The process of finding representations of frugal splitting operators is quite straightforward. Roughly speaking it is as follows: select a primal index; apply Moreau's identity to all backward steps that do not correspond to the primal index; explicitly define the results of the forward evaluations and backward steps; rearrange and identify the primal-dual operator and the generalized primal-dual resolvent representation. We demonstrate the process on the three operator splitting of Davis–Yin [12] which encodes the zeros of $A_1 + A_2 + A_3$ where $A = (A_1, A_2, A_3) \in \mathcal{A}_3^{\{2\}}$ as the fixed points of the operator

$$T_A = \mathrm{J}_{\gamma A_3} \circ (2\,\mathrm{J}_{\gamma A_1} - \mathrm{Id} - \gamma A_2 \circ \mathrm{J}_{\gamma A_1}) + \mathrm{Id} - \mathrm{J}_{\gamma A_1}\,.$$

In order for fixed point iterations of T_A to always be convergent it is further required that A_2 is cocoercive and that the step-size γ is sufficiently small. However, we leave the convergence analysis of frugal splitting operators to Section 7.

To find a representation, we first choose the primal index p in the representation (p, M, N, U, V). Since it is required that $p \notin \{2\}$ we have the choice of either $p = 1$ or $p = 3$. We choose $p = 3$. Applying Moreau's identity to the resolvents of A_i for all $i \neq p$ (in this case only $\mathrm{J}_{\gamma A_1}$) and defining the result of each forward and backward step yields

$$\begin{aligned}
y_1 &= (\mathrm{Id} + \gamma^{-1} A_1^{-1})^{-1}(\gamma^{-1} z),\\
y_2 &= A_2(z - \gamma y_1),\\
y_3 &= (\mathrm{Id} + \gamma A_3)^{-1}(2(z - \gamma y_1) - z - \gamma y_2),\\
T_A z &= y_3 + z - (z - \gamma y_1).
\end{aligned}$$

Rearranging the first three lines such that we only have z on the left and y_1, y_2 and y_3 and unscaled operators on the right yields

$$\begin{aligned} z &\in A_1^{-1} y_1 + \gamma y_1, \\ z &\in A_2^{-1} y_2 + \gamma y_1, \\ \gamma^{-1} z &\in A_3 y_3 + 2y_1 + y_2 + \gamma^{-1} y_3, \\ T_A z &= \gamma y_1 + y_3. \end{aligned}$$

If we define $y = (y_1, y_2, y_3)$, we see that the first three lines can be written as

$$\begin{bmatrix} 1 \\ 1 \\ \gamma^{-1} \end{bmatrix} z \in \Delta_{A,3} y + \begin{bmatrix} \gamma & 0 & 0 \\ \gamma & 0 & 0 \\ 2 & 1 & \gamma^{-1} \end{bmatrix} y = \Phi_{A,3} y + \begin{bmatrix} \gamma & 0 & 1 \\ \gamma & 0 & 1 \\ 1 & 0 & \gamma^{-1} \end{bmatrix} y$$

and T_A can then be written as

$$\begin{aligned} y &= \left(\begin{bmatrix} \gamma & 0 & 1 \\ \gamma & 0 & 1 \\ 1 & 0 & \gamma^{-1} \end{bmatrix} + \Phi_{A,3} \right)^{-1} \begin{bmatrix} 1 \\ 1 \\ \gamma^{-1} \end{bmatrix} z, \\ T_A z &= z - \begin{bmatrix} 1 \end{bmatrix} z + \begin{bmatrix} \gamma & 0 & 1 \end{bmatrix} y. \end{aligned}$$

From this we can easily identify the matrices M, N, U and V in the representation $(3, M, N, U, V)$ by comparing to Definition 4.1.

6. Minimal Lifting

DEFINITION 6.1—LIFTING
The lifting number*, or* lifting*, of a frugal splitting operator $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$ over $\mathcal{A}_n^F$ is the number $d \in \mathbb{N}_+$.*

The lifting number represents how much memory, proportional to the problem variable in (1), is needed to store data between iterations in a fixed point iteration of the splitting operator. For instance, if $\mathcal{H} = \mathbb{R}^N$ and we are trying to find a zero associated with $A \in \mathcal{A}_n^F$ with a frugal splitting operator with lifting number 3, i.e., $T_{(\cdot)} \colon \mathcal{H}^3 \to \mathcal{H}^3$, we need to be able to store a variable in $\mathbb{R}^{3N}$ between iterations[4]. For this reason, we are interested in finding lower bounds for the lifting number and frugal splitting operators that attain these bounds.

[4] It is possible for the internal operations needed to evaluate the splitting operator itself to require additional memory. However, since this is highly problem and implementation dependent, we do not consider this.

DEFINITION 6.2—MINIMAL LIFTING
A frugal splitting operator $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ *over* $\mathcal{A}_n^F$ *has* minimal lifting *if* $d \leq d'$ *for all frugal splitting operators* $T'_{(\cdot)}\colon \mathcal{H}^{d'} \to \mathcal{H}^{d'}$ *over* $\mathcal{A}_n^F$. *Furthermore, we say that* d *is the* minimal lifting *over* $\mathcal{A}_n^F$.

The equivalent representation of a frugal splitting operator in Theorem 5.1 is useful when it comes to examining this minimal lifting. In fact, a lower bound on the lifting number is directly given by Corollary 5.2.

COROLLARY 6.3
Let (p, M, N, U, V) *be a representation of a frugal splitting operator* $T_{(\cdot)}\colon \mathcal{H}^d \to \mathcal{H}^d$ *over* $\mathcal{A}_n^F$ *that satisfies Theorem 5.1, then*
(i) $d \geq \operatorname{rank} U \geq \operatorname{rank} N \geq \operatorname{rank} M$,
(ii) $d \geq \operatorname{rank} U \geq \operatorname{rank} V \geq \operatorname{rank} M$.

Proof. Corollary 5.2 states that there exist matrices $S \in \mathbb{R}^{n\times d}$ and $P \in \mathbb{R}^{d\times n}$ such that $M = SUP$, $N = SU$ and $V = UP$. The results then follow directly from the fact that $U \in \mathbb{R}^{d\times d}$ and that a product of matrices cannot have greater rank than any of its factors. □

Since the kernel rank bounds the lifting number, it is of great interest how small the rank of any valid kernel can be made, which leads us to the following definition.

DEFINITION 6.4—MINIMAL KERNEL RANK
A matrix $M \in \mathbb{R}^{n\times n}$ *is a* minimal p-kernel over $\mathcal{A}_n^F$ *if it is a* p*-kernel over* $\mathcal{A}_n^F$ *and* $\operatorname{rank} M \leq \operatorname{rank} M'$ *for all* p*-kernels* M' *over* $\mathcal{A}_n^F$. *The* minimal p-kernel rank over $\mathcal{A}_n^F$ *is* $\operatorname{rank} M$ *where* M *is a minimal* p*-kernel* M *over* $\mathcal{A}_n^F$.

The existence of a minimal p-kernel over $\mathcal{A}_n^F$ follows from the fact that $\operatorname{rank} M \in \mathbb{N}$ for all real matrices and a p-kernel over $\mathcal{A}_n^F$ exists for all $n \geq 2$, $F \in \{1,\dots,n\}$, and $p \in \{1,\dots,n\} \setminus F$. We can use the minimal p-kernel rank over $\mathcal{A}_n^F$ and Corollary 6.3 to provide a lower bound on the minimal lifting number, but, this is not enough to establish that that lower bound actually can be attained. For that we need to show that it is always possible to construct a frugal splitting operator over $\mathcal{A}_n^F$ from a p-kernel over $\mathcal{A}_n^F$.

PROPOSITION 6.5
Let $F \subset \{1,\dots,n\}$ *and* $p \in \{1,\dots,n\} \setminus F$. *If* M *is a* p*-kernel over* $\mathcal{A}_n^F$ *with* $\operatorname{rank} M = d$ *and matrices* $K \in \mathbb{R}^{n\times d}$ *and* $H \in \mathbb{R}^{d\times n}$ *satisfy* $\operatorname{ran} K = (\ker M)^\perp$ *and* $\ker H = (\operatorname{ran} M)^\perp$, *then* (p, M, MK, HMK, HM) *is a representation of a frugal splitting operator over* $\mathcal{A}_n^F$ *with lifting number* d. *Furthermore, such matrices* K *and* H *exist for all* p*-kernels over* $\mathcal{A}_n^F$.

Proof. We first show that (p, M, MK, HMK, HM) satisfies the conditions in Theorem 5.1.

Theorem 5.1(i) is directly given by the assumption on M. Now, let $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^n$ be such that $HMKx = HMy$. Since $\ker H \cap \operatorname{ran} M = \{0\}$, this implies $MKx = My$, which proves Theorem 5.1(ii). For Theorem 5.1(iii), let $y \in \mathbb{R}^n$ be arbitrary and let $y^{\parallel} \in \ker M$ and $y^{\perp} \in (\ker M)^{\perp}$ be such that $y = y^{\parallel} + y^{\perp}$. Since $\operatorname{ran} K = (\ker M)^{\perp}$, there exists $x \in \mathbb{R}^d$ such that $y^{\perp} = Kx$ and hence $MKx = My^{\perp} = M(y^{\parallel} + y^{\perp}) = My$, multiplying both sides with H from the left finally gives Theorem 5.1(iii).

Finally, let M be an arbitrary p-kernel over $\mathcal{A}_n^F$. Let $\operatorname{rank} M = d$, then the kernel M has d linearly independent columns and d linearly independent rows. Define $K \in \mathbb{R}^{n \times d}$ such that the columns of K are d linearly independent rows of M and define $H \in \mathbb{R}^{d \times n}$ such that the rows of H are d linearly independent columns of M, then $\operatorname{ran} K = (\ker M)^{\perp}$ and $\ker H = (\operatorname{ran} M)^{\perp}$. This concludes the proof. □

With these results in hand, our main minimal lifting result can be stated and proved.

THEOREM 6.6
The minimal lifting number of a frugal splitting operator over $\mathcal{A}_n^F$ is equal to the minimal p-kernel rank over $\mathcal{A}_n^F$ for arbitrary $p \in \{1,\dots,n\} \setminus F$.

Proof. Let $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$ be a frugal splitting operator over $\mathcal{A}_n^F$ with minimal lifting and let $p \in \{1,\dots,n\} \setminus F$. Theorem 5.1 states that $T_{(\cdot)}$ has a representation (p,M,N,U,V) and Corollary 6.3 then directly gives that the lifting d is greater or equal to the minimal p-kernel rank over $\mathcal{A}_n^F$. Proposition 6.5 proves that d is equal to the minimal p-kernel rank over $\mathcal{A}_n^F$ since otherwise we could construct a frugal splitting operator with smaller lifting. Since the choice of $p \in \{1,\dots,n\} \setminus F$ was arbitrary and d is independent of p, the minimal p-kernel rank over $\mathcal{A}_n^F$ is the same for all $p \in \{1,\dots,n\} \setminus F$. □

This theorem reduces the problem of finding the minimal lifting over $\mathcal{A}_n^F$ to finding the minimal p-kernel rank over $\mathcal{A}_n^F$. Furthermore, since Proposition 6.5 proves the existence results necessary for Theorem 6.6 by construction, it provides a clear way of constructing frugal splitting operators with lifting equal to the kernel rank. Hence, even the construction of frugal splitting operators with minimal lifting can be seen as a problem of finding p-kernels over $\mathcal{A}_n^F$ with minimal rank. We will give an example of this in Section 8 and end this section by making Theorem 6.6 concrete by finding the minimal kernel rank over $\mathcal{A}_n^F$ and the corresponding minimal lifting results.

COROLLARY 6.7
Let $n \geq 2$ and $F \subset \{1,\dots,n\}$. The minimal lifting over $\mathcal{A}_n^F$ is $n - |F|$ if $1 \in F$ or $n \in F$, otherwise it is $n - 1 - |F|$, where $|F|$ is the cardinality of F.

Proof. Let M be a p-kernel over $\mathcal{A}_n^F$. It must have the structure

$$M = \left[\begin{array}{c|c|c} L_1 & \mathbf{1} & \mathbf{0} \\ \hline * & l_p & -\mathbf{1} \\ \hline * & * & L_2 \end{array}\right] \in \mathbb{R}^{n \times n}$$

where $L_1 \in \mathbb{R}^{(p-1)\times(p-1)}$ and $L_2 \in \mathbb{R}^{(n-p)\times(n-p)}$ are lower triangular matrices and $l_p > 0$ is a real number. The symbols $*$, $\mathbf{1}$ and $\mathbf{0}$ respectively denote an arbitrary real matrix, a matrix of all ones, and a matrix of all zeros, all of appropriate sizes for their position. Since reordering columns and rows does not change the rank of a matrix there exist matrices

$$M_c = \left[\begin{array}{c|c|c} L_1 & \mathbf{0} & \mathbf{1} \\ \hline * & L_2 & * \\ \hline * & -\mathbf{1} & l_p \end{array}\right] \in \mathbb{R}^{n\times n} \quad \text{and} \quad M_r = \left[\begin{array}{c|c|c} l_p & * & -\mathbf{1} \\ \hline \mathbf{1} & L_1 & \mathbf{0} \\ \hline * & * & L_2 \end{array}\right] \in \mathbb{R}^{n\times n}.$$

such that $\operatorname{rank} M = \operatorname{rank} M_c = \operatorname{rank} M_r$.

Consider the first $n-1$ columns of M_c. Since L_1 and L_2 are lower triangular, we see that the number of linearly independent columns is greater or equal to the number of non-zero diagonal elements of L_1 and L_2. Furthermore, since the diagonal elements of L_1 and L_2 are diagonal elements of M, $M_{i,i} = 0$ if and only if $i \in F$, and $p \notin F$, there are at least $n-1-|F|$ linearly independent columns among the first $n-1$ columns of M_c, hence, $\operatorname{rank} M_c \geq n-1-|F|$. If $1 \notin F$ then $M_{1,1} = (L_1)_{1,1} \neq 0$ and this bound can be attained by selecting the first column of M_c parallel to the last and putting zeros in all other positions that are allowed to be zero. If $1 \in F$ then $M_{1,1} = (L_1)_{1,1} = 0$ and the last column of M_c is not in the span of the first $n-1$ columns and hence $\operatorname{rank} M_c \geq n-|F|$. This bound is attained by putting zeros in all positions of M_c that are allowed to be zero. By considering rows of M_r instead of columns of M_c we can analogously conclude that $\operatorname{rank} M_r \geq n-1-|F|$ and if $n \in F$ (and hence $M_{n,n} = (L_2)_{n-p,n-p} = 0$), then $\operatorname{rank} M_r \geq n-|F|$. These bounds are also similarly attained.

Since $\operatorname{rank} M = \operatorname{rank} M_c = \operatorname{rank} M_r$ we clearly have that $\operatorname{rank} M \geq n-1-|F|$ and if $1 \in F$ or $n \in F$ then $\operatorname{rank} M \geq n-|F|$. Choices of M that attain these bounds can be constructed by reordering the rows and columns of the choices of M_c and M_r that attain their respective bounds. The minimal kernel rank over $\mathcal{A}_n^F$ is then clearly $n-|F|$ if $1 \in F$ or $n \in F$, otherwise it is $n-1-|F|$. The lifting result then follows from Theorem 6.6. □

In the case of frugal resolvent splitting operators, i.e., $F = \emptyset$, this is the same lower bound as the one found by Malitsky and Tam [23]. When considering frugal splitting operators with forward evaluations, i.e., $F \neq \emptyset$, this corollary uncovers an interesting phenomenon where the minimal lifting depends on the evaluation order. A smaller lifting number is possible as long as neither the first nor last evaluation is a forward evaluation. This makes it clear why the three operator splitting of Davis and Yin, see Section 5.5, achieves a lifting number of one while the corresponding methods of Vũ and Condat [10, 35] require a lifting of two. Davis and Yin's method performs the forward evaluation between two resolvent evaluations, while the Vũ–Condat method performs the forward evaluation first.

7. Convergence

We will now consider the convergence of fixed point iterations $z_{k+1} = T_A z_k$ for frugal splitting operators $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$ over $\mathcal{A}_n^F$ and $A \in \mathcal{A}_n^F$. From Corollary 5.2 we know that these iterations can, for each $p \in \{1,\dots,n\} \setminus F$, be written as

$$\begin{cases} y_k = (SUP + \Phi_{A,p})^{-1} S U z_k, \\ z_{k+1} = z_k - U(z_k - P y_k) \end{cases} \quad \text{or equivalently} \quad \begin{cases} SU(z_k - P y_k) \in \Phi_{A,p}\, y_k, \\ U(z_k - P y_k) = z_k - z_{k+1} \end{cases} \tag{10}$$

for some matrices $S \in \mathbb{R}^{n\times d}$, $P \in \mathbb{R}^{d\times n}$ and $U \in \mathbb{R}^{d\times d}$. In certain cases, this iteration can be analyzed with existing theory. For instance, if S is symmetric positive definite, $P = I$ and $U = \theta I$ for some $\theta \in (0,1]$, then this is an averaged fixed point iteration of a resolvent in the metric given by the symmetric kernel $SUP = \theta S$. However, for general frugal splitting operators, SUP is only rarely symmetric, especially when considering frugal splitting operators with minimal lifting. In fact, a minimal p-kernel can be symmetric[5] only when $n = 1$ or $n = 2$, see the proof of Corollary 6.7. For this reason we will derive sufficient convergence conditions without any symmetry requirements on the kernel under the following assumption.

ASSUMPTION 7.1
Let $A = (A_1,\dots,A_n) \in \mathcal{A}_n^F$ be such that the following hold
(i) $\operatorname{zer} \sum_{i=1}^n A_i \neq \emptyset$,
(ii) *A_i is β_i-cocoercive for all $i \in F$.*
For this A, define two diagonal matrices $B \in \mathbb{R}^{n\times n}$ and $B^\dagger \in \mathbb{R}^{n\times n}$ where $B_{i,i} = \beta_i$ and $B^\dagger_{i,i} = \beta_i^{-1}$ for all $i \in F$. All other elements of the matrices are zero.

Cocoercivity of the forward evaluated operators is a standard setting for proving convergence of forward-backward like methods. However, there exist frugal splitting operators whose fixed point iterations converge under only Lipschitz assumptions on the operator used for forward evaluations. An example of this, that does not have minimal lifting, is the forward-reflected-backward splitting [22].

Since inverses of cocoercive operators are strongly monotone, primal-dual operators for tuples that satisfy Assumption 7.1 have the following strong-monotonicity-like property.

LEMMA 7.2
Let $F \subset \{1,\dots,n\}$, $p \in \{1,\dots,n\} \setminus F$ and let $A \in \mathcal{A}_n^F$ satisfy Assumption 7.1, then

$$\langle u - y, x - y\rangle \geq \|x - y\|_B^2$$

for all $x, y \in \mathcal{H}^n$ and for all $u \in \Phi_{A,p} x$, $v \in \Phi_{A,p} y$.

[5] These symmetric minimal kernels correspond to the kernels used in the proximal point method and Douglas–Rachford splitting, respectively.

Proof. With some slight abuse of notation for set-valued operators, we first note that for all $x, y \in \mathcal{H}^n$ we have

$$\begin{aligned}
&\langle \Phi_{A,p} x - \Phi_{A,p} y, x - y \rangle \\
&\quad = \langle \Delta_{A,p} x - \Delta_{A,p} y, x - y \rangle + \langle \Gamma_p x - \Gamma_p y, x - y \rangle \\
&\quad = \langle \Delta_{A,p} x - \Delta_{A,p} y, x - y \rangle \\
&\quad = \langle A_p x_p - A_p y_p, x_p - y_p \rangle + \sum\nolimits_{i=\{1,\dots,n\}\setminus\{p\}} \langle A_i^{-1} x_i - A_i^{-1} y_i, x_i - y_i \rangle
\end{aligned}$$

where the fact that Γ_p is skew-adjoint was used. For all $i \in \{1,\dots,n\} \setminus F$, the operator A_i is monotone and so is A_i^{-1}, while for all $i \in F$, A_i is β_i-cocoercive and hence A_i^{-1} is β_i-strongly monotone. Noting that $p \notin F$ by assumption and using these properties yields

$$\langle \Phi_{A,p} x - \Phi_{A,p} y, x - y \rangle \geq \sum\nolimits_{f \in F} \beta_f \|x_f - y_f\|^2 = \|x - y\|_B^2. \qquad \square$$

This property is used to derive our main convergence theorem.

THEOREM 7.3
Let $F \subset \{1,\dots,n\}$ and let $A \in \mathcal{A}_n^F$ satisfy Assumption 7.1. Let $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$ be a frugal splitting operator over $\mathcal{A}_n^F$ and let (p, SUP, SU, U, UP) be a representation of $T_{(\cdot)}$ according to Corollary 5.2. Let the sequences $\{z_k\}_{k\in\mathbb{N}}$ and $\{y_k\}_{k\in\mathbb{N}}$ be generated by (10) for this representation and some $z_0 \in \mathcal{H}^d$.

Consider the following conditions on a symmetric matrix $Q \in \mathbb{R}^{d\times d}$,

(a) $(I - I_F)(P^T Q - S)U = 0$,

(b) $Q \succ 0$ *and* $W \succ 0$,

where $W = QU + (QU)^T - U^T Q U - \frac{1}{2}(P^T QU - SU)^T B^\dagger (P^T QU - SU)$, $I \in \mathbb{R}^{n\times n}$ is the identity matrix, and $I_F \in \mathbb{R}^{n\times n}$ is the diagonal matrix with $(I_F)_{i,i} = 1$ if $i \in F$, otherwise $(I_F)_{i,i} = 0$. If (a) holds, then

$$\|z_{k+1} - Py\|_Q^2 \leq \|z_k - Py\|_Q^2 - \|z_k - Py_k\|_W^2$$

for all $k \in \mathbb{N}$ and all $y \in \operatorname{zer} \Phi_{A,p}$. If (b) also holds, then

(i) $z_k - Py_k \to 0$,

(ii) $\Phi_{A,p} y_k \ni SU(z_k - Py_k) \to 0$,

(iii) $y_k \rightharpoonup y^\star$,

(iv) $z_k \rightharpoonup Py^\star \in \operatorname{fix} T_A$,

for some $y^\star \in \operatorname{zer} \Phi_{A,p}$ as $k \to \infty$.

Proof. Let $Q \in \mathbb{R}^{d\times d}$ be a symmetric matrix and let $k \in \mathbb{N}$. For arbitrary $y \in \operatorname{zer} \Phi_{A,p}$, we have

$$\begin{aligned}
\|z_{k+1} - Py\|_Q^2 &= \|z_k - Py - U(z_k - Py_k)\|_Q^2 \\
&= \|z_k - Py\|_Q^2 + \|z_k - Py_k\|_{U^T QU}^2 - 2\langle QU(z_k - Py_k), z_k - Py \rangle.
\end{aligned}$$

Insert $Py_k - Py_k$ on the right hand side of the inner product to get

$$\begin{aligned}\|z_{k+1} - Py\|_Q^2 &= \|z_k - Py\|_Q^2 + \|z_k - Py_k\|_{U^TQU}^2 - 2\langle QU(z_k - Py_k), z_k - Py_k\rangle \\ &\quad - 2\langle QU(z_k - Py_k), Py_k - Py\rangle \\ &= \|z_k - Py\|_Q^2 - \|z_k - Py_k\|_{QU+(QU)^T-U^TQU}^2 \\ &\quad - 2\langle P^TQU(z_k - Py_k), y_k - y\rangle.\end{aligned}$$

From (10) we know that $SU(z_k - Py_k) \in \Phi_{A,p}y_k$ and Lemma 7.2 then gives that

$$0 \leq \langle SU(z_k - Py_k), y_k - y\rangle - \|y_k - y\|_B^2.$$

Add two times this inequality to the previous equality to get

$$\begin{aligned}\|z_{k+1} - Py\|_Q^2 &\leq \|z_k - Py\|_Q^2 - \|z_k - Py_k\|_{QU+(QU)^T-U^TQU}^2 - 2\|y_k - y\|_B^2 \\ &\quad - 2\langle P^TQU(z_k - Py_k), y_k - y\rangle + 2\langle SU(z_k - Py_k), y_k - y\rangle \\ &= \|z_k - Py\|_Q^2 - \|z_k - Py_k\|_{QU+(QU)^T-U^TQU}^2 - 2\|y_k - y\|_B^2 \\ &\quad - 2\langle (P^TQU - SU)(z_k - Py_k), y_k - y\rangle.\end{aligned}$$

Assume that condition (a) is satisfied, then $P^TQU - SU = I_F(P^TQU - SU)$ and since $I_F = (2B)^{1/2}(\frac{1}{2}B^\dagger)^{1/2}$, we have

$$\begin{aligned}\|z_{k+1} - Py\|_Q^2 &\leq \|z_k - Py\|_Q^2 - \|z_k - Py_k\|_{QU+(QU)^T-U^TQU}^2 - 2\|y_k - y\|_B^2 \\ &\quad - 2\langle (2^{-1}B^\dagger)^{1/2}(P^TQU - SU)(z_k - Py_k), (2B)^{1/2}(y_k - y)\rangle.\end{aligned}$$

Using Young's inequality then finally results in

$$\begin{aligned}\|z_{k+1} - Py\|_Q^2 &\leq \|z_k - Py\|_Q^2 - \|z_k - Py_k\|_{QU+(QU)^T-U^TQU}^2 \\ &\quad + \|(2^{-1}B^\dagger)^{1/2}(P^TQU - SU)(z_k - Py_k)\|^2 \\ &= \|z_k - Py\|_Q^2 - \|z_k - Py_k\|_W^2\end{aligned} \tag{11}$$

where $W = QU + (QU)^T - U^TQU - \frac{1}{2}(P^TQU - SU)^TB^\dagger(P^TQU - SU)$, which proves the first statement.

From here on we assume that (b) also holds, which implies that both $\|\cdot\|_Q$ and $\|\cdot\|_W$ indeed are norms on $\mathcal{H}^d$. The k used in (11) is arbitrary and the inequality holds for all $k \in \mathbb{N}$. Adding the inequalities for $k = 0, 1, \ldots$ implies

$$\|z_k - Py_k\|_W^2 \to 0,$$

which implies that (i) holds. Statement (ii) follows directly from (i) and (10). Inequality (11) further implies

$$\|z_{k+1} - Py\|_Q^2 \leq \|z_k - Py\|_Q^2$$

for all $k \in \mathbb{N}$ and all $y \in \operatorname{zer} \Phi_{A,p}$ which in turn implies the boundedness of $\{z_k\}_{k\in\mathbb{N}}$ and the convergence of $\{\|z_k - Py\|_Q^2\}_{k\in\mathbb{N}}$ for all $y \in \operatorname{zer} \Phi_{A,p}$. These two facts along with (i) and (ii) will be used to prove (iii). Statement (iv) then follows directly from the weak continuity of P and (iii).

For the proof of (iii), we will first show that $\{y_k\}_{k\in\mathbb{N}}$ is bounded and hence has weak sequential cluster points. Then we will show that these cluster points are in $\operatorname{zer} \Phi_{A,p}$ and lastly we will show that there is at most one cluster point. The convergence $y_k \rightharpoonup y^\star$ for some $y^\star \in \operatorname{zer} \Phi_{A,p}$ then follows from [3, Lemma 2.46].

Since SUP is a p-kernel over $\mathcal{A}_n^F$, the operator $(SUP + \Phi_{A,p})^{-1} \circ SU$ can be evaluated using a finite number of vector additions, scalar multiplications, resolvents and evaluations of the cocoercive operators A_i with $i \in F$. All of these operations are Lipschitz continuous and hence must $(SUP + \Phi_{A,p})^{-1} \circ SU$ be Lipschitz continuous, let us say with constant L. Furthermore, with $y \in \operatorname{zer} \Phi_{A,p}$ we have $y = (SUP + \Phi_{A,p})^{-1} SUPy$ and

$$\|y_k - y\| = \|(SUP + \Phi_{A,p})^{-1} SUz_k - (SUP + \Phi_{A,p})^{-1} SUPy\| \leq L\|z_k - Py\|.$$

As noted above, the sequence $\{z_k\}_{k\in\mathbb{N}}$ is bounded and hence $\{y_k\}_{k\in\mathbb{N}}$ is also bounded and has convergent subsequences [3, Lemma 2.45]. From (ii), the maximal monotonicity of $\Phi_{A,p}$, see Section 2.1, and the weak-strong continuity of maximally monotone operators [3, Proposition 20.37] we can conclude that any weak sequential cluster point of $\{y_k\}_{k\in\mathbb{N}}$ lies in $\operatorname{zer} \Phi_{A,p}$.

What remains to be shown is that $\{y_k\}_{k\in\mathbb{N}}$ possesses at most one weak sequential cluster point. Let $\{y_{k_i}\}_{i\in\mathbb{N}}$ and $\{y_{k_j}\}_{j\in\mathbb{N}}$ be sub-sequences such that $y_{k_i} \rightharpoonup a \in \operatorname{zer} \Phi_{A,p}$ and $y_{k_j} \rightharpoonup b \in \operatorname{zer} \Phi_{A,p}$. Since P as an operator on $\mathcal{H}^n$ is linear and hence weakly continuous, (i) implies that

$$z_{k_i} \rightharpoonup Pa \quad \text{and} \quad z_{k_j} \rightharpoonup Pb.$$

Furthermore, $\{\|z_k - Pa\|_Q^2\}_{k\in\mathbb{N}}$ and $\{\|z_k - Pb\|_Q^2\}_{k\in\mathbb{N}}$ both converge and hence

$$\langle z_k, Pa - Pb\rangle_Q = \tfrac{1}{2}\|z_k - Pb\|_Q^2 - \tfrac{1}{2}\|z_k - Pa\|_Q^2 - \tfrac{1}{2}\|Pb\|_Q^2 + \tfrac{1}{2}\|Pa\|_Q^2 \to \nu$$

for some $\nu \in \mathbb{R}$. In particular, it means that

$$\langle z_{k_i}, Pa - Pb\rangle_Q \to \langle Pa, Pa - Pb\rangle_Q = \nu = \langle Pb, Pa - Pb\rangle_Q \leftarrow \langle z_{k_j}, Pa - Pb\rangle_Q.$$

This implies

$$0 = \langle Pa - Pb, Pa - Pb\rangle_Q = \|Pa - Pb\|_Q^2$$

and hence $Pa = Pb$. Since $y \in \operatorname{zer} \Phi_{A,p}$ implies $y = (SUP + \Phi_{A,p})^{-1} SUPy$ we have

$$a = (SUP + \Phi_{A,p})^{-1} SUPa = (SUP + \Phi_{A,p})^{-1} SUPb = b.$$

This proves that there is at most one weak sequential cluster point. Hence, $\{y_k\}_{k\in\mathbb{N}}$ converges weakly to some $y^\star \in \operatorname{zer} \Phi_{A,p}$, i.e., (iii) holds. □

A few remarks on this theorem are in order. The sequence generated by a fixed point iteration of a frugal splitting operator that satisfies Theorem 7.3 is Fejér monotone, see [3, Definition 5.1] with respect to $P \operatorname{zer} \Phi_{A,p}$ in the Hilbert space given by $\langle \cdot, \cdot \rangle_Q$. This is true even for methods without minimal lifting such as momentum methods, see Section 7.1. We will in particular show that forward-backward splitting with Nesterov-like momentum where the momentum parameter is fixed satisfies Theorem 7.3 and hence is Fejér monotone. Furthermore, we see that condition (a) becomes stricter when the set F on which we make forward evaluations becomes smaller. This puts stronger restrictions on the structure of the frugal splitting operator. For instance, if we assume that S, U and P are square and invertible and $F = \emptyset$, condition (a) states that $Q = (P^T)^{-1}S$ and hence $(P^T)^{-1}S$ must be symmetric. In other cases, we have more freedom in choosing $Q \succ 0$ such that (b) of Theorem 7.3 is satisfied. It should also be noted that, although $W \succ 0$ is a quadratic expression in Q, it can be transformed to an equivalent positive definite condition that is linear in Q by using a Schur complement, i.e.,

$$W \succ 0 \iff \begin{bmatrix} QU + (QU)^T - U^T Q U & U^T (P^T Q - S)^T (\frac{1}{2} B^\dagger)^{1/2} \\ (\frac{1}{2} B^\dagger)^{1/2} (P^T Q - S) U & I \end{bmatrix} \succ 0$$

where $I \in \mathbb{R}^{n \times n}$ is an identity matrix and $(\frac{1}{2} B^\dagger)^{1/2}$ exists since it is a diagonal matrix with non-negative elements. This makes the search for a matrix Q that satisfies (a) and (b) a semi-definite feasibility problem that straightforwardly can be solved numerically.

From the $W \succ 0$ of condition (b) we conclude that U must have full rank, otherwise there exists a non-zero element $x \in \mathbb{R}^d$ such that $Ux = 0$ which implies $x^T W x = 0$ which is a contradiction. This is convenient since, given a frugal splitting operator with representation (p, M, N, U, V) where U is invertible, it is easy to find a representation of the form used in Theorem 7.3, i.e., (p, SUP, SU, U, UP). In fact, this factorization can be expressed only in terms of the matrices N, U and V,

$$S = NU^{-1} \quad \text{and} \quad P = U^{-1}V,$$

which implies that $M = NU^{-1}V$ must hold for such a frugal splitting operator.

There are frugal splitting operators where U is rank deficient but their convergence are of no interest. This is because they are either guaranteed to not converge in general or they can be reduced to a frugal splitting operator whose representation has a full rank U without losing any information. To see this, let (p, SUP, SU, U, UP) be a representation satisfying Corollary 5.2 of a frugal splitting operator $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$. Let $\Pi \in \mathbb{R}^{d \times d}$ be the projection matrix onto $\operatorname{ran} U^T$—the projection is here in the standard Euclidean $\mathbb{R}^d$ space. The projection matrix on the kernel of U is then $\overline{\Pi} = I - \Pi$. Consider the fixed point iteration $z_{k+1} = T_A z_k$ for some $A \in \mathcal{A}_n^F$ and $z_0 \in \mathcal{H}^d$. If we define $z_0^{\parallel} = \Pi z_0$ and $z_0^{\perp} = \overline{\Pi} z_0$ this fixed point

iteration can be written as $z_k = z_k^{\parallel} + z_k^{\perp}$ where

$$\begin{aligned} y_k &= (SUP + \Phi_{A,p})^{-1} S U z_k^{\parallel}, \\ z_{k+1}^{\parallel} &= z_k^{\parallel} - \Pi U(z_k^{\parallel} - P y_k), \\ z_{k+1}^{\perp} &= z_k^{\perp} - \overline{\Pi} U(z_k^{\parallel} - P y_k), \end{aligned}$$

for all $k \in \mathbb{N}$. Without going into details, if $\operatorname{ran} U = \operatorname{ran} U^T$ then $z_k^{\perp} = z_0^{\perp}$ for all $k \in \mathbb{N}$ and only the $\{z_k^{\parallel}\}_{k\in\mathbb{N}}$ sequence is of interest. Furthermore, the $\{z_k^{\parallel}\}_{k\in\mathbb{N}}$ sequence can be recovered from a fixed point iteration of a frugal splitting operator with lifting equal to the rank of U. If instead $\operatorname{ran} U \neq \operatorname{ran} U^T$, it is always possible to find an operator tuple $A \in \mathcal{A}_n^F$ and initial point $z_0 \in \mathcal{H}^d$ such that $z_k^{\parallel} = z_0$, $z_k^{\perp} = -kc$ and $z_k = z_0 - kc$ for all $k \in \mathbb{N}$ and some $c \in \mathcal{H}^d \setminus \{0\}$. Hence, $\{z_k\}_{k\in\mathbb{N}}$ will always diverge for this choice of A and z_0.

7.1 Applications of Theorem 7.3

Theorem 7.3 recovers many well-known convergence results, for instance the results for forward-backward splitting [17, 20], Douglas–Rachford splitting [21], the Chambolle–Pock method [8], and the minimal lifting methods of Ryu [29] and Malitsky–Tam [23]. We will not present all these results here and settle for presenting the result for the three operator splitting of Davis and Yin [12]. We will also present convergence conditions for the fixed point iteration of the forward-backward operator with Nesterov-like momentum [4, 26]. To save space, we will not derive any of the representations and simply state the primal index p and the matrices U, S and P of a representation (p, SUP, SU, U, UP), see Corollary 5.2. Similarly, for the convergence results we just state the matrices Q and W of Theorem 7.3. Detailed derivations of all the listed examples and more can be found in the supplement.

Three Operator Splitting of Davis–Yin The three operator splitting of Davis and Yin has already been presented in Section 5.5 but we restate it here,

$$x_{k+1} = x_k - \mathrm{J}_{\gamma A_1} x_k + \mathrm{J}_{\gamma A_3} \circ (2\mathrm{J}_{\gamma A_1} - \mathrm{Id} - \gamma A_2 \circ \mathrm{J}_{\gamma A_1}) x_k$$

where $x_0 \in \mathcal{H}$, $\gamma > 0$ and $A = (A_1, A_2, A_3) \in \mathcal{A}_3^{\{2\}}$. The representation derived in that section can be factored into a representation of the form $(3, SUP, SU, U, UP)$ where

$$U = \begin{bmatrix} 1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 \\ 1 \\ \gamma^{-1} \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} \gamma & 0 & 1 \end{bmatrix}.$$

To prove convergence, we choose $Q = \begin{bmatrix} \gamma^{-1} \end{bmatrix}$ in Theorem 7.3 and, assuming A satisfies Assumption 7.1, this results in $0 = (I - I_F)(P^T Q - S)U$ and $W = \begin{bmatrix} \gamma^{-1} - \frac{1}{2\beta_2} \end{bmatrix}$.

If $\gamma < 2\beta_2$, then $x_k \rightharpoonup Py^\star$ for $y^\star \in \operatorname{zer}\Phi_{A,3}$ and

$$\mathrm{J}_{\gamma A_3} \circ (2\,\mathrm{J}_{\gamma A_1} - \mathrm{Id} - \gamma A_2 \circ \mathrm{J}_{\gamma A_1}) x_k = \mathrm{J}_{\gamma A_1} x_k + x_{k+1} - x_k \rightharpoonup x^\star \in \operatorname{zer} A_1 + A_2 + A_3$$

and hence also $\mathrm{J}_{\gamma A_1} x_k \rightharpoonup x^\star$. This is the same convergence condition as the one presented in [12].

Forward-Backward with Nesterov-like Momentum A fixed point iteration of a forward-backward operator with Nesterov-like momentum [26] can be written as

$$x_{k+1} = \mathrm{J}_{\gamma A_2}(x_k + \theta(x_k - x_{k-1}) - \gamma A_1(x_k + \theta(x_k - x_{k-1})))$$

for $\lambda > 0$, $\theta \in \mathbb{R}$, $x_0, x_{-1} \in \mathcal{H}$ and $A = (A_1, A_2) \in \mathcal{A}_2^{\{1\}}$. In the proximal-gradient setting, this is also the same update that is used in the FISTA method [4]. Nesterov momentum gained popularity due to it achieving optimal convergence rates in the smooth optimization setting. However, these faster convergence rates require a momentum parameter θ that varies between iterations, something the fixed point iterations considered in this paper will not allow.

We remove the dependency on previous iterations by introducing an extra iterate as

$$\begin{aligned} x_{k+1} &= \mathrm{J}_{\gamma A_2}(x_k + \theta y_k - \gamma A_1(x_k + \theta y_k)), \\ y_{k+1} &= x_{k+1} - x_k \end{aligned}$$

where $x_0, y_0 \in \mathcal{H}$. This is a fixed point iteration of the frugal splitting operator given by the representation $(2, SUP, SU, U, UP)$ where

$$U = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 1-\theta & \theta \\ \gamma^{-1}(1-\theta) & \gamma^{-1}\theta \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The lifting number is two, which is not minimal, even though the kernel SUP has rank one and is minimal. In fact, the kernel is the same as for the ordinary forward-backward splitting. Theorem 7.3 with

$$Q = \gamma^{-1} \begin{bmatrix} 1-\theta & \theta \\ \theta & a \end{bmatrix} \quad \text{and} \quad W = \gamma^{-1} \begin{bmatrix} 1-\theta-\hat{\gamma}-a & \theta(1-\hat{\gamma}) \\ \theta(1-\hat{\gamma}) & a-\theta^2\hat{\gamma} \end{bmatrix}$$

where $a > 0$ and $\hat{\gamma} = \frac{\gamma}{2\beta_1} > 0$ yields the convergence of $x_k \rightharpoonup x^\star \in \operatorname{zer} A_1 + A_2$ and $y_k \rightharpoonup 0$ if A satisfies Assumption 7.1 and both $Q \succ 0$ and $W \succ 0$. It can be verified that $Q - W \succeq 0$ hence it is enough that $W \succ 0$ which is equivalent to

$$0 < a - \theta^2\hat{\gamma} \quad \text{and} \quad 0 < 1 - \theta - \hat{\gamma} - a - \tfrac{\theta^2(1-\hat{\gamma})^2}{a-\theta^2\hat{\gamma}}.$$

If we restrict these results to $\theta > 0$, these conditions hold with $a = \theta^2\hat{\gamma} + \theta(1-\hat{\gamma})$ if

$$0 < 1 - 3\theta - \tfrac{\gamma}{2\beta_1}(\theta - 1)^2.$$

It is easily verified that this condition also implies the existence of an $a > 0$ that makes $Q \succ 0$ and $W \succ 0$ even in the case when $\theta = 0$. In fact, it reduces to the well known result $0 < \gamma < 2\beta_1$ for ordinary forward-backward splitting. As mentioned before, if these conditions hold then Theorem 7.3 gives Fejér monotonicity of $\{(x_k, y_k)\}_{k\in\mathbb{N}}$ w.r.t. $P\operatorname{zer}\Phi_{A,p} = \operatorname{zer}(A_1 + A_2)\times\{0\}$ in the norm $\|\cdot\|_Q$.

8. A New Frugal Splitting Operator With Minimal Lifting

We will now derive a new frugal splitting operator with minimal lifting. The approach we will take is the same as the one outlined in Proposition 6.5, i.e., we will select a primal index p, a kernel M and matrices H and K and form a representation as (p, M, MK, HMK, HM). As long as the matrices satisfy Proposition 6.5, the resulting generalized primal-dual resolvent is a frugal splitting operator with lifting equal to the rank of the kernel. Therefore, if we choose a kernel with minimal rank, the resulting frugal splitting operator will have minimal lifting.

As noted in Section 6, the minimal lifting number of a frugal splitting operator over $\mathcal{A}_n^F$ depends on F. In particular, from Corollary 6.7 we see that it is not only a question regarding whether F is empty or not; the minimal lifting number depends on the actual order of the forward and backward evaluations. For this reason, we choose to construct a frugal splitting operator over $\mathcal{A}_n^F$ where F is such that $1 \notin F$ and $n \notin F$. Corollary 6.7 then guarantees that the minimal lifting in this setting is $n - 1 - |F|$ instead of potentially being $n - |F|$. To further simplify the setup we assume that all the single-valued evaluations come one after another as $F = \{n-f, \dots, n-1\}$ where $f = |F|$ is the number of single-valued operators.

THEOREM 8.1
Let $n, f \in \mathbb{N}$ be such that $n \geq 2$ and $f \leq n-2$. Let $A = (A_1, \dots, A_n) \in \mathcal{A}_n^F$ where $F = \{n-f, \dots, n-1\}$ if $f > 0$ and $F = \emptyset$ if $f = 0$. Let $z_{i,0} \in \mathcal{H}$ for all $i \in \{1, \dots, n-1-f\}$ and

$$
\begin{aligned}
x_{1,k} &= \mathrm{J}_{\lambda A_1} z_{1,k}, \\
x_{i,k} &= \mathrm{J}_{\theta^{-1}\lambda A_i}(x_{1,k} + \theta^{-1} z_{i,k}) && \text{for all } i \in \{2, \dots, n-1-f\}, \\
x_{i,k} &= \lambda A_i x_{1,k} && \text{for all } i \in \{n-f, \dots, n-1\}, \\
\bar{x}_k &= \sum\nolimits_{j=n-f}^{n-1} x_{j,k} + \sum\nolimits_{j=2}^{n-1-f} (z_{j,k} + \theta(x_{1,k} - x_{j,k})), \\
x_{n,k} &= \mathrm{J}_{\lambda A_n}(2x_{1,k} - z_{1,k} - \bar{x}_k), \\
z_{i,k+1} &= z_{i,k} - \theta(x_{i,k} - x_{n,k}) && \text{for all } i \in \{1, \dots, n-1-f\}
\end{aligned}
$$

for all $k \in \mathbb{N}$ where $\gamma > 0$ and $\theta > 0$. If A satisfies Assumption 7.1 and

$$
\frac{\lambda}{2} \sum\nolimits_{i=n-f}^{n-1} \beta_i^{-1} < 2 - \theta(n-1-f)
$$

then $x_{n,k} \rightharpoonup x^\star \in \operatorname{zer} \sum_{i=1}^n A_i$.

Proof. Let $T_{(\cdot)}\colon \mathcal{H}^{n-1-f} \to \mathcal{H}^{n-1-f}$ be the generalized primal-dual resolvent with representation (n, M, MK, HMK, HM) with the matrices

$$M = \left[\begin{array}{c|c|c|c} 1 & & & 1 \\ \hline \mathbf{1} & \frac{1}{\theta} I & & \mathbf{1} \\ \hline \mathbf{1} & & & \mathbf{1} \\ \hline 1 & & & 1 \end{array}\right] \in \mathbb{R}^{n\times n}, \quad K = \left[\begin{array}{c|c} \frac{1}{2} & \\ \hline & I \\ \hline & \\ \hline \frac{1}{2} & \end{array}\right] \in \mathbb{R}^{n\times(n-1-f)}$$

$$\text{and} \quad H = \theta \left[\begin{array}{c|c|c|c} \frac{1}{2+f} & & \frac{1}{2+f}\mathbf{1}^T & \frac{1}{2+f} \\ \hline & I & & \end{array}\right] \in \mathbb{R}^{(n-1-f)\times n}$$

where $\theta > 0$, $I \in \mathbb{R}^{(n-2-f)\times(n-2-f)}$ is the identity matrix, $\mathbf{1}$ are column vectors of ones with appropriate sizes and empty blocks denote zero matrices. Note, the third rows and columns vanish completely when $f = 0$ while the second rows and columns vanish when $f = n-2$. The matrix M is an n-kernel over $\mathcal{A}_n^F$ with rank $n-1-f$, which is minimal, see Corollary 6.7. Furthermore, these matrices satisfy $\operatorname{ran} K = \operatorname{ran} M^T = (\ker M)^\perp$ and $\ker H = (\operatorname{ran} H^T)^\perp = (\operatorname{ran} M)^\perp$ and hence $T_{(\cdot)}$ is a frugal splitting operator by Proposition 6.5. For later use we note that $T_{(\cdot)}$ also can be represented as (n, SUP, SU, U, UP) with $U = HMK$, $S = MK(HMK)^{-1}$, $P = (HMK)^{-1}HM$.

We first consider the case when $\lambda = 1$, the general case when $\lambda > 0$ will be proved below. In this case, with $z_k = (z_{1,k}, \dots, z_{n-1-f,k})$, we note that the update of z_{k+1} in the theorem can be written as

$$z_{k+1} = T_A z_k.$$

It can also be verified that $(y_{1,k}, \dots, y_{n,k}) = (M + \Phi_{A,n})^{-1} MK z_k$ satisfies

$$\begin{aligned} y_{1,k} &= z_{1,k} - x_{1,k}, \\ y_{i,k} &= z_{1,k} + \theta(x_{1,k} - x_{i,k}) && \text{for all } i \in \{2, \dots, n-1-f\}, \\ y_{i,k} &= x_{i,k} && \text{for all } i \in \{n-f, \dots, n\}, \end{aligned}$$

for all $k \in \mathbb{N}$. Furthermore, if we choose

$$Q = \theta^{-1} \left[\begin{array}{c|c} 1 & \\ \hline & I \end{array}\right] \in \mathbb{R}^{(n-1-f)\times(n-1-f)},$$

then $I_F(P^TQ - S)U = (P^TQ - S)U$ where I_F is defined in Theorem 7.3 and condition (a) of Theorem 7.3 holds. Condition (b) of Theorem 7.3 with this Q reads as

$$Q \succ 0 \quad \text{and} \quad W = \left[\begin{array}{c|c} 2 - \theta(n-1-f) - \frac{1}{2}\sum_{i=n-f}^{n-1} \beta_i^{-1} & \\ \hline & \frac{1}{\theta} I \end{array}\right] \succ 0$$

which holds if and only if $\theta > 0$ and $\frac{1}{2}\sum_{i=n-f}^{n-1}\beta_i^{-1} < 2-\theta(n-1-f)$. In this case, Theorem 7.3 gives the convergence of

$$(y_{1,k},\dots,y_{n,k}) \rightharpoonup (y_1^\star,\dots,y_n^\star) \in \operatorname{zer}\Phi_{A,n}$$

and $x_{n,k} = y_{n,k} \rightharpoonup y_n^\star \in \operatorname{zer}\sum_{i=1}^n A_i$.

Finally, consider the general case with arbitrary $\lambda > 0$. We then see that the update of the theorem can be written as

$$z_{k+1} = T_{\lambda A} z_k$$

for all $k \in \mathbb{N}$ where $\lambda A = (\lambda A_1,\dots,\lambda A_n)$. Since A_i is β_i-cocoercive for all $i \in F$, λA_i will be $\beta_i\lambda^{-1}$-cocoercive for all $i \in F$ and hence will λA satisfy Assumption 7.1. Applying Theorem 7.3 to this fixed point iteration with the scaled operator tuple then gives the convergence of

$$(y_{1,k},\dots,y_{n,k}) \to (y_1^\star,\dots,y_n^\star) \in \operatorname{zer}\Phi_{\lambda A,n}$$

and $x_{n,k} = y_{n,k} \to y_n^\star \in \operatorname{zer}\sum_{i=1}^n \lambda A_i$ if $\theta > 0$ and $\frac{\lambda}{2}\sum_{i=n-f}^{n-1}\beta_i^{-1} < 2-\theta(n-1-f)$. This proves the theorem. □

Note that the relaxation factor θ also occurs as a step-size scaling on some of the resolvents, i.e., $\mathrm{J}_{\theta^{-1}\lambda A_i}$ is evaluated for $i \in \{2,\dots,n-1-f\}$ while $\mathrm{J}_{\lambda A_i}$ is evaluated for $i \in \{1,n\}$ This was necessary in order to ensure convergence. We also see that, in order to find a step-size that ensures convergence, $\theta < \frac{2}{n-1-f}$ is needed, i.e., the relaxation parameter must decrease with the number of operators. However, this is counteracted by the fact that the step-size of most of the resolvents are scaled with θ^{-1} and hence will also increase with the number of operators.

There are no step-size restrictions when no forward evaluations are used, $f = 0$. Furthermore, we see that the step-size bound only depends on the sum of the inverse cocoercivity constants. This is natural since all forward steps are evaluated at the same point and the results are simply added together. The update can therefore equivalently be seen as evaluating $\widehat{A} = \sum_{i=n-f}^{n-1} A_i$ instead of each operator individually and $\widehat{A}$ is a $(\sum_{i=n-f}^{n-1}\beta_i^{-1})^{-1}$-cocoercive operator.

8.1 Relation to Other Methods With Minimal Lifting

When $n = 3$, $f = 1$ and $\theta = 1$, this method reduces to the three operator splitting of Davis and Yin, see Section 5.5. When $n = 3$ and $f = 0$, the method is closely related to the three operator resolvent splitting operator of Ryu [29]. That method uses a

frugal splitting operator that is calculated as $(\hat{z}_1, \hat{z}_2) = T_{(A_1,A_2,A_3)}(z_1, z_2)$ where

$$
\begin{aligned}
x_1 &= \mathrm{J}_{\lambda A_1}(z_1),\\
x_2 &= \mathrm{J}_{\lambda A_2}(z_2 + x_1),\\
x_3 &= \mathrm{J}_{\lambda A_3}(-z_1 - z_2 + x_1 + x_2),\\
\hat{z}_1 &= z_1 + \theta(x_3 - x_1),\\
\hat{z}_2 &= z_2 + \theta(x_3 - x_2)
\end{aligned}
$$

and we see that, if we consider the unrelaxed case $\theta = 1$, then this update is the same as our update in Theorem 8.1. For other choices of θ, our proposed method differs in that the step-size in the computation of x_2 is scaled with θ^{-1}. As noted by Malitsky and Tam [23], a straightforward extension of Ryu's method to four operators fails to converge for all $\theta > 0$ in some cases and we found this step-size scaling to be the key that allowed us to establish convergence for $n > 3$.

In the case when $f = 0$ and $n > 2$ is arbitrary, Malitsky and Tam [23] presented a splitting method which they proved had minimal lifting. It uses a frugal splitting operator where $(\hat{z}_1, \ldots, \hat{z}_{n-1}) = T_{(A_1,\ldots,A_n)}(z_1, \ldots, z_{n-1})$ is calculated as

$$
\begin{aligned}
x_1 &= \mathrm{J}_{\gamma A_1}(z_1),\\
x_i &= \mathrm{J}_{\gamma A_i}(z_i - z_{i-1} + x_{i-1}) && \text{for all } i \in \{2, \ldots, n-1\},\\
x_n &= \mathrm{J}_{\gamma A_n}(-z_{n-1} + x_1 + x_{n-1}),\\
\hat{z}_i &= z_i + \theta(x_{i+1} - x_i) && \text{for all } i \in \{1, \ldots, n-1\}.
\end{aligned}
$$

This method was later expanded to include forward evaluations in [2] and the results of this paper prove that the method with forward evaluations also has minimal lifting. One feature of our splitting operator is that it allows $x_{i,k}$ to be calculated in parallel for all $i \in \{2, \ldots, n-1\}$, which cannot be done with the Malitsky–Tam splitting operator since each resolvent depends on the previous one. However, it should be noted that fixed point iterations of the Malitsky–Tam operator can be partially parallelized with $\ldots, x_{i-2,k+1}, x_{i,k}, x_{i+2,k-1}, \ldots$ being computable in parallel. Comparing the step-sizes of the two methods, the Malitsky–Tam method converges for all $\theta \in (0, 1)$ while our method requires that $\theta \in (0, \frac{2}{n-1})$, i.e., our method requires a smaller relaxation parameter for larger n. As mentioned before though, our method actually increases the step-sizes for the resolvents of $A_2, \ldots, A_{n-1}$ with n, which might offset the decreasing relaxation.

Another method similar to ours in the $f = 0$ case is the method presented by Campoy in [7]. It is based on Douglas–Rachford splitting applied to a product space reformulation of the finite sum monotone inclusion problem which results in a split-

ting operator where $(\hat{z}_1,\dots,\hat{z}_{n-1}) = T_A(z_1,\dots,z_{n-1})$ is such that

$$\begin{aligned} x_1 &= \mathrm{J}_{\frac{\gamma}{n-1}A_1}\left(\tfrac{1}{n-1}\textstyle\sum_{j=1}^{n-1} z_j\right) \\ x_i &= \mathrm{J}_{\gamma A_i}(2x_1 - z_{i-1}) && \text{for all } i \in \{2,\dots,n\} \\ \hat{z}_i &= z_i + \theta(x_{i+1} - x_1) && \text{for all } i \in \{1,\dots,n-1\}. \end{aligned}$$

This operator clearly has minimal lifting and its fixed point iteration converges for all $\theta \in (0,2)$, $\gamma > 0$ and $A \in \mathcal{A}_n$. As with our method, it is parallelizable and uses an uneven step-size with the first resolvent using a step-size of $\frac{\gamma}{n-1}$ while the others use a step-size of γ. However, it does not appear possible to rewrite this splitting operator as a special case of ours, or vice versa.

A similar approach to the one used by Campoy was also presented by Condat *et al.* [11] but restricted to the convex optimization case. They presented essentially the same splitting operator as the one from Campoy but with a weighted average instead of the arithmetic average in the first row. However, Condat *et al.* also applied more schemes than just Douglas–Rachford splitting to the product space reformulation, resulting in several different parallelizable splitting operators both with and without forward evaluations. Most notably is perhaps a Davis–Yin based splitting operator over $\mathcal{A}_n^{\{2\}}$ that is parallelizable and has minimal lifting, [11, Equation (212)]. Although many of these methods are similar to ours, we again failed to rewrite either our or any of their methods with minimal lifting as special cases of each other.

The biggest difference between this work and the works of Campoy and Condat *et al.* is perhaps conceptual. While their parallelizable methods are based on applying existing splitting operators to reformulations of the problem, we directly search over all possible frugal splitting operators. Our search is of course in no way exhaustive but is enabled by the representation theorem since it allows us to easily work with the entire class of frugal splitting operators and nothing else.

9. Conclusion

We have presented an explicit parameterization of all frugal splitting operators. The parameterization is in terms of what we call generalized primal-dual resolvents, and we have provided necessary and sufficient conditions for a generalized primal-dual resolvent being a frugal splitting operator. This allows for a unified analysis and both minimal lifting and convergence results that are applicable to all frugal splitting operators. The minimal lifting results of Ryu and Malitsky–Tam were expanded beyond resolvent splitting operators to general frugal splitting operators with forward evaluations and we showed that the lifting number depends on the order of forward and backward evaluations. We further presented a new convergent frugal splitting operator with minimal lifting that allows for most of the forward and/or backward steps to be computed in parallel. In the triple-backward case the method is the same as the minimal lifting method of Ryu if neither method uses relaxation. The

slight difference in how relaxation is introduced is crucial to extend Ryu's method to an arbitrary number of operators. In the double-backward-single-forward case the method reduces to three operator splitting by Davis and Yin.

References

[1] F. Alvarez and H. Attouch. "An Inertial Proximal Method for Maximal Monotone Operators via Discretization of a Nonlinear Oscillator with Damping". *Set-Valued Analysis* **9**:1 (2001), pp. 3–11. DOI: 10.1023/A:1011253113155.

[2] F. J. Aragón-Artacho, Y. Malitsky, M. K. Tam, and D. Torregrosa-Belén. *Distributed Forward-Backward Methods for Ring Networks*. 2022. arXiv: 2112.00274v2 [cs, math]. URL: https://arxiv.org/abs/2112.00274v2 (visited on 2022-09-09).

[3] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Second. CMS Books in Mathematics. Springer International Publishing, 2017. ISBN: 978-3-319-48310-8.

[4] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". *SIAM Journal on Imaging Sciences* **2**:1 (2009), pp. 183–202. DOI: 10.1137/080716542.

[5] R. I. Boţ, E. R. Csetnek, and C. Hendrich. "Inertial Douglas–Rachford Splitting for Monotone Inclusion Problems". *Applied Mathematics and Computation* **256** (2015), pp. 472–487. DOI: 10.1016/j.amc.2015.01.017.

[6] M. N. Bùi and P. L. Combettes. "Warped Proximal Iterations for Monotone Inclusions". *Journal of Mathematical Analysis and Applications* **491**:1 (2020), p. 124315. DOI: 10.1016/j.jmaa.2020.124315.

[7] R. Campoy. "A product space reformulation with reduced dimension for splitting algorithms". *Computational Optimization and Applications* **83**:1 (2022), pp. 319–348. DOI: 10.1007/s10589-022-00395-7. URL: https://doi.org/10.1007/s10589-022-00395-7 (visited on 2022-09-09).

[8] A. Chambolle and T. Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". *Journal of Mathematical Imaging and Vision* **40**:1 (2011), pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

[9] P. L. Combettes and J. Eckstein. "Asynchronous Block-Iterative Primal-Dual Decomposition Methods for Monotone Inclusions". *Mathematical Programming* **168**:1 (2018), pp. 645–672. DOI: 10.1007/s10107-016-1044-0.

[10] L. Condat. "A Primal–Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms". *Journal of Optimization Theory and Applications* **158**:2 (2013), pp. 460–479. DOI: 10.1007/s10957-012-0245-9.

[11] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. *Proximal Splitting Algorithms for Convex Optimization: A Tour of Recent Advances, with New Twists*. 2021. arXiv: 1912.00137 [math]. URL: http://arxiv.org/abs/1912.00137v7 (visited on 2022-09-09).

[12] D. Davis and W. Yin. "A Three-Operator Splitting Scheme and its Optimization Applications". *Set-Valued and Variational Analysis* **25**:4 (2017), pp. 829–858. DOI: 10.1007/s11228-017-0421-z.

[13] Y. Drori and M. Teboulle. "Performance of First-Order Methods for Smooth Convex Minimization: A Novel Approach". *Mathematical Programming* **145**:1 (2014), pp. 451–482. DOI: 10.1007/s10107-013-0653-0.

[14] J. Elliott. *The Characteristic Roots of Certain Real Symmetric Matrices*. MA thesis. Univeristy of Tennessee, 1953. URL: https://trace.tennessee.edu/utk_gradthes/2384.

[15] P. Giselsson. "Nonlinear Forward-Backward Splitting with Projection Correction". *SIAM Journal on Optimization* (2021), pp. 2199–2226. DOI: 10.1137/20M1345062.

[16] P. Giselsson. *Nonlinear Forward-Backward Splitting with Projection Correction*. 2021. arXiv: 1908.07449v3. URL: http://arxiv.org/abs/1908.07449v3.

[17] A. A. Goldstein. "Convex Programming in Hilbert Space". *Bulletin of the American Mathematical Society* **70**:5 (1964), pp. 709–711. DOI: 10.1090/S0002-9904-1964-11178-2.

[18] P. Latafat and P. Patrinos. "Asymmetric Forward–Backward–Adjoint Splitting for Solving Monotone Inclusions Involving Three Operators". *Computational Optimization and Applications* **68**:1 (2017), pp. 57–93. DOI: 10.1007/s10589-017-9909-6.

[19] L. Lessard, B. Recht, and A. Packard. "Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints". *SIAM Journal on Optimization* **26**:1 (2016), pp. 57–95. DOI: 10.1137/15M1009597.

[20] E. S. Levitin and B. T. Polyak. "Constrained Minimization Methods". *USSR Computational mathematics and mathematical physics* **6**:5 (1966), pp. 1–50.

[21] P. L. Lions and B. Mercier. "Splitting Algorithms for the Sum of Two Nonlinear Operators". *SIAM Journal on Numerical Analysis* **16**:6 (1979), pp. 964–979. DOI: 10.1137/0716071.

[22] Y. Malitsky and M. K. Tam. "A Forward-Backward Splitting Method for Monotone Inclusions Without Cocoercivity". *SIAM Journal on Optimization* **30**:2 (2020), pp. 1451–1472. DOI: 10.1137/18M1207260.

[23] Y. Malitsky and M. K. Tam. *Resolvent Splitting for Sums of Monotone Operators with Minimal Lifting*. 2021. arXiv: 2108.02897v1 [math]. URL: http://arxiv.org/abs/2108.02897v1 (visited on 2022-02-07).

[24] M. Morin, S. Banert, and P. Giselsson. "Nonlinear Forward-Backward Splitting with Momentum Correction" (2022). arXiv: 2112.00481v2 [math]. URL: http://arxiv.org/abs/2112.00481v2.

[25] A. Moudafi and M. Oliny. "Convergence of a Splitting Inertial Proximal Method for Monotone Operators". *Journal of Computational and Applied Mathematics* **155**:2 (2003), pp. 447–454. DOI: 10.1016/S0377-0427(02)00906-8.

[26] Y. Nesterov. "A Method For Solving A Convex Programming Problem With Rate of Convergence $O(1/k^2)$". *Soviet Math. Doklady* **v.269**:No.3 (1983), pp. 543–547.

[27] H. Raguet, J. Fadili, and G. Peyré. "A Generalized Forward-Backward Splitting". *SIAM Journal on Imaging Sciences* **6**:3 (2013), pp. 1199–1226. DOI: 10.1137/120872802.

[28] R. T. Rockafellar. "Monotone Operators and the Proximal Point Algorithm". *SIAM Journal on Control and Optimization* **14**:5 (1976), pp. 877–898. DOI: 10.1137/0314056.

[29] E. K. Ryu. "Uniqueness of DRS as the 2 Operator Resolvent-Splitting and Impossibility of 3 Operator Resolvent-Splitting". *Mathematical Programming* **182**:1 (2020), pp. 233–273. DOI: 10.1007/s10107-019-01403-1.

[30] E. K. Ryu, A. B. Taylor, C. Bergeling, and P. Giselsson. "Operator Splitting Performance Estimation: Tight Contraction Factors and Optimal Parameter Selection". *SIAM Journal on Optimization* **30**:3 (2020), pp. 2251–2271. DOI: 10.1137/19M1304854.

[31] E. K. Ryu and B. C. Vũ. "Finding the Forward-Douglas–Rachford-Forward Method". *Journal of Optimization Theory and Applications* **184**:3 (2020), pp. 858–876. DOI: 10.1007/s10957-019-01601-z.

[32] A. B. Taylor, J. M. Hendrickx, and F. Glineur. "Exact Worst-Case Performance of First-Order Methods for Composite Convex Optimization". *SIAM Journal on Optimization* **27**:3 (2017), pp. 1283–1313. DOI: 10.1137/16M108104X.

[33] A. B. Taylor, J. M. Hendrickx, and F. Glineur. "Smooth Strongly Convex Interpolation and Exact Worst-Case Performance of First-Order Methods". *Mathematical Programming* **161**:1 (2017), pp. 307–345. DOI: 10.1007/s10107-016-1009-3.

[34] P. Tseng. "A Modified Forward-Backward Splitting Method for Maximal Monotone Mappings". *SIAM Journal on Control and Optimization* **38**:2 (2000), pp. 431–446. DOI: 10.1137/S0363012998338806.

[35] B. C. Vũ. "A Splitting Algorithm for Dual Monotone Inclusions Involving Cocoercive Operators". *Advances in Computational Mathematics* **38**:3 (2013), pp. 667–681. DOI: 10.1007/s10444-011-9254-8.

Supplementary Material: Derivation of Representations and Convergence Conditions

In this section we will verify the representations and convergence conditions for the frugal splitting operators stated in the paper. For posterity we will also derive representations and convergence conditions for a number of frugal splitting operators not previously mentioned.

We will keep a consistent notation in each of the examples presented. For a representation of a frugal splitting operator $T_{(\cdot)} \colon \mathcal{H}^d \to \mathcal{H}^d$ over $\mathcal{A}_n^F$, the goal is to find matrices M, N, U, V such that

$$\begin{aligned} y &= (M + \Phi_{(\cdot),p})^{-1} N z, \\ T_{(\cdot)} z &= z - Uz - Vy \end{aligned}$$

for some $p \in \{1,\dots,n\}$ and that Theorem 5.1 is satisfied. Such a representation (p,M,N,U,V) can always be factorized in terms of matrices S and P such that

$$M = SUP, \quad N = SU \quad \text{and} \quad V = UP,$$

and where $\operatorname{ran} P \subseteq (\ker U)^\perp$ and $\ker S \supseteq (\operatorname{ran} U)^\perp$, see Corollary 5.2. The convergence conditions of Theorem 7.3 for a fixed point iterations of T_A for some $A \in \mathcal{A}_n^F$ that satisfies Assumption 7.1 is stated in terms of this factorization and can be written as

$$\begin{gathered} Q \succ 0, \\ (I - I_F)(P^T Q - S)U = 0, \\ W = QU + (QU)^T - U^T Q U - \tfrac{1}{2} U^T (P^T Q - S)^T B^\dagger (P^T Q - S) U \succ 0 \end{gathered}$$

where Q is a symmetric matrix that needs to be found for each frugal splitting operator. When constructing new frugal splitting operators, we find it more convenient to work with a factorization of the representation (p,M,N,U,V) in terms of matrices H and K such that

$$N = MK, \quad V = HM \quad \text{and} \quad U = HMK,$$

where $\operatorname{ran} K = (\ker M)^\perp$ and $\ker H = (\operatorname{ran} M)^\perp$, see Proposition 6.5. This allows us to first design a kernel and then easily find N, U and V that results in a first order splitting. Furthermore, some of the frugal splitting operators will be presented without step-size. A variable step-size can be added to these methods simply by scaling the operator tuple in the same way as for our new frugal splitting operator in Theorem 8.1.

Forward-Backward

The forward-backward operator [17, 20] is

$$T_{(A_1,A_2)} = \mathrm{J}_{\gamma A_2} \circ (\mathrm{Id} - \gamma A_1)$$

where $\gamma > 0$ and $A = (A_1, A_2) \in \mathcal{A}_2^{\{1\}}$. To derive a representation we choose primal index $p = 2$. The next step would be to apply the Moreau identity to all backward steps with index $i \neq p$ but since there are no such backward steps we can directly define the results of each forward and backward evaluation

$$\begin{aligned} y_1 &= A_1 z, \\ y_2 &= (\mathrm{Id} + \gamma A_2)^{-1}(z - \gamma y_1), \\ T_A z &= y_2. \end{aligned}$$

We rewrite the first two lines such that the input is on the left and the result of all forward and backward steps are on the right

$$\begin{aligned} z &\in A_1^{-1} y_1, \\ \gamma^{-1} z &\in A_2 y_2 + y_1 + \gamma^{-1} y_2, \\ T_A z &= y_2. \end{aligned}$$

If we define $y = (y_1, y_2) \in \mathcal{H}^2$ the first two lines can be written as

$$\begin{bmatrix} 1 \\ \gamma^{-1} \end{bmatrix} z \in \underbrace{\begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2 \end{bmatrix}}_{\Delta_{A,2}} y + \begin{bmatrix} 0 & 0 \\ 1 & \gamma^{-1} \end{bmatrix} y = \underbrace{\begin{bmatrix} A_1^{-1} & -1 \\ 1 & A_2 \end{bmatrix}}_{\Phi_{A,2}} y + \begin{bmatrix} 0 & 1 \\ 0 & \gamma^{-1} \end{bmatrix} y$$

which yields

$$\begin{aligned} y &= \left(\begin{bmatrix} 0 & 1 \\ 0 & \gamma^{-1} \end{bmatrix} + \Phi_{A,2} \right)^{-1} \begin{bmatrix} 1 \\ \gamma^{-1} \end{bmatrix} z, \\ T_A z &= z - \begin{bmatrix} 1 \end{bmatrix} z + \begin{bmatrix} 0 & 1 \end{bmatrix} y, \end{aligned}$$

and the matrices M, N, U, V in the representation $(2, M, N, U, V)$ are then easily identified by comparing to Definition 4.1 which yields

$$M = \begin{bmatrix} 0 & 1 \\ 0 & \gamma^{-1} \end{bmatrix}, \quad N = \begin{bmatrix} 1 \\ \gamma^{-1} \end{bmatrix}, \quad V = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 \end{bmatrix}.$$

It is seen directly that

$$S = \begin{bmatrix} 1 \\ \gamma^{-1} \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

provides a factorization of this representation as $(2, SUP, SU, U, UP)$. For the convergence conditions, choosing $Q = \begin{bmatrix} \gamma^{-1} \end{bmatrix}$ where I is the 2×2 identity matrix it is

obvious that $Q \succ 0$ and we further have

$$\begin{aligned}(I - I_F)(P^T Q - S)U &= (I - I_F)\left(\gamma^{-1}\begin{bmatrix}0\\1\end{bmatrix} - \begin{bmatrix}1\\\gamma^{-1}\end{bmatrix}\right)\\ &= (I - I_F)\begin{bmatrix}-1\\0\end{bmatrix}\\ &= 0\end{aligned}$$

and

$$\begin{aligned}W &= QU + (QU)^T - U^T QU - \tfrac{1}{2}U^T(P^T Q - S)^T B^{\dagger}(P^T Q - S)U\\ &= \left[\gamma^{-1}\right] + \left[\gamma^{-1}\right] - \left[\gamma^{-1}\right] - \tfrac{1}{2}\begin{bmatrix}-1 & 0\end{bmatrix}\begin{bmatrix}\beta_1^{-1} & 0\\0 & 0\end{bmatrix}\begin{bmatrix}-1\\0\end{bmatrix}\\ &= \left[\gamma^{-1} - \tfrac{1}{2\beta_1}\right].\end{aligned}$$

The final condition for convergence is then $W \succ 0$ which yields the well known result $\gamma < 2\beta_1$.

Douglas–Rachford

The Douglas–Rachford splitting operator [21] is

$$T_{(A_1,A_2)} = \tfrac{1}{2}\,\mathrm{Id} + \tfrac{1}{2}(2\,\mathrm{J}_{\gamma A_2} - \mathrm{Id}) \circ (2\,\mathrm{J}_{\gamma A_1} - \mathrm{Id})$$

where $A = (A_1, A_2) \in \mathcal{A}_2$ and $\gamma > 0$. Consider the evaluation $\hat{z} = T_{(A_1,A_2)}z$ for some $z \in \mathcal{H}$. This can be written as

$$\begin{aligned}x_1 &= \mathrm{J}_{\gamma A_1} z,\\ y_2 &= \mathrm{J}_{\gamma A_2}(2x_1 - z),\\ \hat{z} &= \tfrac{1}{2}z + \tfrac{1}{2}(2y_2 - (2x_1 - z)).\end{aligned}$$

We choose the primal index to $p = 2$ and apply the Moreau identity to the first resolvent,

$$\begin{aligned}y_1 &= \mathrm{J}_{\gamma^{-1}A_1^{-1}}(\gamma^{-1}z),\\ x_1 &= z - \gamma y_1,\\ y_2 &= \mathrm{J}_{\gamma A_2}(2x_1 - z),\\ \hat{z} &= \tfrac{1}{2}z + \tfrac{1}{2}(2y_2 - (2x_1 - z)).\end{aligned}$$

Eliminating x_1 gives

$$\begin{aligned}y_1 &= \mathrm{J}_{\gamma^{-1}A_1^{-1}}(\gamma^{-1}z),\\ y_2 &= \mathrm{J}_{\gamma A_2}(z - 2\gamma y_1),\\ \hat{z} &= \gamma y_1 + y_2.\end{aligned}$$

Using the definition of the resolvent yields

$$\begin{aligned} z &\in \gamma y_1 + A_1^{-1} y_1, \\ \gamma^{-1} z &\in 2y_1 + \gamma^{-1} y_2 + A_2 y_2, \\ \hat{z} &= z - z + \gamma y_1 + y_2. \end{aligned}$$

Regrouping such that $\Phi_{(A_1,A_2),2}$ can be identified finally gives

$$\begin{aligned} z &\in [\gamma y_1 + y_2] + [A_1^{-1} y_1 - y_2], \\ \gamma^{-1} z &\in [y_1 + \gamma^{-1} y_2] + [A_2 y_2 + y_1], \\ \hat{z} &= z - [z] + [\gamma y_1 + y_2] \end{aligned}$$

and we can identify

$$M = \begin{bmatrix} \gamma & 1 \\ 1 & \gamma^{-1} \end{bmatrix}, \quad N = \begin{bmatrix} 1 \\ \gamma^{-1} \end{bmatrix}, \quad V = \begin{bmatrix} \gamma & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 \end{bmatrix}.$$

The representation can be factored as $(2, SUP, SU, U, UP)$ where

$$S = \begin{bmatrix} 1 \\ \gamma^{-1} \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} \gamma & 1 \end{bmatrix}.$$

For the convergence conditions, choosing $Q = \begin{bmatrix} \gamma^{-1} \end{bmatrix}$ yield $Q \succ 0$ and

$$(I - I_F)(P^T Q - S) = (P^T Q - S) = \gamma^{-1} \begin{bmatrix} \gamma \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ \gamma^{-1} \end{bmatrix} = 0$$

and

$$\begin{aligned} W &= QU + (QU)^T - U^T Q U - \tfrac{1}{2} U^T (P^T Q - S)^T B^\dagger (P^T Q - S) U \\ &= Q + Q - Q = Q \end{aligned}$$

and hence is $W \succ 0$, i.e., fixed point iterations of the Douglas–Rachford splitting operator always converge.

Davis–Yin Three Operator Splitting

To find a representation of three operator splitting of Davis–Yin [12],

$$T_{(A_1,A_2,A_3)} = \mathrm{J}_{\gamma A_3} \circ (2\,\mathrm{J}_{\gamma A_1} - \mathrm{Id} - \gamma A_2 \circ \mathrm{J}_{\gamma A_1}) + \mathrm{Id} - \mathrm{J}_{\gamma A_1}.$$

where $\gamma > 0$ and $A = (A_1, A_2, A_3) \in \mathcal{A}_3^{\{2\}}$ we choose $p = 3$. Applying Moreau's identity to the resolvents of A_i for all $i \neq p$ (in this case only $\mathrm{J}_{\gamma A_1}$) and defining the result

of each forward and backward-step yields

$$\begin{aligned} y_1 &= (\mathrm{Id} + \gamma^{-1} A_1^{-1})^{-1}(\gamma^{-1} z), \\ y_2 &= A_2(z - \gamma y_1), \\ y_3 &= (\mathrm{Id} + \gamma A_3)^{-1}(2(z - \gamma y_1) - z - \gamma y_2), \\ T_A z &= y_3 + z - (z - \gamma y_1) \end{aligned}$$

for all $z \in \mathcal{H}$. Rearranging the first three lines such that we only have z on the left and y_1, y_2 and y_3 and unscaled operators on the right yields

$$\begin{aligned} z &\in A_1^{-1} y_1 + \gamma y_1, \\ z &\in A_2^{-1} y_2 + \gamma y_1, \\ \gamma^{-1} z &\in A_3 y_3 + 2y_1 + y_2 + \gamma^{-1} y_3, \\ T_A z &= \gamma y_1 + y_3. \end{aligned}$$

If we define $y = (y_1, y_2, y_3) \in \mathcal{H}^3$ we see that the first three lines can be written as

$$\begin{bmatrix} 1 \\ 1 \\ \gamma^{-1} \end{bmatrix} z \in \Delta_{A,3} y + \begin{bmatrix} \gamma & 0 & 0 \\ \gamma & 0 & 0 \\ 2 & 1 & \gamma^{-1} \end{bmatrix} y = \Phi_{A,3} y + \begin{bmatrix} \gamma & 0 & 1 \\ \gamma & 0 & 1 \\ 1 & 0 & \gamma^{-1} \end{bmatrix} y$$

and T_A can then be written as

$$\begin{aligned} y &= \left(\begin{bmatrix} \gamma & 0 & 1 \\ \gamma & 0 & 1 \\ 1 & 0 & \gamma^{-1} \end{bmatrix} + \Phi_{A,3} \right)^{-1} \begin{bmatrix} 1 \\ 1 \\ \gamma^{-1} \end{bmatrix} z, \\ T_A z &= z - \begin{bmatrix} 1 \end{bmatrix} z + \begin{bmatrix} \gamma & 0 & 1 \end{bmatrix} y. \end{aligned}$$

From this we can easily identify the matrices M, N, U and V in the representation $(3, M, N, U, V)$ by comparing to Definition 4.1.

$$M = \begin{bmatrix} \gamma & 0 & 1 \\ \gamma & 0 & 1 \\ 1 & 0 & \gamma^{-1} \end{bmatrix}, \quad N = \begin{bmatrix} 1 \\ 1 \\ \gamma^{-1} \end{bmatrix}, \quad V = \begin{bmatrix} \gamma & 0 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 \end{bmatrix}.$$

This can be factored as $(3, SUP, SU, U, UP)$ where

$$S = \begin{bmatrix} 1 \\ 1 \\ \gamma^{-1} \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} \gamma & 0 & 1 \end{bmatrix}.$$

The convergence conditions are satified by $Q = \left[\gamma^{-1}\right]$. We see that $Q \succ 0$ and

$$\begin{aligned}(I - I_F)(P^T Q - S)U &= (I - I_F)(P^T Q - S) \\ &= (I - I_F)\left(\gamma^{-1}\begin{bmatrix}\gamma \\ 0 \\ 1\end{bmatrix} - \begin{bmatrix}1 \\ 1 \\ \gamma^{-1}\end{bmatrix}\right) \\ &= 0\end{aligned}$$

and

$$\begin{aligned}W =& QU + (QU)^T - U^T QU - \tfrac{1}{2}U^T(P^T Q - S)^T B^\dagger (P^T Q - S)U \\ &= Q - \tfrac{1}{2}(P^T Q - S)^T B^\dagger (P^T Q - S) \\ &= \left[\gamma^{-1}\right] - \tfrac{1}{2}\begin{bmatrix}0 & -1 & 0\end{bmatrix} B^\dagger \begin{bmatrix}0 \\ -1 \\ 0\end{bmatrix} \\ &= \left[\gamma^{-1} - \tfrac{1}{2\beta_2}\right]\end{aligned}$$

and $W \succ 0$ if $\gamma < 2\beta_2$.

Forward-Backward with Momentum on the Forward Step

Forward-Backward with momentum on the forward step [25] can be written as a fixed point iteration of the frugal splitting operator

$$T_{(A_1,A_2)}(z_1, z_2) = \begin{pmatrix}\hat{z}_1 \\ \hat{z}_2\end{pmatrix} = \begin{pmatrix}\mathrm{J}_{\gamma A_2}(z_1 - \gamma A_1 z_1 + \theta z_2) \\ \hat{z}_1 - z_1\end{pmatrix}$$

where $\gamma > 0$, $\theta \in \mathbb{R}$ and $A = (A_1, A_2) \in \mathcal{A}_2^{\{1\}}$. Note, there are other frugal splitting operators whose fixed point iteration are equivalent to this forward-backward method with momentum. We claim that $(2, SUP, SU, U, UP)$ where

$$U = \begin{bmatrix}1 & 0 \\ 1 & 1\end{bmatrix}, \quad S = \begin{bmatrix}1 & 0 \\ \gamma^{-1}(1-\theta) & \gamma^{-1}\theta\end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix}$$

is a representation of this frugal splitting operator. To show this we first note that

$$\begin{aligned}V = UP = \begin{bmatrix}0 & 1 \\ 0 & 1\end{bmatrix}, \quad M = SUP = \begin{bmatrix}0 & 1 \\ 0 & \gamma^{-1}\end{bmatrix} \quad \text{and} \\ N = SU = \begin{bmatrix}1 & 0 \\ \gamma^{-1}(1-\theta) & \gamma^{-1}\theta\end{bmatrix}\begin{bmatrix}1 & 0 \\ 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 \\ \gamma^{-1} & \gamma^{-1}\theta\end{bmatrix}\end{aligned}$$

and

$$\begin{aligned}Nz - My &\in \Phi_{A,2}y, \\ \hat{z} &= z - Uz + Vy\end{aligned}$$

can then be written as

$$\begin{aligned} z_1 - y_2 &\in A_1^{-1} y_1 - y_2, \\ \gamma^{-1}(z_1 + \theta z_2) - \gamma^{-1} y_2 &\in A_2 y_2 + y_1, \\ \hat{z}_1 &= z_1 - z_1 + y_2, \\ \hat{z}_2 &= z_2 - z_1 - z_2 + y_2 \end{aligned}$$

which after some rearranging gives

$$\begin{aligned} y_1 &\in A_1 z_1, \\ z_1 - \gamma y_1 + \theta z_2 &\in (\mathrm{Id} + \gamma A_2) y_2, \\ \hat{z}_1 &= y_2, \\ \hat{z}_2 &= y_2 - z_1. \end{aligned}$$

Rewriting the second line as a resolvent and combining the first and second gives

$$\begin{aligned} y_2 &= \mathrm{J}_{\gamma A_2}(z_1 - \gamma A_1 z_1 + \theta z_2), \\ \hat{z}_1 &= y_2, \\ \hat{z}_2 &= y_2 - z_1 \end{aligned}$$

which is exactly the frugal splitting operator above. For the convergence, if we choose

$$Q = \gamma^{-1} \begin{bmatrix} 1-\theta & \theta \\ \theta & |\theta| + \epsilon \end{bmatrix}$$

where $\epsilon > 0$ then

$$\begin{aligned} (I - I_F)(P^T Q - S)U &= (I - I_F)\gamma^{-1} \left(\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1-\theta & \theta \\ \theta & (|\theta| + \epsilon) \end{bmatrix} - \begin{bmatrix} \gamma & 0 \\ 1-\theta & \theta \end{bmatrix} \right) U \\ &= (I - I_F)\gamma^{-1} \left(\begin{bmatrix} 0 & 0 \\ 1-\theta & \theta \end{bmatrix} - \begin{bmatrix} \gamma & 0 \\ (1-\theta) & \theta \end{bmatrix} \right) U \\ &= (I - I_F) \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \\ &= 0. \end{aligned}$$

Furthermore, we have

$$\begin{aligned}
QU + (QU)^T - U^T QU &= \gamma^{-1} \begin{bmatrix} 1-\theta & \theta \\ \theta & |\theta| + \epsilon \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + (QU)^T - U^T QU \\
&= \gamma^{-1} \begin{bmatrix} 1 & \theta \\ \theta + |\theta| + \epsilon & |\theta| + \epsilon \end{bmatrix} + (QU)^T - U^T QU \\
&= \gamma^{-1} \begin{bmatrix} 2 & 2\theta + |\theta| + \epsilon \\ 2\theta + |\theta| + \epsilon & 2(|\theta| + \epsilon) \end{bmatrix} \\
&\quad - \gamma^{-1} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \theta \\ \theta + |\theta| + \epsilon & |\theta| + \epsilon \end{bmatrix} \\
&= \gamma^{-1} \begin{bmatrix} 2 & 2\theta + |\theta| + \epsilon \\ 2\theta + |\theta| + \epsilon & 2(|\theta| + \epsilon) \end{bmatrix} \\
&\quad - \gamma^{-1} \begin{bmatrix} 1 + \theta + |\theta| + \epsilon & \theta + |\theta| + \epsilon \\ \theta + |\theta| + \epsilon & |\theta| + \epsilon \end{bmatrix} \\
&= \gamma^{-1} \begin{bmatrix} 1 - \theta - |\theta| - \epsilon & \theta \\ \theta & |\theta| + \epsilon \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
\tfrac{1}{2} U^T (P^T Q - S)^T B^\dagger (P^T Q - S) U &= \tfrac{1}{2} \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \\
&= \gamma^{-1} \begin{bmatrix} \frac{\gamma}{2\beta_1} & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

which gives

$$\begin{aligned}
W &= QU + (QU)^T - U^T QU - \tfrac{1}{2} U^T (P^T Q - S)^T B^\dagger (P^T Q - S) U \\
&= \gamma^{-1} \begin{bmatrix} 1 - \theta - |\theta| - \epsilon & \theta \\ \theta & |\theta| + \epsilon \end{bmatrix} - \gamma^{-1} \begin{bmatrix} \frac{\gamma}{2\beta_1} & 0 \\ 0 & 0 \end{bmatrix} \\
&= \gamma^{-1} \begin{bmatrix} 1 - \theta - |\theta| - \epsilon - \frac{\gamma}{2\beta_1} & \theta \\ \theta & |\theta| + \epsilon \end{bmatrix}.
\end{aligned}$$

The final condition for convergence is then $Q \succ 0$ and $W \succ 0$ which hold if

$$0 < 1 - \theta - \tfrac{\theta^2}{|\theta| + \epsilon} \quad \text{and} \quad 0 < 1 - \theta - \tfrac{\theta^2}{|\theta| + \epsilon} - |\theta| - \epsilon - \tfrac{\gamma}{2\beta_1}.$$

The first condition is clearly implied by the second since $\epsilon > 0$, $\gamma > 0$ and $\beta_1 > 0$. There exists $\epsilon > 0$ such that the second condition hold if

$$0 < 1 - \theta - 2|\theta| - \tfrac{\gamma}{2\beta_1}.$$

This is the same conditions as was derived in [24].

Forward-Backward with Nesterov-like Momentum

The update of forward-backward with Nesterov-like momentum [4, 26] can be seen as the application of the following frugal splitting operator

$$T_{(A_1,A_2)}(z_1,z_2) = \begin{pmatrix} \hat{z}_1 \\ \hat{z}_2 \end{pmatrix} = \begin{pmatrix} \mathrm{J}_{\gamma A_2}(z_1 + \theta z_2 - \gamma A_1(z_1 + \theta z_2)) \\ \hat{z}_1 - z_1 \end{pmatrix}$$

where $\gamma > 0$, $\theta > 0$ and $A = (A_1, A_2) \in \mathcal{A}_2^{\{1\}}$. As with forward-backward with momentum on the forward step presented earlier, there are other frugal splitting operators that also would yield a Nesterov-like momentum update. To derive a representation, we choose the primal index $p = 2$ and note that the frugal splitting operator can be written as

$$\begin{aligned} y_1 &= A_1(z_1 + \theta z_2), \\ y_2 &= (\mathrm{Id} + \gamma A_2)^{-1}(z_1 + \theta z_2 - \gamma y_1), \\ \hat{z}_1 &= y_2, \\ \hat{z}_2 &= y_2 - z_1. \end{aligned}$$

Inverting A_1 and $\mathrm{Id} + \gamma A_2$ yield

$$\begin{aligned} z_1 + \theta z_2 &\in A_1^{-1} y_1, \\ \gamma^{-1} z_1 + \gamma^{-1}\theta z_2 - y_1 &\in A_2 y_2 + \gamma^{-1} y_2, \\ \hat{z}_1 &= z_1 - z_1 + y_2, \\ \hat{z}_2 &= z_2 - z_1 - z_2 + y_2. \end{aligned}$$

Rearranging so that $\Phi_{(A_1,A_2),2}$ can be identified gives

$$\begin{aligned} z_1 + \theta z_2 &\in [y_2] + [A_1^{-1} y_1 - y_2], \\ \gamma^{-1} z_1 + \gamma^{-1}\theta z_2 &\in [\gamma^{-1} y_2] + [A_2 y_2 + y_1], \\ \hat{z}_1 &= z_1 - [z_1] + [y_2], \\ \hat{z}_2 &= z_2 - [z_1 + z_2] + [y_2] \end{aligned}$$

and we can identify

$$M = \begin{bmatrix} 0 & 1 \\ 0 & \gamma^{-1} \end{bmatrix}, \quad N = \begin{bmatrix} 1 & \theta \\ \gamma^{-1} & \gamma^{-1}\theta \end{bmatrix}, \quad V = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

This representation can be factored as$(2, SUP, SU, U, UP)$ where

$$S = \begin{bmatrix} 1 - \theta & \theta \\ \gamma^{-1}(1 - \theta) & \gamma^{-1}\theta \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

For the convergence conditions we choose

$$Q = \gamma^{-1}\begin{bmatrix} 1-\theta & \theta \\ \theta & a \end{bmatrix}$$

where $a > 0$. This yields

$$\begin{aligned}
0 &= (I - I_F)(P^T Q - S)U \\
&= (I - I_F)\left(\gamma^{-1}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1-\theta & \theta \\ \theta & a \end{bmatrix} - \begin{bmatrix} 1-\theta & \theta \\ \gamma^{-1}(1-\theta) & \gamma^{-1}\theta \end{bmatrix}\right) U \\
&= (I - I_F)\left(\gamma^{-1}\begin{bmatrix} 0 & 0 \\ 1-\theta & \theta \end{bmatrix} - \begin{bmatrix} 1-\theta & \theta \\ \gamma^{-1}(1-\theta) & \gamma^{-1}\theta \end{bmatrix}\right) U \\
&= (I - I_F)\begin{bmatrix} -1+\theta & -\theta \\ 0 & 0 \end{bmatrix} U \\
&= (I - I_F)\begin{bmatrix} -1+\theta & -\theta \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \\
&= (I - I_F)\begin{bmatrix} -1 & -\theta \\ 0 & 0 \end{bmatrix} \\
&= 0
\end{aligned}$$

and

$$\begin{aligned}
QU + (QU)^T - U^T QU &= \gamma^{-1}\begin{bmatrix} 1-\theta & \theta \\ \theta & a \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + (QU)^T - U^T QU \\
&= \gamma^{-1}\begin{bmatrix} 1 & \theta \\ \theta + a & a \end{bmatrix} + (QU)^T - U^T QU \\
&= \gamma^{-1}\begin{bmatrix} 2 & 2\theta + a \\ 2\theta + a & 2a \end{bmatrix} - \gamma^{-1}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & \theta \\ \theta + a & a \end{bmatrix} \\
&= \gamma^{-1}\begin{bmatrix} 2 & 2\theta + a \\ 2\theta + a & 2a \end{bmatrix} - \gamma^{-1}\begin{bmatrix} 1+\theta+a & \theta+a \\ \theta+a & a \end{bmatrix} \\
&= \gamma^{-1}\begin{bmatrix} 1-\theta-a & \theta \\ \theta & a \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
\tfrac{1}{2}U^T(P^T Q - S)^T B^{\dagger}(P^T Q - S)U &= \tfrac{1}{2}\begin{bmatrix} -1 & 0 \\ -\theta & 0 \end{bmatrix}\begin{bmatrix} \beta_1^{-1} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} -1 & -\theta \\ 0 & 0 \end{bmatrix} \\
&= \tfrac{1}{2\beta_1}\begin{bmatrix} -1 & 0 \\ -\theta & 0 \end{bmatrix}\begin{bmatrix} -1 & -\theta \\ 0 & 0 \end{bmatrix} \\
&= \tfrac{1}{2\beta_1}\begin{bmatrix} 1 & \theta \\ \theta & \theta^2 \end{bmatrix}.
\end{aligned}$$

Hence we have

$$\begin{aligned} W &= QU + (QU)^T - U^T QU - \tfrac{1}{2}U^T(P^TQ - S)B^\dagger(P^TQ - S)U \\ &= \gamma^{-1}\begin{bmatrix} 1-\theta-a & \theta \\ \theta & a \end{bmatrix} - \tfrac{1}{2\beta_1}\begin{bmatrix} 1 & \theta \\ \theta & \theta^2 \end{bmatrix} \\ &= \gamma^{-1}\begin{bmatrix} 1-\theta-\hat{\gamma}-a & \theta(1-\hat{\gamma}) \\ \theta(1-\hat{\gamma}) & a-\theta^2\hat{\gamma} \end{bmatrix} \end{aligned}$$

where $\hat{\gamma} = \frac{\gamma}{2\beta_1}$. It can be verified that $Q - W \succ 0$ and the final condition for convergence—$Q \succ 0$ and $W \succ 0$—then holds if $W \succ 0$ which is equivalent to

$$0 < a - \theta^2\hat{\gamma} \quad \text{and} \quad 0 < 1 - \theta - \hat{\gamma} - a - \tfrac{\theta^2(1-\hat{\gamma})^2}{a-\theta^2\hat{\gamma}}.$$

A New Frugal Splitting Operator with Minimal Lifting

We will here derive a frugal splitting operator with minimal lifting that serves as the base of the algorithm in Theorem 8.1. This was also done in the proof of Theorem 8.1 but we will here be more detailed and verify the matrix calculations more carefully.

The frugal splitting operator will be over $\mathcal{A}_n^F$ where $F = \{n-f,\dots,n-1\}$ and $f = |F|$ and for notational convenience we let $R = \{2,\dots,n-f-1\}$ with $r = |R|$. The representation of the new splitting is then (n, M, MK, HMK, HM) where

$$M = \left[\begin{array}{c|c|c|c} 1 & & & 1 \\ \hline \mathbf{1} & \frac{1}{\theta}I & & \mathbf{1} \\ \hline \mathbf{1} & & & \mathbf{1} \\ \hline 1 & & & 1 \end{array}\right] \in \mathbb{R}^{n\times n}, \quad K = \left[\begin{array}{c|c} \frac{1}{2} & \\ \hline & I \\ \hline & \\ \hline \frac{1}{2} & \end{array}\right] \in \mathbb{R}^{n\times(1+r)} \quad \text{and}$$

$$H = \theta\left[\begin{array}{c|c|c|c} \frac{1}{2+f} & & \frac{1}{2+f}\mathbf{1}^T & \frac{1}{2+f} \\ \hline & I & & \end{array}\right] \in \mathbb{R}^{(1+r)\times n}$$

where $\theta > 0$, I is the identity matrix in $\mathbb{R}^{r\times r}$, $\mathbf{1}$ are column vectors of ones with appropriate sizes and empty block denotes zero matrices. Since Proposition 6.5 is satisfied by these matrices, the operator corresponding to this representation is a frugal splitting operator. Let $T_{(\cdot)}\colon \mathcal{H}^{(1+r)} \to \mathcal{H}^{(1+r)}$ be this frugal splitting operator and consider the evaluation of $\hat{z} = T_A z$ where $A = (A_1,\dots,A_n) \in \mathcal{A}_n^F$. This can be written as

$$\begin{aligned} y &= (M + \Phi_{A,n})^{-1}MKz \\ \hat{z} &= z - HMKz + HMy. \end{aligned}$$

Using the definition of the inverse of $M + \Phi_{A,n}$ and writing out the matrices explic-

itly gives

$$\left[\begin{array}{c|c} 1 & \\ \hline \mathbf{1} & \frac{1}{\theta} I \\ \hline \mathbf{1} & \\ \hline 1 & \end{array}\right] z \in \left[\begin{array}{c|c|c|c} 1 & & & \\ \hline \mathbf{1} & \frac{1}{\theta} I & & \\ \hline \mathbf{1} & & & \\ \hline 2 & \mathbf{1} & \mathbf{1} & 1 \end{array}\right] y + \begin{bmatrix} A_1^{-1} & & & \\ & \ddots & & \\ & & A_{n-1}^{-1} & \\ & & & A_n \end{bmatrix} y$$

$$\hat{z} = z - \left[\begin{array}{c|c} \theta & \\ \hline \theta\mathbf{1} & I \end{array}\right] z + \theta \left[\begin{array}{c|c|c|c} 1 & & & 1 \\ \hline \mathbf{1} & \frac{1}{\theta} I & & \mathbf{1} \end{array}\right] y$$

Setting $z = (z_1, \ldots, z_{1+r})$, $\hat{z} = (\hat{z}_1, \ldots, \hat{z}_{1+r})$, $y = (y_1, \ldots, y_n)$ and writing out the corresponding equations for these matrix/operator expressions gives

$$\begin{aligned}
z_1 &\in (\mathrm{Id} + A_1^{-1}) y_1, \\
z_1 + \tfrac{1}{\theta} z_i &\in y_1 + (\tfrac{1}{\theta}\,\mathrm{Id} + A_i^{-1}) y_i && \text{for all } i \in R, \\
z_1 &\in y_1 + A_i^{-1} y_i && \text{for all } i \in F, \\
z_1 &\in y_1 + \textstyle\sum_{j=1}^{n-1} y_j + (\mathrm{Id} + A_n) y_n, \\
\hat{z}_1 &= (1-\theta) z_1 + \theta(y_1 + y_n), \\
\hat{z}_i &= -\theta z_1 + \theta(y_1 + y_n + \tfrac{1}{\theta} y_i) && \text{for all } i \in R.
\end{aligned}$$

Rearranging each line yields

$$\begin{aligned}
y_1 &= (\mathrm{Id} + A_1^{-1})^{-1} z_1, \\
y_i &= (\mathrm{Id} + \theta A_i^{-1})^{-1} (\theta[z_1 - y_1] + z_i) && \text{for all } i \in R, \\
y_i &= A_i (z_1 - y_1) && \text{for all } i \in F, \\
y_n &= (\mathrm{Id} + A_n)^{-1} ([z_1 - y_1] - \textstyle\sum_{j=1}^{n-1} y_j), \\
\hat{z}_1 &= z_1 - \theta[z_1 - y_1] + \theta y_n, \\
\hat{z}_i &= y_i - \theta[z_1 - y_1] + \theta y_n && \text{for all } i \in R.
\end{aligned}$$

Applying the Moreau identity to the first two lines and introducing variables x_i for

all $x \in \{1,\dots,n\}$ that contains the results of all primal evaluations yields

$$\begin{aligned}
x_1 &= (\mathrm{Id}+A_1)^{-1} z_1, \\
x_i &= (\mathrm{Id}+\theta^{-1}A_i)^{-1}([z_1 - y_1] + \theta^{-1} z_i) && \text{for all } i \in R, \\
x_i &= A_i(z_1 - y_1) && \text{for all } i \in F, \\
x_n &= (\mathrm{Id}+A_n)^{-1}([z_1 - y_1] - \textstyle\sum_{j=1}^{n-1} y_j), \\
y_1 &= z_1 - x_1, \\
y_i &= \theta[z_1 - y_1] + z_i - \theta x_i = z_i + \theta(x_1 - x_i) && \text{for all } i \in R, \\
y_i &= x_i && \text{for all } i \in F, \\
y_n &= x_n, \\
\hat{z}_1 &= z_1 - \theta[z_1 - y_1] + \theta y_n, \\
\hat{z}_i &= y_i - \theta[z_1 - y_1] + \theta y_n && \text{for all } i \in R.
\end{aligned}$$

Eliminate the y_i variables gives

$$\begin{aligned}
x_1 &= (\mathrm{Id}+A_1)^{-1} z_1, \\
x_i &= (\mathrm{Id}+\theta^{-1}A_i)^{-1}(x_1 + \theta^{-1} z_i) && \text{for all } i \in R, \\
x_i &= A_i x_1 && \text{for all } i \in F, \\
\bar{x} &= \textstyle\sum_{j\in F} x_j + \sum_{j\in R}(z_j + \theta(x_1 - x_j)), \\
x_n &= (\mathrm{Id}+A_n)^{-1}(2x_1 - z_1 - \bar{x}), \\
\hat{z}_1 &= z_1 - \theta(x_1 - x_n), \\
\hat{z}_i &= z_i - \theta(x_i - x_n) && \text{for all } i \in R.
\end{aligned}$$

This concludes the derivations of the frugal splitting operator. To derive conditions for convergence we need a representation of the form (n, SUP, SU, U, UP) and since $U = HMK$ is invertible this is easily calculated as

$$U = HMK = \left[\begin{array}{c|c} \theta & \\ \hline \theta\mathbf{1} & I \end{array}\right] \in \mathbb{R}^{(1+r)\times(1+r)}, \quad U^{-1} = \left[\begin{array}{c|c} \theta^{-1} & \\ \hline -\mathbf{1} & I \end{array}\right] \in \mathbb{R}^{(1+r)\times(1+r)},$$

$$P = U^{-1}HM = \left[\begin{array}{c|c|c} 1 & & 1 \\ \hline & I & \\ \hline & & \end{array}\right] \in \mathbb{R}^{(1+r)\times n}, \quad S = MKU^{-1} = \theta^{-1}\left[\begin{array}{c|c} 1 & \\ \hline & I \\ \hline \mathbf{1} & \\ \hline 1 & \end{array}\right] \in \mathbb{R}^{n\times(1+r)}$$

Choosing

$$Q = \theta^{-1}\left[\begin{array}{c|c} 1 & \\ \hline & I \end{array}\right] \in \mathbb{R}^{(1+r)\times(1+r)}$$

then results in

$$
\begin{aligned}
(I-I_F)(P^TQ-S)U &= (I-I_F)\left(\theta^{-1}\left[\begin{array}{c|c} 1 & \\ \hline & I \\ \hline 1 & \end{array}\right] - \theta^{-1}\left[\begin{array}{c|c} 1 & \\ \hline & I \\ \hline \mathbf{1} & \\ \hline 1 & \end{array}\right]\right)U \\
&= (I-I_F)\theta^{-1}\left[\begin{array}{c|c} & \\ \hline & \\ \hline -\mathbf{1} & \\ \hline & \end{array}\right]\left[\begin{array}{c|c} \theta & \\ \hline \theta\mathbf{1} & I \end{array}\right] \\
&= (I-I_F)\left[\begin{array}{c|c} & \\ \hline & \\ \hline -\mathbf{1} & \\ \hline & \end{array}\right] \\
&= 0.
\end{aligned}
$$

We further have

$$
\begin{aligned}
&QU+(QU)^T-U^TQU \\
&\quad= \theta^{-1}(U+U^T-U^TU) \\
&\quad= \theta^{-1}\left(\left[\begin{array}{c|c} \theta & \\ \hline \theta\mathbf{1} & I \end{array}\right] + \left[\begin{array}{c|c} \theta & \theta\mathbf{1}^T \\ \hline & I \end{array}\right] - \left[\begin{array}{c|c} \theta & \theta\mathbf{1}^T \\ \hline & I \end{array}\right]\left[\begin{array}{c|c} \theta & \\ \hline \theta\mathbf{1} & I \end{array}\right]\right) \\
&\quad= \theta^{-1}\left(\left[\begin{array}{c|c} 2\theta & \theta\mathbf{1}^T \\ \hline \theta\mathbf{1} & 2I \end{array}\right] - \left[\begin{array}{c|c} \theta^2(n-1-f) & \theta\mathbf{1}^T \\ \hline \theta\mathbf{1} & I \end{array}\right]\right) \\
&\quad= \left[\begin{array}{c|c} 2 & \mathbf{1}^T \\ \hline \mathbf{1} & 2\theta^{-1}I \end{array}\right] - \left[\begin{array}{c|c} \theta(1+r) & \mathbf{1}^T \\ \hline \mathbf{1} & \theta^{-1}I \end{array}\right] \\
&\quad= \left[\begin{array}{c|c} 2-\theta(1+r) & \\ \hline & \theta^{-1}I \end{array}\right]
\end{aligned}
$$

and

$$
\begin{aligned}
\tfrac{1}{2}U^T(P^TQ-S)B^\dagger(P^TQ-S)U &= \tfrac{1}{2}\left[\begin{array}{c|c|c|c} & & -\mathbf{1}^T & \\ \hline & & & \end{array}\right]B^\dagger\left[\begin{array}{c|c} & \\ \hline & \\ \hline -\mathbf{1} & \\ \hline & \end{array}\right] \\
&= \tfrac{1}{2}\left[\begin{array}{c|c} \sum_{i\in F}\beta_i^{-1} & \\ \hline & \end{array}\right]
\end{aligned}
$$

which, since $1 + r = n - 1 - f$, results in

$$\begin{aligned} W &= QU + (QU)^T - U^T QU - \tfrac{1}{2} U^T (P^T Q - S) B^\dagger (P^T Q - S) U \\ &= \left[\begin{array}{c|c} 2 - \theta(1+r) & \\ \hline & \frac{1}{\theta} I \end{array}\right] - \tfrac{1}{2} \left[\begin{array}{c|c} \sum_{i \in F} \beta_i^{-1} & \\ \hline & \end{array}\right] \\ &= \left[\begin{array}{c|c} 2 - \theta(n-1-f) - \frac{1}{2} \sum_{i \in F} \beta_i^{-1} & \\ \hline & \frac{1}{\theta} I \end{array}\right]. \end{aligned}$$

We have $Q \succ 0$ and $W \succ 0$ if $\theta > 0$ and

$$0 < 2 - \theta(n-1-f) - \tfrac{1}{2} \sum\nolimits_{i \in F} \beta_i^{-1}.$$

Minimal Lifting Method of Malitsky–Tam

Malitsky and Tam [23] presented a frugal splitting operator over $\mathcal{A}_n$ with minimal lifting, $(\hat{z}_1, \ldots, \hat{z}_{n-1}) = T_A(z_1, \ldots, z_{n-1})$ where

$$\begin{aligned} x_1 &= \mathrm{J}_{\gamma A_1}(z_1), \\ x_i &= \mathrm{J}_{\gamma A_i}(z_i - z_{i-1} + x_{i-1}) && \text{for all } i \in \{2, \ldots, n-1\}, \\ x_n &= \mathrm{J}_{\gamma A_n}(-z_{n-1} + x_1 + x_{n-1}), \\ \hat{z}_i &= z_i + \theta(x_{i+1} - x_i) && \text{for all } i \in \{1, \ldots, n-1\}. \end{aligned}$$

Similarly to the derivation of our new method with minimal lifting, we derive a representation of this frugal splitting operator without any step-sizes, i.e., we set $\gamma = 1$ in the expressions above. As was done in Theorem 8.1, it is straightforward to modify the resulting convergence conditions to include step-sizes. In fact, since this frugal splitting operator has no forward evaluations, the convergence conditions for fixed point iteration will not depend on the step-size. To start, we select primal index $p = n$ and apply the Moreau identity to the $n - 1$ first resolvents which gives

$$\begin{aligned} y_1 &= \mathrm{J}_{A_1^{-1}} z_1, \\ y_i &= \mathrm{J}_{A_i^{-1}}(z_i - z_{i-1} + x_{i-1}) && \text{for all } i \in \{2, \ldots, n-1\}, \\ y_n &= \mathrm{J}_{A_n}(-z_{n-1} + x_1 + x_{n-1}), \\ x_1 &= z_1 - y_1, \\ x_i &= z_i - z_{i-1} + x_{i-1} - y_i && \text{for all } i \in \{2, \ldots, n-1\}, \\ x_n &= y_n, \\ \hat{z}_i &= z_i + \theta(x_{i+1} - x_i) && \text{for all } i \in \{1, \ldots, n-1\}. \end{aligned}$$

Looking at the expression of x_i for $i \in \{2,\dots,n-1\}$ we see

$$\begin{aligned} x_2 &= z_2 - z_1 + x_1 - y_2 = z_2 - y_1 - y_2 \\ x_3 &= z_3 - z_2 + x_2 - y_3 = z_3 - y_1 - y_2 - y_3 \\ &\vdots \\ x_i &= z_i - z_{i-1} + x_{i-1} - y_i = z_i - \sum_{j=1}^{i} y_j \end{aligned}$$

which gives

$$\hat{z}_i = z_i + \theta((z_{i+1} - \sum_{j=1}^{i+1} y_j) - (z_i - \sum_{j=1}^{i} y_j)) = z_i + \theta(z_{i+1} - z_i - y_{i+1})$$

for all $i \in \{1,\dots,n-2\}$ and

$$\hat{z}_{n-1} = z_{n-1} + \theta(y_n - (z_{n-1} - \sum_{j=1}^{n-1} y_j)) = z_{n-1} + \theta(-z_{n-1} + \sum_{j=1}^{n} y_j).$$

Inserting these expressions back in gives

$$\begin{aligned} y_1 &= \mathrm{J}_{A_1^{-1}}(z_1), \\ y_i &= \mathrm{J}_{A_i^{-1}}(z_i - \sum_{j=1}^{i-1} y_j) && \text{for all } i \in \{2,\dots,n-1\}, \\ y_n &= \mathrm{J}_{A_n}(z_1 - y_1 - \sum_{j=1}^{n-1} y_j), \\ \hat{z}_i &= z_i - \theta(z_i - z_{i+1}) + \theta(-y_{i+1}) && \text{for all } i \in \{1,\dots,n-2\}, \\ \hat{z}_{n-1} &= z_{n-1} - \theta z_{n-1} + \theta \sum_{j=1}^{n} y_j. \end{aligned}$$

From this we can identify the representation (n, M, N, U, V) as

$$M = \begin{bmatrix} 1 & & & & 1 \\ 1 & 1 & & & 1 \\ \vdots & \ddots & \ddots & & \vdots \\ 1 & \cdots & 1 & 1 & 1 \\ 1 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ \vdots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

and

$$U = \theta \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}, \quad V = \theta \begin{bmatrix} 0 & -1 & & & & \\ & 0 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & 0 & -1 & \\ 1 & 1 & \cdots & 1 & 1 & 1 \end{bmatrix}$$

where $M \in \mathbb{R}^{n\times n}$, $N \in \mathbb{R}^{n\times(n-1)}$, $U \in \mathbb{R}^{(n-1)\times(n-1)}$ and $V \in \mathbb{R}^{(n-1)\times n}$.

Since U is invertible a factorization of the form (n, SUP, SU, U, UP) must satisfy

$$S = NU^{-1} \quad \text{and} \quad P = U^{-1}V$$

which gives

$$S = \theta^{-1}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ & 1 & \ddots & 1 \\ & & \ddots & \vdots \\ & & & 1 \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{n\times(n-1)} \quad \text{and} \quad P = \begin{bmatrix} 1 & & & & 1 \\ 1 & 1 & & & 1 \\ \vdots & \ddots & \ddots & & \vdots \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}.$$

The convergence conditions are then satisfied by $Q = \theta^{-1}I$ since

$$(I - I_F)(P^TQ - S)U = (S - S)U = 0$$

and

$$\begin{aligned} W &= QU + (QU)^T - U^TQU - \tfrac{1}{2}U^T(P^TQ - S)B^{\dagger}(P^TQ - S)U \\ &= QU + (QU)^T - U^TQU \\ &= \theta^{-1}(U + U^T - U^TU) \\ &= \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} - \theta \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & -1 & & \\ & 1 & \ddots & \\ & & \ddots & -1 \\ & & & 1 \end{bmatrix} \\ &= \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} - \theta \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \\ &= (1-\theta)\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} + \theta \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}. \end{aligned}$$

The tridiagonal matrix is Toeplitz and has eigenvalues $2 + 2\cos(\frac{k\pi}{n})$ for $k \in \{1,\ldots,n-1\}$, see for instance [14], and hence $Q \succ 0$ and $W \succ 0$ for all $0 < \theta < 1$.

Minimal Lifting Method of Ryu

Ryu [29] presented a frugal splitting operator over $\mathcal{A}_3$ with minimal lifting, i.e., the lifting number is two. Let $T_{(\cdot)}\colon \mathcal{H}^2 \to \mathcal{H}^2$ be this frugal splitting operator and let $A = (A_1, A_2, A_3) \in \mathcal{A}_3$. The evaluation $(\hat{z}_1, \hat{z}_2) = T_A(z_1, z_2)$ is defined as

$$\begin{aligned}
x_1 &= \mathrm{J}_{A_1}(z_1),\\
x_2 &= \mathrm{J}_{A_2}(z_2 + x_1),\\
x_3 &= \mathrm{J}_{A_3}(-z_1 - z_2 + x_1 + x_2),\\
\hat{z}_1 &= z_1 + \theta(x_3 - x_1),\\
\hat{z}_2 &= z_2 + \theta(x_3 - x_2).
\end{aligned}$$

To derive a representation of this we select the primal index $p = 3$ and apply the Moreau identity to the first two resolvents,

$$\begin{aligned}
y_1 &= \mathrm{J}_{A_1^{-1}}(z_1),\\
y_2 &= \mathrm{J}_{A_2^{-1}}(z_1 + z_2 - y_1),\\
y_3 &= \mathrm{J}_{A_3}(z_1 - 2y_1 - y_2),\\
\hat{z}_1 &= z_1 - \theta z_1 + \theta(y_1 + y_3),\\
\hat{z}_2 &= z_2 - \theta(z_1 + z_2) + \theta(y_1 + y_2 + y_3).
\end{aligned}$$

From this we can identify

$$M = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad U = \theta \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad V = \theta \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

and $(3, M, N, U, V)$ is then a representation of $T_{(\cdot)}$. A factorization of the form $(3, SUP, SU, U, UP)$ needed for the convergence analysis is easily found since U is invertible and is given by

$$S = \theta^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Choosing $Q = \theta^{-1} I$ yields

$$(I - I_F)(P^T Q - S)U = (P^T Q - S)U = (S - S)U = 0$$

and

$$\begin{aligned} W &= QU + (QU)^T - U^T QU - \tfrac{1}{2}U^T(P^T Q - S)^T B^\dagger (P^T Q - S)U \\ &= \theta^{-1}(U + U^T - U^T U) \\ &= \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \theta \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \\ &= (1-\theta)\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} + \theta \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

and it is clear that the final conditions for convergence, $Q \succ 0$ and $W \succ 0$, hold if $0 < \theta < 1$.

Minimal Lifting Method of Condat *et al.* and Campoy

Condat *et al.* presented a product space technique for reformulating finite sum convex optimization problems as convex problems that can be handled with traditional splitting techniques [11, Parallel versions of the algorithms. Technique 1]. Campoy presented in [7] an similar product space approach but for finite sum monotone inclusion problems over $\mathcal{A}_n$. Applying the Douglas–Rachford splitting operator to these reformulations yields a frugal resolvent splitting operator for the original finite sum problems and, since the product space reformulations have a lifting of $n-1$, so does the splitting operator and hence it has minimal lifting. The frugal splitting operator $T_{(\cdot)}\colon \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ over $\mathcal{A}_n^F$ resulting from Campoy's approach is for $A = (A_1, \ldots, A_n) \in \mathcal{A}_n$ defined as $(\hat{z}_1, \ldots, \hat{z}_{n-1}) = T_A(z_1, \ldots, z_{n-1})$ such that

$$\begin{aligned} x_1 &= \mathrm{J}_{\frac{\gamma}{n-1}A_1}\left(\tfrac{1}{n-1}\textstyle\sum_{j=1}^{n-1} z_j\right) && \\ x_i &= \mathrm{J}_{\gamma A_i}(2x_1 - z_{i-1}) && \text{for all } i \in \{2, \ldots, n\} \\ \hat{z}_i &= z_i + \theta(x_{i+1} - x_1) && \text{for all } i \in \{1, \ldots, n-1\}. \end{aligned}$$

The splitting operator of Condat *et al.* is essentially the same but with a weighted average instead of the arithmetic average being used in the first row. Below we derive a representation and convergence conditions for Campoy's splitting operator but a similar representation for the splitting of Condat *et al.* can be analogously derived.

We choose the primal index as $p = n$ and apply Moreau's identity to all other

resolvents

$$
\begin{aligned}
y_1 &= \mathrm{J}_{\frac{n-1}{\gamma}A_1^{-1}}\left(\tfrac{n-1}{\gamma}\tfrac{1}{n-1}\textstyle\sum_{j=1}^{n-1} z_j\right) \\
y_i &= \mathrm{J}_{\gamma^{-1}A_i^{-1}}(\gamma^{-1}(2x_1 - z_{i-1})) && \text{for all } i \in \{2,\dots,n-1\} \\
y_n &= \mathrm{J}_{\gamma A_n}(2x_1 - z_{n-1}) \\
x_1 &= \tfrac{1}{n-1}\textstyle\sum_{j=1}^{n-1} z_j - \tfrac{\gamma}{n-1} y_1 \\
x_i &= 2x_1 - z_{i-1} - \gamma y_i && \text{for all } i \in \{2,\dots,n-1\} \\
x_n &= y_n \\
\hat{z}_i &= z_i + \theta(x_{i+1} - x_1) && \text{for all } i \in \{1,\dots,n-1\}.
\end{aligned}
$$

Eliminating the x_i variables and rewriting the resolvents yields

$$
\begin{aligned}
y_1 &= \left(\tfrac{\gamma}{n-1}\,\mathrm{Id} + A_1^{-1}\right)^{-1}\left(\tfrac{1}{n-1}\textstyle\sum_{j=1}^{n-1} z_j\right) \\
y_i &= \left(\gamma\,\mathrm{Id} + A_i^{-1}\right)^{-1}\left(\tfrac{2}{n-1}\textstyle\sum_{j=1}^{n-1} z_j - z_{i-1} - \tfrac{2\gamma}{n-1} y_1\right) && \text{for all } i \in \{2,\dots,n-1\} \\
y_n &= \left(\gamma^{-1}\,\mathrm{Id} + A_n\right)^{-1}\left(\tfrac{2\gamma^{-1}}{n-1}\textstyle\sum_{j=1}^{n-1} z_j - \gamma^{-1} z_{n-1} - \tfrac{2}{n-1} y_1\right) \\
\hat{z}_i &= z_i - \theta\left(z_i - \tfrac{1}{n-1}\textstyle\sum_{j=1}^{n-1} z_j\right) + \theta\left(-\tfrac{\gamma}{n-1} y_1 - \gamma y_{i+1}\right) && \text{for all } i \in \{1,\dots,n-2\} \\
\hat{z}_{n-1} &= z_i - \theta\left(\tfrac{1}{n-1}\textstyle\sum_{j=1}^{n-1} z_j\right) + \theta\left(\tfrac{\gamma}{n-1} y_1 + y_n\right).
\end{aligned}
$$

From this the matrices in the representation (n, M, N, U, V) can be identified as

$$
M = \begin{bmatrix}
\frac{\gamma}{n-1} & & & & 1 \\
\frac{2\gamma}{n-1} & \gamma & & & 1 \\
\vdots & & \ddots & & \vdots \\
\frac{2\gamma}{n-1} & & & \gamma & 1 \\
\frac{3-n}{n-1} & -1 & \cdots & -1 & \gamma^{-1}
\end{bmatrix},
$$

$$
N = \frac{1}{n-1}\begin{bmatrix}
1 & \cdots & 1 & 1 & 1 \\
3-n & 2 & \cdots & 2 & 2 \\
2 & 3-n & 2 & \cdots & 2 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
2 & \cdots & 2 & 3-n & 2 \\
\frac{2}{\gamma} & \frac{2}{\gamma} & \cdots & \frac{2}{\gamma} & \frac{3-n}{\gamma}
\end{bmatrix},
$$

$$U = \frac{\theta}{n-1}\begin{bmatrix} n-2 & -1 & \dots & -1 & -1 \\ -1 & n-2 & -1 & \dots & -1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -1 & \dots & -1 & n-2 & -1 \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix}$$

and

$$V = \frac{\theta\gamma}{n-1}\begin{bmatrix} -1 & 1-n & & & & \\ -1 & & 1-n & & & \\ \vdots & & & \ddots & & \\ -1 & & & & 1-n & \\ 1 & & & & & \frac{n-1}{\gamma} \end{bmatrix}$$

where $M \in \mathbb{R}^{n\times n}$, $N \in \mathbb{R}^{n\times(n-1)}$, $U \in \mathbb{R}^{(n-1)\times(n-1)}$ and $V \in \mathbb{R}^{(n-1)\times n}$. Notice that we have

$$U^{-1} = \frac{1}{\theta}\begin{bmatrix} 1 & & & 1 \\ & \ddots & & \vdots \\ & & 1 & 1 \\ -1 & \cdots & -1 & 1 \end{bmatrix}$$

which makes it possible to factor the representation (n,M,N,U,V) as (n,SUP,SU,U,UP) where

$$S = NU^{-1} = \frac{1}{\theta}\begin{bmatrix} 0 & & & & 1 \\ -1 & 0 & & & 1 \\ & -1 & \ddots & & \vdots \\ & & \ddots & 0 & 1 \\ & & & -1 & 1 \\ \gamma^{-1} & \gamma^{-1} & \cdots & \gamma^{-1} & \gamma^{-1} \end{bmatrix} \in \mathbb{R}^{n\times(n-1)}$$

and

$$P = U^{-1}V = \gamma\begin{bmatrix} 0 & -1 & & & & \gamma^{-1} \\ & 0 & -1 & & & \gamma^{-1} \\ & & \ddots & \ddots & & \vdots \\ & & & 0 & -1 & \gamma^{-1} \\ 1 & 1 & \cdots & 1 & 1 & \gamma^{-1} \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}.$$

Notice that $P = \gamma S^T$.

For the convergence theorem we choose $Q = \gamma I$ where $I \in \mathbb{R}^{(n-1)\times(n-1)}$ is the identity matrix. We then have $P^T Q = S$ and $Q \succ 0$ for all $\gamma > 0$, hence, it is enough to show that $W \succ 0$.

$$
\begin{aligned}
W &= UQ + (UQ)^T - U^T QU \\
&= \gamma^{-1}(U + U^T - U^T U) \\
&= \gamma^{-1}((\frac{2\theta}{n-1}\begin{bmatrix} n-2 & -1 & \dots & -1 & 0 \\ -1 & n-2 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -1 & \dots & -1 & n-2 & \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} - U^T U) \\
&= \gamma^{-1}(\frac{2\theta}{n-1}\begin{bmatrix} n-2 & -1 & \dots & -1 & 0 \\ -1 & n-2 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -1 & \dots & -1 & n-2 & \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \\
&\quad - \frac{\theta^2}{n-1}\begin{bmatrix} n-2 & -1 & \dots & -1 & 0 \\ -1 & n-2 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -1 & \dots & -1 & n-2 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}) \\
&= \frac{\theta(2-\theta)}{\gamma(n-1)}\begin{bmatrix} n-2 & -1 & \dots & -1 & 0 \\ -1 & n-2 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -1 & \dots & -1 & n-2 & \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}
\end{aligned}
$$

and we see that $W \succ 0$ for all $\theta \in (0,2)$ and all $\gamma > 0$.

Primal-Dual method of Chambolle–Pock

The primal-dual method made popular by Chambolle–Pock [8] can be seen as a fixed point iteration of the following frugal splitting operator over $\mathcal{A}_2$,

$$
T_{(A_1,A_2)}(z_1,z_2) = \begin{pmatrix} \hat{z}_1 \\ \hat{z}_2 \end{pmatrix} = \begin{pmatrix} \mathrm{J}_{\tau A_1}(z_1 - \tau z_2) \\ \mathrm{J}_{\sigma A_2^{-1}}(z_2 + \sigma(2\hat{z}_1 - x_1)) \end{pmatrix}.
$$

Traditionally, this primal-dual method allows for composition with a linear operator in the monotone inclusion problem but here we assume the linear operator is the identity operator. Since our generalized primal-dual resolvent also makes use of a primal-dual formulation, deriving a representation of this frugal splitting operator

is particularly easy. First we choose the primal index to $p = 1$ since we already have the first resolvent in primal form and the second in dual form. Introduce some intermediate variables

$$\begin{aligned} y_1 &= \mathrm{J}_{\tau A_1}(z_1 - \tau z_2), \\ y_2 &= \mathrm{J}_{\sigma A_2^{-1}}(z_2 - \sigma z_1 + 2\sigma y_1), \\ \hat{z}_1 &= y_1, \\ \hat{z}_2 &= y_2. \end{aligned}$$

Use the definition of a resolvent,

$$\begin{aligned} z_1 - \tau z_2 &\in y_1 + \tau A_1 y_1, \\ z_2 - \sigma z_1 + 2\sigma y_1 &\in y_2 + \sigma A_2^{-1} y_2, \\ \hat{z}_1 &= y_1, \\ \hat{z}_2 &= y_2, \end{aligned}$$

and rearrange to identify $\Phi_{A,1}$

$$\begin{aligned} \tau^{-1} z_1 - z_2 &\in [\tau^{-1} y_1 - y_2] + [A_1 y_1 + y_2], \\ \sigma^{-1} z_2 - z_1 &\in [-y_1 + \sigma^{-1} y_2] + [A_2^{-1} y_2 - y_1], \\ \hat{z}_1 &= z_1 - [z_1] + [y_1], \\ \hat{z}_2 &= z_2 - [z_2] + [y_2]. \end{aligned}$$

From this we can identify

$$M = \begin{bmatrix} \tau^{-1} & -1 \\ -1 & \sigma^{-1} \end{bmatrix}, \quad N = \begin{bmatrix} \tau^{-1} & -1 \\ -1 & \sigma^{-1} \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

This representation can be factorized as $(1, SUP, SU, U, UP)$ where

$$S = \begin{bmatrix} \tau^{-1} & -1 \\ -1 & \sigma^{-1} \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

If we choose $Q = S$ we see that $Q \succ 0$ if $\sigma\tau < 1$ and

$$(I - I_F)(P^T Q - S)U = P^T Q - S = Q - S = 0$$

and

$$\begin{aligned} W &= QU + (QU)^T - U^T QU - \tfrac{1}{2} U^T (P^T Q - S)^T B^\dagger (P^T Q - S)U \\ &= Q + Q - Q \\ &= S. \end{aligned}$$

The convergence conditions with this choice of Q is then $Q \succ 0$ and $W \succ 0$ which hold as long as $S \succ 0$, i.e., as long as $\sigma\tau < 1$.

Projective Splitting

In [16] it was noted that the update of a synchronous version of projective splitting [9] can be seen as a generalized primal-dual resolvent over $\mathcal{A}_n$ with representation $(n, M, M, \theta M, \theta M)$ where $\theta > 0$ and

$$M = \left[\begin{array}{c|c} \mathcal{T} & \mathbf{1} \\ \hline -\mathbf{1}^T & \tau_n^{-1} \end{array}\right] \in \mathbb{R}^{n\times n},$$

$\mathbf{1} \in \mathbb{R}^{n-1}$ is the vector of all ones, $\mathcal{T} \in \mathbb{R}^{(n-1)\times(n-1)}$ is a diagonal matrix with diagonal elements $\mathcal{T}_{i,i} = \tau_i$ for all $i \in \{1,\dots,n-1\}$ and $\tau_i > 0$ for all $i \in \{1,\dots,n\}$. In the projective splitting method the actual value of θ in each iteration is calculated based on the results of all resolvent evaluations but it is quite clear that the operator given by $(n, M, M, \theta M, \theta M)$ is a frugal splitting operator and we will show that fixed point iterations with a fixed θ converge for sufficiently small θ. With $T_{(\cdot)} \colon \mathcal{H}^n \to \mathcal{H}^n$ being this frugal splitting operator, the evaluation of $\hat{z} = T_A z$ for some $z \in \mathcal{H}^n$ and $A \in \mathcal{A}_n$ can be written as

$$\begin{aligned} Mz &\in My + \Phi_{A,n} y = (M + \Gamma_n)y + \Delta_{A,n} y \\ \hat{z} &= z - \theta M(z - y) = (I - \theta M)z + \theta M y \end{aligned}$$

which explicitly becomes

$$\left[\begin{array}{c|c} \mathcal{T} & \mathbf{1} \\ \hline -\mathbf{1}^T & \tau_n^{-1} \end{array}\right] z \in \left(\left[\begin{array}{c|c} \mathcal{T} & \\ \hline & \tau_n^{-1} \end{array}\right] + \left[\begin{array}{ccc|c} A_1^{-1} & & & \\ & \ddots & & \\ & & A_{n-1}^{-1} & \\ \hline & & & A_n \end{array}\right] \right) y$$

$$\hat{z} = z - \theta \left[\begin{array}{c|c} \mathcal{T} & \mathbf{1} \\ \hline -\mathbf{1}^T & \tau_n^{-1} \end{array}\right] z + \theta \left[\begin{array}{c|c} \mathcal{T} & \mathbf{1} \\ \hline -\mathbf{1}^T & \tau_n^{-1} \end{array}\right] y.$$

Setting $z = (z_1,\dots,z_n)$, $\hat{z} = (z_1,\dots,z_n)$, $y = (y_1,\dots,y_n)$ and writing out each line gives

$$\begin{aligned} \tau_i z_i + z_n &\in (\tau_i \,\mathrm{Id} + A_i^{-1}) y_i \quad \text{for all } i \in \{1,\dots,n-1\}, \\ \tau_n^{-1} z_n - \sum\nolimits_{j=1}^{n-1} z_j &\in (\tau_n^{-1} \,\mathrm{Id} + A_n) y_n, \\ \hat{z}_i &= z_i - \theta(\tau_i z_i + z_n) + \theta(\tau_i y_i + y_n) \quad \text{for all } i \in \{1,\dots,n-1\}, \\ \hat{z}_n &= z_n - \theta(\tau_n^{-1} z_n - \sum\nolimits_{j=1}^{n-1} z_j) + \theta(\tau_n^{-1} y_n - \sum\nolimits_{j=1}^{n-1} y_j). \end{aligned}$$

Rewriting these equations using resolvents gives

$$
\begin{aligned}
y_i &= \mathrm{J}_{\tau_i^{-1} A_i^{-1}}(z_i + \tau_i^{-1} z_n) && \text{for all } i \in \{1,\dots,n-1\},\\
y_n &= \mathrm{J}_{\tau_n A_n}(z_n - \tau_n \textstyle\sum_{j=1}^{n-1} z_j),\\
\hat{z}_i &= z_i - \theta(\tau_i z_i + z_n) + \theta(\tau_i y_i + y_n) && \text{for all } i \in \{1,\dots,n-1\},\\
\hat{z}_n &= z_n - \theta(\tau_n^{-1} z_n - \textstyle\sum_{j=1}^{n-1} z_j) + \theta(\tau_n^{-1} y_n - \textstyle\sum_{j=1}^{n-1} y_j).
\end{aligned}
$$

Applying the Moreau identity and introducing variables for the primal evaluation yields

$$
\begin{aligned}
x_i &= \mathrm{J}_{\tau_i A_i}(\tau_i z_i + z_n) && \text{for all } i \in \{1,\dots,n-1\},\\
x_n &= \mathrm{J}_{\tau_n A_n}(z_n - \tau_n \textstyle\sum_{j=1}^{n-1} z_j),\\
y_i &= z_i + \tau_i^{-1} z_n - \tau_i^{-1} x_i && \text{for all } i \in \{1,\dots,n-1\},\\
y_n &= x_n,\\
\hat{z}_i &= z_i - \theta(\tau_i z_i + z_n) + \theta(\tau_i y_i + y_n) && \text{for all } i \in \{1,\dots,n-1\},\\
\hat{z}_n &= z_n - \theta(\tau_n^{-1} z_n - \textstyle\sum_{j=1}^{n-1} z_j) + \theta(\tau_n^{-1} y_n - \textstyle\sum_{j=1}^{n-1} y_j).
\end{aligned}
$$

Eliminating y_i for all $i \in \{1,\dots,n\}$ yields

$$
\begin{aligned}
x_i &= \mathrm{J}_{\tau_i A_i}(\tau_i z_i + z_n) && \text{for all } i \in \{1,\dots,n-1\},\\
x_n &= \mathrm{J}_{\tau_n A_n}(z_n - \tau_n \textstyle\sum_{j=1}^{n-1} z_j),\\
\hat{z}_i &= z_i - \theta(x_i - x_n) && \text{for all } i \in \{1,\dots,n-1\},\\
\hat{z}_n &= z_n - \theta(\tau_n^{-1} - \textstyle\sum_{j=1}^{n-1} \tau_i^{-1}) z_n + \theta(\tau_n^{-1} x_n + \textstyle\sum_{j=1}^{n-1} \tau_i^{-1} x_j).
\end{aligned}
$$

To get convergence conditions we select $Q = \theta^{-1} I$ and factor the representations as (n, SUP, SU, U, UP) where

$$U = \theta M, \quad S = \theta^{-1} I \quad \text{and} \quad P = I$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix. This results in

$$(I - I_F)(P^T Q - S)U = (P^T Q - S)U = (\theta^{-1} I - \theta^{-1} I)U = 0$$

and hence

$$\begin{aligned}
W &= QU + (QU)^T - U^T QU - \tfrac{1}{2}U^T(P^TQ - S)B^\dagger(P^TQ - S)U \\
&= \theta^{-1}(U + U^T - U^TU) \\
&= M + M^T - \theta M^T M \\
&= 2\left[\begin{array}{c|c} \mathcal{T} & \\ \hline & \tau_n^{-1} \end{array}\right] - \theta\left[\begin{array}{c|c} \mathcal{T} & -\mathbf{1} \\ \hline \mathbf{1}^T & \tau_n^{-1} \end{array}\right]\left[\begin{array}{c|c} \mathcal{T} & \mathbf{1} \\ \hline -\mathbf{1}^T & \tau_n^{-1} \end{array}\right] \\
&= 2\left[\begin{array}{c|c} \mathcal{T} & \\ \hline & \tau_n^{-1} \end{array}\right] - \theta\left[\begin{array}{c|c} \mathcal{T}^2 + \mathbf{1}\mathbf{1}^T & \mathcal{T}\mathbf{1} - \tau_n^{-1}\mathbf{1} \\ \hline \mathbf{1}^T\mathcal{T} - \tau_n^{-1}\mathbf{1}^T & \tau_n^{-2} + n - 1 \end{array}\right] \\
&= \theta\left[\begin{array}{c|c} 2\theta^{-1}\mathcal{T} & \\ \hline & 2\theta^{-1}\tau_n^{-1} \end{array}\right] + \theta\left[\begin{array}{c|c} -\mathcal{T}^2 - \mathbf{1}\mathbf{1}^T & -\mathcal{T}\mathbf{1} + \tau_n^{-1}\mathbf{1} \\ \hline -\mathbf{1}^T\mathcal{T} + \tau_n^{-1}\mathbf{1}^T & -\tau_n^{-2} - n + 1 \end{array}\right] \\
&= \theta\left[\begin{array}{c|c} 2\theta^{-1}\mathcal{T} - \mathcal{T}^2 - \mathbf{1}\mathbf{1}^T & -\mathcal{T}\mathbf{1} + \tau_n^{-1}\mathbf{1} \\ \hline -\mathbf{1}^T\mathcal{T} + \tau_n^{-1}\mathbf{1}^T & 2\theta^{-1}\tau_n^{-1} - \tau_n^{-2} - n + 1 \end{array}\right].
\end{aligned}$$

For convergence it is required that $Q \succ 0$ and $W \succ 0$. As long as $\theta > 0$ then $Q \succ 0$ and it is clear that $W \succ 0$ for sufficiently small θ. If $\tau_i = \tau_n^{-1}$ for all $i \in \{1,\dots,n-1\}$, then W simplifies to

$$W = \theta\tau_n^{-1}\left[\begin{array}{c|c} (2\theta^{-1} - \tau_n^{-1})I - \mathbf{1}\mathbf{1}^T & \\ \hline & 2\theta^{-1} - \tau_n^{-1} - n + 1 \end{array}\right].$$

and $W \succ 0$ if

$$\theta < \frac{2}{n - 1 + \tau_n^{-1}}.$$

Note, although τ_i appear as step-sizes in the resolvents it could be argued that they are in fact not proper step-sizes since they appear outside the resolvents as well. If we introduce $\gamma > 0$ and apply this fixed point iteration to $\gamma A = (\gamma A_1, \dots, \gamma A_n)$ instead we get the following fixed point iteration

$$\begin{aligned}
x_i &= \mathrm{J}_{\tau_i \gamma A_i}(\tau_i z_i + z_n) && \text{for all } i \in \{1,\dots,n-1\}, \\
x_n &= \mathrm{J}_{\tau_n \gamma A_n}(z_n - \tau_n \textstyle\sum_{j=1}^{n-1} z_j), \\
\hat{z}_i &= z_i - \theta(x_i - x_n) && \text{for all } i \in \{1,\dots,n-1\}, \\
\hat{z}_n &= z_n - \theta(\tau_n^{-1} - \textstyle\sum_{j=1}^{n-1} \tau_i^{-1})z_n + \theta(\tau_n^{-1}x_n + \textstyle\sum_{j=1}^{n-1} \tau_i^{-1} x_j)
\end{aligned}$$

which converges for all $\gamma > 0$, as long as the choice of τ_i and θ are such that the Q and W above are positive definite. Hence, even if we might have to choose τ_i and θ small, the step-sizes in the resolvents can always be made arbitrarily large.

LUNDS UNIVERSITET

Fixpunktsiterationer för monotona inklusionsproblem

Martin Morin
Institutionen för Reglerteknik

Populärvetenskaplig sammanfattning av doktorsavhandlingen *Fixed Point Iterations for Finite Sum Monotone Inclusions*, november 2022. Avhandlingen kan laddas ner från: `http://www.control.lth.se/publications`

Matematik är språket som används för att modellera världen. I vardagen använder vi modeller för att till exempel beräkna restider utifrån kända hastighetsbegränsningar eller planera budgetar utifrån utgifter och inkomster. Matematiska modeller är också en av de främsta grundstenarna inom många ingenjörs-, och forskningsfält. Med dem kan vi bland annat simulera deformationen av en bil under en krasch och därmed bygga säkrare bilar och analysera störningar på elnätet för att göra det mer robust. Matematiska modeller är verktyget som tillåter oss förstå hur olika kvantiteter interagerar.

Med tiden har många modeller blivit allt mer komplexa och detaljerade, så komplexa att de inte längre går att analysera för hand. Dagens ingenjörer och forskare förlitar sig därför på datorer för att utföra sina beräkningar och utvecklingen av beräkningsmetoder anpassade för implementation på en dator har därför blivit av största vikt. Denna avhandling fokuserar på beräkningsmetoder för att lösa en typ av problem som kallas *monotona inklusionsproblem*. Dessa problem är vanligt förekommande inom en rad olika fält såsom statistisk analys, bild- och signalbehandling, modern maskininlärning och design av optimala lösningar, t.ex. hitta snabbaste GPS rutten eller eller bästa positionen av en Wi-Fi sändare.

Även i denna avhandling är modellering en grundsten som används för att analysera och utforska dessa beräkningsmetoder. Med hjälp av modeller i form av *fixpunktsiterationer* kan vi undersöka hur olika delar av en metod eller ett problem påverkar hur snabbt en lösning kan beräknas. Detta kan låta oss lösa problem snabbare och effektivare men gör också det möjligt att designa nya beräkningsmetoder i hopp om att lösa större och mer komplexa problem. Modellerna kan dock säga mer än bara vad som är möjligt, de kan också säga vad som inte är möjligt med en viss teknik eller typ av beräkningsmetod. Denna vetskap kan hjälpa ingenjörer att inte slösa tid på att försöka göra det omöjliga och styra forskare till de områden där nya tekniker behövs. Genom att bidra till förståelsen för de beräkningsmetoder som används av dagens ingenjörer och forskare hjälper avhandlingen till med att öppna upp för morgondagens tekniska applikationer och vetenskapliga upptäckter.